ASSOCIATION RULES

Relative causality and data mining.

INTRODUCTION

Example: Imagine we have a set of sets such as, for example:

$$\{\{a, b, c\}, \{a, b, d, \}, \{a, c, e, f\}, \{i, g, k, l\}, \{a, b, h, g, n\}, \{c, f, i, m\}\}$$

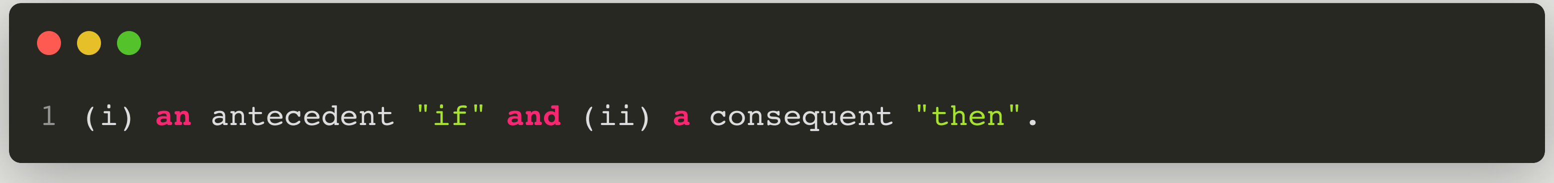
The subsets inside the bigger set can be referred as "transactions" if one imagine different purchases at a store.

Question: in the previous example what were the most frequent items purchased?

Answer: In that example we had a total of 6 transactions among which 4 transactions had the item "a" purchased and also 3 transactions had both items "a" and "b".

INTRODUCTION

Definition: An association rule has two parts



```
(i) an antecedent "if" and (ii) a consequent "then".
```

An antecedent is an item found within the data; a consequent is an item found in combination with the antecedent.

Thus, association rules can be established by algorithm that can search and count the frequent *if-then* patterns in a database.

Main Goal: apply efficient machine learning algorithms for analyzing associations or co-occurrences in a database.

During this lesson we are going to study and apply two important algorthms: APRIORI and ECLAT.

THE APRIORI ALGORITHM

Proposed by R. Agrawal and R. Srikant in 1994 for finding frequent itemsets in a dataset for boolean association rule. Name of the algorithm is Apriori because it uses prior knowledge of frequent itemset properties. We apply an iterative approach or level-wise search where k-frequent itemsets are used to find k+1 itemsets.

The idea is to search for frequent *if-then* patterns; the algorithm uses the concepts of **support** and **confidence** in order to identify the most frequent (and thus relevant) relationships.

SUPPORT: IS A MEASURE OF HOW FREQUENTLY THE ITEMS APPEAR IN THE DATASET OF ALL TRANSACTIONS.

CONFIDENCE: REPRESENTS THE NUMBER OF TIMES THE *IF-THEN* STATEMENTS ARE FOUND TRUE.

LIFT: IS METRIC THAT CAN HELP COMPARE THE CONFIDENCE VS. THE *EXPECTED* CONFIDENCE.

EXAMPLE

Support: Chance you pick someone that took DATA 301 by random i.e. a probability value:

$$\mathrm{P(course\ taken = DATA301)} = \frac{\text{The number of students who took DATA 301}}{\text{Total number of students}}$$

Confidence: Chance you pick someone that took DATA 310, given they're in another course. This is a *conditional* probability, i.e.

$$\mathrm{P(course\ taken = DATA301 | course\ taken = DATA201)}$$

Lift: Improvement in chance when you contrast support to confidence such as the ratio

$$\frac{\mathrm{P(course\ taken = DATA301 | course\ taken = DATA201)}}{\mathrm{P(course\ taken = DATA301)}}$$

EXAMPLE

TID	Items
T1	i1, i2, i5
T2	i2, i4
T3	i2, i3
T4	i1, i2, i4
T5	i1, i3
T6	i2, i3
T7	i1, i3
T8	i1, i2, i3, i5
T9	i1, i2, i3

So, the minimum support count is 2, and the minimum confidence is 60%.

Step 1

Itemset	Support
i1	6
i2	7
i3	6
i4	2
i5	2

Step 2

Itemset	Support
i1, i2	4
i1, i3	4
i1, i4	1
i1, i5	2
i2, i3	4
i2, i4	2
i2, i5	2
i3, i4	0
i3, i5	1
i4, i5	0

Step 3

Itemset	Support
i1, i2, i3	2
i1, i2, i5	2

EXAMPLE

By using the principle that

Confidence(A->B)=Support_count(A∩B)/Support_count(A)

where **A->B** means **if A then B,** we have the following association rules listed in the order of the **Lift**:

- Confidence[(i1 and i2)->(i3)] = Support(i1 and i2 and i3)/Support(i1 and i2) = 2/4 x 100 = 50%
- Confidence[(i1 and i3)->(i2)] = Support(i1 and i2 and i3)/Support(i1 and i3) = 2/4 x 100 = 50%
- Confidence[(i2 and i3)->(i1)] = Support(i1 and i2 and i3)/Support(i2 and i3) = 2/4 x 100 = 50%
- Confidence[(i1) -> (i2 and i3)] = Support(i1 and i2 and i3)/Support(i1) = 2/6 x 100 = 33%
- Confidence[(i2) -> (i1 and i3)] = Support(i1 and i2 and i3)/Support(i2) = 2/7 x 100 = 28%
- Confidence[(i3) -> (i1 and i2)] = Support(i1 and i2 and i3)/Support(i3) = 2/6 x 100 = 33%

Thus, the first three rules can be considered as the most important ones or have some significant relevance in this data set.

THE APRIORI ALGORITHM STEPS

i. Choose a minimum support and confidence: very important for large datasets with many observations.

ii. Take all possible subsets that meet minimum support.

iii. Calculate all confidence with those subsets; retain those that meet minimum confidence.

iv. Report remaining rules (generally in order with highest lift first).

Critical Thinking/Discussion Prompt:

How do we decide whether **Item A => Item B** or rather **Item B => Item A** ?

Compare:

$$\mathcal{P}\left(B|A\right) \text{ vs } \mathcal{P}\left(A|B\right)$$

ANOTHER ALGORITHM: ECLAT

ECLAT is an acronym for Equivalence class Clustering and bottom-up Lattice Traversal.

Main Idea: use Transaction Id Sets (tidsets) intersections to compute the support value of a candidate.

How it works:

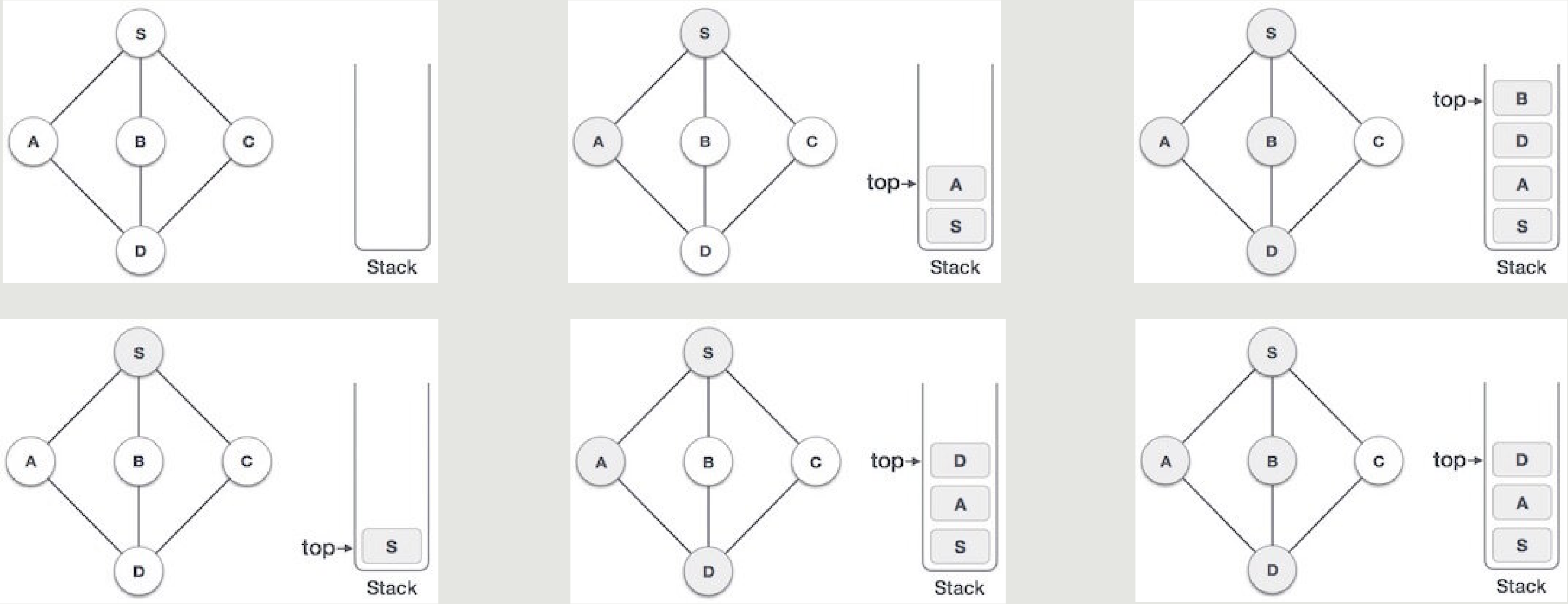
1. Uses a vertical data representation to organize transactions.
2. A depth first search is used to find item sets.
3. Uses equivalence classes to find patterns.
4. Filters out pairs that don't meet minimum support.
5. Intersects transaction lists to determine support for item sets.

DEPTH FIRST SEARCH

1. Visit the adjacent unvisited vertex. Mark it as visited. Display it. Push it in a stack.

2. If no adjacent vertex is found, pop up a vertex from the stack. (It will pop up all the vertices from the stack that do not have adjacent vertices.)

3. Repeat Rule 1 and Rule 2 until the stack is empty.

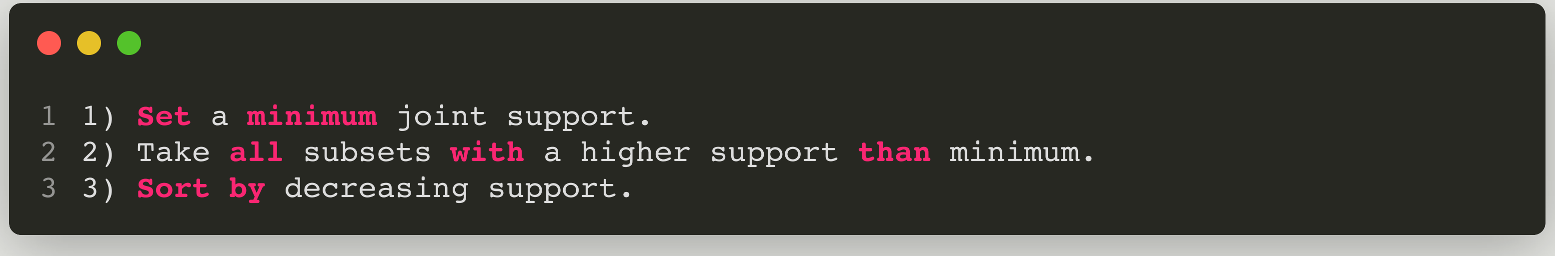


Critical thinking: What happens next?

Source: https://www.tutorialspoint.com/data_structures_algorithms/depth_first_traversal.htm

ECLAT

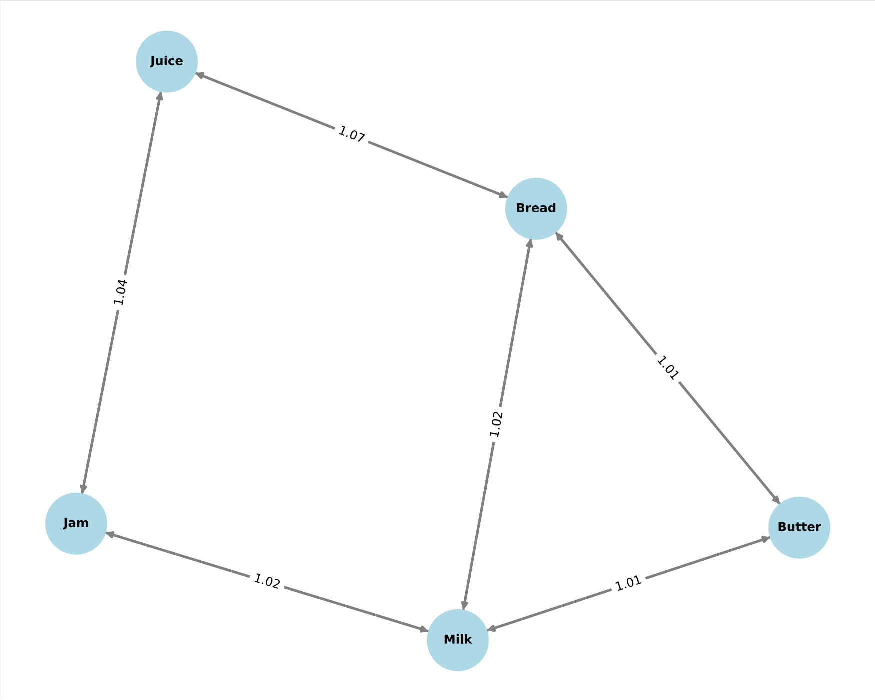
ECLAT steps:



```
1) Set a minimum joint support.
2) Take all subsets with a higher support than minimum.
3) Sort by decreasing support.
```

Example: Frequently bought items in a grocery store.

TID	Bread	Butter	Milk	Juice	Jam
T1	1	1	0	0	1
T2	0	1	0	1	0
T3	0	1	1	0	0
T4	1	1	0	1	0
T5	1	0	1	0	0
T6	0	1	1	0	0
T7	1	0	1	0	0
T8	1	1	1	0	1
T9	1	1	1	0	0



ECLAT

k = 1 and minimum support = 2

Item	TIDSET
Bread	{T1, T4, T5, T7, T8, T9}
Butter	{T1, T2, T3, T4, T6, T8, T9}
Milk	{T3, T5, T6, T7, T8, T9}
Juice	{T2, T4}
Jam	{T1, T8}

k = 2

ITEM	TIDSET
{Bread, Butter}	{T1, T4, T8, T9}
{Bread, Milk}	{T5, T7, T8, T9}
{Bread, Juice}	{T4}
{Bread, Jam}	{T1, T8}
{Butter, Milk}	{T3, T6, T8, T9}
{Butter, Juice}	{T2, T4}
{Butter, Jam}	{T1, T8}
{Milk, Jam}	{T8}

k = 3

ITEM	TIDSET
{Bread, Butter, Milk}	{T8, T9}
{Bread, Butter, Jam}	{T1, T8}

k = 4 only one transaction:

ITEM:{Bread, Butter, Milk and Jam}

TIDSET:{T8}

ECLAT

The minimum support is 2 and we infer the following associations:

ITEMS BOUGHT	RECOMMENDED PRODUCTS
Bread	Butter
Bread	Milk
Bread	Jam
Butter	Milk
Butter	Jam
Bread and Butter	Milk
Bread and Butter	Jam