Pincer-Search: An Efficient Algorithm for Discovering the Maximum Frequent Set

Dao-I Lin Zvi M. Kedem

TKDE 2002 (pp: 553-566)

Presented by: Yi-Hung Wu
Date: 2002/9/18

Problem

• Finding the Maximum Frequent Sets (MFS)
  – MFS could be long
  – Two-way search: bottom-up and top-down
• Property 1

If an itemset is infrequent, all its supersets must be infrequent and if an itemset is frequent, all its subsets must be frequent and they need not be examined further.
Finding Frequent Sets

- Apriori algorithm
  - Join procedure
  - Prune procedure

Solution (1/3)

- Main Idea
  - Use of the information gathered in one direction to prune more candidates/passes in the other direction
  - Two way search by using the MFCS
    - All the maximal sets that are not known to be infrequent
Solutions (2/3)

- **Main Issues**
  - How to update the MFCS efficiently?
    - MFCS-gen
  - How to generate the correct candidate set?
    - Join procedure as in Apriori
    - Recovery procedure
    - New prune procedure

Solutions (3/3)

- **The Basic Pincer-Search Algorithm**
- **The Adaptive Pincer-Search Algorithms**
  - Delay the maintenance of the MFCS
  - Generate the candidates in MFS without counting
Performance Evaluation (1/2)

- **Scattered DB**
  - Many short MFS
- **Concentrated DB**
  - Few long MFS

Performance Evaluation (2/2)

- **Census DB**
  - Very long MFS
  - Scattered distribution
- **Stock Market DB**
  - 3000 stocks
  - Up/down
  - 60 minutes
  - 15 transactions
Conclusion Remarks

- **Contribution**
  - A new and efficient algorithm for mining maximal frequent sets for applications with long patterns

- **Weakness**
  - Long patterns in a scattered distribution

- **Related Work**
  - Dynamic Item Counting (DIC) [SIGMOD97]
  - Max-Miner [SIGMOD98]

Paper Scoring

- **Scores** \{bad, marginal, good, excellent\}
  - Originality: good
  - Technical Depth: good
  - Impact/Practicability: good
  - Readability: good
  - Overall: good
Related Work

• **Dynamic Itemset Counting and Implication Rules for Market Basket Data**
  – S. Brin, R. Motwani, J. D. Ullman, S. Tsur

---

**Candidate group**

\[ h(g) = \{1\}, \ t(g) = \{2, 3, 4\} \]

**Counting:**

\[ h(g), h(g) \cup t(g), h(g) \cup i, \forall i \in t(g) \]

---

**Related Work**

• **Efficiently Mining Long Patterns from Databases**
  – R. J. Bayardo

---

**Figure 5:** After 2M transactions.

**Figure 4:** After M transactions.