Data Mining for CS336

- **Simplest question:** Find sets of items that appear together "frequently" in baskets
- ***Support*** for itemset I: Number of baskets containing all items in I
  - (Often expressed as a fraction of the total number of baskets)
- Given a ***support threshold*** s, then sets of items that appear in at least s baskets 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

Each row is a basket

Support of {Beer, Bread} = 2

Example: frequent itemset

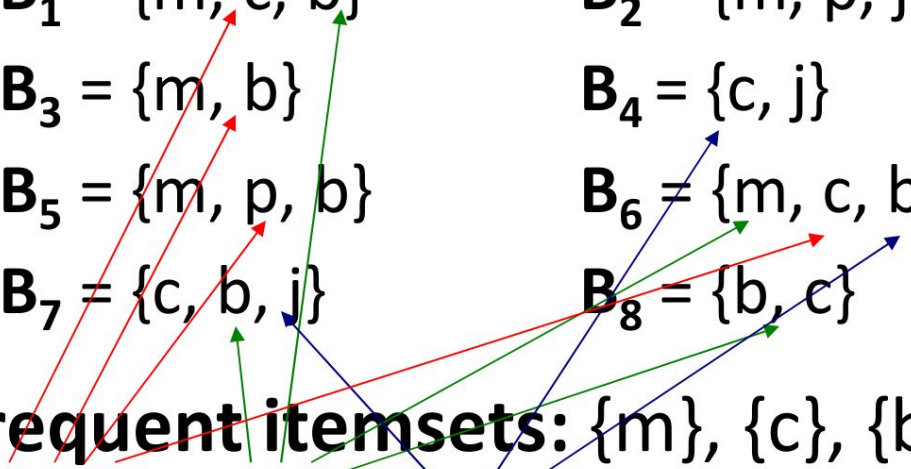
- **Items** = {milk, coke, pepsi, beer, juice}
- **Support threshold** = 3 baskets

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}



- **Frequent itemsets:** {m}, {c}, {b}, {j}, {m,b} , {b,c} , {c,j}.

[2] Association Rule and its Confidence

If-then rules about the contents of baskets

- $\{i_1, i_2, \ldots, i_k\} \rightarrow j$ means: "if a basket contains all of $i_1, \ldots, i_k$ then it is ***likely*** to contain j"

- ***Confidence*** of this association rule is the probability of j given $I = \{i_1, \ldots, i_k\}$

$$\text{conf}(I \rightarrow j) = \frac{\text{support}(I \cup j)}{\text{support}(I)}$$

[3] Mining Association Rules of high confidence

Step1: Find all frequent itemsets via A-Priori algorithm

Step2: For each frequent itemset I, conduct rule generation:

(2.1) For every subset A of I, generate a rule $A \rightarrow I \setminus A$

(Since I is frequent, A is also frequent)

(2.2) Compute the confidence for each rule

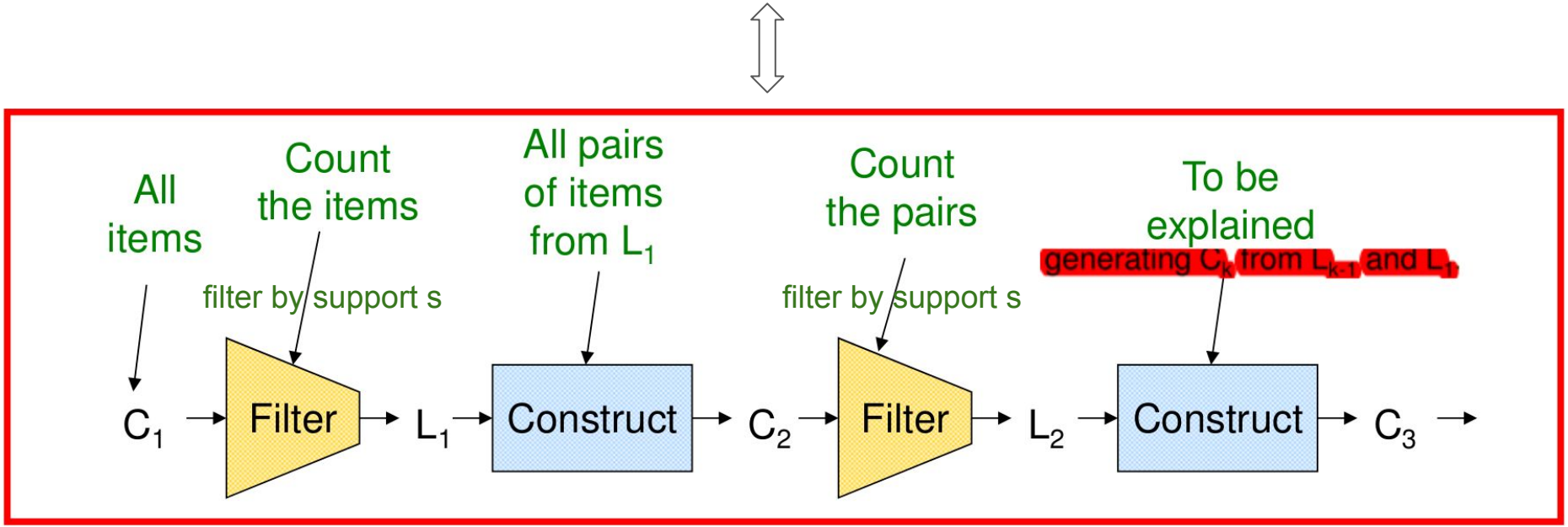
$$\text{confidence}(A \rightarrow I \setminus A) = \frac{\text{support}(A) \cup \text{support}(I \setminus A)}{\text{support}(A)} = \frac{\text{support}(I)}{\text{support}(A)}$$

(2.3) output the rules above the confidence threshold

[4] A-Priori Algorithm for finding frequent itemsets

For k = 1, 2,... N, where N is the total number of items, k is the itemset size, **construct two sets of size k**:

- **C_k** = the set of candidate frequent itemsets (size k) = those that might be frequent itemsets (support >= s) based on information from the pass of k-1
- **L_k** = the set of truly frequent itemsets (size k)



- **Hypothetical steps of the A-Priori algorithm**
  - C_1 = { {b} {c} {j} {m} {n} {p} }
  - Count the support of itemsets in C_1
  - Prune non-frequent: L_1 = { b, c, j, m }
  - Generate C_2 = { {b,c} {b,j} {b,m} {c,j} {c,m} {j,m} }
  - Count the support of itemsets in C_2
  - Prune non-frequent: L_2 = { {b,m} {b,c} {c,m} {c,j} }
  - Generate C_3 = { {b,c,m} {b,c,j} {b,m,j} {c,m,j} } **
  - Count the support of itemsets in C_3
  - Prune non-frequent: L_3 = { {b,c,m} }