Unsupervised Learning: Association Rules

Market Basket Analysis and Apriori Algorithm

Sarwan Ali

Department of Computer Science
Georgia State University

Understanding Association Rules

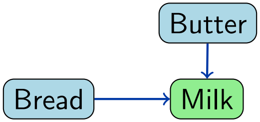
Today's Learning Journey

What are Association Rules?

- **Association Rules** identify relationships between different items in a dataset
- Discover patterns of the form: "If X, then Y"
- Most commonly used in **Market Basket Analysis**
- Help businesses understand customer purchasing behavior

Real-World Example

"People who buy bread and butter also tend to buy milk"



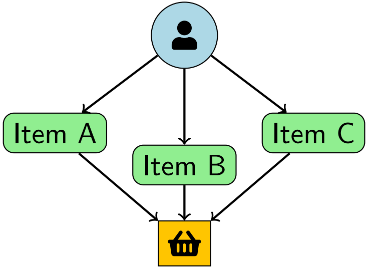
Market Basket Analysis

Definition:

- Technique to identify items frequently bought together
- Analyzes transactional data
- Reveals hidden patterns in customer behavior

Applications:

- Product placement in stores
- Cross-selling strategies
- Recommendation systems
- Inventory management



Essential Terminology

Itemset

A collection of one or more items. Example: {Bread, Milk}

Transaction

A set of items purchased together in a single purchase

Frequent Itemset

An itemset that appears in at least a minimum number of transactions

Association Rule

An implication of the form $X \Rightarrow Y$ where X and Y are itemsets

Transaction ID	**Items**
T1	{Bread, Milk, Butter}
T2	{Bread, Eggs}
T3	{Milk, Butter, Cheese}
T4	{Bread, Milk, Eggs, Butter}

Support, Confidence, and Lift

Support: Frequency of itemset occurrence

$$\text{Support}(X) = \frac{\text{Transactions containing } X}{\text{Total transactions}}$$

Confidence: Strength of implication

$$\text{Confidence}(X \Rightarrow Y) = \frac{\text{Support}(X \cup Y)}{\text{Support}(X)}$$

Lift: Measure of rule interestingness

$$\text{Lift}(X \Rightarrow Y) = \frac{\text{Confidence}(X \Rightarrow Y)}{\text{Support}(Y)}$$

Interpretation

- **Support**: How often does the rule occur?
- **Confidence**: How reliable is the rule?
- **Lift**: How much better than random?

Lift $= 1$: Independence
Lift > 1: Positive correlation
Lift < 1: Negative correlation

Example Calculation

TID	Items
1	{Bread, Milk}
2	{Bread, Eggs}
3	{Milk, Eggs}
4	{Bread, Milk, Eggs}
5	{Bread, Eggs}

For rule: Bread $\Rightarrow$ Eggs

$$\text{Support}(\{Bread\}) = \frac{4}{5} = 0.8 \tag{1}$$

$$\text{Support}(\{Eggs\}) = \frac{4}{5} = 0.8 \tag{2}$$

$$\text{Support}(\{Bread, Eggs\}) = \frac{3}{5} = 0.6 \tag{3}$$

$$\text{Confidence}(Bread \Rightarrow Eggs) = \frac{0.6}{0.8} = 0.75 \tag{4}$$

$$\text{Lift}(Bread \Rightarrow Eggs) = \frac{0.75}{0.8} = 0.9375 \tag{5}$$

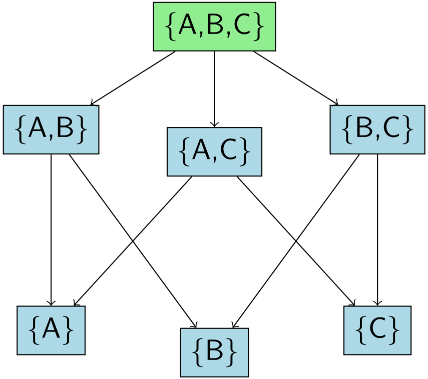
Apriori Algorithm Overview

Purpose: Find all frequent itemsets efficiently
Key Principle - Apriori Property:

Downward Closure

If an itemset is frequent, then all of its subsets are also frequent.

Contrapositive: If an itemset is infrequent, then all of its supersets are also infrequent.



This property allows us to prune the search space significantly!

Apriori Algorithm Steps

1. **Initialize:** Start with 1-itemsets
2. **Generate Candidates:** Create candidate k-itemsets from frequent (k-1)-itemsets
3. **Prune:** Remove candidates that have infrequent subsets
4. **Count Support:** Scan database to count support of candidates
5. **Filter:** Keep only frequent itemsets (support $\geq$ min_support)
6. **Repeat:** Continue until no more frequent itemsets found

$$L_1 \xrightarrow{\text{generate}} C_2 \xrightarrow{\text{prune}} L_2 \xrightarrow{\text{generate}} C_3 \xrightarrow{\text{prune}} L_3$$

Apriori Algorithm Pseudocode

Algorithm

```
def apriori(transactions, min_support):
    # Step 1: Find frequent 1-itemsets
    L1 = find_frequent_1_itemsets(transactions, min_support)
    L = [L1]
    k = 2
    while L[k-2] is not empty:
        Ck = generate_candidates(L[k-2]) # Step 2: Generate candida.
        Ck = prune_candidates(Ck, L[k-2]) # Step 3: Prune candidates
        # Step 4: Count support
        for transaction in transactions:
            increment_count(Ck, transaction)
        # Step 5: Filter frequent itemsets
        Lk = filter_frequent(Ck, min_support)
        L.append(Lk)
        k += 1
    return L
```

Apriori Example - Dataset

Transaction Database:

TID	Items
100	{I1, I2, I5}
200	{I2, I4}
300	{I2, I3}
400	{I1, I2, I4}
500	{I1, I3}
600	{I2, I3}
700	{I1, I3}
800	{I1, I2, I3, I5}
900	{I1, I2, I3}

Parameters:

- Total transactions: 9
- Minimum support threshold: 2 ($\approx$ 22.2%)

Step 1: Find Frequent 1-Itemsets

Candidate 1-itemsets (C_1):

Itemset	Support
{I1}	6
{I2}	7
{I3}	6
{I4}	2
{I5}	2

Frequent 1-itemsets (L_1):

Itemset	Support
{I1}	6
{I2}	7
{I3}	6
{I4}	2
{I5}	2

Result

All 1-itemsets are frequent (support ≥ 2)

Step 2: Generate and Prune 2-Itemsets

Candidate 2-itemsets (C_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

Frequent 2-itemsets (L_2):

Itemset	Support
{I1, I2}	4
{I1, I3}	4
{I1, I5}	2
{I2, I3}	4
{I2, I4}	2
{I2, I5}	2

Pruned: {I1, I4}, {I3, I4}, {I3, I5}, {I4, I5} (support < 2)

Step 3: Generate 3-Itemsets

Generate C_3 from L_2:

- Join {I1, I2} and {I1, I3} $\rightarrow$ {I1, I2, I3}
- Join {I1, I2} and {I1, I5} $\rightarrow$ {I1, I2, I5}
- Join {I1, I3} and {I1, I5} $\rightarrow$ {I1, I3, I5}
- Join {I2, I3} and {I2, I4} $\rightarrow$ {I2, I3, I4}
- Join {I2, I3} and {I2, I5} $\rightarrow$ {I2, I3, I5}
- Join {I2, I4} and {I2, I5} $\rightarrow$ {I2, I4, I5}

After Pruning (C_3):

Itemset	**Support**
{I1, I2, I3}	2
{I1, I2, I5}	2

Frequent 3-itemsets (L_3):

Itemset	**Support**
{I1, I2, I3}	2
{I1, I2, I5}	2

Note: Other candidates pruned because subsets not in L_2

From Frequent Itemsets to Rules

Process:

1. For each frequent itemset I, generate all possible rules
2. For rule $X \Rightarrow Y$ where $X \cup Y = I$:
   - Calculate confidence $= \frac{\text{support}(I)}{\text{support}(X)}$
   - Keep rule if confidence $\geq$ minimum confidence threshold

Example with $\{$I1, I2, I3$\}$ (support = 2):

Rule	Confidence	Valid?
$\{I1\} \Rightarrow \{I2, I3\}$	$\frac{2}{6} = 0.33$	✓
$\{I2\} \Rightarrow \{I1, I3\}$	$\frac{2}{7} = 0.29$	✓
$\{I3\} \Rightarrow \{I1, I2\}$	$\frac{2}{6} = 0.33$	✓
$\{I1, I2\} \Rightarrow \{I3\}$	$\frac{2}{4} = 0.50$	✓
$\{I1, I3\} \Rightarrow \{I2\}$	$\frac{2}{4} = 0.50$	✓
$\{I2, I3\} \Rightarrow \{I1\}$	$\frac{2}{4} = 0.50$	✓

Assuming minimum confidence $= 0.25$

Real-World Applications

Retail & E-commerce:

- Product recommendation
- Store layout optimization
- Promotional strategies
- Inventory management

Entertainment:

- Movie/music recommendations
- Content bundling
- User behavior analysis

Healthcare:

- Drug interaction analysis
- Treatment pattern discovery
- Symptom correlation

Web Analytics:

- Website navigation patterns
- Cross-selling opportunities
- User session analysis

Success Story

Amazon's "Customers who bought this item also bought" feature reportedly increases sales by 10-30%

Advantages and Limitations

Advantages:

- Easy to understand and interpret
- No need for labeled data
- Scalable to large datasets
- Provides actionable insights
- Well-established methodology

Limitations:

- Computationally expensive
- Generates many redundant rules
- Sensitive to parameter selection
- May miss rare but important patterns
- Assumes independence between transactions

Performance Considerations

- Time complexity: $O(2^n)$ in worst case
- Space complexity depends on number of frequent itemsets
- Multiple database scans required

Algorithm Variations

FP-Growth Algorithm:

- Uses FP-tree data structure
- Requires only 2 database scans
- More memory efficient
- Faster than Apriori for dense datasets

ECLAT (Equivalence Class Transformation):

- Uses vertical data representation
- Intersection-based approach
- Good for sparse datasets

Other Improvements:

- Hash-based itemset counting
- Transaction reduction
- Partitioning algorithms
- Sampling techniques

Advanced Metrics

Beyond Support, Confidence, and Lift:

Conviction:

$$\text{Conviction}(X \Rightarrow Y) = \frac{1 - \text{Support}(Y)}{1 - \text{Confidence}(X \Rightarrow Y)}$$

Cosine Similarity:

$$\text{Cosine}(X, Y) = \frac{\text{Support}(X \cup Y)}{\sqrt{\text{Support}(X) \times \text{Support}(Y)}}$$

Jaccard Coefficient:

$$\text{Jaccard}(X, Y) = \frac{\text{Support}(X \cup Y)}{\text{Support}(X) + \text{Support}(Y) - \text{Support}(X \cup Y)}$$

These metrics help identify truly interesting and non-redundant rules.

Key Takeaways

Association Rules

- Powerful technique for discovering relationships in transactional data
- Foundation of market basket analysis and recommendation systems
- Uses support, confidence, and lift as key measures

Apriori Algorithm

- Systematic approach to finding frequent itemsets
- Leverages downward closure property for efficiency
- Generates candidate itemsets level by level

Practical Impact

- Widely used in retail, e-commerce, and web analytics
- Drives billions of dollars in additional revenue
- Continues to evolve with new algorithms and applications

Next Steps

What's Coming Next:

- Clustering algorithms (K-means, Hierarchical)
- Dimensionality reduction techniques
- Anomaly detection methods

Practice Exercises:

- Implement Apriori algorithm from scratch
- Analyze real transaction datasets
- Compare different interestingness measures
- Optimize algorithm performance

Questions?