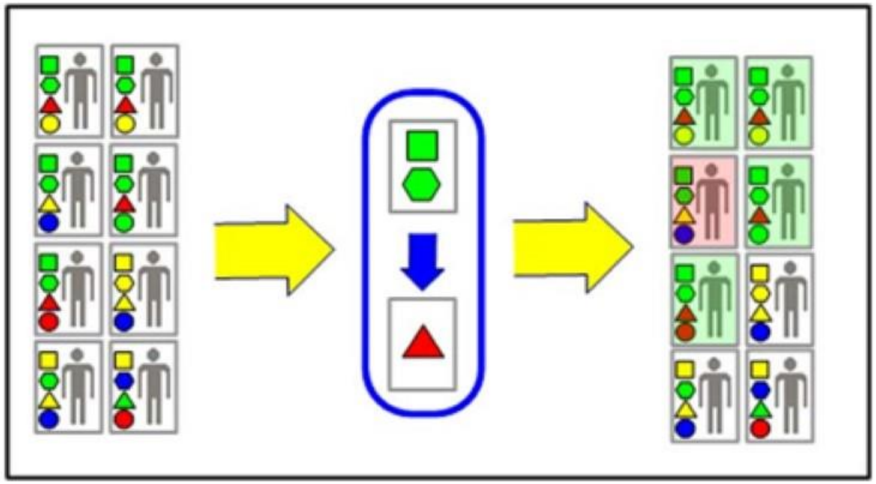
Association rule mining

Outline

- Basket Analysis,
- Frequent Itemset,
- Rules Generation,
- Rules Evaluation,
- Libraries,
- Mlxtend API,
- Exercises.

Basket Analysis



1. What are the patterns among customer purchases (events)?
2. Creation of rules in a form $X \rightarrow Y$ (antecedants $\rightarrow$ consequents) where X and Y are *disjoint* sets.
3. {Bread} $\rightarrow$ {Butter} people who buy bread are also likely to buy butter.
4. Purpose: promotion campaigns, shop organization, etc.
5. Rule mining contains **two steps**: frequent itemset generation and (based on that), the final rule creation.

Frequent Itemset

1. Set of sets containing products (events) that often co-occur together.
2. An itemset is "frequent" if it meets a user-specified support threshold. If the **support** threshold is set to 0.5 (50%), a frequent itemset is a set of items that occur together in at least 50% of all transactions in the database.
3. Frequent itemsets can be generated by Apriori method.

Transaction ID	Items Bought
2000	A,B,C
1000	A,C
4000	A,D
5000	B,E,F

Min. support 50%

Frequent Itemset	Support
{A}	75%
{B}	50%
{C}	50%
{A,C}	50%

Generation of association rules

1.Based on frequent itemsets the rules are being generated.

2.After rules generation, the chosen measure evaluates them (confidence, lift, or support again).

3.Most often: support is first used to find frequent (significant) itemsets, hen confidence is used in a second step to produce rules from the frequent itemsets that exceed a min. confidence threshold.

If {A,B,C,D} is a frequent itemset, candidate rules:

$ABC \rightarrow D$,	$ABD \rightarrow C$,	$ACD \rightarrow B$,	$BCD \rightarrow A$,
$A \rightarrow BCD$,	$B \rightarrow ACD$,	$C \rightarrow ABD$,	$D \rightarrow ABC$
$AB \rightarrow CD$,	$AC \rightarrow BD$,	$AD \rightarrow BC$,	$BC \rightarrow AD$,
$BD \rightarrow AC$,	$CD \rightarrow AB$,		

4.If the frec lso frequent, so rules like $A \rightarrow B$ are also generated but not using this set.

5.Frequent itemsets of size 1 are not considered.

Rules evaluation - measures

1.Support - frequency of rule/itemset, the probability of seeing such a rule/itemset in the database. ***D*** *- database (set of transactions),* ***t*** *- transaction,* ***A*** *- itemset.*

$$support\left(A\right)=\frac{\left|\left\{t\in D;A\subseteq t\right\}\right|}{\left|D\right|}$$

$$\text{support}(A \rightarrow C) = \text{support}(A \cup C), \quad \text{range: } [0, 1]$$

2.Confidence - the probability of seeing C in a transaction given that it also contains A. Antisymmetric. Commonly used in rule mining as a threshold. Sensitive to the frequency of A. The higher, the better.

$$\text{confidence}(A \rightarrow C) = \frac{\text{support}(A \rightarrow C)}{\text{support}(A)}, \quad \text{range: } [0, 1]$$

3.Lift - how often A and C occur together when we assume that they are statistically independent. If A and C are independent, the Lift score will be exactly 1. Symmetric. The higher, the better.

$$\text{lift}(A \rightarrow C) = \frac{\text{confidence}(A \rightarrow C)}{\text{support}(C)}, \quad \text{range: } [0, \infty]$$

Rules evaluation - example

Transaction 1	🍎 🍺 🍚 🍗
Transaction 2	🍎 🍺 🍚
Transaction 3	🍎 🍺
Transaction 4	🍎 🍐
Transaction 5	🍼 🍺 🍚 🍗
Transaction 6	🍼 🍺 🍚
Transaction 7	🍼 🍺
Transaction 8	🍼 🍐

Support {🍎} = $\frac{4}{8}$ = 0.5

port{Beer} = ?

port{Apple, Beer} = ?

Confidence {🍎 → 🍺} = $\frac{\text{Support \{🍎, 🍺\}}}{\text{Support \{🍎\}}}$

Confidence{Apple → Beer} = **?**

Lift {🍎 → 🍺} = $\frac{\text{Support \{🍎, 🍺\}}}{\text{Support \{🍎\} x Support \{🍺\}}}$

Lift{Apple → Beer} = **?**

Rules evaluation - example

Transaction 1	🍎 🍺 🍚 🍗
Transaction 2	🍎 🍺 🍚
Transaction 3	🍎 🍺
Transaction 4	🍎 🍐
Transaction 5	🍼 🍺 🍚 🍗
Transaction 6	🍼 🍺 🍚
Transaction 7	🍼 🍺
Transaction 8	🍼 🍐

Support {🍎} = $\frac{4}{8}$ = 0.5

port{Beer} = 6/8 = **0.75**

port{Apple, Beer} = 3/8 = **0.375**

Confidence {🍎 → 🍺} = $\frac{\text{Support \{🍎, 🍺\}}}{\text{Support \{🍎\}}}$

Confidence{Apple → Beer} =
= 0.375/0.5 = **0.75**

Lift {🍎 → 🍺} = $\frac{\text{Support \{🍎, 🍺\}}}{\text{Support \{🍎\} x Support \{🍺\}}}$

Lift{Apple → Beer} = 0.375/(0.5*0.75)=
= **1**

`mlxtend.frequent_patterns.apriori`

1. `from mlxtend.frequent_patterns import apriori.`
2. Works on pandas `DataFrame` – helps in result manipulation.
3. The data has to be One-hot encoded.

API

apriori(df, min_support=0.5, use_colnames=False)

Get frequent itemsets from a one-hot DataFrame **Parameters**

- `df` : pandas DataFrame

  pandas DataFrame in one-hot encoded format. For example Apple Bananas Beer Chicken Milk Rice 0 1 0 1 1 0 1 1 1 0 1 0 0 1 2 1 0 1 0 0 0 3 1 1 0 0 0 0 4 0 0 1 1 1 1 5 0 0 1 0 1 1 6 0 0 1 0 1 0 7 1 1 0 0 0 0

- `min_support` : float (default: 0.5)

  A float between 0 and 1 for minumum support of the itemsets returned. The support is computed as the fraction transactions_where_item(s)_occur / total_transactions.

- `use_colnames` : bool (default: False)

  If true, uses the DataFrames' column names in the returned DataFrame instead of column indices.
  Returns

pandas DataFrame with columns ['support', 'itemsets'] of all itemsets that are >= min_support.

mlxtend.frequent_patterns.association_rules

1. Generates association rules from frequent itemset, providing several measures: **support, confidence, lift**, leverage, conviction (not working properly).
2. Input - Pandas dataframes of frequent itemsets, returned by the `apriori` method.

API

association_rules(df, metric='confidence', min_threshold=0.8)

Generates a DataFrame of association rules including the metrics 'score', 'confidence', and 'lift'

Parameters

- `df` : pandas DataFrame

  pandas DataFrame of frequent itemsets with columns ['support', 'itemsets']

- `metric` : string (default: 'confidence')

  Metric to evaluate if a rule is of interest. Supported metrics are 'support', 'confidence', 'lift', 'leverage', and 'conviction' These metrics are computed as follows: - support(A->C) = support(A+C) [aka 'support'], range: [0, 1] - confidence(A->C) = support(A+C) / support(A), range: [0, 1] - lift(A->C) = confidence(A->C) / support(C), range: [0, inf] - leverage(A->C) = support(A->C) - support(A)*support(C), range: [-1, 1] - conviction = [1 - support(C)] / [1 - confidence(A->C)], range: [0, inf]

- `min_threshold` : float (default: 0.8)

  Minimal threshold for the evaluation metric to decide whether a candidate rule is of interest.

Returns

pandas DataFrame with columns "antecedent support", "consequent support", "support", "confidence", "lift", "leverage", "conviction" of all rules for which metric(rule) >= min_threshold.