Chapter 6: Data Mining Methods: Unsupervised Methods

Sunil Regmi, Lecturer

DoAI, Kathmandu University

Outline

- **Association Rules Mining**
  - The Apriori algorithm
  - Evaluation of association rules (interestingness measures)
- **Cluster Analysis**
  - K-means algorithm
  - K-medoids algorithm
  - Expectation Maximization (EM) algorithm
  - DBSCAN algorithm
  - Distance-based agglomerative and divisive clustering
  - Cluster validation: Intrinsic and extrinsic methods

Association Rules Mining

- Association analysis is a popular method for discovering interesting relationships between variables in large databases.
- Example: Market basket analysis
  - Analyzes customer buying habits by finding associations between items purchased together
  - Example: $\{\text{Diaper}\} \rightarrow \{\text{Beer}\}$
- Applications:
  - Basket data analysis, cross-marketing, catalog design, loss-leader analysis, web log analysis, DNA sequence analysis, etc.

Market-Basket Transactions

- Given a set of transactions, find rules that will predict the occurrence of an item based on the occurrences of other items in the transaction

TID	Items
1	Bread, Milk
2	Bread, Diaper, Beer, Eggs
3	Milk, Diaper, Beer, Coke
4	Bread, Milk, Diaper, Beer
5	Bread, Milk, Diaper, Coke

Table: Market-Basket transactions

- Example of Association Rules
  - $\{\text{Diaper}\} \rightarrow \{\text{Beer}\}$,
  - $\{\text{Milk, Bread}\} \rightarrow \{\text{Eggs, Coke}\}$,
  - $\{\text{Beer, Bread}\} \rightarrow \{\text{Milk}\}$,
- Implication means co-occurrence, not causality!

Frequent Itemset and Market-Basket Transactions

Definition: Frequent Itemset

- **Itemset**
  - A collection of one or more items
    - Example: {Milk, Bread, Diaper}
  - k-itemset
    - An itemset that contains k items
- **Support count** (σ)
  - Frequency of occurrence of an itemset
  - E.g. $\sigma(\{\text{Milk, Bread, Diaper}\}) = 2$
- **Support**
  - Fraction of transactions that contain an itemset
  - E.g. $s(\{\text{Milk, Bread, Diaper}\}) = 2/5$
- **Frequent Itemset**
  - An itemset whose support is greater than or equal to a *minsup* threshold

TID	Items
1	Bread, Milk
2	Bread, Diaper, Beer, Eggs
3	Milk, Diaper, Beer, Coke
4	Bread, Milk, Diaper, Beer
5	Bread, Milk, Diaper, Coke

Table: Market-Basket Transactions

Definition: Association Rule

- **Association Rule**
  - An implication expression of the form $X \rightarrow Y$, where X and Y are itemsets
  - Example: $\{\text{Milk, Diaper}\} \rightarrow \{\text{Beer}\}$
- **Rule Evaluation Metrics**
  - **Support** (s)
    - Fraction of transactions that contain both X and Y
  - **Confidence** (c)
    - Measures how often items in Y appear in transactions that contain X
- **Example:**
  - $\{\text{Milk, Diaper}\} \Rightarrow \{\text{Beer}\}$

$$s = \frac{\sigma(\{\text{Milk, Diaper, Beer}\})}{|\mathcal{T}|} = \frac{2}{5} = 0.4$$

$$c = \frac{\sigma(\{\text{Milk, Diaper, Beer}\})}{\sigma(\{\text{Milk, Diaper}\})} = \frac{2}{3} \approx 0.67$$

Association Rule Mining: Goal

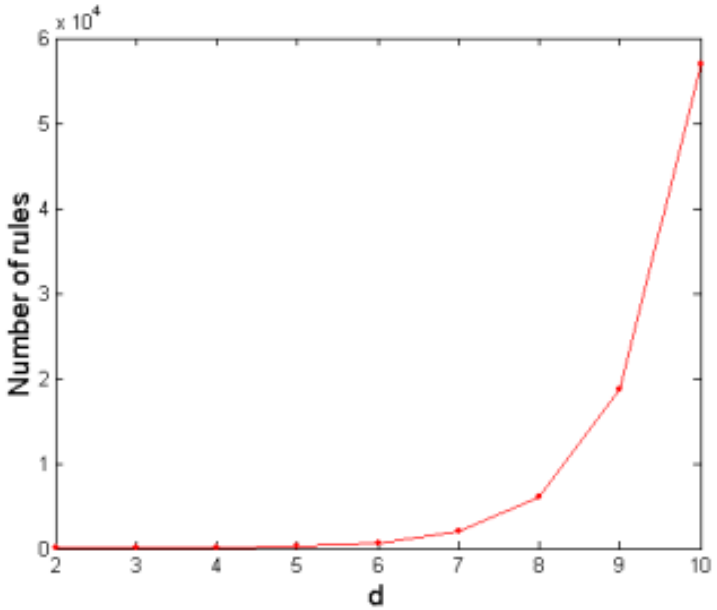
- **Given a set of transactions** T, the goal is to find all rules with:
  - Support $\geq$ *minsup* threshold
    - *minsup*: Minimum support threshold, the fraction of transactions containing the itemset (e.g., *minsup* $= 0.3$ means $\geq 30\%$ of transactions include the itemset)
  - Confidence $\geq$ *minconf* threshold
    - *minconf*: Minimum confidence threshold, the fraction of transactions with the antecedent that also contain the consequent (e.g., *minconf* $= 0.7$ means $\geq 70\%$ of such transactions include the consequent)

Association Rule Mining: Brute-Force Approach

- **Brute-force approach**:
  - List all possible association rules
  - Compute support and confidence for each rule
  - Prune rules failing *minsup* and *minconf* thresholds
  - $\Rightarrow$ Computationally prohibitive!

Computational Complexity

- Given d unique items:
  - Total number of itemsets $= 2^d$
  - Total number of possible association rules:



$$R = \sum_{k=1}^{d-1}\left[\binom{d}{k} \times \sum_{j=1}^{d-k}\binom{d-k}{j}\right]$$

$$= 3^d - 2^{d+1} + 1$$

If d=6, R = 602 rules

Mining Association Rules

TID	Items
1	Bread, Milk
2	Bread, Diaper, Beer, Eggs
3	Milk, Diaper, Beer, Coke
4	Bread, Milk, Diaper, Beer
5	Bread, Milk, Diaper, Coke

Example of Rules:

{Milk,Diaper} → {Beer} (s=0.4, c=0.67)
{Milk,Beer} → {Diaper} (s=0.4, c=1.0)
{Diaper,Beer} → {Milk} (s=0.4, c=0.67)
{Beer} → {Milk,Diaper} (s=0.4, c=0.67)
{Diaper} → {Milk,Beer} (s=0.4, c=0.5)
{Milk} → {Diaper,Beer} (s=0.4, c=0.5)

Observations:

- All the above rules are binary partitions of the same itemset:
  {Milk, Diaper, Beer}
- Rules originating from the same itemset have identical support but can have different confidence
- Thus, we may decouple the support and confidence requirements

Introduction to Apriori Algorithm

- A basic method used in data analysis to find groups of items that often appear together in large sets of data.
- Helps to discover useful patterns or rules about how items are related, particularly valuable in market basket analysis.
- Example:
  - In a grocery store, if many customers buy bread and butter together, the store can:
    - Place these items closer.
    - Create special offers.
  - Helps the store sell more and make customers happy.

Apriori Principle

Definition

If an itemset is **frequent**, then all of its **subsets** must also be frequent.

- If {c, d, e} is frequent, then:

  All subsets: $\{c, d\}, \{c, e\}, \{d, e\}, \{c\}, \{d\}, \{e\}$ must also be frequent.

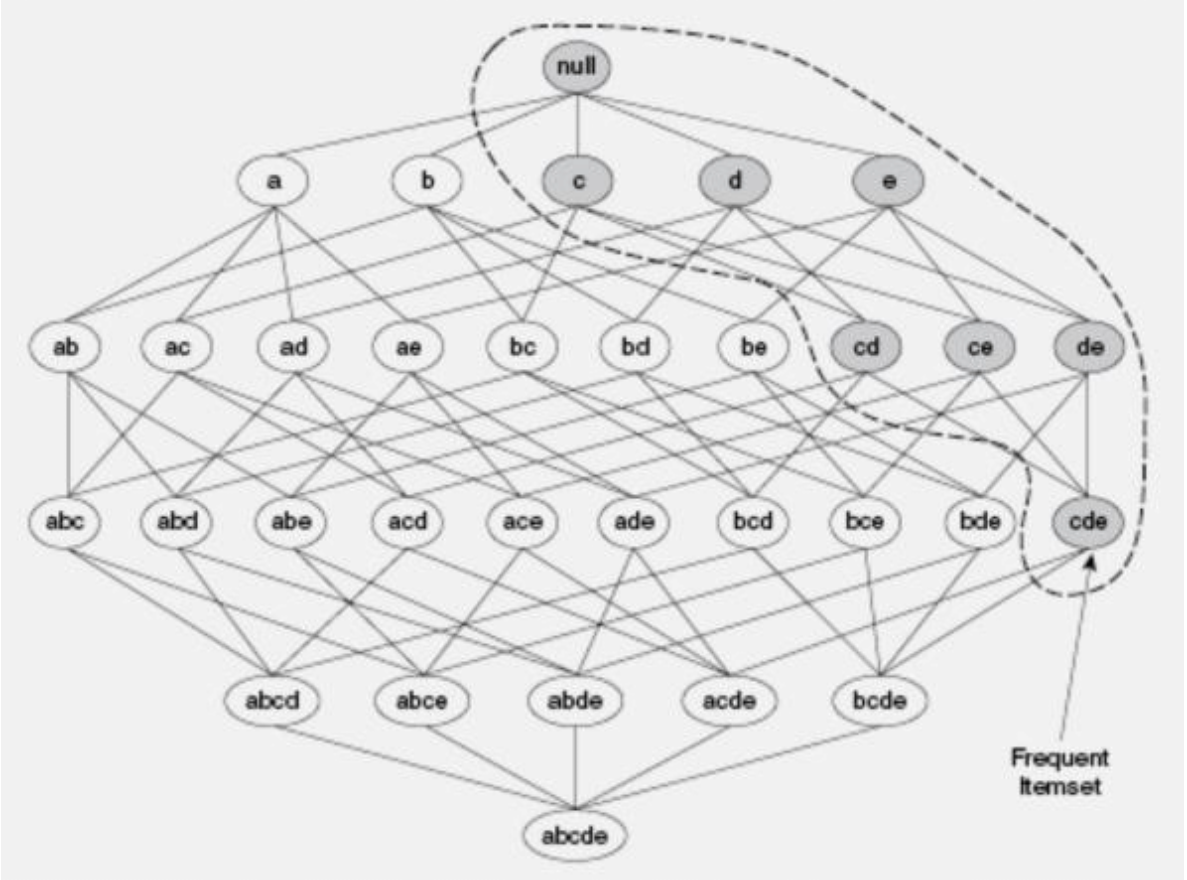
- This property is used to prune the search space.

Illustration of Apriori Principle

- Let {c, d, e} be a frequent itemset.
- Any transaction containing {c, d, e} must also contain:

$$\{c\}, \{d\}, \{e\}, \{c, d\}, \{c, e\}, \{d, e\}$$

- So all of these subsets must appear frequently.



Support-Based Pruning

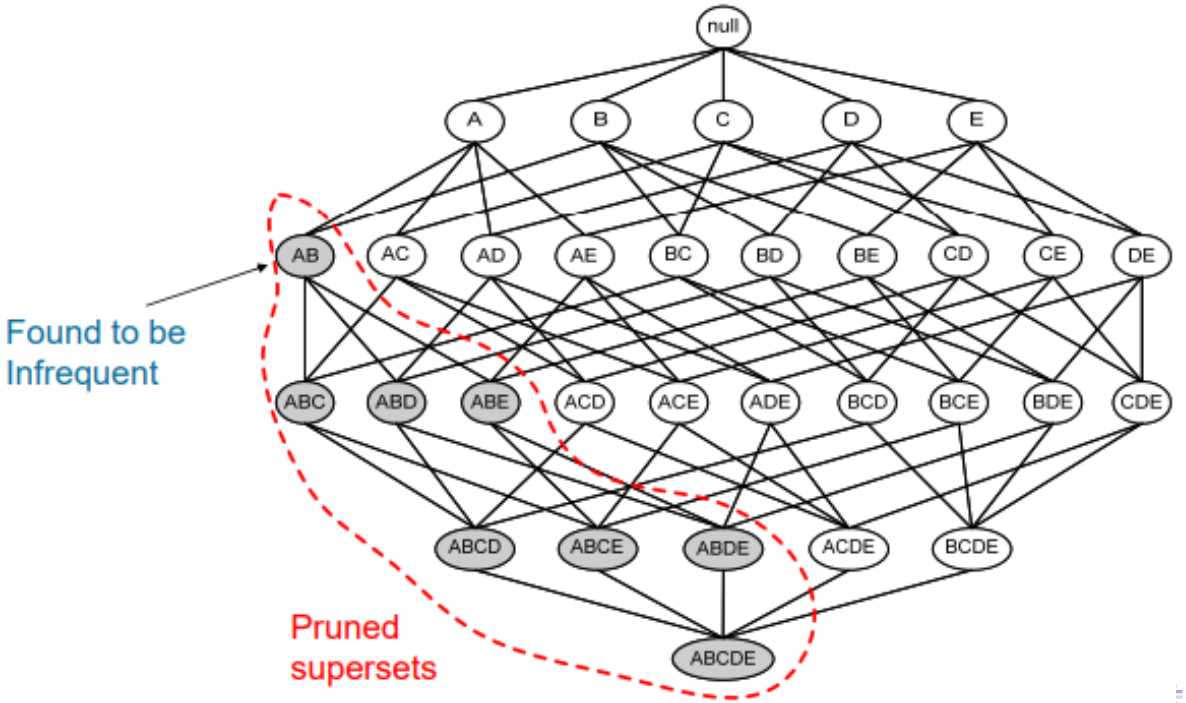
- Conversely, if {a, b} is infrequent, then any superset:

$$\{a, b, x\}, \{a, b, y\}, \ldots \text{ will also be infrequent.}$$

- This is called the **anti-monotone property** of support:

$$\text{support}(X) \geq \text{support}(X \cup Y)$$

- This allows pruning of entire branches of the candidate search space.



Apriori Algorithm Steps

1. Let $k = 1$
2. Generate frequent itemsets of length 1
3. Repeat until no new frequent itemsets:
   - 3.1 Generate candidate itemsets of length $k + 1$
   - 3.2 Prune candidates containing infrequent subsets
   - 3.3 Count support of candidates by scanning the database
   - 3.4 Eliminate infrequent candidates

Apriori Iteration Cycle

- Start with frequent 1-itemsets: **L1**
- Use L1 to generate C2 (candidate 2-itemsets)
- Prune C2 to get L2 (frequent 2-itemsets)
- Repeat:

$$L_k \Rightarrow C_{k+1} \Rightarrow \text{Prune} \Rightarrow L_{k+1}$$

- Stop when no more frequent itemsets are found.

How the Apriori Algorithm Works

- The Apriori algorithm operates through a systematic process involving several key steps:
  1. Identifying Frequent Itemsets
  2. Creating Possible Item Groups
  3. Removing Infrequent Item Groups
  4. Generating Association Rules

Step 1: Identifying Frequent Itemsets

- The algorithm starts by looking through all the data to count how many times each single item appears (1-itemsets).
- Uses a rule called *minimum support*:
  - A number that tells us how often an item or group of items needs to appear to be important.
  - If an item appears often enough (count $\geq$ minimum support), it is called a *frequent itemset*.

Step 2: Creating Possible Item Groups

- After finding frequent 1-item groups, the algorithm combines them to create pairs of items (2-item groups).
- Checks which pairs are frequent by seeing if they appear enough times in the data.
- This process continues step-by-step, making groups of 3 items, then 4 items, and so on.
- Stops when it can't find any bigger groups that happen often enough.

Step 3: Removing Infrequent Item Groups

- Uses a helpful rule to save time:
  - If a group of items does not appear often enough, any larger group that includes these items will also not appear often. (refer to above figures.)
- The algorithm does not check those larger groups, avoiding wasted time and making the process faster.

Step 4: Generating Association Rules

- The algorithm makes rules to show how items are related.
- Checks these rules using:
  - *Support*: Fraction of transactions containing the rule's items.
  - *Confidence*: How often the rule's consequent appears when the antecedent is present.
  - *Lift*: Measures how much more often the rule occurs compared to if the items were independent.
- Finds the strongest rules based on these metrics.

Apriori Algorithm

1: $C_1 \leftarrow$ all 1-itemsets (candidates)
2: $F_1 \leftarrow$ frequent 1-itemsets with support $\geq$ minsup
3: $k \leftarrow 2$
4: **while** $F_{k-1} \neq \emptyset$ **do**
5: $\quad C_k \leftarrow$ candidates generated from F_{k-1}
6: $\quad$ **for** each transaction t **do**
7: $\quad\quad$ **for** each candidate $c \in C_k$ **do**
8: $\quad\quad\quad$ **if** $c \subseteq t$ **then** increment count of c
9: $\quad\quad\quad$ **end if**
10: $\quad\quad$ **end for**
11: $\quad$ **end for**
12: $\quad F_k \leftarrow$ candidates in C_k with support $\geq$ minsup
13: $\quad k \leftarrow k + 1$
14: **end while**
15: **return** all frequent itemsets F_k

- Uses anti-monotone property: if an itemset is infrequent, all supersets are infrequent.

Dataset and Parameters

Transactions (Grocery Shop)

T1	Bread, Butter, Milk
T2	Bread, Milk
T3	Butter, Milk
T4	Bread, Butter
T5	Bread, Milk

- Minimum Support Threshold: **50%** (itemset must appear in at least 3 out of 5 transactions)
- Minimum Confidence Threshold: **70%**

Extracting Itemsets from Transactions

- **1-itemsets:** All single items appearing in transactions.
  Example: $\{Bread\}, \{Milk\}, \{Diaper\}, \{Beer\}, \{Coke\}$
- **2-itemsets:** All pairs of items that appear together in at least one transaction.
  Example:
  $\{Bread, Milk\}, \{Bread, Diaper\}, \{Milk, Diaper\}, \{Beer, Coke\}, \ldots$
- **3-itemsets:** All triplets of items appearing together in transactions.
  Example: $\{Bread, Milk, Diaper\}, \{Milk, Diaper, Beer\}, \ldots$

Step 1: Frequent 1-Itemsets

Item	Support Count	Support %
Bread	4	80%
Butter	3	60%
Milk	4	80%

- All items meet minimum support threshold (50%).
- Frequent 1-itemsets: {Bread}, {Butter}, {Milk}.

Step 3: Candidate 3-Itemsets

- Generate 3-itemset candidate from frequent 2-itemsets:

$$\{Bread, Butter, Milk\}$$

- Calculate support:

Itemset	Support Count	Support %
{Bread, Butter, Milk}	2	40%

- Does not meet minimum support threshold.
- No frequent 3-itemsets.

Step 4: Generate Association Rules from Frequent Itemsets

- Rules are generated from frequent itemsets with confidence 70%.

Rule 1: Bread $\Rightarrow$ Butter

- Support($\{$Bread, Butter$\}$) = 2
- Support($\{$Bread$\}$) = 4
- Confidence = 2/4 = 50% (Fails threshold)

Rule 2: Butter $\Rightarrow$ Bread

- Support($\{$Bread, Butter$\}$) = 2
- Support($\{$Butter$\}$) = 3
- Confidence = 2/3 66.7% (Fails threshold)

Rule 3: Bread $\Rightarrow$ Milk

- Support($\{$Bread, Milk$\}$) = 3
- Support($\{$Bread$\}$) = 4
- Confidence = 3/4 = 75% (Passes threshold)

More Association Rules

Rule 4: Milk $\Rightarrow$ Bread

- Support($\{$Bread, Milk$\}$) $= 3$
- Support($\{$Milk$\}$) $= 4$
- Confidence $= 3/4 = 75\%$ (Passes threshold)

Rule 5: Butter $\Rightarrow$ Milk

- Support($\{$Butter, Milk$\}$) $= 3$
- Support($\{$Butter$\}$) $= 3$
- Confidence $= 3/3 = 100\%$ (Passes threshold)

Rule 6: Milk $\Rightarrow$ Butter

- Support($\{$Butter, Milk$\}$) $= 3$
- Support($\{$Milk$\}$) $= 4$
- Confidence $= 3/4 = 75\%$ (Passes threshold)

Summary

- Frequent itemsets are generated by pruning candidates that don't meet support.
- Association rules are generated from frequent itemsets with high confidence.
- In this example, strong rules include:

$$\text{Bread} \Rightarrow \text{Milk}, \quad \text{Milk} \Rightarrow \text{Bread}, \quad \text{Butter} \Rightarrow \text{Milk}$$

- Apriori helps find useful shopping patterns efficiently.

Why Evaluate Association Rules?

- Not all generated rules are useful or meaningful.
- Evaluation helps identify **interesting** and **valuable** rules.
- Measures quantify strength, significance, and usefulness of rules.

Basic Metrics: Support and Confidence

- **Support** of rule $X \Rightarrow Y$:

$$\text{support}(X \Rightarrow Y) = P(X \cup Y)$$

  - How frequently X and Y appear together.

- **Confidence** of rule $X \Rightarrow Y$:

$$\text{confidence}(X \Rightarrow Y) = P(Y|X) = \frac{P(X \cup Y)}{P(X)}$$

  - Probability of Y given X.

- *Limitations:* Confidence can be misleading without considering overall frequency of Y.

Lift (Interest)

- Measures how much more often X and Y occur together than expected if they were independent.

$$\text{lift}(X \Rightarrow Y) = \frac{P(X \cup Y)}{P(X) \times P(Y)}$$

- Interpretation:
  - lift $= 1$: X and Y are independent.
  - lift > 1: Positive correlation (rule is meaningful).
  - lift < 1: Negative correlation (rule is unlikely).

Conviction

- Measures the degree of implication of the rule, focusing on how often Y fails when X occurs.

$$\text{conviction}(X \Rightarrow Y) = \frac{1 - P(Y)}{1 - \text{confidence}(X \Rightarrow Y)}$$

- Interpretation:
  - Higher conviction (>1) means stronger implication.
  - Conviction $= 1$ means independence.
  - Conviction < 1 indicates inverse implication.

Other Measures

- **Leverage**:

$$\text{leverage}(X \Rightarrow Y) = P(X \cup Y) - P(X)P(Y)$$

  - Difference between observed and expected co-occurrence.

- **J-Measure**:

$$J(X \Rightarrow Y) = P(X \cup Y) \log \frac{P(Y|X)}{P(Y)} + P(X \cup \neg Y) \log \frac{P(\neg Y|X)}{P(\neg Y)}$$

  - Information-theoretic measure.

- **Kulczynski Measure**, **Gini Index**, and others.

Summary

- **Support** and **Confidence** are basic and widely used but have limitations.
- **Lift** adjusts for the base frequency of Y and reveals true associations.
- **Conviction** gives an asymmetric perspective focusing on rule implication failure.
- Choosing the right measure depends on the application and goals.