Association Rule

General Concepts

Association Rules: Key Concepts

Key Concept	DESCRIPTION
Items	Basic units of analysis, often representing products, features, or events.
Transaction	Groups of items occurring together, like a grocery basket or website visit session.
Itemset	A sub-group of items within a transaction
Frequent itemset	An itemset occurring in a significant percentage of transactions (above a minimum support threshold).
Association Rule	A rule linking two itemsets (antecedent and consequent), typically with measures like support, confidence, and lift.

Ref:

Association Rules: Key Metrics

Task	DESCRIPTION	Formula	Range
Support	Percentage of transactions containing the itemset. Indicates how frequent an itemset is in all transaction.	$\text{support}(x) = \frac{\text{transaction containing } x}{\text{total transactions}}$ $\text{support}(x \rightarrow y) = \frac{\text{transaction containing x and y}}{\text{total transactions}}$	range: [0,1]
Confidence	Measures the likelihood that item Y is bought when item X is bought. X is antecedent and Y is consequent The confidence of a rule x -> y is the probability of seeing the consequent in a transaction given that it also contains the antecedent. Note that the metric is not symmetric or directed; for instance, the confidence for x -> y is different than the confidence for y->x.	$confidence(x \rightarrow y) = \frac{\text{support}(x \cup y)}{\text{support}(x)}$	range: [0,1]
Lift	Indicates the strength of the association between items. If they were statistically independent. • **Lift=1**: The probability of occurrence of antecedent and consequent is independent of each other. • **Lift > 1**: It determines the degree to which the two item-sets are dependent to each other. • **Lift < 1**: It tells us that one item is a substitute for other items, which means one item has a negative effect on another.	$lift(x \rightarrow y) = \frac{\text{support}(x \cup y)}{\text{support}(x) * \text{support}(y)}$	range: [0,∞]
Leverage	Measures the difference between the observed support of XUY and the expected support if X and Y were independent. A leverage value of 0 indicates independence.	$\text{leverage}(x \rightarrow y) = \text{support}(x \cup y) - (\text{support}(x) * \text{support}(y))$	range: [−1,1]
Conviction	Measures the degree to which the presence of X increases the probability of Y. Inf means highly dependent. Similar to lift, if items are independent, the conviction is 1	$conviction(x \rightarrow y) = \frac{1 - \text{support}(y)}{1 - \text{confidence}(x \rightarrow y)}$	range: [0,∞]

Ref: https://rasbt.github.io/mlxtend/user_guide/frequent_patterns/association_rules/

Association Rules: Itemsets Example

Transaction ID	Items
1	A, B, C, D, E
2	A, B, C, F
3	A, B, D, E
4	A, B, C, D, E, F
5	B, C, E, F

1-Itemsets Support

Itemset	Support
{A}	4
{B}	5
{C}	4
{D}	3
{E}	4
{F}	3

2-Itemsets Support

Itemset	Support
{A, B}	4
{A, C}	3
{A, D}	2
{A, E}	3
{A, F}	2
{B, C}	4
{B, D}	3
{B, E}	4
{B, F}	3
{C, D}	2
{C, E}	3
{C, F}	3
{D, E}	3
{D, F}	1
{E, F}	3

3-Itemsets Support

Itemset	Support
{A, B, C}	3
{A, B, D}	2
{A, B, E}	3
{A, B, F}	2
{A, C, D}	2
{A, C, E}	2
{A, C, F}	1
{A, D, E}	2
{A, D, F}	1
{A, E, F}	1
{B, C, D}	2
{B, C, E}	3
{B, C, F}	3
{B, D, E}	2
{B, D, F}	1
{B, E, F}	3
{C, D, E}	2
{C, D, F}	1
{C, E, F}	2
{D, E, F}	1

4-Itemsets Support

Itemset	Support
{A, B, C, D}	2
{A, B, C, E}	2
{A, B, C, F}	1
{A, B, D, E}	2
{A, B, D, F}	1
{A, B, E, F}	1
{A, C, D, E}	1
{A, C, D, F}	1
{A, C, E, F}	1
{A, D, E, F}	1
{B, C, D, E}	2
{B, C, D, F}	1
{B, C, E, F}	2
{B, D, E, F}	1
{C, D, E, F}	1

5-Itemsets Support

Itemset	Support
{A, B, C, D, E}	2
{A, B, C, D, F}	1
{A, B, C, E, F}	1
{A, B, D, E, F}	1
{B, C, D, E, F}	1

6-Itemsets Support

Itemset	Support
{A, B, C, D, E, F}	1

Association Rule

types of Itemset Mining

Association Rules: Frequent Itemset Mining (FIM)

Advantages of Frequent Itemset Mining

1. **Discovery of Patterns and Insights**
    - Reveals hidden relationships among items in large datasets, enabling actionable insights for decision-making
    - Identifies frequently co-occurring items, useful in applications like market basket analysis and recommendation systems

2. **Scalability to Large Datasets**
    - Modern algorithms (e.g., Apriori, FP-Growth) efficiently handle vast amounts of data by pruning irrelevant Itemsets
    - Reduces the search space by leveraging concepts like support thresholds and itemset closure

3. **Application Diversity -** Widely applicable across various domains like retail (basket analysis), web usage mining, bioinformatics (gene association), and fraud detection

4. **Data Compression -** Frequent Itemsets summarize large datasets by capturing the most relevant associations, reducing the need to examine the entire data repeatedly

5. **Basis for Advanced Techniques -** Serves as a foundation for other data mining tasks, such as association rule mining, sequence mining, and clustering

6. **Customizability -** Support thresholds allow users to customize the frequency criterion to their dataset or business needs, focusing on meaningful patterns

7. **Improved Decision-Making -** Provides critical information for strategy formulation, such as identifying frequently purchased product combinations for cross-selling

Disadvantages of Frequent Itemset Mining

1. **High Computational Cost -** Generating all frequent itemsets can be computationally expensive, especially for dense datasets with a low support threshold.

2. **Exponential Growth of Candidates -** The number of possible itemsets grows exponentially with the number of items, making the process resource-intensive.

3. **Difficulty with Rare Patterns -** Rare but valuable patterns are often missed due to high support thresholds.

4. **Redundancy in Results -** Generates many redundant itemsets (e.g., subsets of larger frequent itemsets), which may not provide new information.

5. **Parameter Sensitivity -** The results heavily depend on the chosen support threshold. Setting it too high might miss valuable patterns; setting it too low may overwhelm with irrelevant ones.

6. **Scalability Challenges for High-Dimensional Data -** Mining datasets with a high number of unique items (e.g., text data or bioinformatics) can be challenging due to the combinatorial explosion of itemsets.

7. **Loss of Temporal or Sequential Information -** Standard frequent itemset mining does not consider the order of transactions, limiting its application in temporal or sequence-sensitive contexts.

8. **Overfitting and Noise -** Patterns may reflect noise or overfit the specific dataset, leading to irrelevant or non-generalizable results.

9. **Interpretability Issues -** The sheer volume of frequent itemsets in large datasets can overwhelm users, making it difficult to interpret and prioritize actionable insights.

10. **Dependence on Data Quality -** Requires clean, well-processed data; otherwise, results may be inaccurate or misleading.

Association Rules: Frequent Itemset Mining (FIM)

Transaction ID	Items
1	A, B, C, D, E
2	A, B, C, F
3	A, B, D, E
4	A, B, C, D, E, F
5	B, C, E, F

Algorithms

- Apriori
- FP-Growth

Some of the frequent itemset with minimum support 0.6, meaning occur 3 out of 5 transaction.

Frequent Itemset	Occurrence	Support
{'B'}	5	1
{'E'}	4	0.8
{'C'}	4	0.8
{'A'}	4	0.8
{'E', 'B'}	4	0.8
{'C', 'B'}	4	0.8
{'A', 'B'}	4	0.8
{'D'}	3	0.6
{'D', 'E'}	3	0.6
{'D', 'B'}	3	0.6
{'A', 'D'}	3	0.6
{'E', 'C'}	3	0.6
{'A', 'E'}	3	0.6
{'A', 'C'}	3	0.6
{'D', 'E', 'B'}	3	0.6
{'A', 'D', 'E'}	3	0.6
{'A', 'D', 'B'}	3	0.6
{'C', 'E', 'B'}	3	0.6
{'A', 'E', 'B'}	3	0.6
{'C', 'A', 'B'}	3	0.6
{'A', 'D', 'E', 'B'}	3	0.6
{'F'}	3	0.6
{'F', 'C'}	3	0.6
{'F', 'B'}	3	0.6
{'F', 'B', 'C'}	3	0.6

Association Rules: Maximal Frequent Itemset Mining (MFIM)

Transaction ID	Items
1	A, B, C, D, E
2	A, B, C, F
3	A, B, D, E
4	A, B, C, D, E, F
5	B, C, E, F

minimum support threshold be 2 transactions (40%)

Frequent Itemset	Occurrence	Support
{A, B, C, D, E}	2	0.4
{A, B, C, F}	2	0.4
{B, C, E, F}	2	0.4

A **Maximal Frequent Itemset** is a frequent itemset that is **not a subset** of any other frequent itemset. Max patterns provide a more condensed and focused view of the most significant patterns in the dataset.

Advantages

- **Compact Representation -** Reduces the size of the frequent itemset output significantly compared to listing all frequent itemsets.
- **Improved Efficiency -** Reduces computation and storage costs since only the largest patterns are stored and analysed.
- **Simplified Rules -** Useful for high-level pattern analysis without diving into detailed subsets

Disadvantages

- **Loss of Subset Information:** Does not explicitly provide information about smaller frequent subsets, which may still be useful for specific applications.
- **Post-Processing Complexity:** If smaller subsets are required later, additional computations are necessary.
- **Not Suitable for All Applications:** In some domains (e.g., recommendation systems), the detailed structure of frequent subsets is more valuable than just the largest patterns.

Algorithms

- FP-Max

Association Rules: Closed Frequent Itemset Mining (CFIM)

Transaction ID	Items
1	A, B, C, D, E
2	A, B, C, F
3	A, B, D, E
4	A, B, C, D, E, F
5	B, C, E, F

All below Itemset are closed, because no superset has the same support.

Frequent Itemset	Occurrence	Support
{A, C}	3	0.6
{B, C}	4	0.8
{A, C, D}	2	0.4

Closed patterns are frequent itemsets that cannot be extended by adding any more items without decreasing their support. These patterns offer a compact representation of frequent itemsets by eliminating redundant combinations.

Advantages

- **Reduced Output Size:** Compared to listing all frequent itemsets, closed frequent itemsets significantly reduce redundancy.
- **Preserves Frequency Information:** Despite reducing redundancy, all necessary support information is retained.
- **Efficient for Rule Generation:** Association rules can be derived directly from closed frequent itemsets.
- **Scalable:** Works well for datasets with a high number of frequent itemsets, where full frequent mining would be computationally expensive.

Disadvantages

- **Loss of Granular Information:** Smaller subsets that are frequent but redundant (with the same support as a closed itemset) are not explicitly listed.
- **Post-Processing Complexity:** If detailed frequent itemsets are needed later, additional processing is required.
- **Algorithm Complexity:** Identifying "closed" itemsets can add overhead to the mining process compared to simple frequent itemset mining.
- **Not Suitable for All Use Cases:** Domains that rely on all frequent subsets for analysis (e.g., recommendation systems) may find this representation insufficient.

Algorithms

- Close
- CHARM

Association Rules: Multi Level Itemset Mining (MLIM)

Transaction ID	Item	Location
1	Apple	A
1	Banana	B
2	Apple	A
2	Banana	B
3	Apple	A
3	Banana	B
3	Orange	A
4	Apple	B
4	Banana	A
4	Orange	B

Multi-Level Itemset Mining is an extension of traditional frequent itemset mining, where the items in the dataset are organized into a hierarchical taxonomy or multiple levels of granularity (e.g., categories, subcategories, or attributes). The goal is to discover frequent itemsets at various levels of abstraction.

Key Features
- **Hierarchical Taxonomy**: Items are structured in a hierarchy.
- **Granularity**: Enables mining patterns at multiple levels of detail (e.g., broader categories or specific items).
- **Support Thresholds**: Different support thresholds can be applied at different levels to find meaningful patterns.
- **Drill-Down and Roll-Up**: Patterns can be generalized (roll-up) or specified (drill-down) depending on the level.

Advantages
- **Broader Insights**: Reveals patterns across multiple levels of abstraction, providing a comprehensive view.
- **Relevance at Different Levels**: Useful in domains where both high-level (categories) and low-level (specific items) patterns matter.

Disadvantages
- **Complexity:** Mining multiple levels increases computational and algorithmic complexity.
- **Parameter Sensitivity**: Setting appropriate support thresholds for each level is challenging and domain-specific.
- **Redundant Patterns**: High-level patterns may overlap with low-level patterns, making it hard to interpret results.
- **Data Requirements**: Requires well-defined taxonomies or hierarchies for items, which might not always be available.

Minimum Support: 0.2

support	itemsets
0.75	((Apple, A))
0.25	((Apple, B))
0.25	((Banana, A))
0.75	((Banana, B))
0.25	((Orange, A))
0.25	((Orange, B))
0.75	((Banana, B), (Apple, A))
0.25	((Orange, A), (Apple, A))
0.25	((Banana, A), (Apple, B))
0.25	((Apple, B), (Orange, B))
0.25	((Banana, A), (Orange, B))
0.25	((Orange, A), (Banana, B))
0.25	((Orange, A), (Banana, B), (Apple, A))
0.25	((Banana, A), (Apple, B), (Orange, B))

Algorithms
- Multi-Level Apriori
- Multi-Level FP-Growth

Association Rules: High Utility Itemset Mining (HUIM)

Transaction ID	Items Purchased	Quantity	Unit Price	Utility (Quantity × Price)
1	A, B, C	2, 3, 1	$5, $2, $3	$10, $6, $3
2	A, C, D	1, 2, 1	$5, $3, $4	$5, $6, $4
3	B, C, D	2, 1, 2	$2, $3, $4	$4, $3, $8
4	A, B	1, 2	$5, $2	$5, $4
5	C, D	3, 1	$3, $4	$9, $4

Minimum Utility Threshold (MUT): $10. Itemsets with value greater then MUT are selected

High Utility Itemset	Utility
{C, D}	$34
{A, B}	$25
{A, C}	$24
{C}	$21
{A}	$20

Calculate the utility of each itemset:

- The utility of an itemset is the sum of the utility values (Quantity × Price) of the items in that itemset across all transactions where the itemset appears.

High Utility Itemset Mining focuses on discovering **itemsets** (sets of items) that contribute significantly to a business objective, such as profit, revenue, or other utility measures. Unlike traditional frequent itemset mining, HUIM considers the **weight** (utility) of items (e.g., price, quantity, or importance) and not just their occurrence.

Difference from Frequent Itemset Mining:

- Frequent itemsets focus only on how often items appear together.
- HUIM considers both frequency and **utility values**

Advantage

- **Real-World Relevance:** HUIM aligns closely with real-world applications, focusing on value rather than frequency alone.
- **Profit Maximization:** Useful for businesses to identify the most profitable item combinations.
- **Flexible Utility Models:** Can incorporate various utility measures such as revenue, satisfaction, or importance.
- **Scalability:** Modern algorithms like HUI-Miner and EFIM make mining large datasets feasible.

Disadvantages

- **Increased Complexity:** Mining high utility itemsets is computationally more expensive than traditional frequent itemset mining.
- **Sensitive to Utility Threshold:** Selecting an appropriate utility threshold can be challenging and affects the results significantly.
- **No Anti-Monotonicity Property:** Unlike frequent itemset mining, HUIM does not follow the downward-closure property (i.e., if an itemset is not high utility, its subsets may still be high utility).
- **Handling Negative Utility:** Dealing with negative or conflicting utilities adds to the complexity.

Algorithms

- Two-Phase
- UPGrowth
- HUI-Miner
- FHM
- EFIM

Association Rules: High Average Utility Itemset Mining (HUIM)

Transaction ID	Items Purchased	Quantity	Unit Price	Utility (Quantity × Price)	Total Utility
1	A, B, C	2, 3, 1	$5, $2, $3	$10, $6, $3	$19
2	A, C, D	1, 2, 1	$5, $3, $4	$5, $6, $4	$15
3	B, C, D	2, 1, 2	$2, $3, $4	$4, $3, $8	$15
4	A, B	1, 2	$5, $2	$5, $4	$9
5	C, D	3, 1	$3, $4	$9, $4	$13

Minimum Utility Threshold (MUT): $20

Itemset	Total Utility	Average Utility (Total Utility / Size of Itemset)
{A}	30	30
{A, C}	51	25.5
{A, B}	50	25
{A, B, C}	71	23.67
{A, D}	46	23
{A, C, D}	67	22.33
{A, B, D}	66	22
{A, B, C, D}	87	21.75
{C}	21	21
{B, C}	41	20.5

High Average Utility Itemset Mining (HAUIM) is a specialized technique in data mining that identifies itemsets with high average utility.

Key Features
- **Normalized by Itemset Size**: Average utility is calculated as the total utility of the itemset divided by the number of items in the itemset.
- **Utility-Oriented**: Focuses on mining itemsets that maximize the average utility (e.g., profit or significance) rather than just frequency

Advantage
- **Real-World Relevance**: Identifies patterns that are economically or operationally valuable rather than simply frequent.
- **Concise Patterns**: Promotes smaller, high-value itemsets by normalizing utility by size.
- **Applicable to Diverse Domains**: Useful in retail, e-commerce, healthcare, and financial analysis.
- **Overcomes Frequent Itemset Limitations**: Avoids patterns that are frequent but of low utility.

Disadvantages
- **Computational Complexity**: Requires additional calculations for utility and average utility, making it computationally intensive.
- **Parameter Sensitivity**: Results depend on threshold settings (e.g., minimum utility), which may need domain-specific tuning.
- **Complexity of Preprocessing**: Assigning utilities and preparing the data may require significant preprocessing.
- **Scalability Issues**: Handling very large datasets with high utility complexity can be challenging.

Algorithms
- UPGrowth (Need to be adapted)
- HAUI-Miner
- FHM
- EFIM (Need to be adapted)

Association Rules: Rare Itemset Mining (RIM)

Transaction ID	Items
1	A, B, C, D, E
2	A, B, C, F
3	A, B, D, E
4	A, B, C, D, E, F
5	B, C, E, F

Focuses on discovering **infrequent itemsets** in a transactional dataset. Itemsets that occur less frequently than a specified **maximum support threshold** but are still significant in specific contexts.

Advantages

- **Anomaly Detection:** Helps identify anomalies or outliers, which are often overlooked in frequent itemset mining.
- **Niche Insights:** Reveals insights about rare behaviours or patterns that may be of significant value.
- **Applications in Critical Domains:** Useful in fields like fraud detection, rare disease analysis, and intrusion detection.
- **Complements Frequent Mining:** Provides a broader understanding of data by uncovering rare patterns alongside frequent ones.

Disadvantages

- **Sparse Data:** Rare itemsets often appear in sparse datasets, making them harder to identify and analyse.
- **High Computational Complexity:** Mining rare itemsets requires more computational resources due to the larger search space.
- **Lack of Anti-Monotonicity:** Rare itemsets do not follow the downward-closure property (i.e., subsets of rare itemsets might not be rare), leading to additional challenges in pruning the search space.
- **Potential Overfitting:** Focusing too much on rare patterns can lead to overemphasis on noise or insignificant patterns.

Maximum Support Threshold, 2 i.e. itemset in at most 2 transactions
Sample itemsets

Itemset	Support
{D, C}	2
{C, A}	2
{B, A}	2
{E, A}	2
{D, A, C}	2
{D, B, A}	2

Algorithms

- Apriori-Inverse
- Rare-Itemset-Miner

Association Rules: Negative Itemset Mining (RIM) - TBD

Transaction ID	Items
1	A, B, C, D, E
2	A, B, C, F
3	A, B, D, E
4	A, B, C, D, E, F
5	B, C, E, F

- When item A is purchased, F is rarely purchased.

Negative Itemset Mining focuses on discovering **negative associations** or **correlations** between items in a dataset. This is useful for identifying cases where the absence of one item is correlated with the presence of another. If customers buy A, they avoid buying F. If C is not purchased, then D is often purchased.

Advantages

- **Unveils Complex Patterns:** Identifies relationships that traditional frequent itemset mining might overlook.
- **Actionable Insights:** Useful for decision-making, such as recommending alternative products or detecting product cannibalization.
- **Enhanced Marketing Strategies:** Helps create bundles or discounts for negatively associated items.
- **Identifies Substitutes:** Reveals substitutive relationships between products or services.

Disadvantages

- **Complexity:** Computationally intensive due to the need to consider both presence and absence of items.
- **Interpretation Challenges:** Negative associations can be harder to interpret and act upon.
- **Data Dependence**: Requires a well-preprocessed dataset to ensure reliable results.
- **Scalability Issues:** For large datasets, mining negative itemsets can become infeasible.

Algorithms

- Negative Association Rule Mining with Apriori (NARM-Apriori)
- FP-Growth with Negative Rules

Association Rules: Generalized Sequential Pattern Mining (GSPM)

Transaction ID	Items
1	A, B, C, D, E
2	A, B, C, F
3	A, B, D, E
4	A, B, C, D, E, F
5	B, C, E, F

Minimum Support: 2

Sequential Patterns:

- {A → B} (Support = 4)
- {A → C} (Support = 3)
- {B → C → E} (Support = 2)
- {A → B → D → E} (Support = 2)

Sequential pattern mining involves discovering patterns that capture the temporal order of events or items in a sequence. Applicable in various domains such as time-series analysis, web clickstreams, and customer behaviour tracking.
Unlike basic sequence mining, GSPM includes generalizations such as constraints (time gaps, taxonomy, etc.) and handles more complex relationships among items or events.

Key Features

- **Sequence Discovery**: Finds patterns in ordered sequences, such as customer purchase histories, web clickstreams, or medical records.
- **Incorporates Constraints**: Time constraints, item taxonomy, or minimum/maximum gap constraints between items in sequences.
- **Flexibility**: Can discover patterns in diverse domains like retail, bioinformatics, or web analytics.

Advantages

- **Rich Information Extraction:** Captures sequential patterns that reveal customer behavior, trends, or workflows.
- **Application Versatility:** Applicable in various fields, including marketing, healthcare, and manufacturing.

Disadvantages

- **High Computational Cost:** Complex algorithms may struggle with large datasets or long sequences.
- **Difficulty in Parameter Tuning:** Requires careful tuning of parameters like minimum support, time gap, or pattern length.
- **Risk of Overfitting:** May generate too many patterns, including less meaningful ones.
- **Complex Implementation:** Constraints like time gaps or hierarchies increase the complexity of implementation.

Algorithms

- Apriori-based Approaches
  - GSP
  - SPADE
- Pattern-Growth-based Approaches
  - FreeSpan
  - PrefixSpan

Association Rules: Periodic Pattern Mining (PPM)

Transaction ID	Timestamp	Items Purchased
1	Day 1	A, B
2	Day 2	A, C
3	Day 3	A, B
4	Day 4	A, B, C
5	Day 5	A, B

Patterns with a periodicity of 2 days

Patterns:
- {A → B}: Periodicity = 2 days, Support = 4.
- {A → C}: Periodicity = 3 days, Support = 2.

Periodic Pattern Mining focuses on discovering patterns that occur regularly or periodically in a dataset. It is particularly useful in scenarios where periodicity plays a crucial role, such as time-series data or event-driven systems.

Key Features
- **Periodicity**: identifies patterns with a consistent recurrence interval or approximate periodic behaviour.
- **Threshold-Based**: Patterns are identified if they meet minimum support and periodicity constraints.
- **Event-Based**: Useful for event-driven data, such as system logs, customer behaviour, or biological cycles.

Advantages
- **Pattern Regularity:** Detects repeating patterns, providing insights into recurring trends and events.
- **Scalability**: Many algorithms are optimized for handling large datasets efficiently.
- **Time-Aware**: Works effectively with time-series data by utilizing temporal relationships.

Disadvantages
- **Parameter Sensitivity**: Requires careful tuning of parameters such as minimum periodicity, support, or noise tolerance.
- **Data Sparsity**: Performance and accuracy can degrade in sparse datasets with irregular occurrences.
- **Complexity**: For datasets with multiple periodic patterns or noise, the algorithms may struggle with high computational costs.
- **Overfitting Risk**: May identify spurious patterns that appear periodic but lack practical relevance.

Algorithms
- Periodic Pattern Mining (PPM)
- PerMiner

Association Rules: Correlated Itemset Mining (CIM)

Transaction ID	Items Purchased
1	A, B, C
2	A, B, D
3	A, C, D
4	A, B, C, D
5	B, C, D

Support > 0.4

Itemset	Support	Lift	Leverage
{A, B}	0.6	0.9375	-0.04
{B, C}	0.8	1.25	0.16
{C, D}	0.8	1.25	0.16
{A, B, C}	0.4	1.042	0.08

- The itemset **{B, C}** is strongly correlated with Lift = 1.25 and Leverage = 0.16.
- The itemset **{A, B}** is weakly correlated with Lift < 1 and negative Leverage.

Correlated Itemset Mining focuses on finding itemsets where items exhibit a strong correlation, rather than simply appearing together frequently. Unlike traditional frequent itemset mining, which considers only support (frequency), correlated itemset mining incorporates statistical measures like **lift**, **confidence**, or **leverage** to ensure that the items are not only frequent but also related.

Key Features

- **Correlation over Frequency**: identifies itemsets with statistically significant relationships, filtering out itemsets that appear together frequently but by chance.
- **Measure-Driven**: Uses metrics such as **lift**, **leverage**, **conviction**, or **all-confidence** to evaluate the strength of the relationship between items.
- **Flexible Thresholds**: Users can define thresholds for correlation measures, making the mining process customizable to specific use cases.
- **Reduced Noise**: Eliminates redundant or uncorrelated itemsets, providing more meaningful results.

Advantages

- **Higher Quality Patterns**: Focuses on significant relationships, leading to actionable insights.
- **Reduced Spurious Associations**: Avoids misleading itemsets that are frequent but not strongly related.

Disadvantages

- **Computational Complexity:** Calculating correlation metrics for all potential itemsets can be resource-intensive, especially for large datasets.
- **Metric Selection**: The choice of correlation metric (e.g., lift, leverage) may influence results and require domain knowledge.
- **Threshold Sensitivity**: Setting appropriate thresholds for correlation measures is challenging and may vary by dataset.
- **Possible Loss of Information**: Itemsets that are frequent but not correlated may still hold value in certain contexts but are ignored.

Algorithms

- COFI
- CMAR (Classification Based on Multiple Association Rules)
- Correlation-Based FP-Growth:

Association Rule

Key Algorithms

Association Rules: Key Algorithms

ALGORITHM	DESCRIPTION & APPLICATION	ADVANTAGES	DISADVANTAGES
Apriori	The Apriori algorithm uses the join and prune step iteratively to identify the most frequent itemset in the given dataset. Prior knowledge (apriori) of frequent itemset properties is used in the process	1. Explainable & interpretable results 2. Exhaustive approach based on the confidence and support.	1. Requires defining the expected number of clusters or mixture components in advance 2. The covariance type needs to be defined for the mixture of component
FP-growth	Frequent Pattern growth (FP-growth) is an improvement on the Apriori algorithm for finding frequent itemsets. It generates a conditional FP-Tree for every item in the data.	1. Explainable & interpretable results 2. Smaller memory footprint than the Apriori algorithm	1. More complex algorithm to build than Apriori 2. Can result in many (incremental) overlapping/trivial itemsets
FP-Max	A variant of Frequent pattern growth that is focused on finding maximal itemsets.	1. Explainable & Interpretable results 2. Smaller memory footprint than the Apriori and FP-growth algorithms	1. More complex algorithm to build than Apriori
Eclat	Equivalence Class Clustering and Bottom-Up Lattice Traversal (Eclat) applies a DepthFirst Search of a graph procedure. This is a more efficient and scalable version of the Apriori algorithm	1. Explainable & interpretable results 2. Computational faster compared to the Apriori algorithm	1. Can provide only a subset of results in contrast to the Apriori algorithm and its variants
Hypergeometric Networks	HNet learns the Association from datasets with mixed data types (discrete and continuous variables) and with unknown functions. Associations are statistically tested using the hypergeometric distribution for finding frequent itemset	1. Explainable & Interpretable results 2. More robust against spurious associations as it uses statistical inferences 3. Can associate discrete (itemsets) in combination with continuous measurements 4. Can handle missing values	1. Computationally intensive for very large datasets

Ref: https://images.datacamp.com/image/upload/v1669897283/Marketing/Blog/Unsupervised_Machine_Learning_Models.pdf