Apriori Algorithm for mining Association Rules

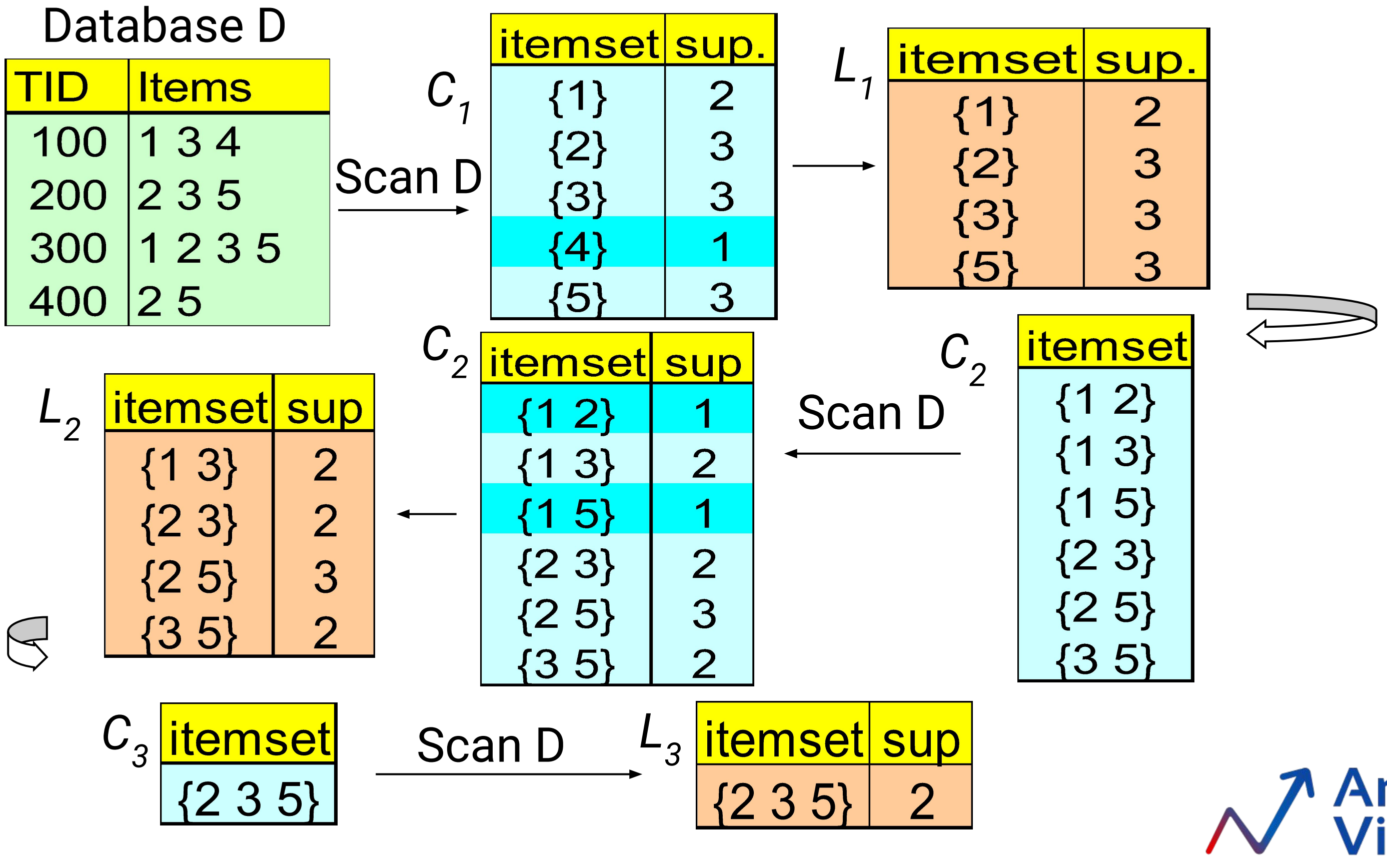
Analytics Vidhya

Apriori Algorithm

- There are many association rule mining algorithms

- Most Popular: Apriori Algorithm
  - Identifies the frequent individual items in the database
  - Extends them to larger and larger item sets if those itemsets appear sufficiently often in the database

Apriori Algorithm - Example



Database D

TID	Items
100	1 3 4
200	2 3 5
300	1 2 3 5
400	2 5

Scan D

C_1

itemset	sup.
{1}	2
{2}	3
{3}	3
{4}	1
{5}	3

L_1

itemset	sup.
{1}	2
{2}	3
{3}	3
{5}	3

C_2

itemset
{1 2}
{1 3}
{1 5}
{2 3}
{2 5}
{3 5}

Scan D

C_2

itemset	sup
{1 2}	1
{1 3}	2
{1 5}	1
{2 3}	2
{2 5}	3
{3 5}	2

L_2

itemset	sup
{1 3}	2
{2 3}	2
{2 5}	3
{3 5}	2

C_3

itemset
{2 3 5}

Scan D

L_3

itemset	sup
{2 3 5}	2

Analytics Vidhya

Generating Candidate Itemsets C_4

- Suppose these are the only 3-itemsets all have >10% support:

  {1, 2, 3}

  {1, 5, 7}

  {5, 6, 8}

  {5, 6, 11}

  {16, 17, 18}

- How do we generate candidate 4-itemsets that *might* have 10% support?

Generating Candidate Itemsets C_4

- Suppose these are the only 3-itemsets all have >10% support:

  {1, 2, 3}

  {1, 5, 7}

  {5, 6, 8}

  {5, 6, 11}

  {16, 17, 18}

Brute Force:

- Note all the items involved: {1, 2, 3, 5, 6, 7, 8, 11, 16, 17, 18}

- Generate all subsets of 4 of these:
  {1,2,3,5}, {1,2,3,6}, {1,2,3,7}, {1,2,3,8}, {1,2,3,11}, {1,2,3,16} etc …

  there are 330 possible subsets in this case!

Generating Candidate Itemsets C_4

- Suppose these are the only 3-itemsets all have >10% support:

  {1, 2, 3}

  {1, 5, 7}

  {5, 6, 8}

  {5, 6, 11}

  {16, 17, 18}

- We can easily see that {1,2,3,5} couldn't have 10% support – because {1,2,5} is *not* one of our 3-itemsets

- Same goes for several other of these subsets

Apriori Trick

{1, 2, 3}

{1, 5, 7}

{5, 6, 8}

{5, 6, 11}

{16, 17, 18}

- Enforce that subsets are always arranged in an order (or similar), as they are already on the left
- **Only** generate *k*+1-itemset candidates from *k*-itemsets that differ in the last item.
- So, in this case, the only candidate 4-itemset would be:

{5, 6, 8, 11}

Apriori Trick

This trick

- Guarantees to capture the itemsets that have enough support
- Will still generate some candidates that don’t have enough support, so we still have to check them in the ‘pruning’ step,
- So for example we need to check if {5, 6, 8,11} has support greater than 10% or not
- If it does, algorithm will stop here as there is just 1 large itemset and no possibility of a 5-large itemset

Recommendation based on Association Rule Mining

- Simplest approach
  - Transform 5-point ratings into binary ratings (1 = above user average)

	Item1	Item2	Item3	Item4	Item5
Alice	1	0	0	0	?
User1	1	0	1	0	1
User2	1	0	1	0	1
User3	0	0	0	1	1
User4	0	1	1	0	0

- Mine rules such as
  - Item1 → Item5
    - support (2/4), confidence (2/2) (without Alice)

- Make recommendations for Alice (basic method)
  - Determine "relevant" rules based on Alice's transactions (the above rule will be relevant as Alice bought Item1)
  - Determine items not already bought by Alice
  - Sort the items based on the rules' confidence values

Association Rule Mining: Formal Definition

- Commonly used for shopping behavior analysis
  - aims at detection of rules such as

    "If a customer purchases baby food then he also buys diapers in 70% of the cases"

- Association rule mining algorithms
  - can detect rules of the form $X \rightarrow Y$ (e.g., beer → diapers) from a set of sales transactions $D = \{t_1, t_2, \ldots t_n\}$
  - Here X is called antecedent & Y is called consequent & X,Y have no items in common
  - Each transaction from D will have information regarding the set of items bought together
  - measure of quality: support, confidence
    - used e.g. as a threshold to cut off unimportant rules