Apriori Algorithm

Rakesh Agrawal
Ramakrishnan Srikant
(description by C. Faloutsos)
Association rules - idea

[Agrawal+SIGMOD93]

- Consider ‘market basket’ case:
  (milk, bread)
  (milk)
  (milk, chocolate)
  (milk, bread)

- Find ‘interesting things’, eg., rules of the form:
  milk, bread -> chocolate | 90%
Association rules - idea

In general, for a given rule
\[ I_j, I_k, \ldots, I_m \rightarrow I_x \mid c \]

‘s’ = support: how often people buy \( I_j, \ldots, I_m, I_x \)

‘c’ = ‘confidence’ (how often people by \( I_x \), given that they have bought \( I_j, \ldots, I_m \))

Eg.: \( s = 20\% \)
\[ c = \frac{20}{40} = 50\% \]
Association rules - idea

Problem definition:

- **given**
  - a set of ‘market baskets’ (=binary matrix, of N rows/baskets and M columns/products)
  - min-support ‘s’ and
  - min-confidence ‘c’

- **find**
  - all the rules with higher support and confidence
Association rules - idea

Closely related concept: “large itemset”

\[ I_j, I_k, \ldots I_m, I_x \]

is a ‘large itemset’, if it appears more than ‘min-support’ times

Observation: once we have a ‘large itemset’, we can find out the qualifying rules easily

Thus, we focus on finding ‘large itemsets’
Association rules - idea

Naive solution: scan database once; keep $2^{|I|}$ counters

Drawback?

Improvement?
Association rules - idea

Naive solution: scan database once; keep $2^{2|I|}$ counters

Drawback? $2^{1000}$ is prohibitive...

Improvement? scan the db $|I|$ times, looking for 1-, 2-, etc itemsets

Eg., for $|I|=4$ items only (a,b,c,d), we have...
What itemsets do you count?

- **Anti-monotonicity**: Any superset of an infrequent itemset is also infrequent (SIGMOD ’93).
  - If an itemset is infrequent, don’t count any of its extensions.

- **Flip the property**: All subsets of a frequent itemset are frequent.

- **Need not count any candidate that has an infrequent subset** (VLDB ’94)
  - Simultaneously observed by Mannila et al., KDD ’94

- Broadly applicable to extensions and restrictions.
Apriori Algorithm: Breadth First Search

say, min-sup = 10

120

{}
Apriori Algorithm: Breadth First Search

say, min-sup = 10

120

{}

a   b   c   d

80   70   5   30
Apriori Algorithm: Breadth First Search

say, min-sup = 10

\[
\begin{array}{cccc}
80 & 70 & 5 & 30 \\
\end{array}
\]
Apriori Algorithm: Breadth First Search
Apriori Algorithm: Breadth First Search
Apriori Algorithm: Breadth First Search
Apriori Algorithm: Breadth First Search

Diagram:

- {} (root)
- a
  - a b
  - a d
- b
- c
- d
- a b d
Subsequent Algorithmic Innovations

- Reducing the cost of checking whether a candidate itemset is contained in a transaction:
  - TID intersection.
  - Database projection, FP Growth

- Reducing the number of passes over the data:
  - Sampling & Dynamic Counting

- Reducing the number of candidates counted:
  - For maximal patterns & constraints.

- Many other innovative ideas …
Impact

- Concepts in Apriori also applied to many generalizations, e.g., taxonomies, quantitative Associations, sequential Patterns, graphs, ...
- Over 3600 citations in Google Scholar.