Beyond Market Baskets

Advanced Pattern Mining

Duration: 1 Hour | **Target Audience:** Data Science/Computer Science students

Prerequisites: Understanding of basic association rule mining, Apriori algorithm, support/confidence concepts

Part 1: Introduction & Learning Objectives

ADVANCED ASSOCIATION RULE MINING

Beyond Simple Itemsets

Capturing the Complexity of Real-World Data

Opening Insight

Real-World Data is Richer

The Basics:

We know how to find simple rules like {Milk} → {Bread}.

The Reality:

Real data involves hierarchies, demographics, and quantities.

Sophisticated Examples:

- Customers who buy premium smartphones (specific) also buy extended warranties (specific).
- Customers aged 25-34 with income \$50-75K buy 2-4 books monthly.

WHY ADVANCED MINING?

Graduating from Basic Patterns

Limitation:

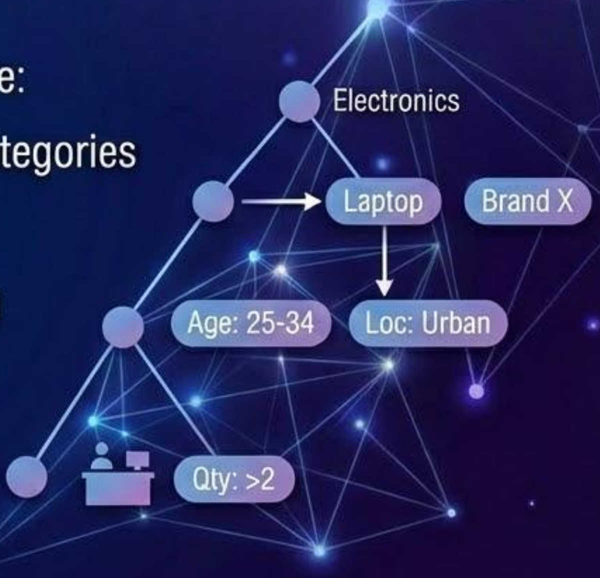
Basic Apriori treats all items as flat and equal (Milk is the same ‘level’ as a Laptop).

Flat & Equal Structure

Opportunity:

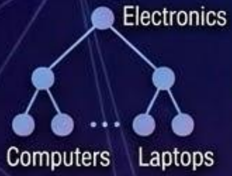
We need patterns that capture:

- **Granularity**: Product categories vs. specific brands.
- **Context**: Who is buying (Age, Location).
- **Quantity**: How much they are buying.



LEARNING OBJECTIVES

By the end of this module, you will be able to:



Multi-Level Mining:
Mine patterns across concept hierarchies (e.g., Electronics → Computers → Laptops).

Multidimensional Mining:
Include dimensions beyond just 'items' (e.g., Age, Income, Time).

Quantitative Data:
Handle numeric attributes (e.g., Age ranges, Salary brackets) in rules.

Rare & Negative Patterns:
Discover what happens infrequently or what items conflict (e.g., People who buy X never buy Y).

Pattern Compression:
Reduce redundancy to find the most concise rule set.

Constraint-Based Mining:
Apply constraints to focus only on relevant patterns.

Part 2: Mining Various Kinds of Patterns

Multilevel Association Rules

Beyond Simple Itemsets

Mining patterns across different levels of abstraction.

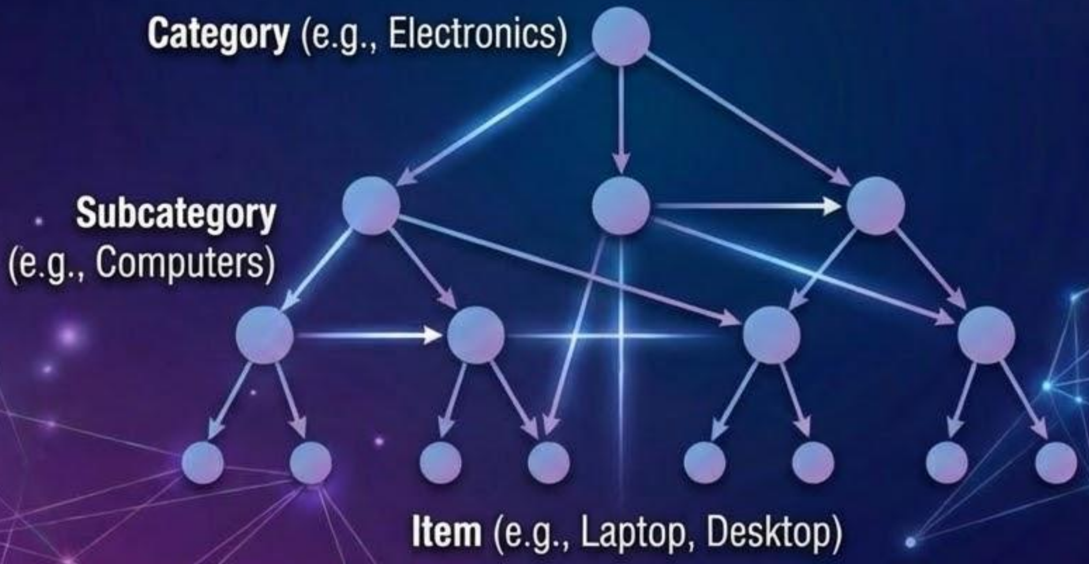


Table 5.1 Task-relevant data, *D*.

TID	Items Purchased
T100	Apple 15″ MacBook Pro, HP Photosmart 7520 printer
T200	Microsoft Office Professional 2020, Microsoft Surface Mobile Mouse
T300	Logitech MX Master 2S Wireless Mouse, Gimars GEL Wrist Rest
T400	Dell Studio XPS 16 Notebook, Canon PowerShot SX70 HS Digital Camera
T500	Apple iPad Air (10.5-inch, Wi-Fi, 256GB), Norton Security Premium
...	...

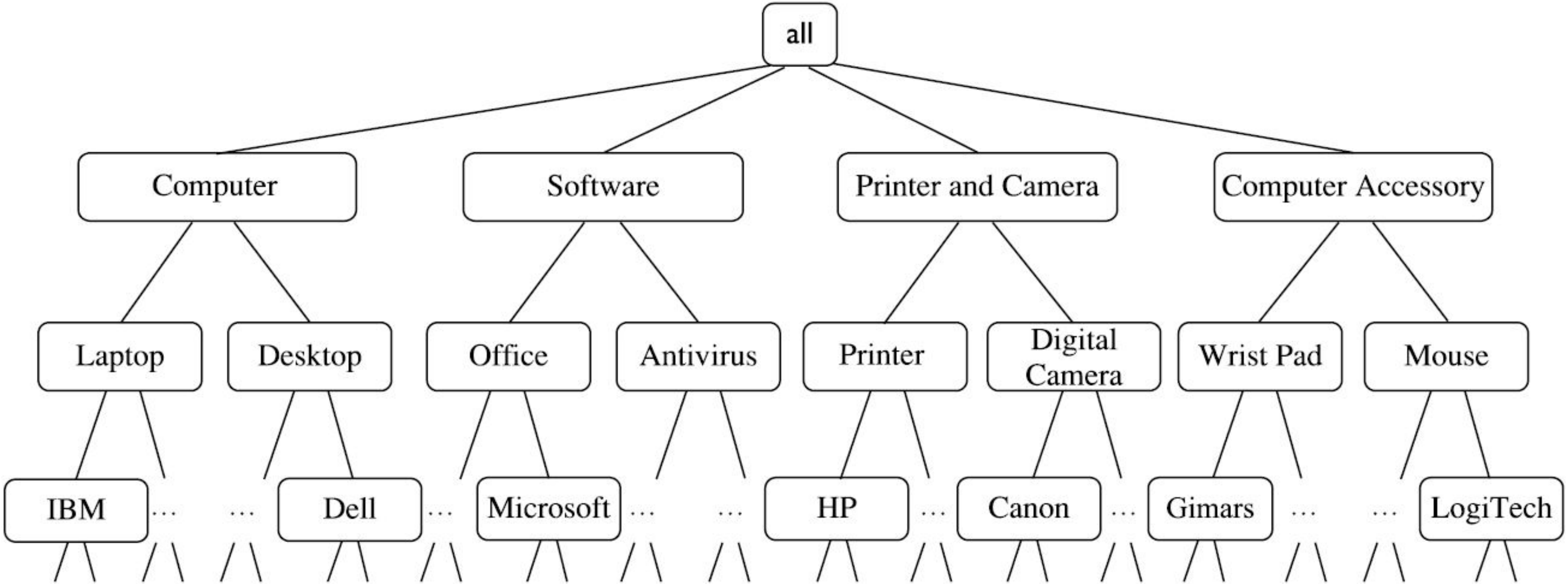


FIGURE 5.1

Concept hierarchy for computer items of an e-store.

The Limitation of Single-Level Mining

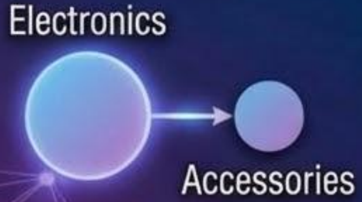
Missing the Forest for the Trees

The Problem:

Traditional mining (e.g., standard Apriori) finds simple rules like {Laptop} → {Mouse} but misses insights at other granularities.

Missed Opportunities:

Higher Level:



Strategic Trend

Lower Level:

Specific Cross-sell

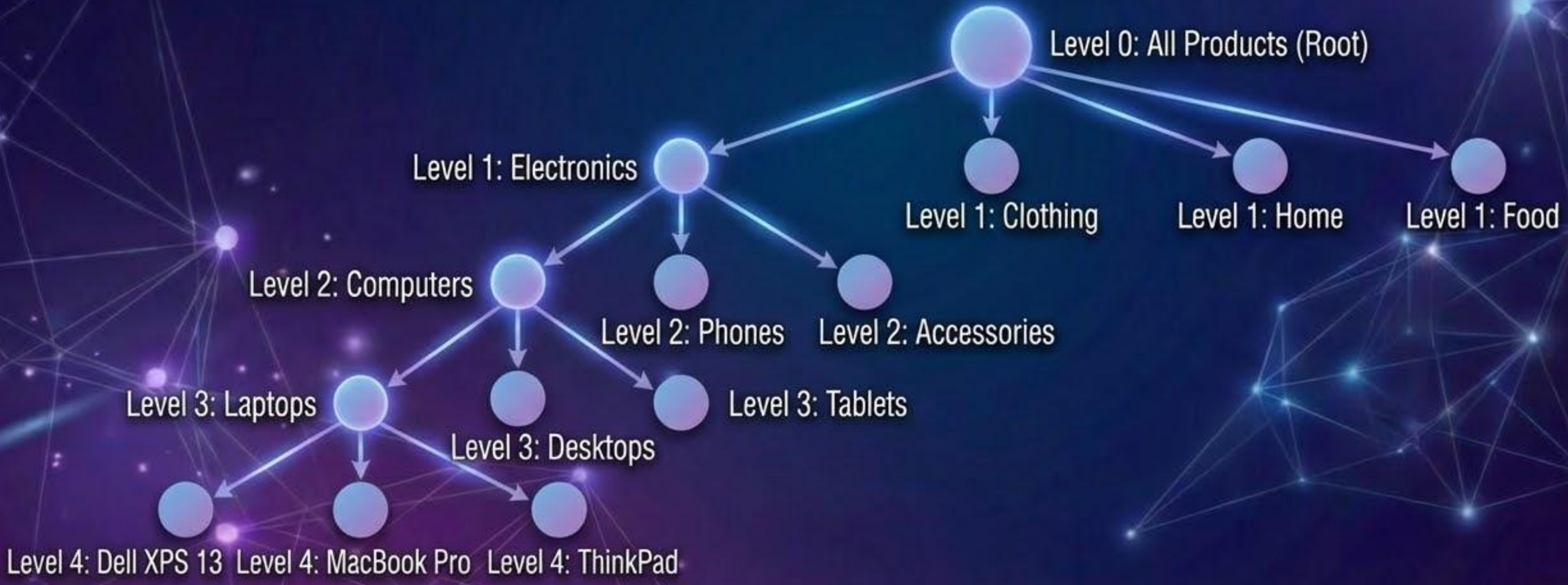
Solution:

We must incorporate Concept Hierarchies into our mining process.

CONCEPT HIERARCHIES IN PATTERN MINING

Structuring the Data

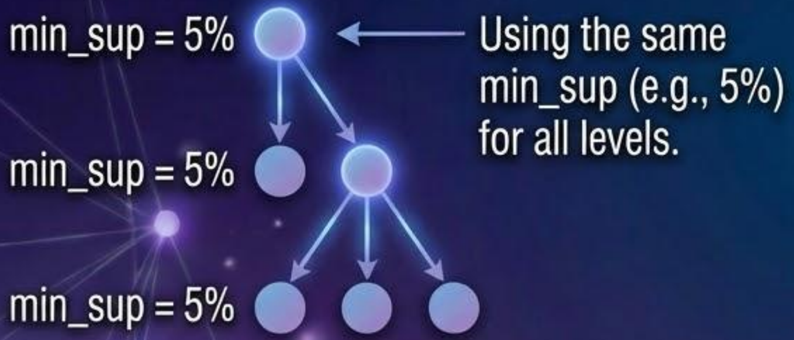
Data naturally exists in layers. A typical Product Hierarchy looks like this:



Approaches to Minimum Support

Rigid vs. Flexible Thresholds

UNIFORM MINIMUM SUPPORT (Rigid):



- **Problem:** “Electronics” appears often, but “Dell XPS 13” is rare.
- **Result:** You miss specific patterns because the threshold is too high for leaf nodes.

REDUCED MINIMUM SUPPORT (Flexible):

- **Benefit:** Allows discovery at appropriate granularities without noise.

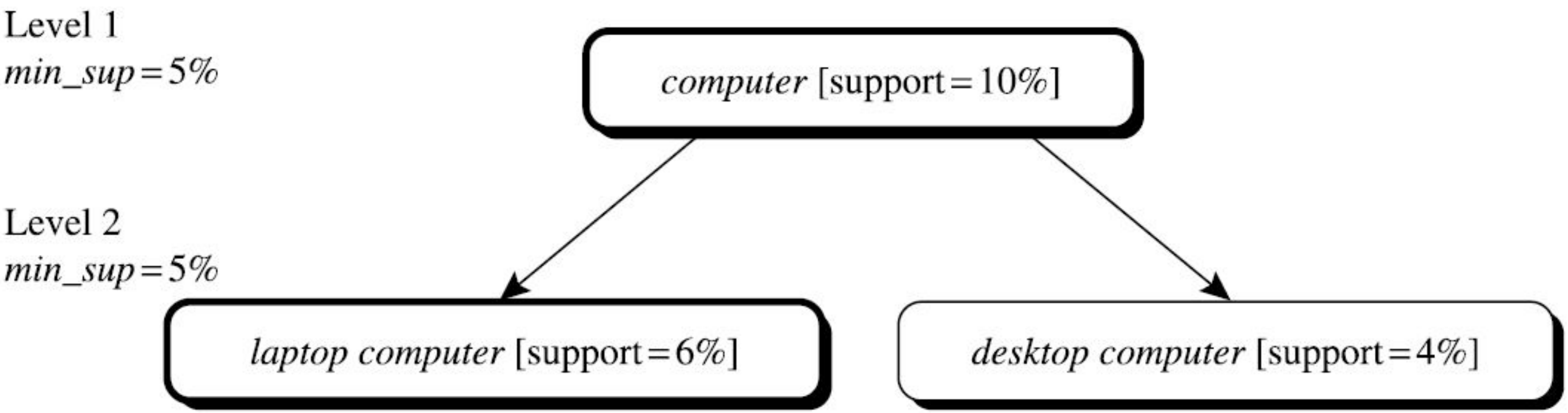


FIGURE 5.2

Multilevel mining with uniform support.

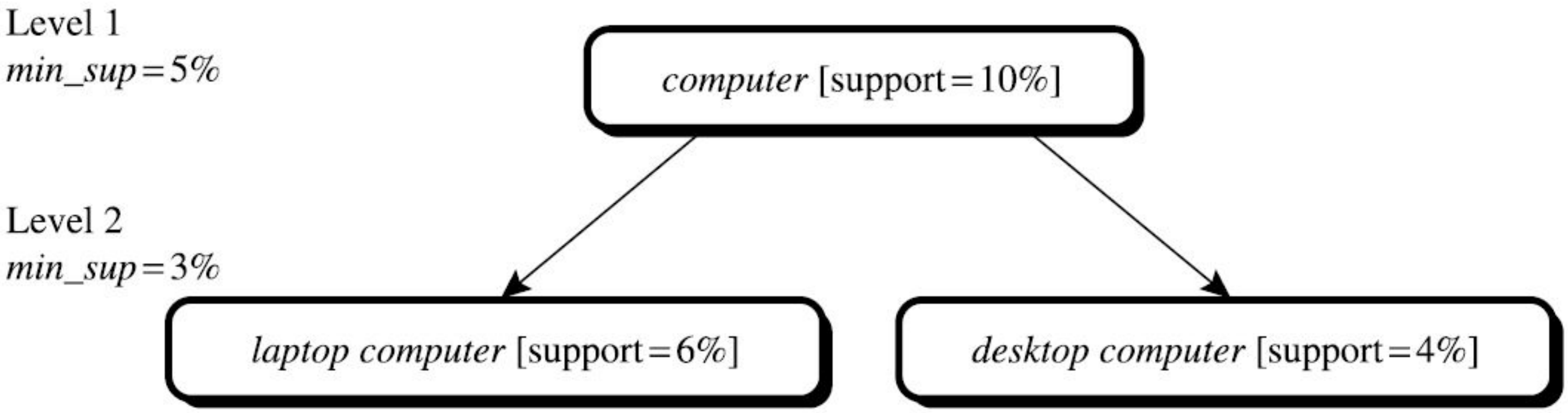


FIGURE 5.3

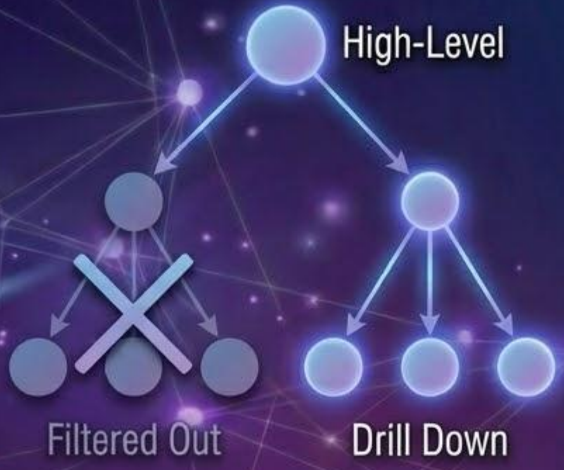
Multilevel mining with reduced support.

MINING STRATEGIES

Navigating the Hierarchy

TOP-DOWN:

Start from high-level categories, filter promising branches, then drill down.



BOTTOM-UP:

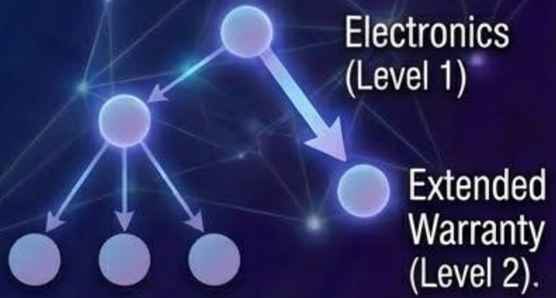
Start with specific items, generalize upwards to find broader trends.

LEVEL-CROSSING:

Finding rules that exist across different levels.

Example: {Electronics} (Level 1) → {Extended Warranty} (Level 2).

Context: A broad purchase category triggering a specific add-on.



Real-World Example: Amazon Hierarchy

Books → Fiction → Mystery → Agatha Christie

HIGH-LEVEL RULE:

{Mystery Books} → {Book Lights}

- **Insight**: Genre-based accessory cross-sell.

LOW-LEVEL RULE:

{Agatha Christie} → {British Mystery Collections}

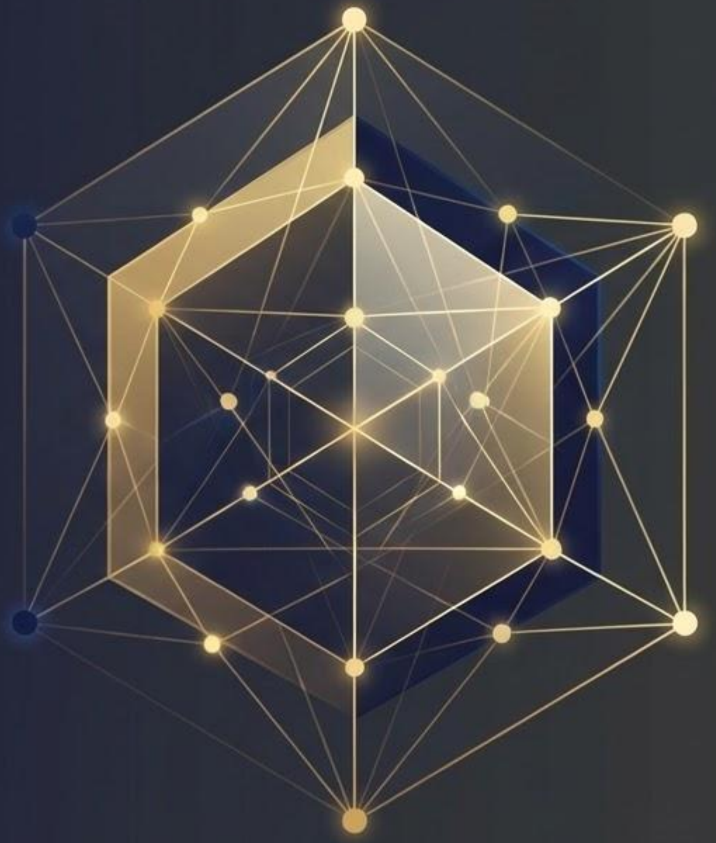
- **Insight**: Author-specific recommendation.

Mining Multidimensional Associations

Section 5.1.2

Beyond Single-Dimensional Patterns

Focus: Incorporating multiple dimensions (attributes) into mining.



BEYOND SINGLE DIMENSIONS

Adding Context to the Cart

TRADITIONAL (SINGLE-DIMENSIONAL):

- Focuses only on items bought together.

{bread} → {butter}

(Item → Item).

MULTIDIMENSIONAL:

- Incorporate attributes like Age, Location, and Income.

{age='25-34', location='urban'} → {car='SUV'}

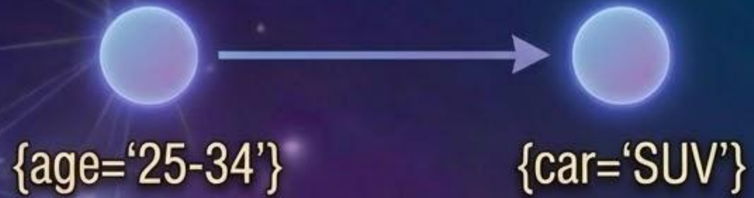
- **Insight:** Captures Attributes → Attribute relationships.

TYPES OF MULTIDIMENSIONAL RULES

Categorizing the Insights

INTER-DIMENSION RULES:

- Different dimensions in the antecedent (left) and consequent (right).



- **Example:** {age='25-34'} → {car='SUV'} (Customer Demographics → Product Choice).

HYBRID-DIMENSION RULES:

- A mix of dimensions on both sides.

- **Example:** {age='25-34', bread} → {butter, location='suburban'}
- **Context:** A specific demographic buying specific items in a specific location.

THE APPROACH: TRANSFORMATION

Flattening the Data

METHOD:

- Treat each attribute-value pair as a unique ‘item’.

Original Record:
(age=25, income=50K,
buys=bread)

TRANSFORMATION PROCESS & EXECUTION:

- Transformed Itemset:
{age_25, income_50K, buys_bread}

Transformed Itemset:
{age_25, income_50K,
buys_bread}

- **Execution:** Once transformed, apply the standard Apriori Algorithm to find frequent patterns.

CHALLENGES & SOLUTIONS

Managing Complexity

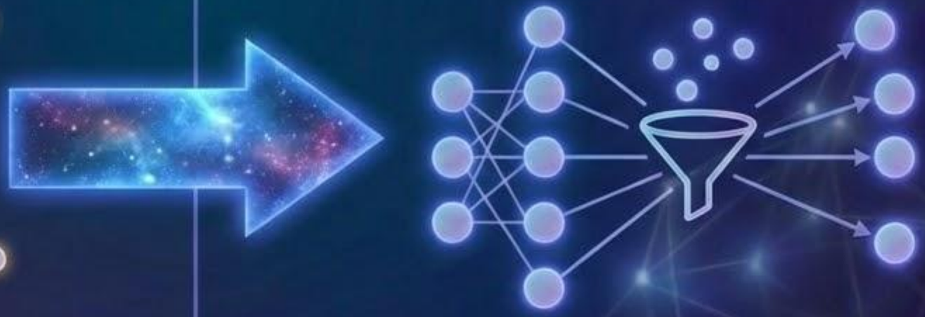
THE CHALLENGE: Combinatorial Explosion.

- Age: 7 Categories (Teen, 20s, 30s...)
- Income: 5 Brackets.
- Location: 4 Types.
- Products: 1,000s of items.

Result: The search space becomes massive.

THE SOLUTION: Use Dimension Constraints.

- Restrict mining to specific dimensions of interest (e.g., only Age and Product).

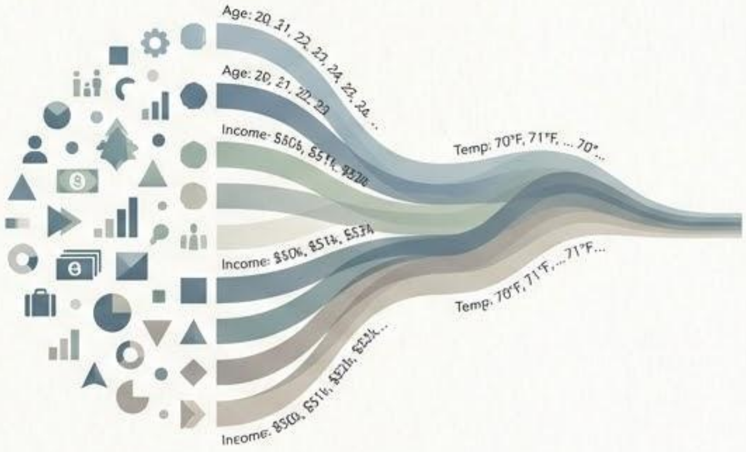


- Filter out combinations that don’t make business sense.

SECTION 5.1.3

MINING QUANTITATIVE ASSOCIATION RULES

Handling Numeric Data



Focus: From Categories to Continuous Values.

Context: Dealing with Age, Income, Temperature, and Price.

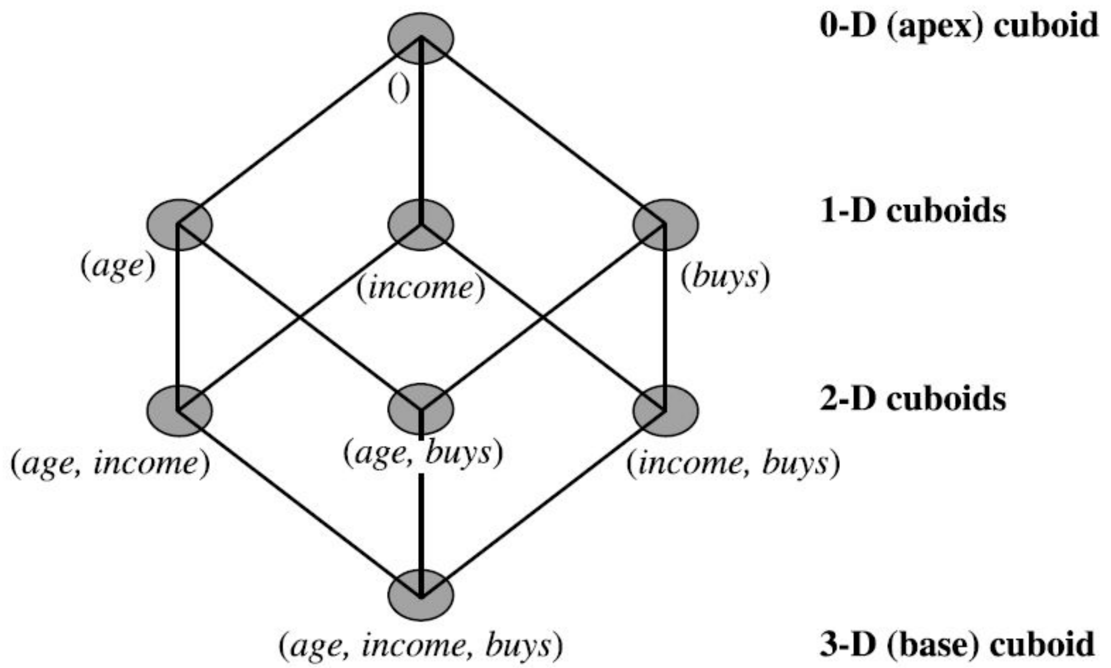


FIGURE 5.4

Lattice of cuboids, making up a 3-D data cube. Each cuboid represents a different group-by. The base cuboid contains the three predicates *age, income*, and *buys*.

The Challenge of Numeric Attributes

Discrete vs. Continuous

The Problem: Discrete Items

Standard association mining (Apriori) works on discrete items (e.g., Apple, Milk).

?

The Reality: Continuous Data

Age: 25, 26, 27...

Income: $52,000, $52,500...

Temperature: 98.6°F, 99.1°F...

The Goal:

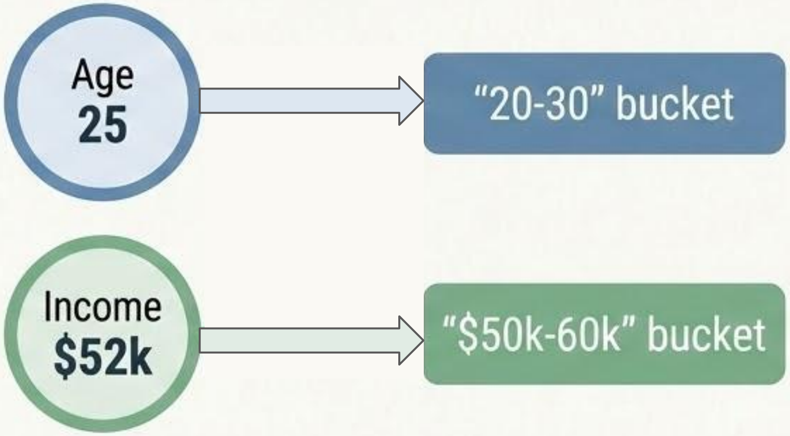
To find rules that incorporate these values meaningfully without treating every unique number as a separate item.

Approach 1: Discretization (Binning)

Turning Numbers into Categories

Method:

Group numeric values into ranges (bins).



Types of Discretization:

Static: Pre-defined bins (e.g., 0-10, 10-20, 20-30). Simple but rigid.

Dynamic: Data-driven bins based on distribution (e.g., quartiles). Adapts to the data.

Optimized: Finding bins that maximize the "interestingness" of the resulting rules.

Approach 2: Quantitative Rule Mining Algorithms

Finding the Sweet Spot

Goal: To dynamically find the most significant ranges, rather than forcing data into pre-set bins.

Resulting Rules:

$\{\textbf{age} \in [25, 35], \textbf{salary} \in [50k, 70k]\} \rightarrow \{\textbf{car}=\text{“luxury”}\}$

Methods:

Statistical Tests: Determine if a range is statistically significant.

Clustering: Group numeric values that tend to occur together in the dataset.

Example: Medical Data

Precision Saves Lives

Simple Rule:

{Diabetes} → {Heart Disease}

Problem: Too generic.

Quantitative Rule:

{Age=60-70, BMI=30-35, Diabetes} → {Heart Disease}

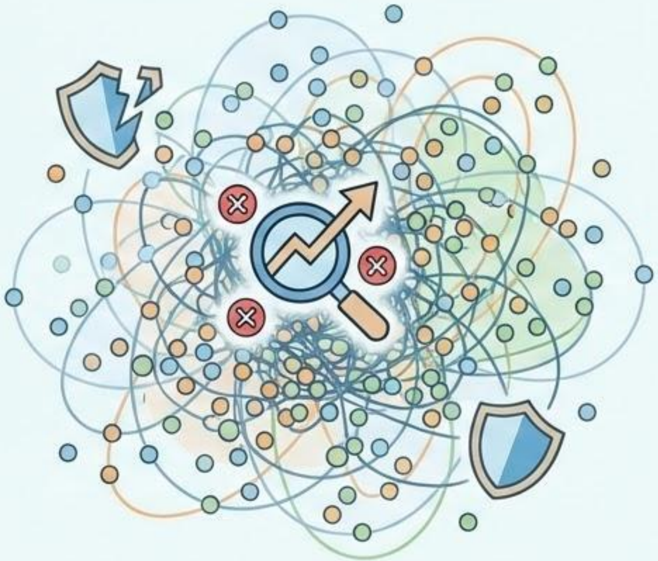
Confidence: 85%

Insight: Identifies high-risk groups with precision, enabling targeted intervention.

The Curse Strikes Again!

Managing 1000+ Products & 50+ Attributes

When standard algorithms choke on complexity.



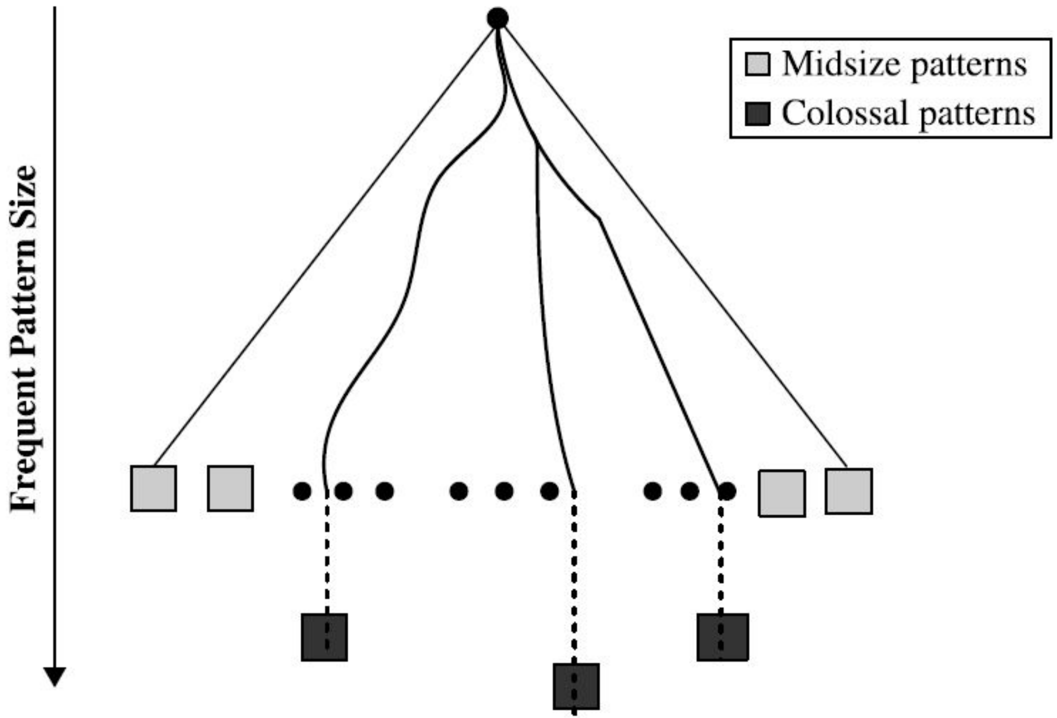


FIGURE 5.5

A high-dimensional data set may contain a small set of colossal patterns but exponentially many midsize patterns.

The Problem
Density, Dimensionality, & Length

The Curse: High dimensionality makes standard algorithms slow or useless.

Key Challenges:

Density

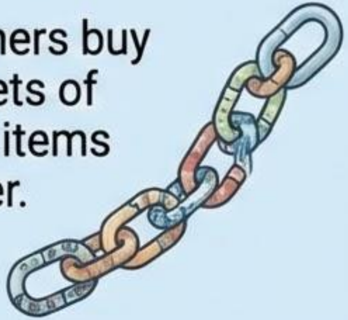
Many items per transaction (e.g., Supermarket baskets with 20+ items).

Dimensionality

A catalog with with huge potential items (100,000+).

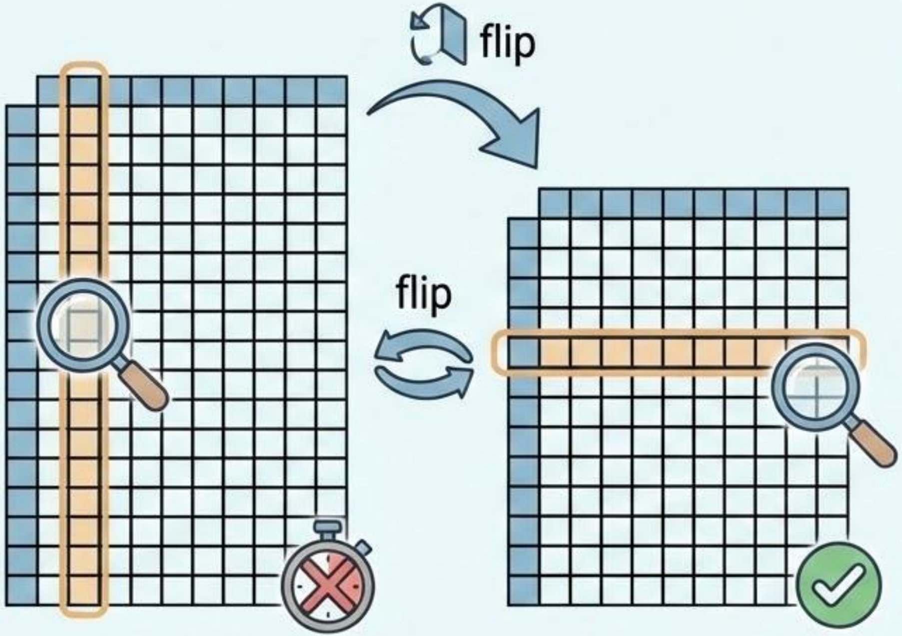
Long Patterns

Customers buy large sets of related items together.



Solution 1 - Row Enumeration

Flipping the Perspective



Standard View
(Column-based)

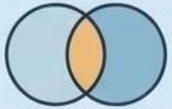
Flipped View
(Row-based)

Concept

Instead of iterating through items (columns), **iterate through transactions** (rows).

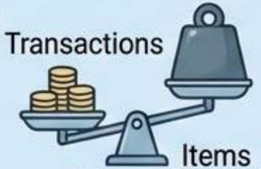
Method

Find common transaction sets first, then identify the items they share.



Benefit

Efficient when the number of items is much larger than the number of transactions (e.g., Gene data).



Solution 2 - Vertical Format (Eclat)

Inverting the Data

Data Transformation

Standard Format

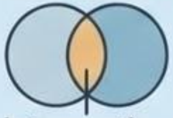
Transaction 1 → {Item A, Item B}.

convert

Vertical Format

Mechanism

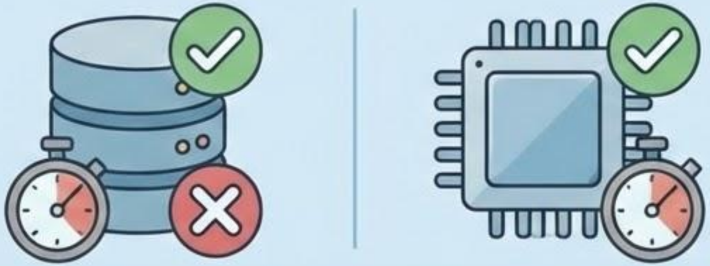
Intersects transaction lists (TID-lists) to calculate support instantly.



Intersection (Support)

Benefit

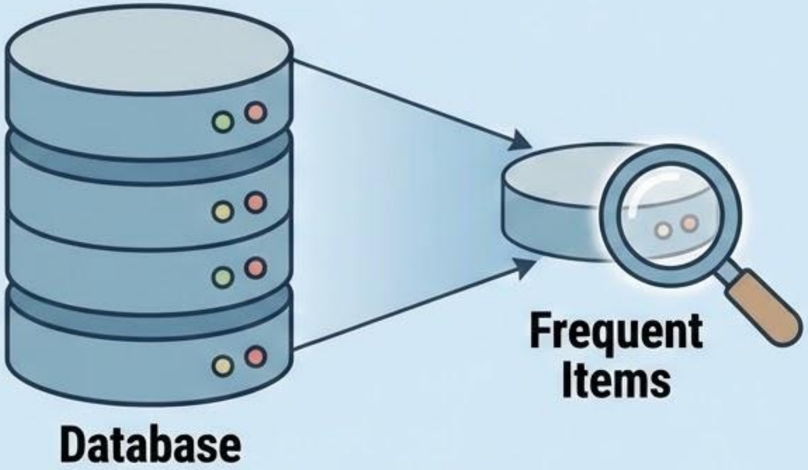
Very efficient for dense data; avoids scanning the whole database repeatedly.



Solution 3 - Projection & FP-Growth

Compression & Reduction

Projection-based

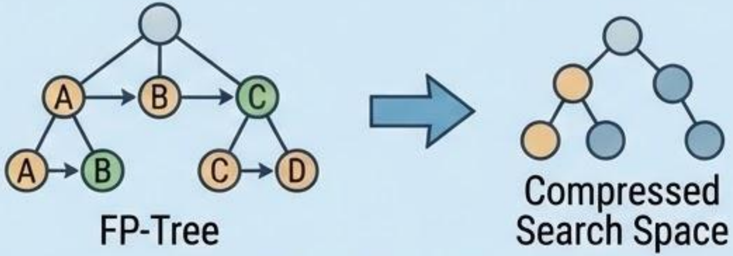


Projects the database onto frequent items to reduce dimensionality progressively.

FP-Growth Adaptation

Dense, High-dimensional Data

Works exceptionally well for dense, high-dimensional data.



Tree Compression: Becomes highly effective when many items co-occur frequently, shrinking the search space.

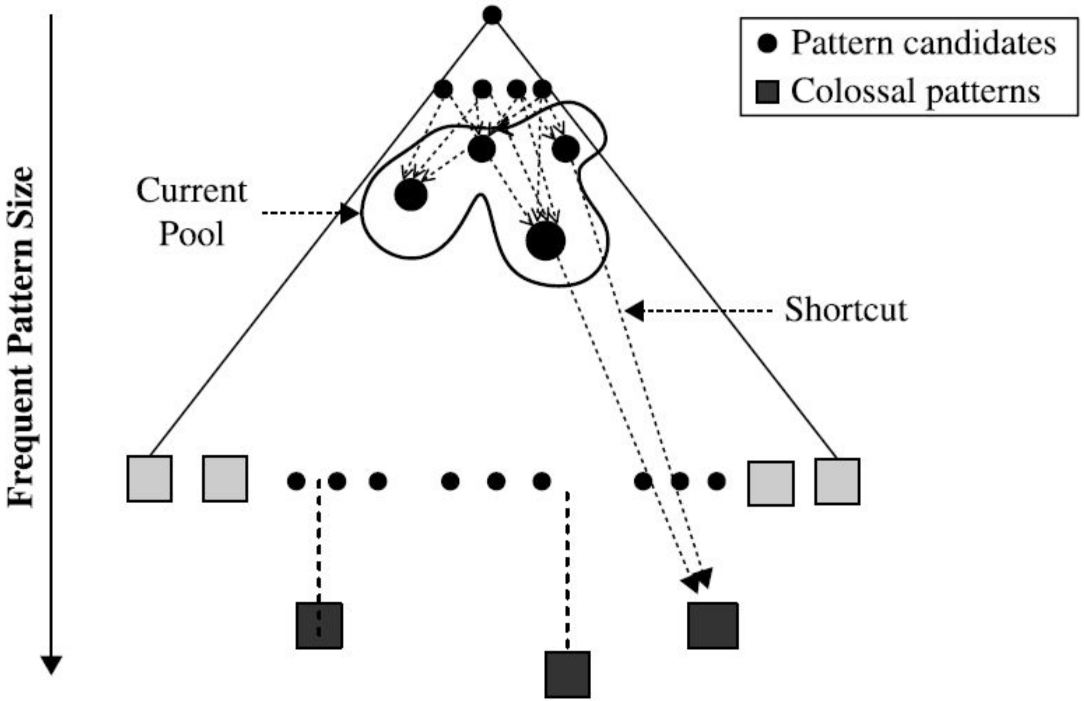


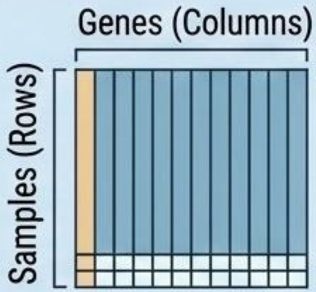
FIGURE 5.6

Pattern tree traversal: Candidates are taken from a pool of patterns, which results in shortcuts through pattern space to the colossal patterns.

Real-World Example: Gene Expression

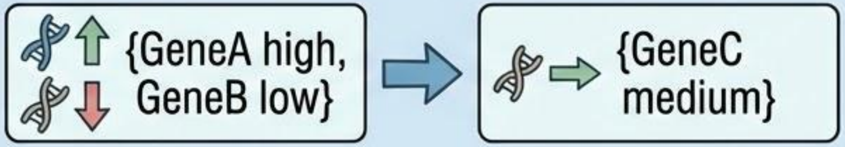
10,000 Genes × 100 Samples

The Data



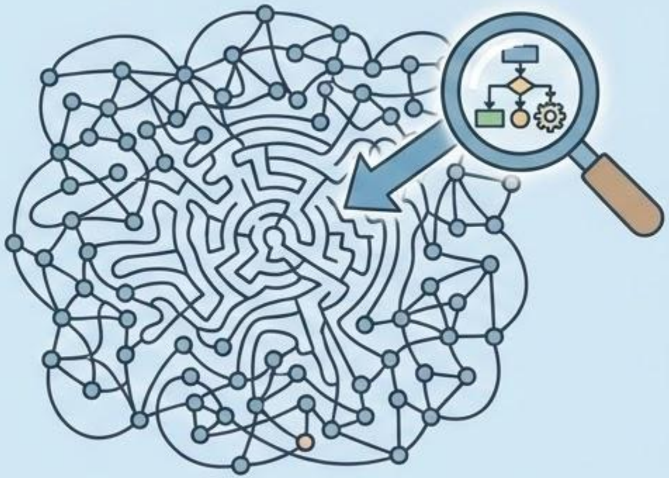
A massive number of columns (Genes) but few rows (Samples).

The Goal



Finding rules like:
{GeneA high, GeneB low} → {GeneC medium}.

The Constraint



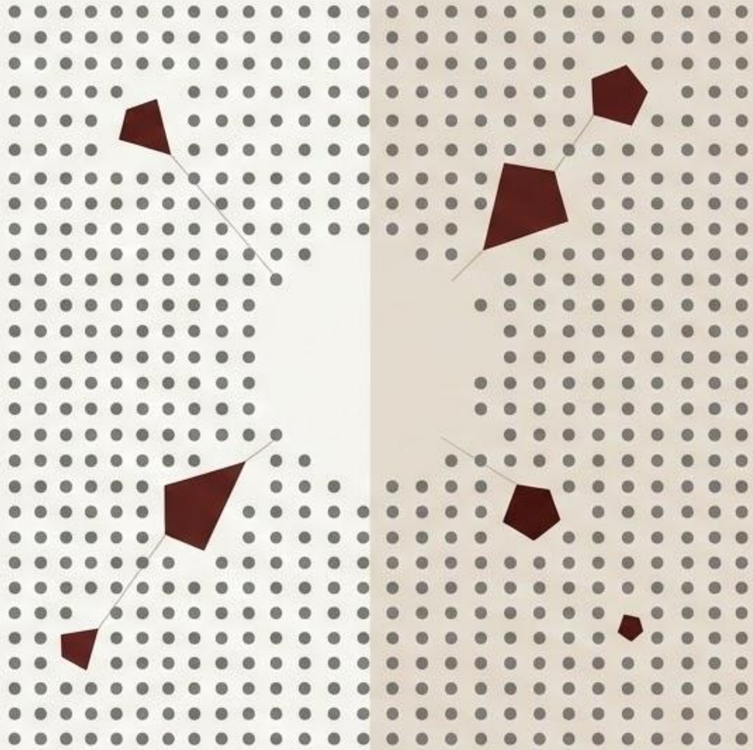
Massive Search Space

A massive search space that requires specialized algorithms (like Row Enumeration) to solve.

Section 5.1.5

The Hidden Value in Outliers

Rare Events and Absent Items



Context: Why the least frequent patterns are often the most critical.

Rare Patterns

Low Support, High Interest

Definition: Patterns that occur very infrequently (low support) but have very high confidence.

Luxury Goods

Support: 0.001%
(Very rare)

Confidence: 90%
(Almost certain)

Critical Events

Rare but critical!

Why Mine Rare Patterns?

Beyond the Average

Exceptional Cases Matter:
Vital for Fraud Detection and handling Medical Emergencies.

High-Value Patterns:
Identifies niche markets like luxury goods or specialty items that yield high margins.

Early Detection:
Early Detection: Emerging trends often start as rare patterns before becoming mainstream.

Methods for Mining Rare Patterns

Handling the Noise

Method 1: Lower min_sup:
Pros: Simple.
Cons: Generates too many uninteresting patterns (noise).

Method 2: Multiple min_sup:
Assigns different thresholds for different items based on their nature.
Truffles: min_sup = 0.001%, Bread: min_sup = 1.0%

Method 3: Constraint-based:
Only mine patterns that contain specific rare items of interest.

Negative Patterns

The Power of Absence

Definition: Rules indicating what items are not present.

Examples:

Coffee and Tea:
{Coffee} → {¬ Tea} (Coffee buyers avoid tea).

Diapers and Beer (Weekday):
{Diapers} → {¬ Beer} (On weekdays, contradicting the famous beer-diaper example).

Types:

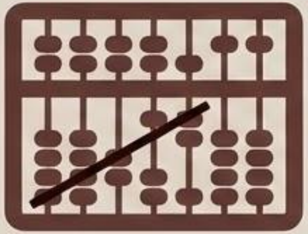
Negation of Item: ¬A (Not A).

Negation of Itemset: ¬(A, B) (Not both A and B together).

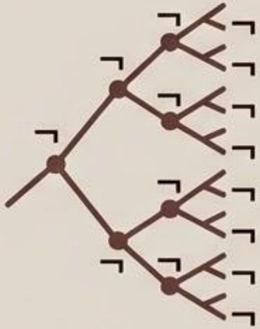
Challenges & Applications

Why it's Hard / Why it's Useful

Mining Challenges:



Calculation:
How do you efficiently count "transactions not containing A"?



Explosion:
For n items, there are 2^n possible negative itemsets.

Applications:

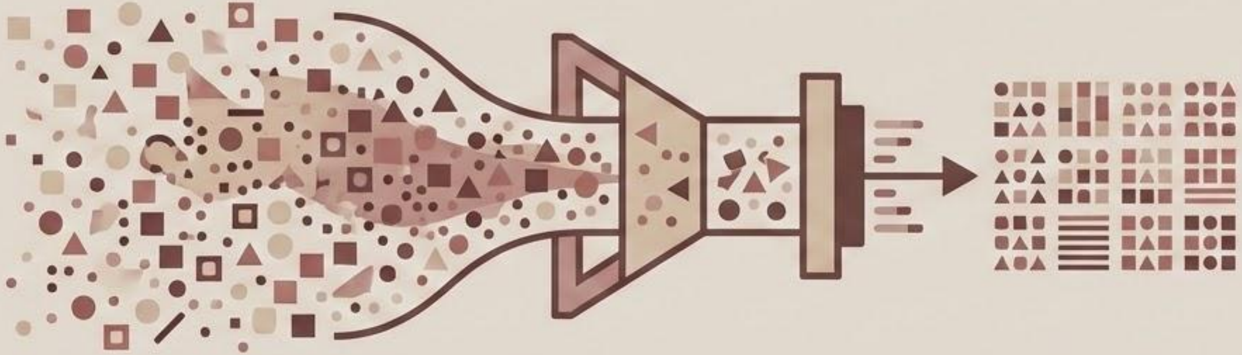
Product Cannibalization:
Buying A reduces the chance of buying B.

Medical
Drug A should not be taken with Drug B.

System Diagnostics:
When Component X works, Component Y fails.

Taming the Pattern Flood

Clustering & Compression Strategies



When 10,000 patterns are too many to handle.

The Problem: Pattern Flood

Too Much of a Good Thing

Scenario: Algorithms like Apriori often find 10,000+ patterns.

Issue: Most of these are Redundant or variations of the same underlying trend.

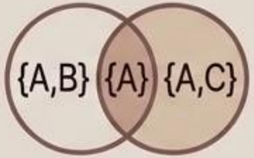
Example: {Milk}, {Milk, Bread}, {Milk, Bread, Butter} might all represent the same "Breakfast Shopper" behavior.

Goal: Represent the entire set of patterns with a few high-quality "Representatives."

Pattern Clustering Approach

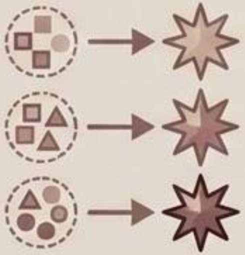
Grouping Similar Insights

Method: Treat patterns as objects and cluster them based on similarity.



Similarity Measure: Jaccard Similarity

$$\text{Similarity}(\{A,B\}, \{A,C\}) = \frac{|\{A,B\} \cap \{A,C\}|}{|\{A,B\} \cup \{A,C\}|} = \frac{\{1\}}{\{3\}}$$



Process:

- Cluster similar patterns together.
- Select the most representative pattern from each cluster (e.g., the one with the highest support or length).

Closed vs. Maximal Itemsets

Exact Compression Strategies

Closed Itemsets:

Lossless Compression: You can reconstruct the exact support of all subsets.
Cons: Can still result in a large number of patterns.

Maximal Itemsets:

Aggressive Compression: Keeps only the longest frequent patterns.
Cons: Lossy. You lose the specific support information of the subsets.
Best For: "What are all the largest possible combinations?"

Table 5.2 Subset of frequent itemsets.

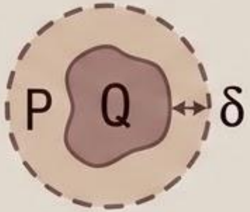
ID	Itemsets	Support
P_1	$\{b, c, d, e\}$	205,227
P_2	$\{b, c, d, e, f\}$	205,211
P_3	$\{a, b, c, d, e, f\}$	101,758
P_4	$\{a, c, d, e, f\}$	161,563
P_5	$\{a, c, d, e\}$	161,576

Example 5.7. Shortcomings of closed itemsets and maximal itemsets for compression. Table 5.2 shows a subset of frequent itemsets on a large data set, where *a, b, c, d, e, f* represent individual items. There is no nonclosed itemset here; therefore we cannot use closed frequent itemsets to compress the data. The only maximal frequent itemset is P_3. However, we observe that itemsets P_2, P_3, and P_4 are significantly different with respect to their support counts. If we were to use P_3 to represent a compressed version of the data, we would lose this support count information entirely. Consider the two pairs (P_1, P_2) and (P_4, P_5). From visual inspection, the patterns within each pair are very similar with respect to their support and expression. Therefore intuitively, P_2, P_3, and P_4, collectively, should serve as a better compressed version of the data. □

δ-Clustering (Delta-Cover)

Approximate Compression

Concept:

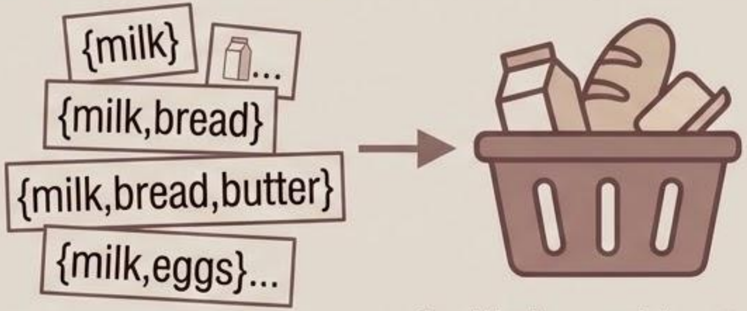


A pattern P covers pattern Q if the support of P is “close enough” to the support of Q (within a tolerance δ).

Advantage: Flexible Compression. It allows for a much smaller set of representatives by ignoring minor fluctuations in support counts.

Real-World Need:

Instead of listing: Just give me:



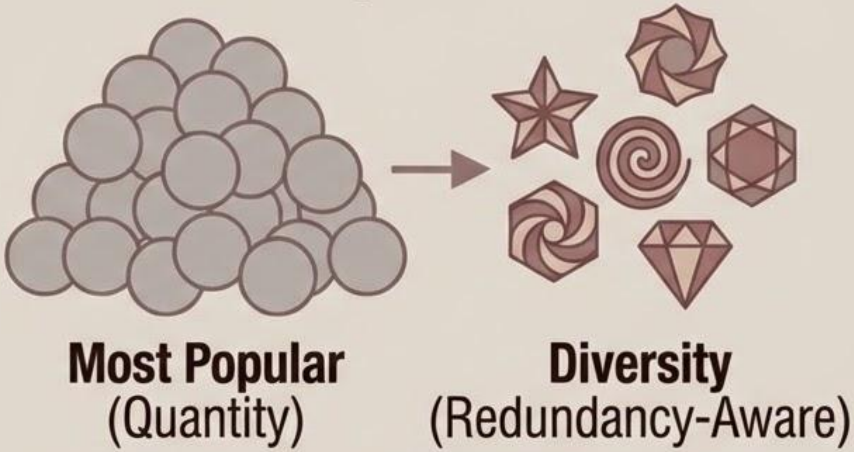
Instead of listing: {milk,bread,butter} (which effectively covers the others).

Extracting Redundancy-Aware Top-k Patterns
Section 5.2.2

Diversity Over Quantity

Solving the “Top-k” Problem

Why the "most popular" patterns are often the most boring.



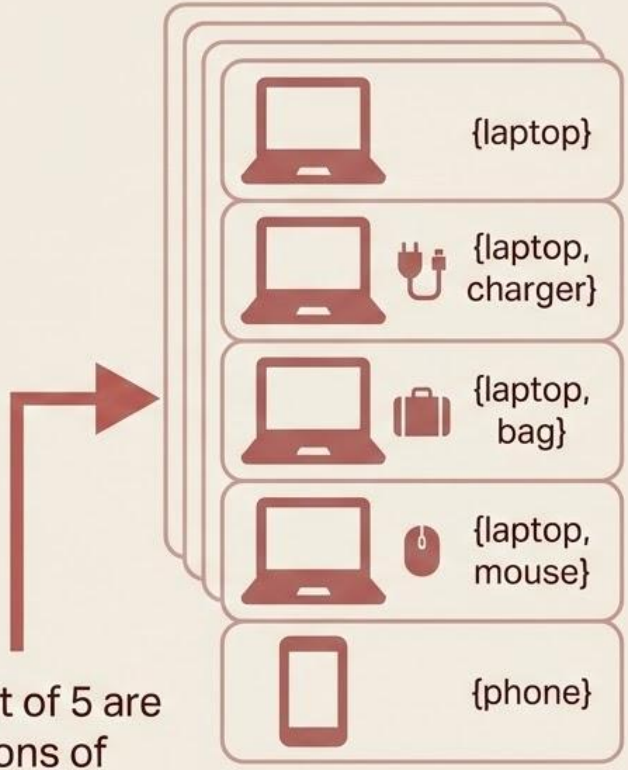
The Problem with Traditional Top-*k*

High Support ≠ High Value

Traditional Approach:
Select the k patterns with the absolute highest support.

The Result: Redundancy!

Top-5 List:



Issue: 4 out of 5 are just variations of "Laptop." The user learns nothing new.

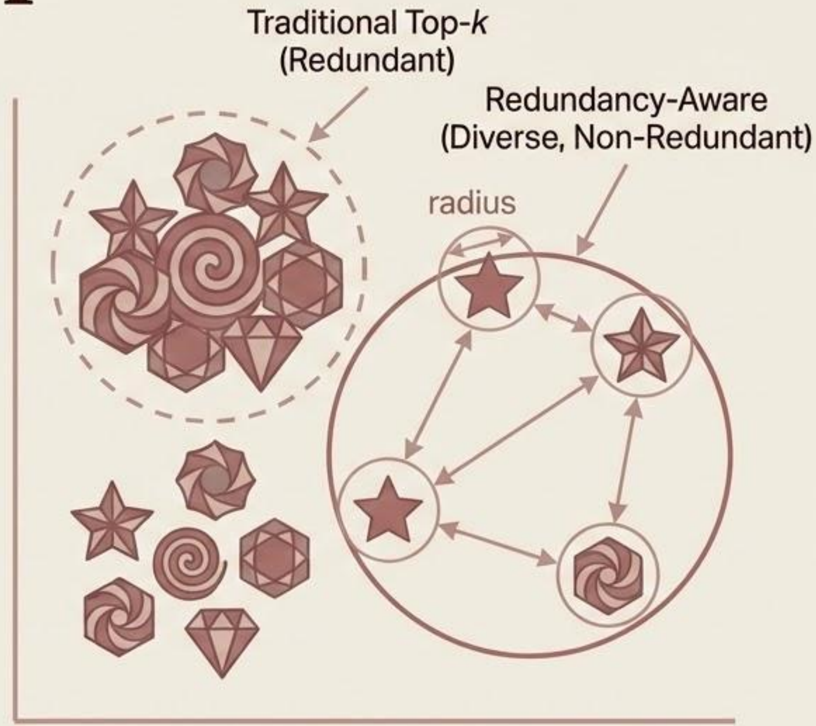
Redundancy-Aware Top-*k*

Seeking Diversity

Goal: Find a set of patterns that are Diverse and Non-Redundant, even if they have slightly lower support.

Approaches:

- (Maximal Marginal Relevance): Balances Relevance (Support) and Novelty (Difference from selected).
- **Coverage-based**: Selects patterns that cover the maximum number of unique transactions.
- **Pattern Distance**: Selects patterns that are geometrically “far” from each other. $Distance({A,B}, {A,C}) = 1 - Jaccard Similarity



Larger Size = Higher Support
Greater Distance = More Diverse

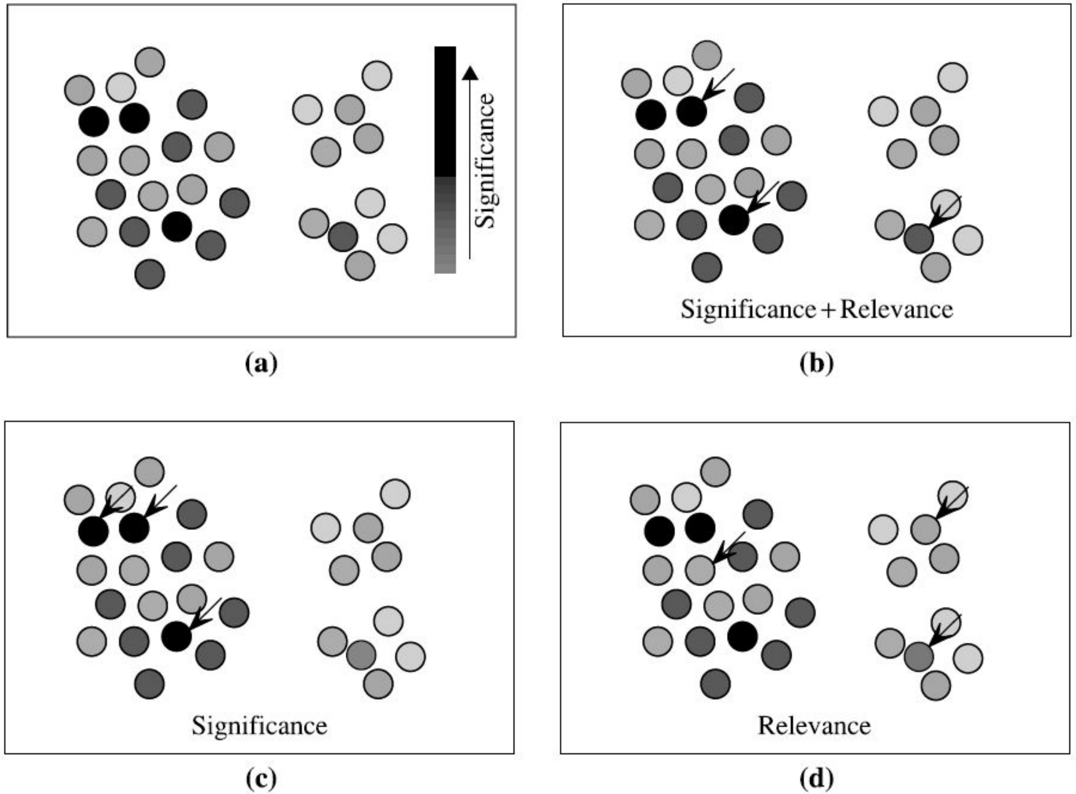


FIGURE 5.7

Conceptual view comparing top-*k* methodologies (where gray levels represent pattern significance, and the closer that two patterns are displayed, the more redundant they are to one another): (a) original patterns, (b) redundancy-aware top-*k* patterns, (c) traditional top-*k* patterns, and (d) *k*-summarized patterns.

Example 5.9. Redundancy-aware top-k strategy vs. other top-k strategies. Fig. 5.7 illustrates the intuition behind *redundancy-aware top-k patterns* vs. *traditional top-k patterns* and *k-summarized patterns*. Suppose we have the frequent patterns set shown in Fig. 5.7(a), where each circle represents a pattern of which the significance is colored in grayscale. The distance between two circles reflects the redundancy of the two corresponding patterns: The closer the circles are, the more redundant the respective patterns are to one another. Let's say we want to find three patterns that will best represent the given set, that is, $k = 3$. Which three should we choose?

Arrows are used to show the patterns chosen if using redundancy-aware top-k patterns (Fig. 5.7b), traditional top-k patterns (Fig. 5.7c), or k-summarized patterns (Fig. 5.7d). In Fig. 5.7(c), the **traditional top-k strategy** relies solely on significance: It selects the three most significant patterns to represent the set.

In Fig. 5.7(d), the ***k*-summarized pattern strategy** selects patterns based solely on nonredundancy. It detects three clusters and finds the most representative patterns to be the "centermost"' pattern from each cluster. These patterns are chosen to represent the data. The selected patterns are considered "summarized patterns" in the sense that they represent or "provide a summary" of the clusters they stand for.

By contrast, in Fig. 5.7(b) the **redundancy-aware top-*k* patterns** make a trade-off between significance and redundancy. The three patterns chosen here have high significance and low redundancy. Observe, for example, the two highly significant patterns that, based on their redundancy, are displayed next to each other. The redundancy-aware top-*k* strategy selects only one of them, taking into consideration that two would be redundant. To formalize the definition of redundancy-aware top-*k* patterns, we need to define the concepts of significance and redundancy. □

Algorithm Sketch

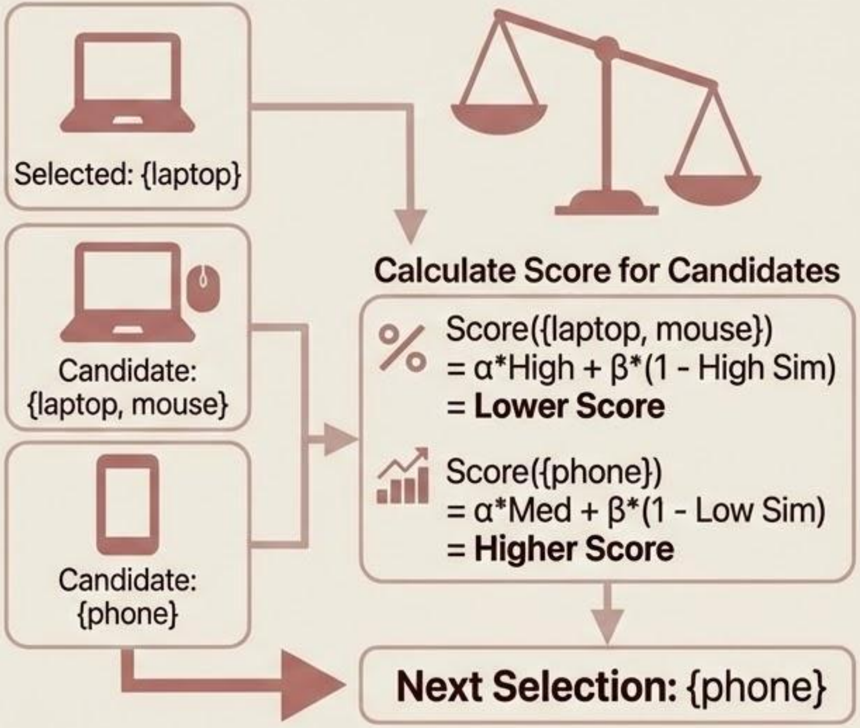
Balancing Score

Step-by-Step:
Step 1: Start with the highest support pattern.
Step 2: For subsequent selections, calculate a composite score:

$$Score = \alpha \times \text{Support} + \beta \times (1 - \text{Max Similarity to Selected})$$

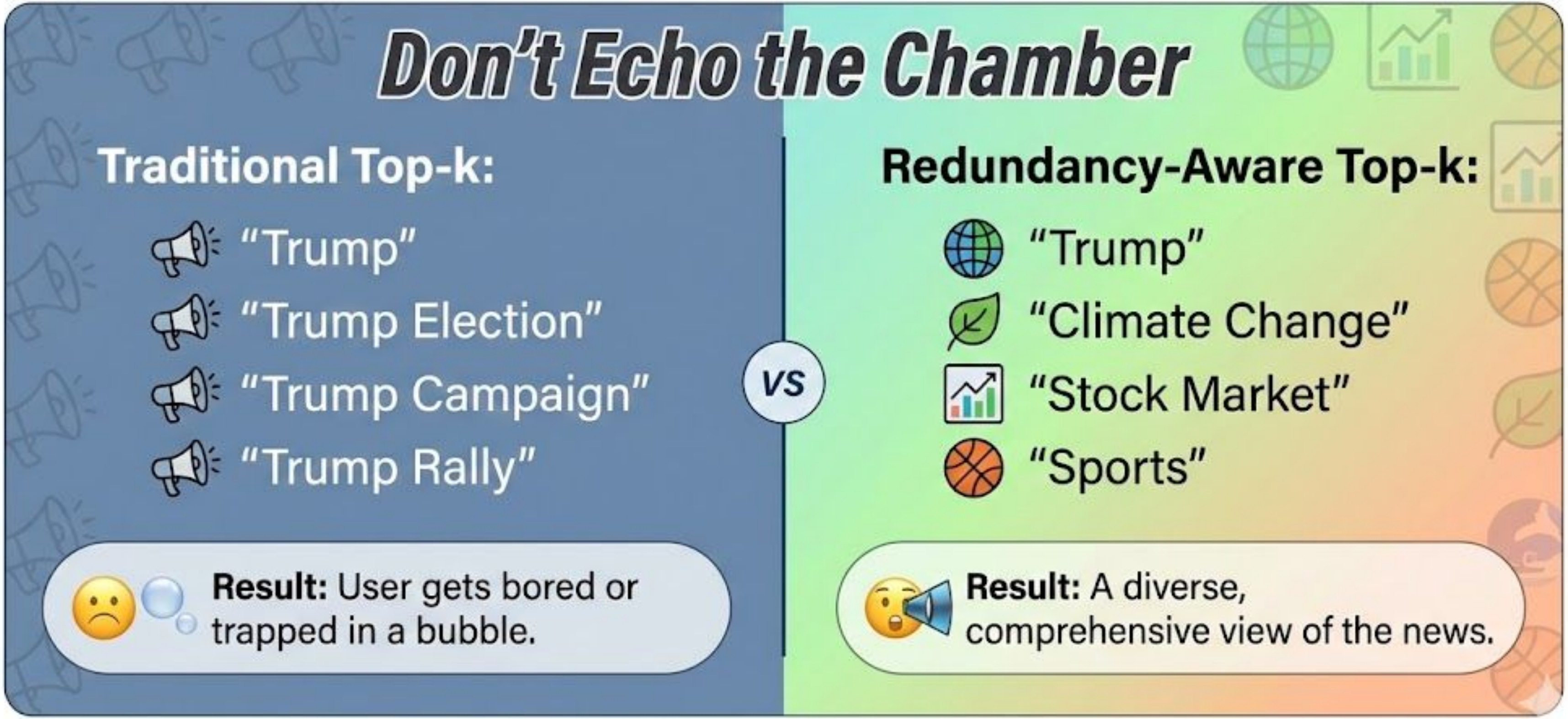
Step 3: Select the pattern with the highest combined score.

Mechanism:



As you pick more 'Laptop' patterns, the Similarity penalty increases, forcing the algorithm to pick 'Phone' or 'Table' next.

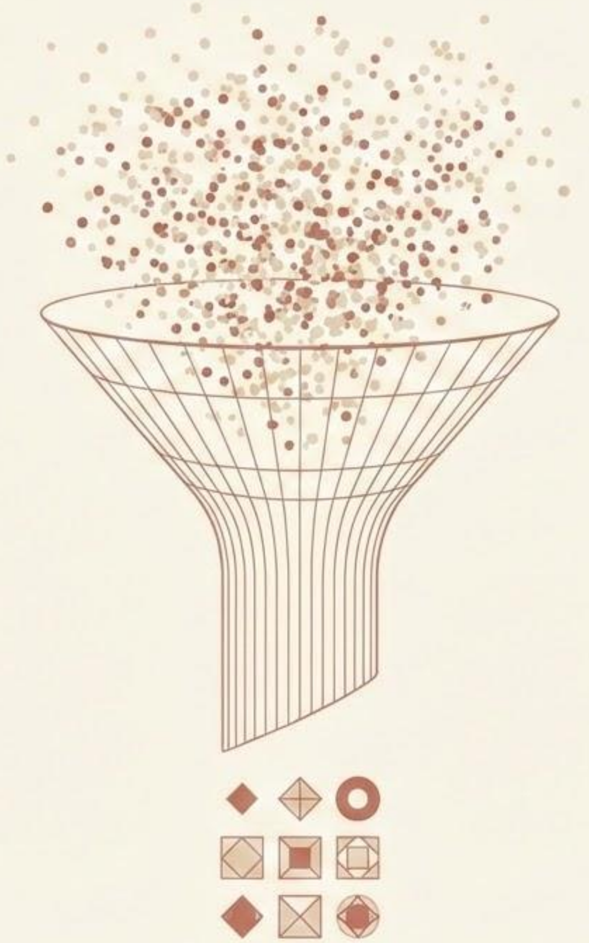
Real-World Example: News Recommendations



Mining with Purpose

Focusing the Search with User Knowledge

Why mine everything when you only care about specific patterns?



Why Constraints? Targeted Discovery

Slide 2

The Motivation:

Users often know exactly what they are looking for.

The Benefit:

Instead of sifting through thousands of irrelevant rules, constraints allow us to focus the algorithm on specific subsets of data or item types.

Efficiency:

Drastically reduces the search space and processing time.

Types of Constraints (1/2)

Filtering the Input

Slide 3

Knowledge Constraints:

Mine only patterns containing specific attributes.

Example: Patterns containing ‘organic’ or ‘gluten-free’.

Data Constraints:

Mine only from a specific subset of the database.

Example: Transactions from 2023 only.

Dimension/Level Constraints:

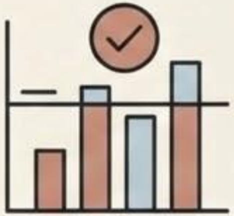
Mine only at a specific level of the hierarchy.

Example: Product Category level (e.g., ‘Dairy’) rather than individual items (e.g., ‘Milk 2%’).

Types of Constraints (2/2)

Filtering the Output

Slide 4



Rule Constraints:

Mine only rules that meet specific statistical thresholds.

Example: Confidence > 80%.

Interestingness Constraints:

Mine only patterns that are statistically "interesting."

Example: Lift > 2.0.

Classification of Constraints

The Mechanics of Pruning

Slide 5

Anti-monotonic Constraints:

If a constraint fails for an itemset, it fails for all supersets.

Example: min_sup. If {A,B} is infrequent, {A,B,C} is guaranteed to be infrequent.
Benefit: Enables massive Pruning.

Monotonic Constraints:

If a constraint holds for an itemset, it holds for all supersets.

Example: contains("electronics"). If {A} is electronic, {A,B} contains electronics.

Convertible Constraints:

Can be turned into anti-monotonic constraints by ordering the items.

Example: avg(price) > $50.

Table 5.3 Characterization of commonly used pattern pruning constraints.

Constraint	Antimonotonic	Monotonic	Succinct
$v \in S$	no	yes	yes
$S \supseteq V$	no	yes	yes
$S \subseteq V$	yes	no	yes
$min(S) \leq v$	no	yes	yes
$min(S) \geq v$	yes	no	yes
$max(S) \leq v$	yes	no	yes
$max(S) \geq v$	no	yes	yes
$count(S) \leq v$	yes	no	no
$count(S) \geq v$	no	yes	no
$sum(S) \leq v$ $(\forall a \in S, a \geq 0)$	yes	no	no
$sum(S) \geq v$ $(\forall a \in S, a \geq 0)$	no	yes	no
$range(S) \leq v$	yes	no	no
$range(S) \geq v$	no	yes	no
$avg(S)\ \theta\ v, \theta \in \{\leq, \geq\}$	convertible	convertible	no
$support(S) \geq \xi$	yes	no	no
$support(S) \leq \xi$	no	yes	no
$all\_confidence(S) \geq \xi$	yes	no	no
$all\_confidence(S) \leq \xi$	no	yes	no

Constraint Pushing

Optimization Strategy

Slide 6

The Strategy:

Apply constraints during the mining process, not after.

The Goal:

Prune the search space as early as possible.

The Result:

Dramatic speedup in algorithm performance.

Example SQL-like Constraint Language:

```sql
MINE PATTERNS AS rules
FROM transactions
WHERE items CONTAINS ("laptop" OR "tablet")
HAVING SUPPORT >= 0.01 AND CONFIDENCE >= 0.7
   AND CONSEQUENT CONTAINS ("accessory")
```

Part 4: Summary & Applications

Choosing the Right Tool

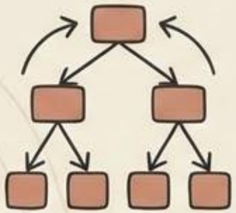
Algorithm Selection, Real-World Use Cases, and Future Trends

Wrapping up Advanced Association Rule Mining.

Algorithm Selection Guide (Data Characteristics)

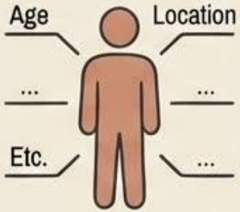
Matching Method to Data

Slide 2



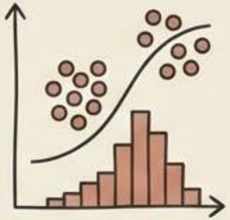
Hierarchical Data:

Use Multilevel Mining with reduced support thresholds.



Customer Profiling:

Use Multidimensional Mining to include age, location, etc.



Numeric Data:

Use Quantitative Association Rules with discretization or clustering.

Dense Data:

Use High-Dimensional Methods (FP-Growth, Vertical Formats) to handle complexity.

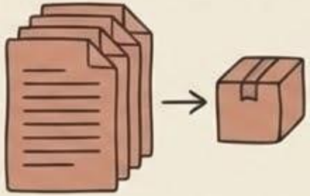
Algorithm Selection Guide (Goal-Oriented)

Matching Method to Objective

Slide 3

Rare Events:

Use Multiple Minimum Support or Constraint-Based Mining.



Compressed Results:

Use Pattern Clustering or Redundancy-Aware Top-k.

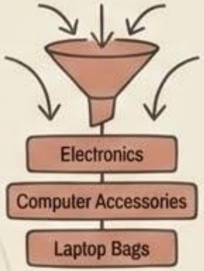
Focused Mining:

Use Constraint-Based Approaches to target specific items (e.g., "Only organic food").

Real-World Applications: E-Commerce

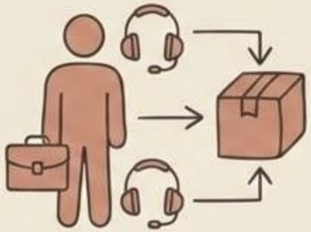
Selling Smarter

Slide 4



Multilevel:

Pattern: "Electronics -> Computer Accessories -> Laptop Bags"
Value: Cross-selling at the right granularity.



Multidimensional:

Pattern: "Young urban professionals -> Premium headphones"
Value: Targeted marketing based on demographics.

Rare Patterns:

Pattern: "Limited edition items -> Collector bundles"
Value: Identifying niche, high-value customer segments.

Real-World Applications: Healthcare

Saving Lives with Data

Slide 5

Quantitative Rules:

Pattern: "Age 60+, BP > 140/90 -> Stroke risk"
Value: Risk stratification and preventative care.

Rare Patterns:

Pattern: "Rare disease symptom combinations"
Value: Early diagnosis of obscure conditions.

Negative Patterns:

Pattern: "Drug A with Drug B -> Adverse reaction"
Value: Preventing dangerous drug interactions.

Real-World Applications: Finance

Managing Risk & Portfolio

Slide 6

High-Dimensional Mining:

Pattern: "1000+ stock correlations"
Value: Identifying hidden dependencies between assets.

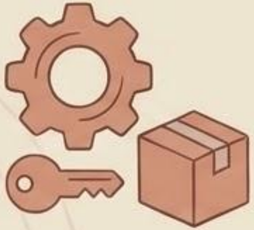
Constraint-Based Mining:

Pattern: "Only patterns involving tech stocks"
Value: Focused portfolio analysis for sector-specific strategies.

Key Insights

The Big Picture

Slide 7



One size doesn't fit all:
Different data needs different methods.

Constraints are friends:
Use domain knowledge to guide the mining process; it saves time and improves relevance.

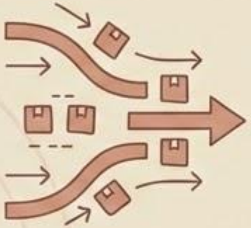
Quality over Quantity:
It is better to have 10 interesting patterns than 1,000 redundant ones.

Context matters:
The same pattern means different things in different contexts (e.g., a "Rare" pattern is noise in retail but a crisis in banking).

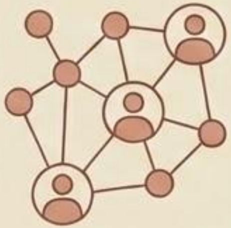
Future Directions

Where We Are Going

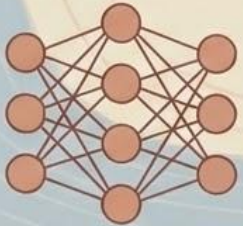
Slide 8



Streaming Pattern Mining: Finding patterns in real-time data streams (IoT, Sensors).



Graph-Based Patterns: Analyzing relationships beyond simple transactions (Social Networks).



Deep Learning Integration: Using Neural Networks for complex pattern discovery.

Interactive Mining: Keeping the "Human-in-the-loop" for exploration.