Data Mining
Mining Association Rules

Joe Burdis
Fall 2024
CUNY Graduate Center

Contents

Reference: Mining of Massive Datasets, Chapter 6. http://mmds.org

Motivating Examples

- Identify items that are often bought together in a supermarket.
    - People who buy diaper and milk often buy beer also.
    - Women’s shoes are often bought together with men’s clothes. *
- Amazon: Customers who viewed this item also viewed ...
- Medical researchers want to discover side effects of drugs.
- Netflix, Spotify, YouTube, Tiktok...

How can we identify significant association rules from data?

- Define association rules
- Define significant association
- Develop a corresponding algorithm

*: Diapers, Beer, and Data Science in Retail. https://canworksmart.com/diapers-beer-retail-predictive-analytics/

The Market-Basket Model

Consider Amazon recommendation as an example.

- The **market** is the set of all items sold on Amazon.
- Each piece of user data is a **basket** of items viewed by a particular customer.
- **Association rules**: if a basket contains {x, y, z}, then it is highly likely to contain {v, w} also.
- Applications of association rules:
  - Display highly related items on the same page.
  - To boost sales of one item, run sale on its associated items.

What Qualifies as a Significant Association?

Step 1: Find sets of items that appear together frequently.

Mathematical Model:

- The **support** of an itemset I: Number of baskets containing I.
- The support is sometimes expressed as a fraction of the total number of baskets.
- Given a support threshold s, those itemsets with support >= s are called **frequent itemsets**.

TID	*Items*
1	Bread, Coke, Milk
2	Beer, Bread
3	Beer, Coke, Diaper, Milk
4	Beer, Bread, Diaper, Milk
5	Coke, Diaper, Milk

Support of {Beer, Bread} = 2

Example of Frequent Itemsets

$B_1 = \{m, c, b\}$ $B_2 = \{m, p, j\}$

$B_3 = \{m, b\}$ $B_4 = \{c, j\}$

$B_5 = \{m, p, b\}$ $B_6 = \{m, c, b, j\}$

$B_7 = \{c, b, j\}$ $B_8 = \{b, c\}$

- Which itemsets are most frequent?

  {m, b} has support = 4, {c, b} has support 4, {m} has support 5, {b} has support 6.

- Is p => m a significant association rule?
- Is b => m a significant association rule?

What Qualifies as a Significant Association?

Step 2: If an association rule I => j is significant, it implies that "if a basket contains I, then it is likely to contain j also."

- Define the **confidence** of the association rule I => j as

$$\mathrm{conf}(I \to j) = \frac{\mathrm{support}(I \cup j)}{\mathrm{support}(I)}$$

- The confidence can be considered as the conditional probability

  P(basket contains {j} | basket contains I), i.e., the probability of that a basket contains {j} given that it contains I.

Example of Confidence

$B_1 = \{m, c, b\}$ $B_2 = \{m, p, j\}$
$B_3 = \{m, b\}$ $B_4 = \{c, j\}$
$B_5 = \{m, p, b\}$ $B_6 = \{m, c, b, j\}$
$B_7 = \{c, b, j\}$ $B_8 = \{b, c\}$

- What is the confidence of rule b => m? supp({m, b})/supp({b}) = 4 / 6
- What is the confidence of rule (m, b) => c? supp({m, b, c})/supp({m, b}) = 2 / 4

What Qualifies as a Significant Association?

Step 3: A significant association rule should be a high confidence rule. Moreover, this association should occur frequently relative to the suggested item.

- The rule X => milk may have high confidence for many itemsets X, because milk is just purchased very often and thus the confidence is likely high.
- Define the **interest** of an association rule I => j as the difference between its confidence and the fraction of baskets that contains j:

$$\text{Interest}(I \rightarrow j) = \text{conf}(I \rightarrow j) - \Pr[j]$$

Example of Confidence and Interest

B_1 = {m, c, b} B_2 = {m, p, j}

B_3 = {m, b} B_4 = {c, j}

B_5 = {m, p, b} B_6 = {m, c, b, j}

B_7 = {c, b, j} B_8 = {b, c}

- What is the interest of rule **(m, b) => c**? ½ - ⅝ = -⅛
- What is the interest of rule **b => m**? 2/3 - ? = ?

Mining Association Rules

Problem: Find all association rules with support >= s and confidence >= c.

- Step 1: Final all frequent itemsets I with threshold s. (Nontrivial for big data)
- Step 2: Generate association rules:
    - For every subset A of I, generate a rule A => I \ A.
    - Since I is frequent, A is also frequent.
- Step 3: Output rules above the confidence threshold.
    - The algorithm should check small rules first: If {A, B} => {C, D} has confidence >= c, then {A, B, C} => D also has confidence >= c. (Why?)

Proof:

C <= Conf1 = supp({A, B, C, D}) / supp({A, B}) <= supp({A, B, C, D}) / supp({A, B, C}) = Conf2

Example

B1 = {m, c, b}
B2 = {m, p, j}
B3 = {m, b}
B4 = {c, j}
B5 = {m, p, b}
B6 = {m, c, b, j}
B7 = {c, b, j}
B8 = {b, c}

- Which itemsets have support >= 3?
- Moreover, which rules have confidence >= 0.75?

m,c,b,p,j
{m},..,{j}

{m},{c},{b},{j}
{m,b},{c,b},{c,j}

m->b
c->b
j->c

Compacting the Output

To reduce the number of rules in the output, we can choose to only keep:

- Maximal frequent itemsets: no immediate superset is frequent
- Closed itemsets: no immediate superset has the same support

	Support	Maximal(s=3)	Closed
~~A~~	~~4~~	~~No~~	~~No~~
B	5	No	Yes
~~C~~	~~3~~	~~No~~	~~No~~
AB	4	Yes	Yes
~~AC~~	~~2~~	~~No~~	~~No~~
BC	3	Yes	Yes
~~ABC~~	~~2~~	~~No~~	~~Yes~~

Finding Frequent Itemsets

The hardest problem often turns out to be finding frequent pairs {i, j}.

- The probability of being frequent drops exponentially with the size of the set.

Let's focus on finding frequent itemsets of size 2.

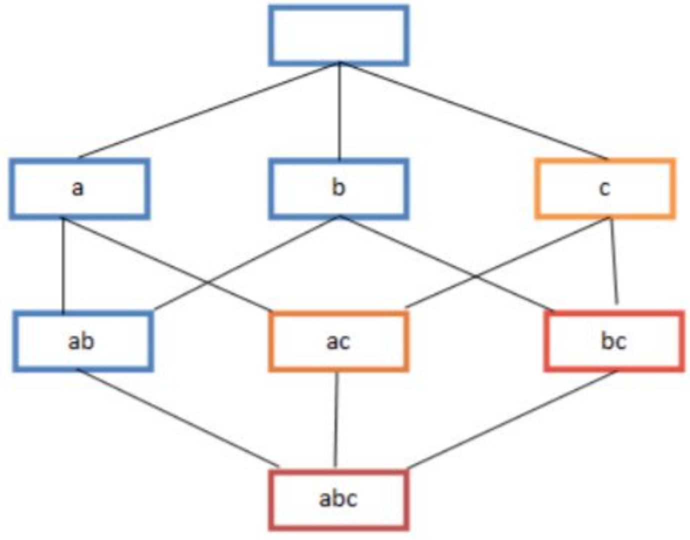
- Walmart has about 100k items on its website.
- Number of all subsets of size 2: $10^5(10^5-1)/2 = 2.5*10^9$.
- If we count every pair with a 4-byte integer: 20 GB memory is needed.
- Use a sparse matrix may reduce the memory requirement, but we can do better with **A-Priori algorithm**.

A-Priori Algorithm

Key idea: monotonicity

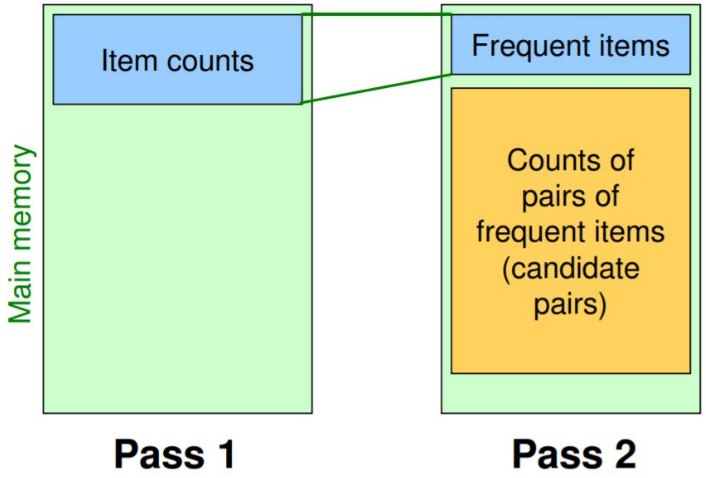
- If an itemset I has support greater than s, then so does every subset J of I.
- If an item i appears less than s times, then no set containing i has support >= s.

Conclusion: frequent item pairs only come from two frequent items.



A-Priori Algorithm for Frequent Item Pairs

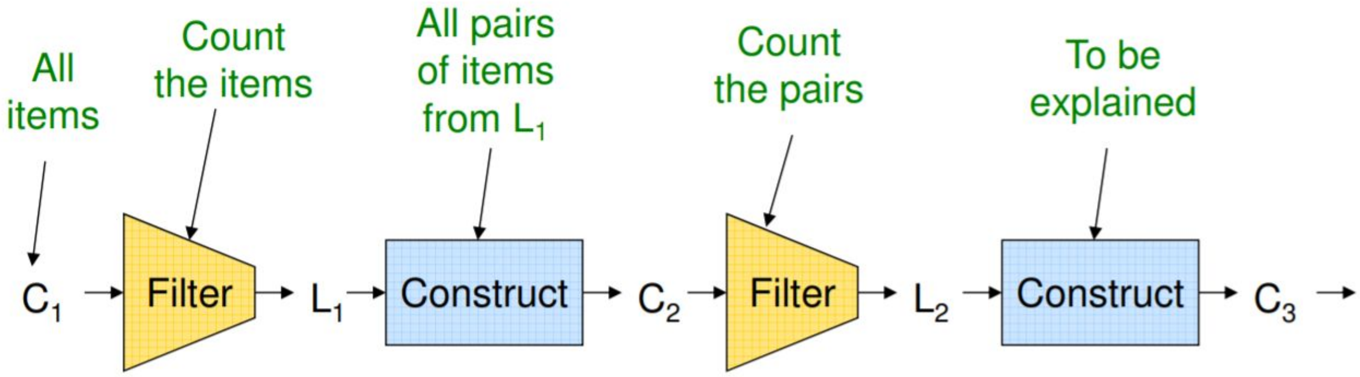
- During the first pass of data, count the occurrences of each individual item.
    - Requires only memory proportional to the number of items
- Keep only those frequent items that appear >= s times.
- During the second pass of data, count only those pairs where both elements are frequent.
    - Requires additional memory proportional to the square of frequent items.
- Output pairs with confidence >= c.



A Priori Algorithm for Itemsets of Size k

For each k, construct two sets of k-tuples:

- C_k: candidate k-tuples generated from frequent itemsets of size k-1.
- L_k: the set of truly frequent k-tuples.



Hypothetical Steps of the A-Priori Algorithm

- C1 = { {b} {c} {j} {m} {n} {p} }
- Count the support of itemsets in C1
- Prune non-frequent: L1 = { b, c, j, m }

- Generate C2 = { {b,c} {b,j} {b,m} {c,j} {c,m} {j,m} }
- Count the support of itemsets in C2
- Prune non-frequent: L2 = { {b,m} {b,c} {c,m} {c,j} }

- Generate C3 = { {b,c,m} {b,c,j} {b,m,j} {c,m,j} }
- Count the support of itemsets in C3
- Prune non-frequent: L3 = { {b,c,m} }

Extensions to A Priori Algorithm

- High-level association rules
  - Men over 65 like ...
  - {baked goods, milk products} => preserved goods
- Varying s and c based on the size of itemsets
- Faster algorithm
- Algorithms requires fewer passes of data