Association Rule Mining

Apriori Algorithm

What is Association Rule Learning?

People who bought also bought …

Association Rule Mining

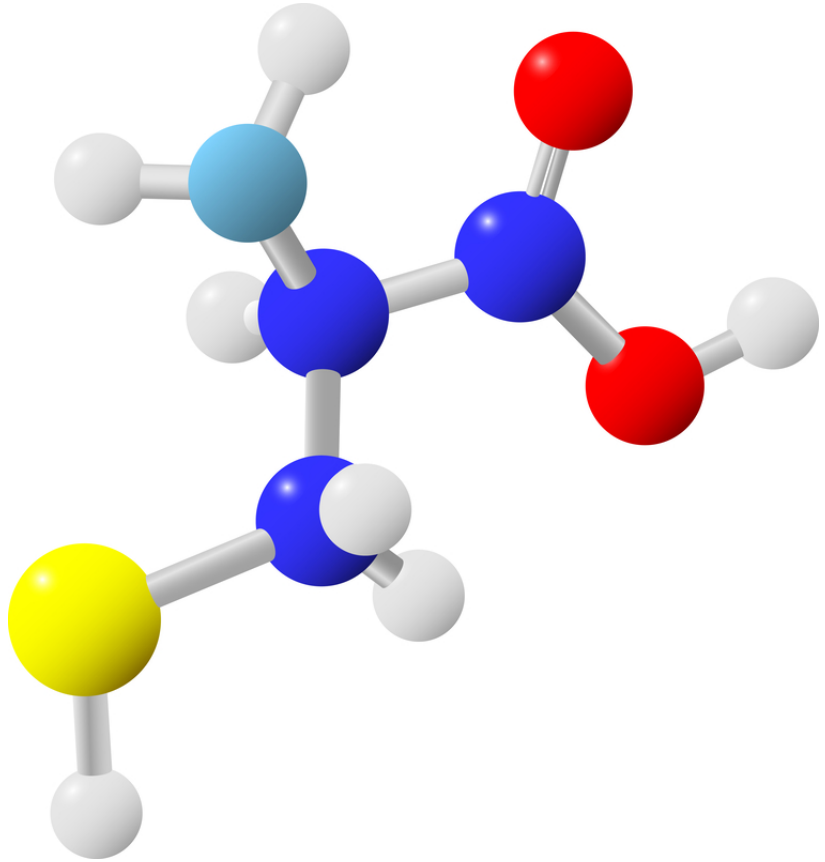
- **Definition: Looking for frequent associations or correlations among sets of items in transactional databases.**
- **Simple *If/Then* statements that demonstrate relationships with a probability**
  - **"If a customer buys bread, then he's 70% likely to buy milk."**
- **A rule has two parts: the antecedent (*if*) and the consequent (*then*)**

Mathematically written as:

$$A \rightarrow B$$ **(A *implies* B)**

Association Rule Usage

- **Medical Diagnosis - associate symptoms to illness**
- **Census Data Analysis - planning efficient public services**
- **Protein Sequences - understanding amino acid sequences in protein functioning**



Association Rule Usage

- **Often used for targeted marketing in retail businesses - Market Basket Analysis**
  - **Shoppers who purchase oil filters also tend to purchase sunglasses.**

Movie Recommendation

User ID	Movies liked
46892	Movie1, Movie2, Movie3, Movie4
31266	Movie1, Movie2
85658	Movie1, Movie2, Movie4
15698	Movie1, Movie2
12876	Movie2, Movie4
45682	Movie1, Movie3

Potential Rules:

Movie1 → Movie2

Movie2 → Movie4

Movie1 → Movie3

Market Basket Optimization

Transaction ID	Products Purchased
78935	Burgers, French Fries, Salad
88648	Burgers, French Fries, Ketchup
79926	Salad, Fruits
48676	Pasta, Fruits, Butter, Salad
98751	Burgers, Pasta, French Fries
68542	Fruits, Apple Juice, Salad
78945	Burgers, French Fries, Ketchup, Mayo

Potential Rules:

Burgers → French Fries

Salad → Fruits

Burgers, French Fries → Ketchup

Association Rule Mining

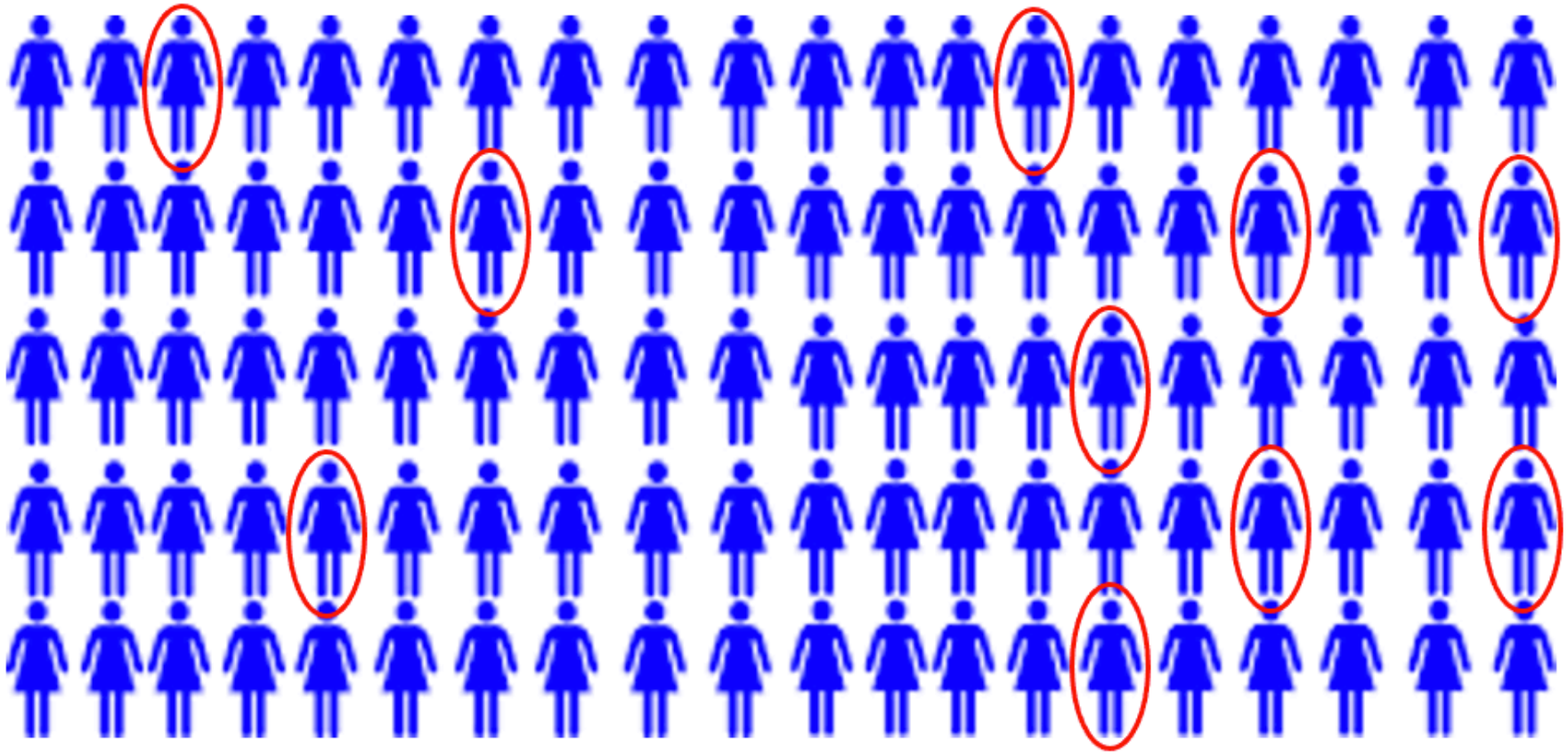
- Step 1 - Support
  - How often do {A,B} occur in the population of transactions?

Association Rule - Support

- Movie Recommendation: support(M) = $\frac{\#\ user\ watchlists\ containing\ M}{\#\ user\ watchlists}$

- Market Basket Optimization: support(I) = $\frac{\#\ transactions\ containing\ I}{\#\ transactions}$

Association Rule - Support



Support = 10/100 = 10%

Association Rule Mining

- Step 1 - Support
  - How often do {A,B} occur in the population of transactions?
  - Itemsets that pass the support threshold are called **frequent itemsets**

→ - Step 2 - Confidence
  - For each potential association rule (A → B), what percentage of the occurrences of A also include B?

Association Rule - confidence

- Movie Recommendation:

$$confidence(M1 \rightarrow M2) = \frac{\#\ user\ watchlists\ containing\ M1\ and\ M2}{\#\ user\ watchlists\ containing\ M1}$$

- Market Basket Optimization:

$$confidence(I1 \rightarrow I2) = \frac{\#\ user\ watchlists\ containing\ I1\ and\ I2}{\#\ user\ watchlists\ containing\ I1}$$

Association Rule - Confidence

Confidence: 7/40 = 17.5%

Association Rule Mining

- One drawback of the confidence measure is that it might misrepresent the importance of an association.

- A high confidence measure only accounts for how popular diapers are, but not beer.

- If beer is also very popular in general, there will be a higher chance that a transaction containing diapers will also contain beer, thus inflating the confidence measure.

Association Rule Mining

- Step 1 - Support
  - How often do {A,B} occur in the population of transactions?

- Step 2 - Confidence
  - For each potential association rule (A → B), what percentage of the occurrences of A also include B?

→ • Step 3 - Lift
  - How much more frequently does A occur with B than without B?

Association Rule Mining

Lift

- how likely is item B purchased when item A is purchased, *while controlling for how popular item B is.*

- A lift of 1 implies no association between items. A lift value greater than 1 means that item B is *likely* to be bought if item A is bought, while a value less than 1 means that item B is *more likely* to be bought without item A.

Association Rule - Lift

- Movie Recommendation:

$$lift(M1 \rightarrow M2) = \frac{confidence\ (M1 \rightarrow M2)}{support\ (M2)}$$

- Market Basket Optimization:

$$lift(I1 \rightarrow I2) = \frac{confidence\ (I1 \rightarrow I2)}{support\ (I2)}$$

Association Rule - Lift

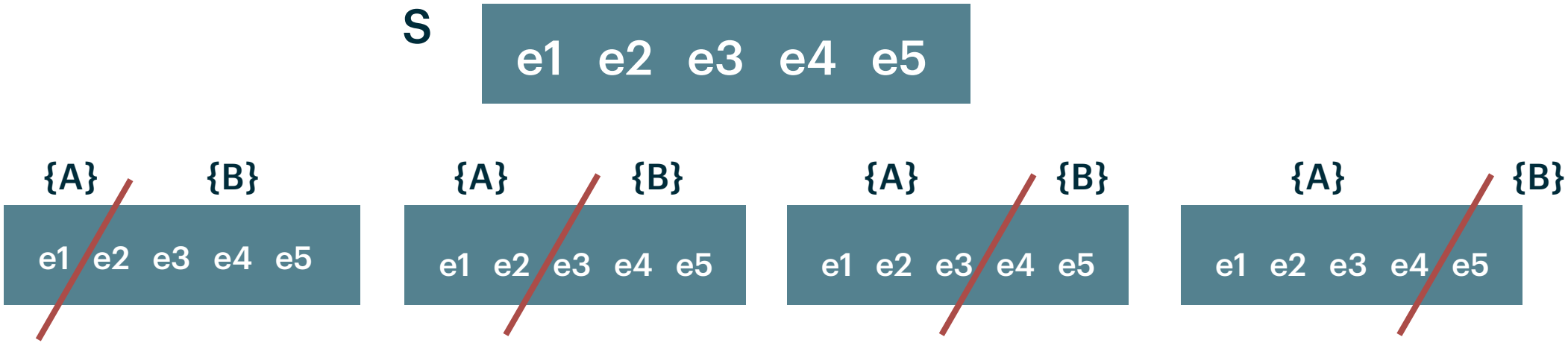
Lift = 17.5% / 10% = 1.75

Association Rule - Algorithm

- Step1: Set a minimum support and confidence
  - Minimum support and confidence are thresholds set by the business
- Step2: Start with subsets of size 1, then size 2 - incrementing up to the number of unique items.
- Step3: Determine the subsets having higher support than the minimum - called **frequent** subsets.
- Step4: Determine all possible rules of the **frequent** subsets (candidate rules) and identify those having higher than minimum confidence
- Step5: sort the rules by decreasing lift

A → B (A implies B)

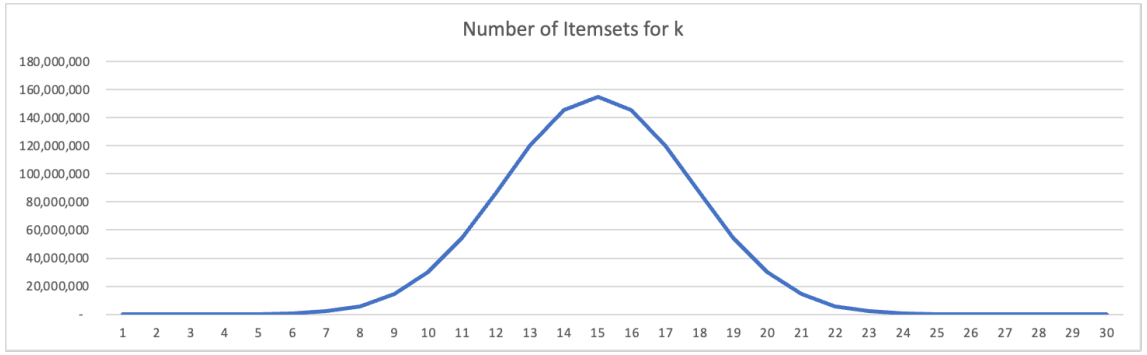
- The items in **A** and in **B**, written as {A,B}, are called an **itemset** {e1, e2, e3, e4 ...}
- If there are **n** unique item ids, an itemset can consist of from **1** to **n** elements
- Which we will partition as two-set partitions: **{A}** and **{B}**



Number of Itemsets

Number of unique itemsets of size k can we generate:

$$ {}_nC_k = \frac{n!}{k!(n-k)!} $$



n = number of unique elements

k = number of elements in itemset

itemsets for k = 1, 2, 3 . . . n (n=30) = 1,073,741,824

Finding Association Rules - Brute Force

for k = 1 to n - 1 *(where n = number of items)*

generate k-itemsets of using n items

for each k-itemset

compute support

if support threshold is met and k > 1 *(no rules for 1-itemsets)*

generate rule candidates

for each rule candidate

compute confidence

if confidence threshold is met

add candidate to rule list

if none of k-itemsets are frequent

STOP

Apriori principles

- Any subset of a frequent itemset must be frequent
- Any superset of a non-frequent itemset must be non-frequent

Finding Association Rules - Apriori

for k = 1 to n *(where n = number of items)*
generate k-itemsets of using only frequent itemsets from k-1 *(unless k == 1)*
for each k-itemset
compute support
if support threshold is met
add itemset to frequent_itemset list
if k > 1 *(no rules for 1-itemsets)*
generate rule candidates
for each rule candidate
compute confidence
if confidence threshold is met
add candidate to rule list
if none of k-itemsets are frequent
STOP

Run-time Comparison

- **30 items, 5 samples of 20 transactions, Single-threaded Python script:**

File	Rules Found	Apriori Time	Brute Force	Speed Differenc
sales1	43	0.00474	0.37225	78.49x
sales2	16	0.00487	0.37210	76.45x
sales3	125	0.01611	2.94327	182.66x
sales4	6	0.00146	0.38256	262.66x
sales5	29	0.00966	0.39230	40.60x

Apriori Example

5 Items, 7 Transactions

T1	Bread	Milk		
T2	Bread	Milk	Banana	
T3	Eggs	Banana		
T4	Eggs	Bread	Milk	Banana
T5	Milk	Banana		
T6	Milk	Banana	Soup	
T7	Eggs	Bread		

Minimum Support >= 20%

1-itemsets	Support
{Eggs}	3 - 42.9%
{Bread}	4 - 57.1%
{Milk}	5 - 71.4%
{Bananas}	5 - 71.4%
~~{Soup}~~	~~1 - 14.3%~~

Frequent 1-itemsets	Support
{Eggs}	3 -
{Bread}	4 - 57.1%
{Milk}	5 - 71.4%
{Bananas}	5 - 71.4%

Apriori Example

Minimum Support >= 20%

Frequent 1-itemsets	Support
{Eggs}	3 -
{Bread}	4 - 57.1%
{Milk}	5 - 71.4%
{Bananas}	5 - 71.4%

2-itemsets	Support
{Eggs,Bread}	2 -
~~{Eggs,Milk}~~	~~1 - 14.3%~~
{Eggs,Bananas}	2 -
{Bread,Milk}	3 -
{Bread,Bananas}	2 -
{Milk,Bananas}	4 - 57.1%

Frequent 2-itemsets	Support
{Eggs,Bread}	2 - 28.6%
{Eggs,Bananas}	2 - 28.6%
{Bread,Milk}	3 - 42.9%
{Bread,Bananas}	2 - 28.6%
{Milk,Bananas}	4 - 57.1%

Brute Force = 10 2-itemsets

Apriori Example

Minimum Support >= 20%

Frequent 2-itemsets	Support
{Eggs,Bread}	2 -
{Eggs,Bananas}	2 -
{Bread,Milk}	3 -
{Bread,Bananas}	2 -
{Milk,Bananas}	4 - 57.1%

3-itemsets	Support
~~{Eggs,Bread,Bananas}~~	~~1 - 14.3%~~
{Bread,Milk,Bananas}	2 -

Brute Force = 10 3-itemsets

Frequent 3-itemsets	Support
{Bread,Milk,Bananas}	2 - 28.6%

Apriori Example

Minimum Support >= 20%

All Frequent itemsets	Support
{Eggs}	3 - 42.9%
{Bread}	4 - 57.1%
{Milk}	5 - 71.4%
{Bananas}	5 - 71.4%
{Eggs,Bread}	2 - 28.6%
{Eggs,Bananas}	2 - 28.6%
{Bread,Milk}	3 - 42.9%
{Bread,Bananas}	2 - 28.6%
{Milk,Bananas}	4 - 57.1%
{Bread,Milk,Bananas}	2 - 28.6%

Candidate Rules	Confidence
{Eggs} → {Bread}	66.7%
{Bread} → {Eggs}	50.1%
{Eggs} → {Bananas}	66.7%
~~{Bananas} → {Eggs}~~	~~40.1%~~
{Bread} → {Milk}	75.1%
{Milk} → {Bread}	60.1%
{Bread} → {Bananas}	50.1%
~~{Bananas} → {Bread}~~	~~40.1%~~
{Milk} → {Bananas}	80%
{Bananas} → {Milk}	80%

Minimum Confidence >= 50%

Candidate Rules	Confidence
{Bread} → {Milk,Bananas}	50.1%
{Milk,Bananas} → {Bread}	50.1%
~~{Milk} → {Bread,Bananas}~~	~~40.1%~~
{Bread,Bananas} → {Milk}	100%
~~{Bananas} → {Bread,Milk}~~	~~40.1%~~
{Bread,Milk} → {Bananas}	66.7%

Apriori Example

Minimum Support >= 20%

All Frequent itemsets	Support
{Eggs}	3 - 42.9%
{Bread}	4 - 57.1%
{Milk}	5 - 71.4%
{Bananas}	5 - 71.4%
{Eggs,Bread}	2 - 28.6%
{Eggs,Bananas}	2 - 28.6%
{Bread,Milk}	3 - 42.9%
{Bread,Bananas}	2 - 28.6%
{Milk,Bananas}	4 - 57.1%
{Bread,Milk,Bananas}	2 - 28.6%

Candidate Rules	Confidence	Lift
{Eggs} → {Bread}	66.7%	1.1681
{Bread} → {Eggs}	50.1%	1.1678
{Eggs} → {Bananas}	66.7%	0.9342
~~{Bananas} → {Eggs}~~	~~40.1%~~	
{Bread} → {Milk}	75.1%	1.0518
{Milk} → {Bread}	60.1%	1.0525
{Bread} → {Bananas}	50.1%	0.7017
~~{Bananas} → {Bread}~~	~~40.1%~~	
{Milk} → {Bananas}	80%	1.1204
{Bananas} → {Milk}	80%	1.1204

Minimum Confidence >= 50%

Candidate Rules	Confidence	Lift
{Bread} → {Milk,Bananas}	50.1%	0.8774
{Milk,Bananas} → {Bread}	50.1%	1.0720
~~{Milk} → {Bread,Bananas}~~	~~40.1%~~	
{Bread,Bananas} → {Milk}	100%	1.4006
~~{Bananas} → {Bread,Milk}~~	~~40.1%~~	
{Bread,Milk} → {Bananas}	66.7%	0.9342

Apriori Example

Derived Association Rules:

{Eggs} → {Bread} [28.6%, 66.7%, 1.1681]
{Bread} → {Eggs} [28.6%, 50.1%, 1.1678]
{Eggs} → {Bananas} [28.6%, 66.7%, 0.9342]
{Bread} → {Milk} [42.9%, 75.1%, 1.0518]
{Milk} → {Bread} [42.9%, 60.1%, 1.0525]
{Bread} → {Bananas} [28.6%, 50.1%, 0.7017]
{Milk} → {Bananas} [57.1%, 80%, 1.1204]
{Bananas} → {Milk} [57.1%, 80%, 1.1204]
{Bread} → {Milk,Bananas} [28.6%, 50.1%, 08774]
{Milk,Bananas} → {Bread} [28.6%, 50.1%, 1.0720]
{Bread,Bananas} → {Milk} [28.6%, 100%, 1.4006]
{Bread,Milk} → {Bananas} [28.6%, 66.7%, 0.9342]

Minimum Support >= 20%, Minimum Confidence >= 50%