CSE 5243 INTRO. TO DATA MINING

Mining Frequent Patterns and Associations: Basic Concepts

(Chapter 6)

Huan Sun, CSE@The Ohio State University

Slides adapted from Prof. Jiawei Han @UIUC, Prof. Srinivasan Parthasarathy @OSU
Mining Frequent Patterns, Association and Correlations: Basic Concepts and Methods

- Basic Concepts
- Efficient Pattern Mining Methods
- Pattern Evaluation
- Summary
Pattern Discovery: Basic Concepts

- What Is Pattern Discovery? Why Is It Important?
- Basic Concepts: Frequent Patterns and Association Rules
- Compressed Representation: Closed Patterns and Max-Patterns
What Is Pattern Discovery?

- **Motivation examples:**
  - What products were often purchased together?
  - What are the subsequent purchases after buying an iPad?
  - What code segments likely contain copy-and-paste bugs?
  - What word sequences likely form phrases in this corpus?
What Is Pattern Discovery?

- **Motivation examples:**
  - What products were often purchased together?
  - What are the subsequent purchases after buying an iPad?
  - What code segments likely contain copy-and-paste bugs?
  - What word sequences likely form phrases in this corpus?

- **What are patterns?**
  - **Patterns:** A set of items, subsequences, or substructures that occur frequently together (or strongly correlated) in a data set
  - Patterns represent *intrinsic* and *important properties* of datasets
What Is Pattern Discovery?

- Motivation examples:
  - What products were often purchased together?
  - What are the subsequent purchases after buying an iPad?
  - What code segments likely contain copy-and-paste bugs?
  - What word sequences likely form phrases in this corpus?

- What are patterns?
  - Patterns: A set of items, subsequences, or substructures that occur frequently together (or strongly correlated) in a data set
  - Patterns represent intrinsic and important properties of datasets

- Pattern discovery: Uncovering patterns from massive data sets
Pattern Discovery: Why Is It Important?

- Finding inherent regularities in a data set
- Foundation for many essential data mining tasks
  - Association, correlation, and causality analysis
  - Mining sequential, structural (e.g., sub-graph) patterns
  - Pattern analysis in spatiotemporal, multimedia, time-series, and stream data
  - Classification: Discriminative pattern-based analysis
  - Cluster analysis: Pattern-based subspace clustering
Pattern Discovery: Why Is It Important?

- Finding inherent regularities in a data set
- Foundation for many essential data mining tasks
  - Association, correlation, and causality analysis
  - Mining sequential, structural (e.g., sub-graph) patterns
  - Pattern analysis in spatiotemporal, multimedia, time-series, and stream data
  - Classification: Discriminative pattern-based analysis
  - Cluster analysis: Pattern-based subspace clustering
- Broad applications
  - Market basket analysis, cross-marketing, catalog design, sale campaign analysis, Web log analysis, biological sequence analysis
Basic Concepts: k-Itemsets and Their Supports

- **Itemset**: A set of one or more items
Basic Concepts: k-Itemsets and Their Supports

- **Itemset**: A set of one or more items
- **k-itemset**: \( X = \{x_1, \ldots, x_k\} \)
  - Ex. \{Beer, Nuts, Diaper\} is a 3-itemset

<table>
<thead>
<tr>
<th>Tid</th>
<th>Items bought</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>Beer, Nuts, Diaper</td>
</tr>
<tr>
<td>20</td>
<td>Beer, Coffee, Diaper</td>
</tr>
<tr>
<td>30</td>
<td>Beer, Diaper, Eggs</td>
</tr>
<tr>
<td>40</td>
<td>Nuts, Eggs, Milk</td>
</tr>
<tr>
<td>50</td>
<td>Nuts, Coffee, Diaper, Eggs, Milk</td>
</tr>
</tbody>
</table>
Basic Concepts: k-Itemsets and Their Supports

- **Itemset**: A set of one or more items
- **k-itemset**: \( X = \{x_1, ..., x_k\} \)
  - Ex. \{Beer, Nuts, Diaper\} is a 3-itemset

- **(absolute) support (count)** of \( X \), \( \sup\{X\} \):
  Frequency or the number of occurrences of an itemset \( X \)
  - Ex. \( \sup\{\text{Beer}\} = 3 \)
  - Ex. \( \sup\{\text{Diaper}\} = 4 \)
  - Ex. \( \sup\{\text{Beer, Diaper}\} = 3 \)
  - Ex. \( \sup\{\text{Beer, Eggs}\} = 1 \)

<table>
<thead>
<tr>
<th>Tid</th>
<th>Items bought</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>Beer, Nuts, Diaper</td>
</tr>
<tr>
<td>20</td>
<td>Beer, Coffee, Diaper</td>
</tr>
<tr>
<td>30</td>
<td>Beer, Diaper, Eggs</td>
</tr>
<tr>
<td>40</td>
<td>Nuts, Eggs, Milk</td>
</tr>
<tr>
<td>50</td>
<td>Nuts, Coffee, Diaper, Eggs, Milk</td>
</tr>
</tbody>
</table>
Basic Concepts: k-Itemsets and Their Supports

- **Itemset**: A set of one or more items
- **k-itemset**: \( X = \{x_1, ..., x_k\} \)
  - Ex. \{Beer, Nuts, Diaper\} is a 3-itemset

- **(absolute) support (count)** of \( X \), \( \text{sup}\{X\} \): Frequency or the number of occurrences of an itemset \( X \)
  - Ex. \( \text{sup}\{\text{Beer}\} = 3 \)
  - Ex. \( \text{sup}\{\text{Diaper}\} = 4 \)
  - Ex. \( \text{sup}\{\text{Beer, Diaper}\} = 3 \)
  - Ex. \( \text{sup}\{\text{Beer, Eggs}\} = 1 \)

- **(relative) support**, \( s\{X\} \): The fraction of transactions that contains \( X \) (i.e., the probability that a transaction contains \( X \))
  - Ex. \( s\{\text{Beer}\} = 3/5 = 60\% \)
  - Ex. \( s\{\text{Diaper}\} = 4/5 = 80\% \)
  - Ex. \( s\{\text{Beer, Eggs}\} = 1/5 = 20\% \)

<table>
<thead>
<tr>
<th>Tid</th>
<th>Items bought</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>Beer, Nuts, Diaper</td>
</tr>
<tr>
<td>20</td>
<td>Beer, Coffee, Diaper</td>
</tr>
<tr>
<td>30</td>
<td>Beer, Diaper, Eggs</td>
</tr>
<tr>
<td>40</td>
<td>Nuts, Eggs, Milk</td>
</tr>
<tr>
<td>50</td>
<td>Nuts, Coffee, Diaper, Eggs, Milk</td>
</tr>
</tbody>
</table>
Basic Concepts: Frequent Itemsets (Patterns)

- An itemset (or a pattern) \( X \) is \textit{frequent} if the support of \( X \) is no less than a \textit{minsup} threshold \( \sigma \)
Basic Concepts: Frequent Itemsets (Patterns)

- An itemset (or a pattern) $X$ is **frequent** if the support of $X$ is no less than a $\text{minsup}$ threshold $\sigma$.

- Let $\sigma = 50\%$ ($\sigma$: $\text{minsup}$ threshold)

  For the given 5-transaction dataset:
  - All the frequent 1-itemsets:
    - Beer: 3/5 (60%); Nuts: 3/5 (60%)
    - Diaper: 4/5 (80%); Eggs: 3/5 (60%)
  - All the frequent 2-itemsets:
    - $\{\text{Beer, Diaper}\}$: 3/5 (60%)
  - All the frequent 3-itemsets?
    - None

<table>
<thead>
<tr>
<th>Tid</th>
<th>Items bought</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>Beer, Nuts, Diaper</td>
</tr>
<tr>
<td>20</td>
<td>Beer, Coffee, Diaper</td>
</tr>
<tr>
<td>30</td>
<td>Beer, Diaper, Eggs</td>
</tr>
<tr>
<td>40</td>
<td>Nuts, Eggs, Milk</td>
</tr>
<tr>
<td>50</td>
<td>Nuts, Coffee, Diaper, Eggs, Milk</td>
</tr>
</tbody>
</table>
Basic Concepts: Frequent Itemsets (Patterns)

- An itemset (or a pattern) $X$ is *frequent* if the support of $X$ is no less than a *minsup* threshold $\sigma$.

- Let $\sigma = 50\%$ ($\sigma$: minsup threshold)
  For the given 5-transaction dataset:
  - All the frequent 1-itemsets:
    - Beer: 3/5 (60%); Nuts: 3/5 (60%)
    - Diaper: 4/5 (80%); Eggs: 3/5 (60%)
  - All the frequent 2-itemsets:
    - {Beer, Diaper}: 3/5 (60%)
  - All the frequent 3-itemsets?
    - None

<table>
<thead>
<tr>
<th>Tid</th>
<th>Items bought</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>Beer, Nuts, Diaper</td>
</tr>
<tr>
<td>20</td>
<td>Beer, Coffee, Diaper</td>
</tr>
<tr>
<td>30</td>
<td>Beer, Diaper, Eggs</td>
</tr>
<tr>
<td>40</td>
<td>Nuts, Eggs, Milk</td>
</tr>
<tr>
<td>50</td>
<td>Nuts, Coffee, Diaper, Eggs, Milk</td>
</tr>
</tbody>
</table>

- Do these itemsets (shown on the left) form the complete set of frequent $k$-itemsets (patterns) for any $k$?

- **Observation**: We may need an efficient method to mine a complete set of frequent patterns.
From Frequent Itemsets to Association Rules

- Comparing with itemsets, rules can be more telling
  - Ex. Diaper $\rightarrow$ Beer
    - Buying diapers may likely lead to buying beers
From Frequent Itemsets to Association Rules

- Ex. Diaper $\rightarrow$ Beer: Buying diapers may likely lead to buying beers

- How strong is this rule? (support, confidence)

- Measuring association rules: $X \rightarrow Y (s, c)$
  - Both $X$ and $Y$ are itemsets
From Frequent Itemsets to Association Rules

- Ex. Diaper → Beer: Buying diapers may likely lead to buying beers
- How strong is this rule? (support, confidence)
- Measuring association rules: \( X \rightarrow Y \) \((s, c)\)
  - Both \( X \) and \( Y \) are itemsets
  - Support, \( s \): The probability that a transaction contains \( X \cup Y \)
    - Ex. \( s\{\text{Diaper, Beer}\} = 3/5 = 0.6 \) (i.e., 60%)

<table>
<thead>
<tr>
<th>Tid</th>
<th>Items bought</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>Beer, Nuts, Diaper</td>
</tr>
<tr>
<td>20</td>
<td>Beer, Coffee, Diaper</td>
</tr>
<tr>
<td>30</td>
<td>Beer, Diaper, Eggs</td>
</tr>
<tr>
<td>40</td>
<td>Nuts, Eggs, Milk</td>
</tr>
<tr>
<td>50</td>
<td>Nuts, Coffee, Diaper, Eggs, Milk</td>
</tr>
</tbody>
</table>
From Frequent Itemsets to Association Rules

- **Ex.** *Diaper → Beer*: Buying diapers may likely lead to buying beers

- **How strong is this rule?** *(support, confidence)*

- **Measuring association rules**: $X \rightarrow Y$ *(s, c)*
  - Both $X$ and $Y$ are itemsets
  - **Support**, $s$: The probability that a transaction contains $X \cup Y$
    - Ex. $s\{\text{Diaper, Beer}\} = 3/5 = 0.6$ (i.e., 60%)
  - **Confidence**, $c$: The *conditional probability* that a transaction containing $X$ also contains $Y$
    - Calculation: $c = \text{sup}(X \cup Y) / \text{sup}(X)$
    - Ex. $c = \text{sup}\{\text{Diaper, Beer}\}/\text{sup}\{\text{Diaper}\} = ¾ = 0.75

<table>
<thead>
<tr>
<th>Tid</th>
<th>Items bought</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>Beer, Nuts, Diaper</td>
</tr>
<tr>
<td>20</td>
<td>Beer, Coffee, Diaper</td>
</tr>
<tr>
<td>30</td>
<td>Beer, Diaper, Eggs</td>
</tr>
<tr>
<td>40</td>
<td>Nuts, Eggs, Milk</td>
</tr>
<tr>
<td>50</td>
<td>Nuts, Coffee, Diaper, Eggs, Milk</td>
</tr>
</tbody>
</table>

**Diagram:**
- Containing both
- Containing beer
- Containing diaper
- $\{\text{Beer}\} \cup \{\text{Diaper}\} = \{\text{Beer, Diaper}\}$
Mining Frequent Itemsets and Association Rules

- Association rule mining
  - Given two thresholds: $\text{minsup}$, $\text{minconf}$
  - Find all of the rules, $X \rightarrow Y$ (s, c)
    - such that, $s \geq \text{minsup}$ and $c \geq \text{minconf}$
Mining Frequent Itemsets and Association Rules

- **Association rule mining**
  - Given two thresholds: \( \text{minsup}, \text{minconf} \)
  - Find all of the rules, \( X \rightarrow Y \) (s, c)
    - such that, \( s \geq \text{minsup} \) and \( c \geq \text{minconf} \)

- Let \( \text{minsup} = 50\% \)
  - Freq. 1-itemsets: Beer: 3, Nuts: 3, Diaper: 4, Eggs: 3
  - Freq. 2-itemsets: \{Beer, Diaper\}: 3

<table>
<thead>
<tr>
<th>Tid</th>
<th>Items bought</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>Beer, Nuts, Diaper</td>
</tr>
<tr>
<td>20</td>
<td>Beer, Coffee, Diaper</td>
</tr>
<tr>
<td>30</td>
<td>Beer, Diaper, Eggs</td>
</tr>
<tr>
<td>40</td>
<td>Nuts, Eggs, Milk</td>
</tr>
<tr>
<td>50</td>
<td>Nuts, Coffee, Diaper, Eggs, Milk</td>
</tr>
</tbody>
</table>
Mining Frequent Itemsets and Association Rules

- **Association rule mining**
  - Given two thresholds: \( \text{minsup}, \text{minconf} \)
  - Find all of the rules, \( X \rightarrow Y \) (s, c)
    - such that, \( s \geq \text{minsup} \) and \( c \geq \text{minconf} \)

- Let \( \text{minsup} = 50\% \)
  - Freq. 1-itemsets: Beer: 3, Nuts: 3, Diaper: 4, Eggs: 3
  - Freq. 2-itemsets: \{Beer, Diaper\}: 3

- Let \( \text{minconf} = 50\% \)
  - Beer \( \rightarrow \) Diaper (60\%, 100\%)
  - Diaper \( \rightarrow \) Beer (60\%, 75\%)

<table>
<thead>
<tr>
<th>Tid</th>
<th>Items bought</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>Beer, Nuts, Diaper</td>
</tr>
<tr>
<td>20</td>
<td>Beer, Coffee, Diaper</td>
</tr>
<tr>
<td>30</td>
<td>Beer, Diaper, Eggs</td>
</tr>
<tr>
<td>40</td>
<td>Nuts, Eggs, Milk</td>
</tr>
<tr>
<td>50</td>
<td>Nuts, Coffee, Diaper, Eggs, Milk</td>
</tr>
</tbody>
</table>
Mining Frequent Itemsets and Association Rules

- **Association rule mining**
  - Given two thresholds: \( \text{minsup}, \text{minconf} \)
  - Find all of the rules, \( X \rightarrow Y \) (s, c)
    - such that, \( s \geq \text{minsup} \) and \( c \geq \text{minconf} \)

- Let \( \text{minsup} = 50\% \)
  - Freq. 1-itemsets: Beer: 3, Nuts: 3, Diaper: 4, Eggs: 3
  - Freq. 2-itemsets: \{Beer, Diaper\}: 3

- Let \( \text{minconf} = 50\% \)
  - Beer \( \rightarrow \) Diaper (60\%, 100\%)
  - Diaper \( \rightarrow \) Beer (60\%, 75\%)

(Q: Are these all rules?)

<table>
<thead>
<tr>
<th>Tid</th>
<th>Items bought</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>Beer, Nuts, Diaper</td>
</tr>
<tr>
<td>20</td>
<td>Beer, Coffee, Diaper</td>
</tr>
<tr>
<td>30</td>
<td>Beer, Diaper, Eggs</td>
</tr>
<tr>
<td>40</td>
<td>Nuts, Eggs, Milk</td>
</tr>
<tr>
<td>50</td>
<td>Nuts, Coffee, Diaper, Eggs, Milk</td>
</tr>
</tbody>
</table>
Mining Frequent Itemsets and Association Rules

- **Association rule mining**
  - Given two thresholds: *minsup*, *minconf*
  - Find all of the rules, \( X \rightarrow Y \) (s, c)
    - such that, \( s \geq \text{minsup} \) and \( c \geq \text{minconf} \)

- Let \( \text{minsup} = 50\% \)
  - Freq. 1-itemsets: Beer: 3, Nuts: 3, Diaper: 4, Eggs: 3
  - Freq. 2-itemsets: \{Beer, Diaper\}: 3

- Let \( \text{minconf} = 50\% \)
  - Beer \( \rightarrow \) Diaper (60\%, 100\%)
  - Diaper \( \rightarrow \) Beer (60\%, 75\%)

<table>
<thead>
<tr>
<th>Tid</th>
<th>Items bought</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>Beer, Nuts, Diaper</td>
</tr>
<tr>
<td>20</td>
<td>Beer, Coffee, Diaper</td>
</tr>
<tr>
<td>30</td>
<td>Beer, Diaper, Eggs</td>
</tr>
<tr>
<td>40</td>
<td>Nuts, Eggs, Milk</td>
</tr>
<tr>
<td>50</td>
<td>Nuts, Coffee, Diaper, Eggs, Milk</td>
</tr>
</tbody>
</table>

- **Observations:**
  - Mining association rules and mining frequent patterns are very close problems
  - Scalable methods are needed for mining large datasets
Challenge: There Are Too Many Frequent Patterns!

- A long pattern contains a combinatorial number of sub-patterns
- How many frequent itemsets does the following TDB\textsubscript{1} contain?
  - TDB\textsubscript{1}: T\textsubscript{1}: \{a\textsubscript{1}, ..., a\textsubscript{50}\}; T\textsubscript{2}: \{a\textsubscript{1}, ..., a\textsubscript{100}\}
  - Assuming (absolute) $minsup = 1$
  - Let's have a try
    - 1-itemsets: \{a\textsubscript{1}\}: 2, \{a\textsubscript{2}\}: 2, ..., \{a\textsubscript{50}\}: 2, \{a\textsubscript{51}\}: 1, ..., \{a\textsubscript{100}\}: 1,
    - 2-itemsets: \{a\textsubscript{1}, a\textsubscript{2}\}: 2, ..., \{a\textsubscript{1}, a\textsubscript{50}\}: 2, \{a\textsubscript{1}, a\textsubscript{51}\}: 1 ..., ..., \{a\textsubscript{99}, a\textsubscript{100}\}: 1,
      ... ..., ...
    - 99-itemsets: \{a\textsubscript{1}, a\textsubscript{2}, ..., a\textsubscript{99}\}: 1, ..., \{a\textsubscript{2}, a\textsubscript{3}, ..., a\textsubscript{100}\}: 1
    - 100-itemset: \{a\textsubscript{1}, a\textsubscript{2}, ..., a\textsubscript{100}\}: 1
Challenge: There Are Too Many Frequent Patterns!

- A long pattern contains a combinatorial number of sub-patterns
- How many frequent itemsets does the following TDB contain?
  - TDB\(_1\): T\(_1\): \{a\(_1\), ..., a\(_{50}\)\}; T\(_2\): \{a\(_1\), ..., a\(_{100}\)\}
  - Assuming (absolute) \(\text{minsup} = 1\)
  - Let’s have a try
    - 1-itemsets: \{a\(_1\)\}: 2, \{a\(_2\)\}: 2, ..., \{a\(_{50}\)\}: 2, \{a\(_{51}\)\}: 1, ..., \{a\(_{100}\)\}: 1
    - 2-itemsets: \{a\(_1\), a\(_2\)\}: 2, ..., \{a\(_1\), a\(_{50}\)\}: 2, \{a\(_1\), a\(_{51}\)\}: 1 ..., ..., \{a\(_99\), a\(_{100}\)\}: 1
    - ..., ..., ..., ...
    - 99-itemsets: \{a\(_1\), a\(_2\), ..., a\(_{99}\)\}: 1, ..., \{a\(_2\), a\(_3\), ..., a\(_{100}\)\}: 1
    - 100-itemset: \{a\(_1\), a\(_2\), ..., a\(_{100}\)\}: 1
- The total number of frequent itemsets:
  \[
  \binom{100}{1} + \binom{100}{2} + \binom{100}{3} + \cdots + \binom{100}{100} = 2^{100} - 1
  \]
Expressing Patterns in Compressed Form: Closed Patterns

- How to handle such a challenge?
- Solution 1: **Closed patterns**: A pattern (itemset) $X$ is **closed** if $X$ is **frequent**, and there exists **no super-pattern** $Y \supset X$, with the same support as $X$
Expressing Patterns in Compressed Form: Closed Patterns

- How to handle such a challenge?

- Solution 1: **Closed patterns**: A pattern (itemset) $X$ is closed if $X$ is frequent, and there exists no super-pattern $Y \supset X$, with the same support as $X$

  - Let Transaction DB $TDB_1$: $T_1: \{a_1, \ldots, a_{50}\}; \ T_2: \{a_1, \ldots, a_{100}\}$

  - Suppose $\text{minsup} = 1$. How many closed patterns does $TDB_1$ contain?
    - Two: $P_1: \{a_1, \ldots, a_{50}\}: 2$; $P_2: \{a_1, \ldots, a_{100}\}: 1$

Why?
Expressing Patterns in Compressed Form: Closed Patterns

- How to handle such a challenge?

- **Solution 1:** **Closed patterns:** A pattern (itemset) $X$ is *closed* if $X$ is *frequent*, and there exists no *super-pattern* $Y \supset X$, with the same support as $X$.

  - Let Transaction DB $TDB_1$: $T_1: \{a_1, \ldots, a_{50}\}; \ T_2: \{a_1, \ldots, a_{100}\}$

  - Suppose $\text{minsup} = 1$. How many closed patterns does $TDB_1$ contain?
    - Two: $P_1: \"\{a_1, \ldots, a_{50}\}: 2\"; \ P_2: \"\{a_1, \ldots, a_{100}\}: 1\"

- **Closed pattern** is a *lossless compression* of frequent patterns

  - Reduces the # of patterns but does not lose the support information!

  - You will still be able to say: $\"\{a_2, \ldots, a_{40}\}: 2\", \"\{a_5, a_{51}\}: 1\"$
Expressing Patterns in Compressed Form: Max-Patterns

- Solution 2: **Max-patterns**: A pattern $X$ is a max-pattern if $X$ is frequent and there exists no frequent super-pattern $Y \supset X$
Solution 2: **Max-patterns**: A pattern \( X \) is a *max-pattern* if \( X \) is frequent and there exists no frequent super-pattern \( Y \supseteq X \)

**Difference from close-patterns?**
- Do not care the real support of the sub-patterns of a max-pattern
- Let Transaction DB \( \text{TDB}_1: T_1: \{a_1, ..., a_{50}\}; T_2: \{a_1, ..., a_{100}\} \)
- Suppose \( \text{minsop} = 1 \). How many max-patterns does \( \text{TDB}_1 \) contain?
  - One: \( P: \{a_1, ..., a_{100}\}; 1 \)
Expressing Patterns in Compressed Form: Max-Patterns

- Solution 2: **Max-patterns**: A pattern $X$ is a max-pattern if $X$ is frequent and there exists no frequent super-pattern $Y \supset X$

- Difference from close-patterns?
  - Do not care the real support of the sub-patterns of a max-pattern
  - Let Transaction DB $TDB_1$: $T_1: \{a_1, \ldots, a_{50}\}; T_2: \{a_1, \ldots, a_{100}\}$
  - Suppose $\minsup = 1$. How many max-patterns does $TDB_1$ contain?
    - One: $P: \{"a_1, \ldots, a_{100}\}: 1$”

- Max-pattern is a lossy compression!
  - We only know $\{a_1, \ldots, a_{40}\}$ is frequent
  - But we do not know the real support of $\{a_1, \ldots, a_{40}\}, \ldots$, any more!
  - Thus in many applications, close-patterns are more desirable than max-patterns
Mining Frequent Patterns, Association and Correlations: Basic Concepts and Methods

- Basic Concepts
- Efficient Pattern Mining Methods
  - The Apriori Algorithm
  - Application in Classification
- Pattern Evaluation
- Summary
Efficient Pattern Mining Methods

- The Downward Closure Property of Frequent Patterns
- The Apriori Algorithm
- Extensions or Improvements of Apriori
- Mining Frequent Patterns by Exploring Vertical Data Format
- FPGrowth: A Frequent Pattern-Growth Approach
- Mining Closed Patterns
The Downward Closure Property of Frequent Patterns

- Observation: From TDB₁: $T_1: \{a_1, \ldots, a_{50}\}; \ T_2: \{a_1, \ldots, a_{100}\}$
  - We get a frequent itemset: $\{a_1, \ldots, a_{50}\}$
  - Also, its subsets are all frequent: $\{a_1\}, \{a_2\}, \ldots, \{a_{50}\}, \{a_1, a_2\}, \ldots, \{a_1, \ldots, a_{49}\}, \ldots$
  - There must be some hidden relationships among frequent patterns!
The Downward Closure Property of Frequent Patterns

- Observation: From TDB\(_1\): T\(_1\): \{a_1, \ldots, a_{50}\}; T\(_2\): \{a_1, \ldots, a_{100}\}
  - We get a frequent itemset: \{a_1, \ldots, a_{50}\}
  - Also, its subsets are all frequent: \{a_1\}, \{a_2\}, \ldots, \{a_{50}\}, \{a_1, a_2\}, \ldots, \{a_1, \ldots, a_{49}\}, \ldots
  - There must be some hidden relationships among frequent patterns!

- The downward closure (also called “Apriori”) property of frequent patterns
  - If \{beer, diaper, nuts\} is frequent, so is \{beer, diaper\}
  - Every transaction containing \{beer, diaper, nuts\} also contains \{beer, diaper\}
  - **Apriori:** Any subset of a frequent itemset must be frequent

A sharp knife for pruning!
The Downward Closure Property of Frequent Patterns

- **Observation:** From TDB$_1$: $T_1$: $\{a_1, \ldots, a_{50}\}$; $T_2$: $\{a_1, \ldots, a_{100}\}$
  - We get a frequent itemset: $\{a_1, \ldots, a_{50}\}$
  - Also, its subsets are all frequent: $\{a_1\}$, $\{a_2\}$, $\ldots$, $\{a_{50}\}$, $\{a_1, a_2\}$, $\ldots$, $\{a_1, \ldots, a_{49}\}$, $\ldots$
  - There must be some hidden relationships among frequent patterns!

- The **downward closure** (also called “Apriori”) property of frequent patterns
  - If $\{\text{beer, diaper, nuts}\}$ is frequent, so is $\{\text{beer, diaper}\}$
  - Every transaction containing $\{\text{beer, diaper, nuts}\}$ also contains $\{\text{beer, diaper}\}$
  - **Apriori:** Any subset of a frequent itemset must be frequent

- **Efficient mining methodology**
  - If any subset of an itemset $S$ is infrequent, then there is no chance for $S$ to be frequent—why do we even have to consider $S$!?
Apriori Pruning and Scalable Mining Methods

- **Apriori pruning principle**: If there is any itemset which is infrequent, its superset should not even be generated!
  - (Agrawal & Srikant @VLDB’94, Mannila, et al. @ KDD’94)

- **Scalable mining Methods**: Three major approaches
  - Level-wise, join-based approach:
    - Apriori (Agrawal & Srikant@VLDB’94)
  - Vertical data format approach:
    - Eclat (Zaki, Parthasarathy, Ogihara, Li @KDD’97)
  - Frequent pattern projection and growth:
    - FPgrowth (Han, Pei, Yin @SIGMOD’00)
Apriori: A Candidate Generation & Test Approach

- Outline of Apriori (level-wise, candidate generation and test)
  - Initially, scan DB once to get frequent 1-itemset
  - Repeat
    - Generate length-(k+1) candidate itemsets from length-k frequent itemsets
    - Test the candidates against DB to find frequent (k+1)-itemsets
    - Set k := k + 1
  - Until no frequent or candidate set can be generated
  - Return all the frequent itemsets derived
The Apriori Algorithm (Pseudo-Code)

$C_k$: Candidate itemset of size $k$

$F_k$: Frequent itemset of size $k$

$K := 1$;
$F_k := \{\text{frequent items}\};$  // frequent 1-itemset

While ($F_k \neq \emptyset$) do {

  // when $F_k$ is non-empty
  $C_{k+1} := \text{candidates generated from } F_k;$  // candidate generation
  Derive $F_{k+1}$ by counting candidates in $C_{k+1}$ with respect to $TDB$ at $\text{minsup}$;
  $k := k + 1$
}

return $\bigcup_k F_k$  // return $F_k$ generated at each level
The Apriori Algorithm—An Example

Database TDB

<table>
<thead>
<tr>
<th>Tid</th>
<th>Items</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>A, C, D</td>
</tr>
<tr>
<td>20</td>
<td>B, C, E</td>
</tr>
<tr>
<td>30</td>
<td>A, B, C, E</td>
</tr>
<tr>
<td>40</td>
<td>B, E</td>
</tr>
</tbody>
</table>

**minsup = 2**

**1st scan**

- **$C_1$**
  - Itemset | sup |
  - {A}     | 2   |
  - {B}     | 3   |
  - {C}     | 3   |
  - {D}     | 1   |
  - {E}     | 3   |

**$F_1$**

- Itemset | sup |
  - {A}    | 2   |
  - {B}    | 3   |
  - {C}    | 3   |
  - {E}    | 3   |

**2nd scan**

- **$C_2$**
  - Itemset | sup |
  - {A, B}  | 1   |
  - {A, C}  | 2   |
  - {A, E}  | 1   |
  - {B, C}  | 2   |
  - {B, E}  | 3   |
  - {C, E}  | 2   |

- **$F_2$**
  - Itemset |
  - {A, C}  |
  - {B, C}  |
  - {B, E}  |
  - {A, E}  |

**3rd scan**

- **$C_3$**
  - Itemset |
  - {B, C, E} |

- **$F_3$**
  - Itemset | sup |
  - {B, C, E} | 2   |
The Apriori Algorithm—An Example

Database TDB

<table>
<thead>
<tr>
<th>Tid</th>
<th>Items</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>A, C, D</td>
</tr>
<tr>
<td>20</td>
<td>B, C, E</td>
</tr>
<tr>
<td>30</td>
<td>A, B, C, E</td>
</tr>
<tr>
<td>40</td>
<td>B, E</td>
</tr>
</tbody>
</table>

**minsup = 2**

1st scan

- **C₁**
  - Itemset: {A} sup: 2
  - Itemset: {B} sup: 3
  - Itemset: {C} sup: 3
  - Itemset: {D} sup: 1
  - Itemset: {E} sup: 3

- **F₁**
  - Itemset: {A} sup: 2
  - Itemset: {B} sup: 3
  - Itemset: {C} sup: 3
  - Itemset: {E} sup: 3

2nd scan

- **C₂**
  - Itemset: {A} sup: 2
  - Itemset: {A, C} sup: 2
  - Itemset: {A, E} sup: 1
  - Itemset: {B} sup: 3
  - Itemset: {B, C} sup: 2
  - Itemset: {B, E} sup: 3
  - Itemset: {C} sup: 3
  - Itemset: {C, E} sup: 2

- **F₂**
  - Itemset: {A, B} sup: 1
  - Itemset: {A, C} sup: 2
  - Itemset: {A, E} sup: 1
  - Itemset: {B, C} sup: 2
  - Itemset: {B, E} sup: 3
  - Itemset: {C, E} sup: 2

3rd scan

- **C₃**
  - Itemset: {B, C, E} sup: 2

- **F₃**
  - Itemset: {B, C, E} sup: 2

**minsup = 2**

Why?
How to generate candidates?

- Step 1: self-joining $F_k$
- Step 2: pruning
Apriori: Implementation Tricks

- How to generate candidates?
  - Step 1: self-joining $F_k$
  - Step 2: pruning

- Example of candidate-generation
  - $F_3 = \{abc, abd, acd, ace, bcd\}$
  - Self-joining: $F_3 * F_3$
    - $abcd$ from $abc$ and $abd$
    - $acde$ from $acd$ and $ace$
Apriori: Implementation Tricks

- How to generate candidates?
  - Step 1: self-joining $F_k$
  - Step 2: pruning
- Example of candidate-generation
  - $F_3 = \{abc, abd, acd, ace, bcd\}$
  - Self-joining: $F_3 \times F_3$
    - $abcd$ from $abc$ and $abd$
    - $acde$ from $acd$ and $ace$
  - Pruning:
    - $acde$ is removed because $ade$ is not in $F_3$
  - $C_4 = \{abcd\}$
Suppose the items in $F_{k-1}$ are listed in an order.

Step 1: self-joining $F_{k-1}$
- Insert into $C_k$
- Select $p.item_1, p.item_2, \ldots, p.item_{k-1}, q.item_{k-1}$
- From $F_{k-1}$ as $p$, $F_{k-1}$ as $q$
- Where $p.item_1 = q.item_1, \ldots, p.item_{k-2} = q.item_{k-2}, p.item_{k-1} < q.item_{k-1}$

Step 2: pruning
- For all itemsets $c$ in $C_k$
  - For all $(k-1)$-subsets $s$ of $c$
    - If ($s$ is not in $F_{k-1}$) then delete $c$ from $C_k$
Apriori Adv/Disadv

- **Advantages:**
  - Uses large itemset property
  - Easily parallelized
  - Easy to implement

- **Disadvantages:**
  - Assumes transaction database is memory resident
  - Requires up to m database scans
Classification based on Association Rules (CBA)

- **Why?**
  - Can effectively uncover the correlation structure in data
  - AR are typically quite scalable in practice
  - Rules are often very intuitive
    - Hence classifier built on intuitive rules is easier to interpret

- **When to use?**
  - On large dynamic datasets where class labels are available and the correlation structure is unknown.
  - Multi-class categorization problems
  - E.g. Web/Text Categorization, Network Intrusion Detection
Backup Slides
Mining Frequent Patterns, Association and Correlations: Basic Concepts and Methods

☐ Basic Concepts

☐ Efficient Pattern Mining Methods

☐ Pattern Evaluation

☐ Summary
Summary

- Basic Concepts
  - What Is Pattern Discovery? Why Is It Important?
  - Basic Concepts: Frequent Patterns and Association Rules
  - Compressed Representation: Closed Patterns and Max-Patterns

- Efficient Pattern Mining Methods
  - The Downward Closure Property of Frequent Patterns
  - The Apriori Algorithm
  - Extensions or Improvements of Apriori
  - Mining Frequent Patterns by Exploring Vertical Data Format
  - FP-Growth: A Frequent Pattern-Growth Approach
  - Mining Closed Patterns

- Pattern Evaluation
  - Interestingness Measures in Pattern Mining
  - Interestingness Measures: Lift and $\chi^2$
  - Null-Invariant Measures
  - Comparison of Interestingness Measures
Recommended Readings (Basic Concepts)

- R. Agrawal, T. Imielinski, and A. Swami, “Mining association rules between sets of items in large databases”, in Proc. of SIGMOD'93
- R. J. Bayardo, “Efficiently mining long patterns from databases”, in Proc. of SIGMOD'98
Recommended Readings
( Efficient Pattern Mining Methods )

- J. Han, J. Pei, and Y. Yin, “Mining frequent patterns without candidate generation”, SIGMOD’00
- M. J. Zaki and Hsiao, “CHARM: An Efficient Algorithm for Closed Itemset Mining”, SDM’02
- J. Wang, J. Han, and J. Pei, “CLOSET+: Searching for the Best Strategies for Mining Frequent Closed Itemsets”, KDD’03
- C. C. Aggarwal, M.A., Bhuiyan, M. A. Hasan, “Frequent Pattern Mining Algorithms: A Survey”, in Aggarwal and Han (eds.): Frequent Pattern Mining, Springer, 2014
Recommended Readings (Pattern Evaluation)

- E. Omiecinski. Alternative Interest Measures for Mining Associations. TKDE’03
- P.-N. Tan, V. Kumar, and J. Srivastava. Selecting the Right Interestingness Measure for Association Patterns. KDD'02
- T. Wu, Y. Chen and J. Han, Re-Examination of Interestingness Measures in Pattern Mining: A Unified Framework, Data Mining and Knowledge Discovery, 21(3):371-397, 2010