Leveraging MaxFP-Growth Algorithm for Market Basket Analysis

An academic exploration of efficient association rule mining techniques for retail transaction data

Introduction & Objectives

Market Basket Analysis

Discover patterns in customer purchasing behavior by identifying which items are frequently bought together, enabling targeted cross-selling strategies

FP-Growth Algorithm

More efficient than Apriori by eliminating candidate generation, using a compact FP-tree structure to mine patterns

MaxFP Extension

Focus on maximal frequent itemsets to reduce redundancy and provide a concise representation of the pattern space

Our goal: Implement and evaluate the MaxFP-Growth algorithm on retail transaction data to discover actionable association rules for business decision-making

Key Terminology

Basic Concepts

- **Transaction:** A set of items purchased together (one row in retail.dat)
- **Item:** An integer code representing a product (no descriptions in FIMI)
- **Itemset:** A collection of items, e.g., {39, 48}
- **Support:** Absolute (count) or relative (count/total transactions)

Pattern Types

- **Frequent itemset:** Itemset with support ≥ minimum threshold
- **Maximal frequent itemset:** Not a subset of any other frequent itemset
- **Closed itemset:** No superset has the same support (reference only)

Association Rule (A → B)

Implies that when A occurs, B likely occurs

Confidence

$conf(A \to B) = support(A \cup B) / support(A)$

Lift

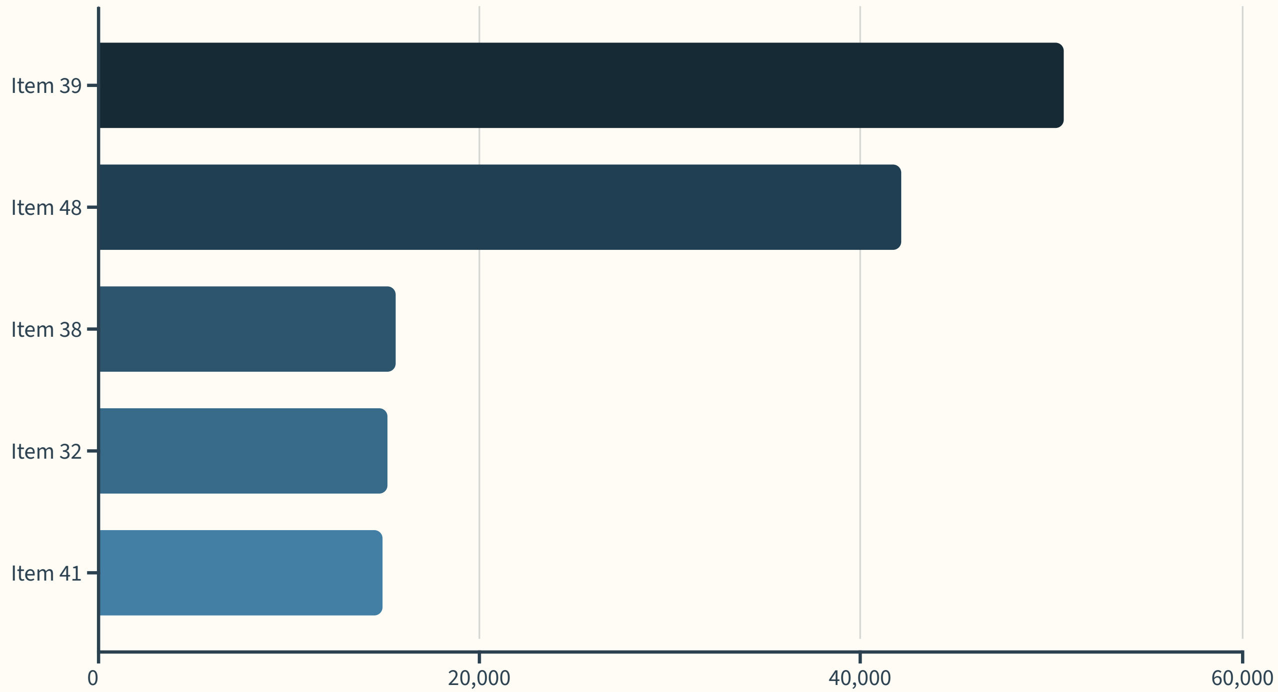
$lift(A \to B) = conf(A \to B) / support(B)$

Values ≥1 indicate positive correlation

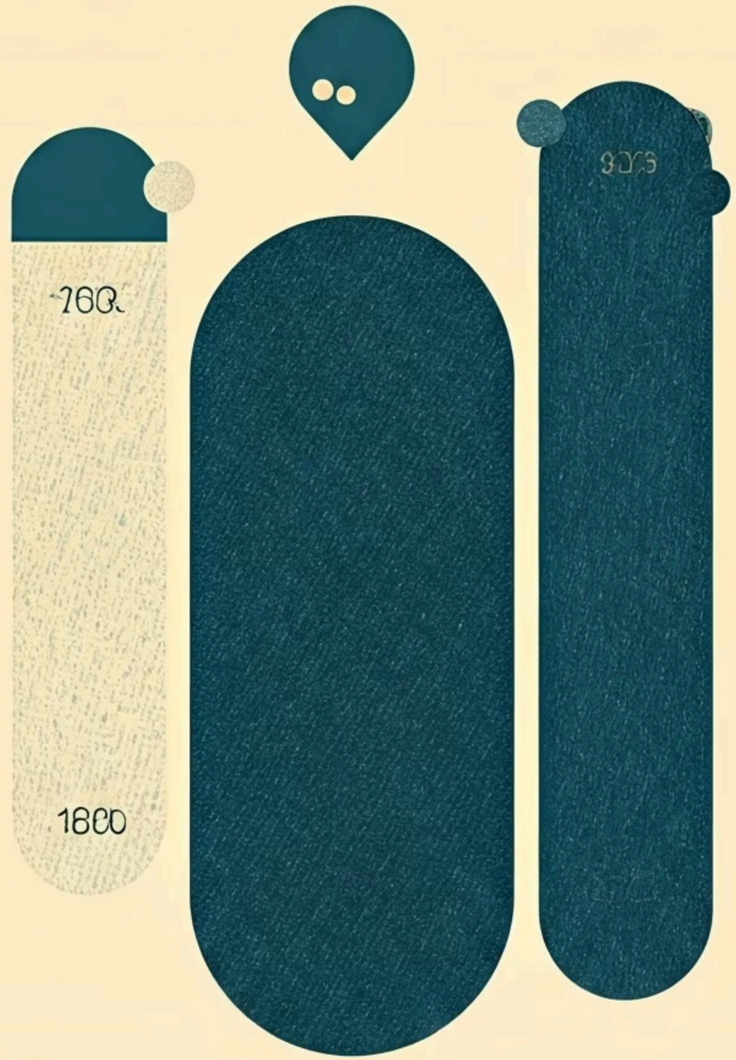
Dataset Characteristics

FIMI retail.dat Statistics

- Total transactions: 88,162
- Unique items: 16,470
- Basket size: mean 10.31, median 8, min 1, max 76



Note the significant skew in item frequencies: items 39 and 48 appear in over 47% of transactions, creating challenges for meaningful rule extraction



Data Preprocessing

Standard FIMI Preprocessing Steps

1. Parse transactions by splitting on whitespace
2. Remove sentinel values (-1, -2) and empty strings
3. Eliminate duplicate items within each transaction
4. Convert items to string format for compatibility with libraries
5. Use sparse matrix representation for one-hot encoding to conserve memory

Support threshold conversion for our dataset (N=88,162):

min_support	support_count
0.005	441
0.01	882
0.02	1,764

FP-Growth Algorithm

1. Initial Scanning

Count support of individual items and retain only those meeting the minimum threshold

2. FP-Tree Construction

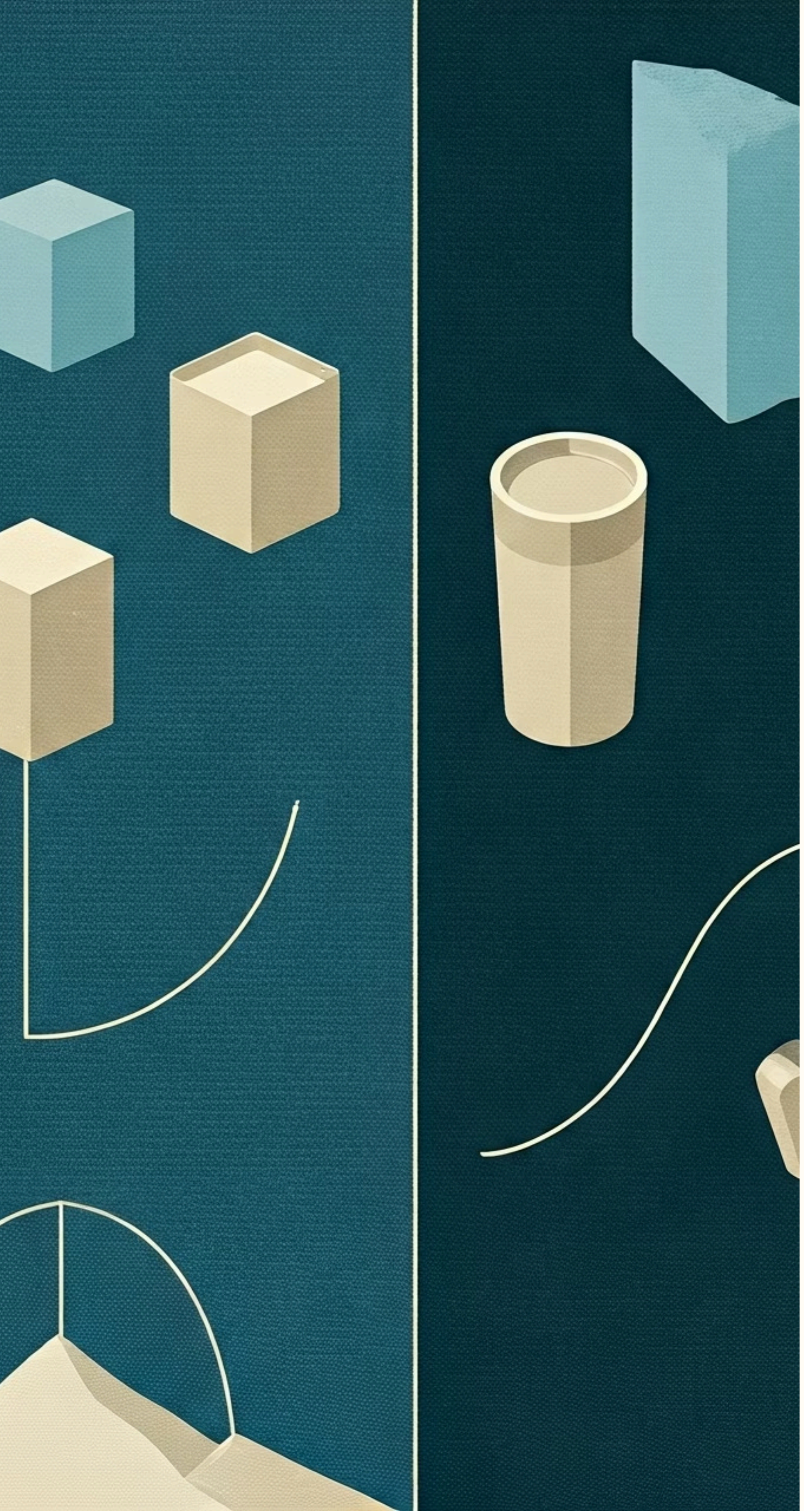
Sort items by descending frequency within each transaction and build a prefix tree with counts at nodes

3. Conditional Pattern Mining

For each item (least to most frequent), extract conditional pattern bases and recursively build conditional FP-trees

4. Pattern Generation

Combine prefix patterns to generate complete frequent itemsets



MaxFP-Growth Implementation

Two Valid Approaches:

Post-Processing Approach

Run standard FP-Growth to find all frequent itemsets, then filter to keep only maximal itemsets (those not a subset of any other frequent itemset)

- Easier to implement
- Meets assignment requirements

Direct Mining Approach

Use specialized algorithms like FPmax or MAFIA that directly mine maximal patterns with stronger pruning techniques

- More efficient for large datasets
- Not required for this assignment

Key insight: While maximal itemsets provide a concise representation of the pattern space, they don't preserve support information for subsets, which may limit rule generation options.

Experimental Design & Parameters

Parameter Selection Rationale

- **min_support = 0.005-0.01**: Balances between finding meaningful patterns and computational efficiency
- **min_confidence = 0.6-0.8**: Ensures rules have reasonable predictive power
- **min_lift = 1.2**: Filters out rules that merely reflect the influence of extremely popular items
- **max_length = 3-4**: Focuses on interpretable patterns while avoiding combinatorial explosion

Parameter Effects

↑ min_support: ↓ patterns & rules, ↑ performance

↑ min_confidence: eliminates weak rules but may retain misleading high-support/low-lift rules

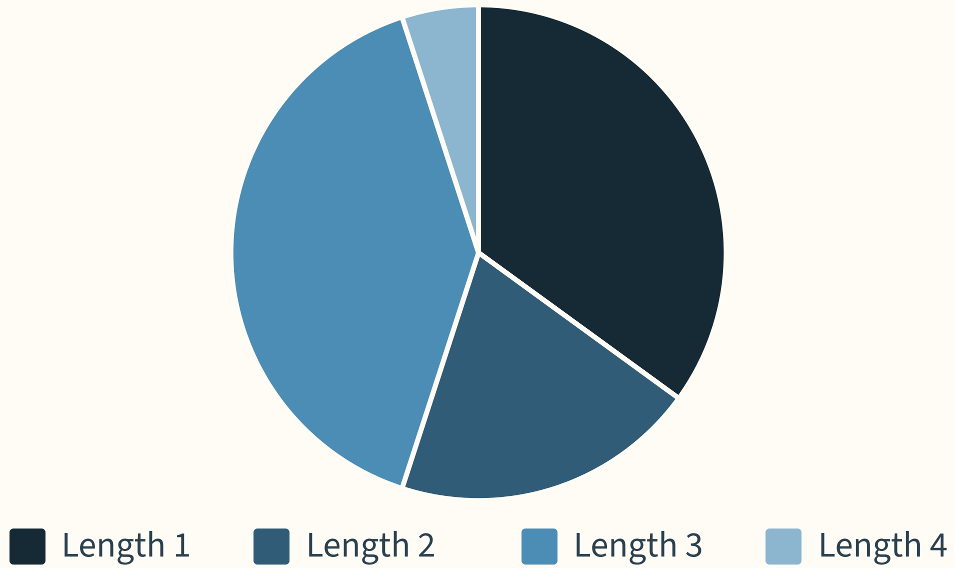
↑ min_lift: keeps truly useful rules but may remove reasonable patterns with rare consequents

↓ max_length: improves interpretability but may miss higher-order interactions

For sparse retail data with super-frequent items (like our dataset), carefully balancing these parameters is crucial to extract meaningful patterns without being overwhelmed by trivial or misleading associations.

Results & Analysis

Pattern Discovery Statistics



Distribution of 20 Maximal Frequent Itemsets

Top Association Rules by Lift

Rule	Support	Confidence	Lift
36 → {38, 39}	2.60%	0.662	4.798
170 → {38, 39}	2.70%	0.652	4.719
{170, 39} → 38	2.70%	0.981	4.711
170 → 38	4.05%	0.978	4.699
110 → 38	3.64%	0.975	4.685

Key insight: Items 38, 39, 170, and 110 form a strongly correlated cluster (lift ~4.7), suggesting a significant cross-selling opportunity.

Business Recommendations & Limitations

1

Cross-Selling Opportunities

The strong association cluster (items 38, 39, 170, 110) with lift values ~4.7 represents a prime opportunity for bundle promotions or strategic product placement

2

Model Validation

Implement 80/20 train/test split validation to ensure rule stability and statistical significance testing (Fisher/Chi-square) to confirm non-random associations

3

Limitations

FIMI dataset lacks item descriptions, making business interpretation challenging; super-frequent items (39, 48) can create misleading rules with high confidence but lift near 1.0

4

Future Directions

Explore closed itemsets for more efficient pattern representation; implement direct MaxFP mining algorithms; develop minimal non-redundant rule bases for cleaner insights