# **Study Notes: Data Mining - Chapter 4**

## **Data Warehousing and Online Analytical Processing (OLAP)**

### **1. What is a Data Warehouse?**

* **Definition:** A **decision support database** maintained separately from operational databases.
* **Key Characteristics (Inmon’s Definition):**
  + **Subject-Oriented:** Focuses on specific subjects like sales, customers.
  + **Integrated:** Consolidates data from multiple sources with standardized formats.
  + **Time-Variant:** Stores historical data for long-term analysis.
  + **Nonvolatile:** Data is stable and only updated periodically.

### **2. Data Warehouse Architecture**

* **Multi-Tiered Structure:**
  1. **Bottom Tier:** Data warehouse database (relational database system).
  2. **Middle Tier:** OLAP server (ROLAP/MOLAP).
  3. **Top Tier:** Front-end tools (querying, reporting, and analysis).
* **Three Data Warehouse Models:**
  + **Enterprise Warehouse:** Covers entire organization.
  + **Data Mart:** Subset of data warehouse for specific departments (e.g., Marketing).
  + **Virtual Warehouse:** Views created dynamically from operational databases.

### **3. OLTP vs. OLAP**

| Feature | OLTP (Online Transaction Processing) | OLAP (Online Analytical Processing) |
| --- | --- | --- |
| Purpose | Manage transactions (CRUD operations) | Analyze data for decision-making |
| Operations | Read/write frequent, small queries | Complex, read-heavy queries |
| Data Scope | Operational, current data | Historical, aggregated data |
| Performance Focus | Fast query execution | High computational efficiency |

### **4. Data Warehouse Modeling: Data Cubes and OLAP**

* **Multidimensional Data Model:** Data is stored in a **data cube**, allowing multiple perspectives of the same data.
* **Components:**
  + **Fact Table:** Contains measures (e.g., sales amount).
  + **Dimension Tables:** Define perspectives (e.g., time, location, product).
* **Schemas for Data Warehouses:**
  + **Star Schema:** Fact table linked to multiple dimension tables.
  + **Snowflake Schema:** Normalized version of the star schema.
  + **Fact Constellation:** Multiple fact tables sharing dimension tables.

### **5. Data Cube Operations**

* **Roll-up (Drill-up):** Aggregates data to a higher level.
* **Drill-down:** Breaks down data into finer details.
* **Slice:** Selects a single dimension subset.
* **Dice:** Selects multiple dimensions to form a subcube.
* **Pivot (Rotate):** Changes the view of data for better analysis.

### **6. Efficient Data Cube Computation**

* **Precomputing Aggregates:**
  + **Materialization Strategies:**
    - **No Materialization:** Compute on-the-fly (slow).
    - **Full Materialization:** Precompute all cuboids (storage-intensive).
    - **Partial Materialization:** Compute only frequently used cuboids.
  + **Compute Cube Operator:**
    - SQL-like syntax:
    - SELECT item, city, year, SUM(sales)  
      FROM sales  
      CUBE BY item, city, year;

### **7. OLAP Server Architectures**

* **ROLAP (Relational OLAP):** Uses relational databases; scalable but slower.
* **MOLAP (Multidimensional OLAP):** Uses specialized storage; fast but storage-intensive.
* **HOLAP (Hybrid OLAP):** Combines ROLAP and MOLAP for balance.

### **8. Indexing OLAP Data**

* **Bitmap Indexing:** Efficient for low-cardinality attributes (e.g., gender, category).
* **Join Indexing:** Precomputes joins between fact and dimension tables.

### **9. OLAP Query Optimization**

* **Choosing Materialized Cuboids:** Optimize query processing by selecting precomputed summaries.
* **Efficient Query Processing Strategies:**
  + Use the smallest relevant cuboid.
  + Prune unnecessary computations.
  + Apply indexing techniques.

### **10. Online Analytical Mining (OLAM)**

* **Integrates OLAP with Data Mining.**
* **Advantages:**
  + Uses **cleaned, structured data** from data warehouses.
  + Enables **interactive mining** (drill-down into patterns).
  + Enhances data visualization.

## **Summary**

* **Data warehouses** provide structured, historical data for decision-making.
* **OLAP operations** allow efficient analysis of multidimensional data.
* **Schemas (Star, Snowflake, Fact Constellation)** organize warehouse data.
* **Data cube materialization** improves performance but has trade-offs.
* **OLAP servers (ROLAP, MOLAP, HOLAP)** differ in performance vs. storage trade-offs.
* **OLAM** enhances OLAP with data mining for deeper insights.

# **Study Notes: Data Mining - Chapter 6**

## **Frequent Pattern Mining, Association Rules, and Correlation Analysis**

### **1. What is Frequent Pattern Mining?**

* **Definition:** A frequent pattern is a set of items, sequences, or structures that **appear frequently** in a dataset.
* **First introduced** by Agrawal, Imielinski, and Swami (1993) in the context of **association rule mining**.
* **Applications:**
  + Market Basket Analysis
  + Web Log Analysis
  + DNA Sequence Analysis
  + Social Network Mining

### **2. Market Basket Analysis**

* **Goal:** Identify **associations** between items frequently bought together.
* **Example:**
  + **Association Rule:** {Laptop} → {Mouse} (If a laptop is bought, a mouse is also likely bought)
  + **Support:** Probability that both items appear together in transactions.
  + **Confidence:** Probability that a transaction containing {Laptop} also contains {Mouse}.

### **3. Key Measures for Association Rules**

1. **Support:**  
   Measures how frequently an itemset appears in the dataset.
2. **Confidence:**  
   Measures how often **B** appears in transactions containing **A**.
3. **Lift:**  
   Measures how much more likely **A and B** occur together compared to **independent** occurrences.  
   * **Lift > 1:** A and B are positively correlated.
   * **Lift < 1:** A and B are negatively correlated.

### **4. Frequent Itemsets and Rule Generation**

* **Frequent Itemset:** A set of items appearing together in at least **min\_support** transactions.
* **Strong Rules:** Rules that meet **min\_support** and **min\_confidence**.
* **Example:**
  + **Transactions:**
  + {Milk, Bread, Diaper} {Milk, Bread} {Milk, Diaper} {Bread, Diaper}
  + **Frequent Itemsets (min\_support = 50%):**
  + {Milk, Bread} → 50% {Milk, Diaper} → 50%

### **5. Apriori Algorithm (Breadth-First Search)**

* **Key Idea:** Uses the **downward closure property** – if an itemset is **frequent**, then all its subsets must also be frequent.
* **Steps:**
  1. Find frequent **1-itemsets**.
  2. Generate candidate **2-itemsets** from 1-itemsets.
  3. Keep itemsets with **min\_support**.
  4. Repeat for **k-itemsets** until no more frequent itemsets exist.
* **Limitations:**
  + Multiple database scans.
  + High computational cost for large datasets.

### **6. FP-Growth Algorithm (Depth-First Search)**

* **Key Idea:** Uses **tree structures (FP-Tree)** to avoid candidate generation.
* **Steps:**
  1. Construct an **FP-Tree** (compressed representation of transactions).
  2. Use **recursive pattern growth** to mine frequent itemsets.
  + **Advantages over Apriori:**
    - Faster (avoids candidate generation).
    - Uses less memory.

### **7. Correlation Analysis & Alternative Interestingness Measures**

* **Limitations of Confidence:** Can be misleading when items are **independent**.
* **Alternative Measures:**
  + **Kulczynski Measure:**
  + **Cosine Similarity:**
  + **Chi-Square Test:**
  + **Interest Factor:**

### **8. Null Transactions and Null-Invariance**

* **Null Transactions:** Transactions that do not contain **A** or **B**.
* **Issue:** Some measures (like Lift) are **not null-invariant**, meaning they are affected by the number of null transactions.
* **Null-Invariant Measures:** Kulczynski, Cosine, Jaccard.

## **Summary**

* **Frequent pattern mining** identifies relationships between items in transactions.
* **Support, confidence, and lift** are key metrics for association rules.
* **Apriori algorithm** uses **candidate generation**, while **FP-Growth** eliminates it with **tree-based mining**.
* **Correlation measures** like **Kulczynski, Chi-Square, and Interest Factor** provide alternative interestingness criteria.

# **Study Notes: Data Mining - Chapter 7**

## **Advanced Frequent Pattern Mining**

### **1. Pattern Mining: A Road Map**

* **Traditional pattern mining** focuses on frequent itemsets and association rules.
* **Advanced pattern mining** extends this to:
  + **Multi-level and multi-dimensional pattern mining**
  + **Constraint-based mining**
  + **Mining rare, negative, and colossal patterns**
  + **Compressed or approximate pattern mining**
  + **Pattern exploration and semantic annotation**

### **2. Multi-Level and Multi-Dimensional Pattern Mining**

#### **Multi-Level Association Rules**

* **Definition:** Association rules that span different levels of abstraction (e.g., categories vs. subcategories).
* **Example:**
  + {Milk} → {Bread} (higher level)
  + {2% Milk} → {Wheat Bread} (lower level)
* **Mining Strategy:**
  + **Top-down approach:** Compute frequent itemsets level by level.
  + **Flexible min-support:** Different thresholds for different levels.

#### **Multi-Dimensional Association Rules**

* **Single-dimensional rule:**
* **Multi-dimensional rule:**
* **Handling Different Attributes:**
  + **Categorical attributes:** Use data cube approaches.
  + **Quantitative attributes:** Use discretization, clustering.

#### **Quantitative Association Rules**

* **Deals with numeric attributes like price, age, salary.**
* **Common Techniques:**
  + **Static discretization:** Predefined intervals.
  + **Dynamic discretization:** Based on data distribution.
  + **Clustering-based methods:** Groups similar values.

### **3. Rare Patterns and Negative Association Rules**

* **Rare Patterns:** Patterns that occur below the traditional min-support threshold but are still interesting.
  + Example: Buying **diamonds and luxury watches** together.
* **Negative Association Rules:** Items that rarely appear together.
  + Example: {SUV} → NOT {Hybrid Car}.
* **Mining Strategies:**
  + Lowering support for rare items.
  + Identifying statistically significant negative correlations.

### **4. Constraint-Based Frequent Pattern Mining**

* **Why Constraints?**
  + Finding **all** patterns is unrealistic due to **explosion of results**.
  + Users specify constraints to **narrow the search**.
* **Types of Constraints:**
  + **Knowledge-type constraints:** Define the type of patterns to mine (e.g., association vs. clustering).
  + **Data constraints:** Filter data before mining (e.g., region = “North America”).
  + **Rule constraints:** Define conditions for acceptable rules (e.g., min\_confidence > 60%).
  + **Interestingness constraints:** Require certain statistical properties (e.g., lift > 1.5).

### **5. Mining High-Dimensional and Colossal Patterns**

* **High-dimensional data challenges:**
  + Too many attributes → exponentially large search space.
  + **Sparse data** makes frequent pattern mining inefficient.
* **Colossal Patterns:**
  + Very large frequent patterns (e.g., size > 50 items).
  + **Pattern-Fusion Strategy:**
    - **Merges smaller patterns** instead of discovering all frequent itemsets.
    - **Jump search space efficiently** to find colossal patterns.

### **6. Mining Compressed or Approximate Patterns**

* **Why Compression?**
  + Too many frequent patterns → redundancy.
  + Need **compact representations** without losing key information.
* **Compression Techniques:**
  + **δ-Clusters:** Patterns that share similar transactions.
  + **Maximal frequent patterns:** Largest frequent patterns without sub-pattern redundancy.
  + **Closed frequent patterns:** Patterns with no super-patterns having the same support.

### **7. Semantic Pattern Annotation and Exploration**

* **Frequent patterns without context may not be useful.**
* **Semantic Annotation:** Assign meaning to patterns based on:
  + **Co-occurrence with other patterns.**
  + **Context in transactions.**
  + **User-defined meta-rules.**
* **Example:** In medical databases, {diabetes, hypertension} → {stroke} may be more meaningful with patient demographics.

## **Summary**

* **Advanced frequent pattern mining** extends traditional association rules with **multi-level, multi-dimensional**, and **constraint-based** approaches.
* **Rare and negative patterns** can reveal **unexpected insights**.
* **Colossal patterns** require specialized **pattern-fusion** techniques.
* **Pattern compression** improves interpretability.
* **Semantic annotation** adds meaning to frequent patterns.

Let me know if you need any modifications! 🚀