Set Data: Frequent Pattern Mining

Instructor: Yizhou Sun

yzsun@ccs.neu.edu

February 24, 2016
Announcement

- **Midterm**
  - Next Wednesday (3/2), 2-hour (6:10-8:10pm) in class
  - Closed-book exam, and one A4 size reference sheet is allowed
  - Bring a calculator (NO cell phone)
  - Cover to last lecture (today’s lecture will be in homework #4)

- Homework #3 is due on 3/4
- Homework #4 is out on 3/4
# Methods to Learn

<table>
<thead>
<tr>
<th></th>
<th>Matrix Data</th>
<th>Text Data</th>
<th>Set Data</th>
<th>Sequence Data</th>
<th>Time Series</th>
<th>Graph &amp; Network</th>
<th>Images</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Classification</strong></td>
<td>Decision Tree; Naïve Bayes; Logistic Regression SVM; kNN</td>
<td></td>
<td></td>
<td></td>
<td>HMM</td>
<td>Label Propagation*</td>
<td>Neural Network</td>
</tr>
<tr>
<td><strong>Clustering</strong></td>
<td>K-means; hierarchical clustering; DBSCAN; Mixture Models; kernel k-means*</td>
<td>PLSA</td>
<td></td>
<td></td>
<td></td>
<td>SCAN*; Spectral Clustering*</td>
<td></td>
</tr>
<tr>
<td><strong>Frequent Pattern Mining</strong></td>
<td></td>
<td>Apriori; FP-growth</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Prediction</strong></td>
<td>Linear Regression</td>
<td></td>
<td></td>
<td></td>
<td>Autoregression</td>
<td>Collaborative Filtering</td>
<td></td>
</tr>
<tr>
<td><strong>Similarity Search</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>DTW</td>
<td>P-PageRank</td>
<td></td>
</tr>
<tr>
<td><strong>Ranking</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>PageRank</td>
<td></td>
</tr>
</tbody>
</table>
Mining Frequent Patterns, Association and Correlations

- Basic Concepts
- Frequent Itemset Mining Methods
- Pattern Evaluation Methods
- Summary
Set Data

- A data point corresponds to a set of items

<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>
What Is Frequent Pattern Analysis?

• **Frequent pattern**: a pattern (a set of items, subsequences, substructures, etc.) that occurs frequently in a data set
  • First proposed by Agrawal, Imielinski, and Swami [AIS93] in the context of frequent itemsets and association rule mining

• **Motivation**: Finding inherent regularities in data
  • What products were often purchased together?—Beer and diapers?!
  • What are the subsequent purchases after buying a PC?
  • What kinds of DNA are sensitive to this new drug?
Why Is Freq. Pattern Mining Important?

• Freq. pattern: An intrinsic and important property of datasets
• Foundation for many essential data mining tasks
  • Association, correlation, and causality analysis
  • Sequential, structural (e.g., sub-graph) patterns
  • Pattern analysis in spatiotemporal, multimedia, time-series, and stream data
  • Classification: discriminative, frequent pattern analysis
  • Cluster analysis: frequent pattern-based clustering
• Broad applications
Basic Concepts: Frequent Patterns

- **itemset**: A set of one or more items
- **k-itemset** $X = \{x_1, \ldots, x_k\}$
- **(absolute) support**, or, **support count** of $X$: Frequency or occurrence of an itemset $X$
- **(relative) support**, $s$, is the fraction of transactions that contains $X$ (i.e., the probability that a transaction contains $X$)
- An itemset $X$ is **frequent** if $X$’s support is no less than a $\text{minsup}$ threshold

<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>

Customer buys beer  
Customer buys both  
Customer buys diaper
Basic Concepts: Association Rules

- Find all the rules \( X \rightarrow Y \) with minimum support and confidence
  - support, \( s \), probability that a transaction contains \( X \cup Y \)
  - confidence, \( c \), conditional probability that a transaction having \( X \) also contains \( Y \)

Let \( \text{minsup} = 50\% \), \( \text{minconf} = 50\% \)

Freq. Pat.: Beer:3, Nuts:3, Diaper:4, Eggs:3, \{Beer, Diaper\}:3

- **Strong Association rules**
  - \( \text{Beer} \rightarrow \text{Diaper} \) (60%, 100%)
  - \( \text{Diaper} \rightarrow \text{Beer} \) (60%, 75%)
Closed Patterns and Max-Patterns

• A long pattern contains a combinatorial number of sub-patterns, e.g., \( \{a_1, \ldots, a_{100}\} \) contains \( 2^{100} - 1 = 1.27 \times 10^{30} \) sub-patterns!

• Solution: Mine closed patterns and max-patterns instead

• An itemset \( X \) is closed if \( X \) is frequent and there exists no super-pattern \( Y \supset X \), with the same support as \( X \) (proposed by Pasquier, et al. @ ICDT’99)

• An itemset \( X \) is a max-pattern if \( X \) is frequent and there exists no frequent super-pattern \( Y \supset X \) (proposed by Bayardo @ SIGMOD’98)

• Closed pattern is a lossless compression of freq. patterns
  • Reducing the # of patterns and rules
Closed Patterns and Max-Patterns

- Exercise. DB = \{<a_1, ..., a_{100}>, < a_1, ..., a_{50}>\}
  - Min_sup = 1.

- What is the set of closed pattern(s)?
  - <a_1, ..., a_{100}>: 1
  - < a_1, ..., a_{50}>: 2

- What is the set of max-pattern(s)?
  - <a_1, ..., a_{100}>: 1

- What is the set of all patterns?
  - !!
Computational Complexity of Frequent Itemset Mining

• How many itemsets are potentially to be generated in the worst case?
  • The number of frequent itemsets to be generated is sensitive to the minsup threshold
  • When minsup is low, there exist potentially an exponential number of frequent itemsets
  • The worst case: $M^N$ where $M$: # distinct items, and $N$: max length of transactions
Mining Frequent Patterns, Association and Correlations

- Basic Concepts
- Frequent Itemset Mining Methods
- Pattern Evaluation Methods
- Summary
Scalable Frequent Itemset Mining Methods

• Apriori: A Candidate Generation-and-Test Approach
  • Improving the Efficiency of Apriori

• FP-Growth: A Frequent Pattern-Growth Approach

• ECLAT: Frequent Pattern Mining with Vertical Data Format

• Generating Association Rules
The Apriori Property and Scalable Mining Methods

• The Apriori property of frequent patterns
  • Any nonempty subsets of a frequent itemset must be frequent
  • If \{beer, diaper, nuts\} is frequent, so is \{beer, diaper\}
  • i.e., every transaction having \{beer, diaper, nuts\} also contains \{beer, diaper\}

• Scalable mining methods: Three major approaches
  • Apriori (Agrawal & Srikant@VLDB’94)
  • Freq. pattern growth (FPgrowth—Han, Pei & Yin @SIGMOD’00)
  • Vertical data format approach (Eclat)
Apriori: A Candidate Generation & Test Approach

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

- **Method**:
  - *Initially*, scan DB once to get frequent 1-itemset
  - Generate length (k+1) candidate itemsets from length k frequent itemsets
  - **Test** the candidates against DB
  - **Terminate** when no frequent or candidate set can be generated
From Frequent k-1 Itemset To Frequent k-Itemset

$C_k$: Candidate itemset of size k

$L_k$: frequent itemset of size k

- From $L_{k-1}$ to $C_k$ (Candidates Generation)
  - The join step
  - The prune step
- From $C_k$ to $L_k$
  - Test candidates by scanning database
Candidates Generation

• How to generate candidates $C_k$?
  • **Step 1**: self-joining $L_{k-1}$
    • Two length $k$-1 itemsets $l_1$ and $l_2$ can join, only if the first $k$-2 items are the same, and for the last term, $l_1[k-1] < l_2[k-1]$ (why?)
  • **Step 2**: pruning

• Why we need pruning for candidates?
  • **How**?
    • Again, use Apriori property
    • A candidate itemset can be safely pruned, if it contains infrequent subset

Assume a pre-specified order for items, e.g., alphabetical order
• Example of Candidate-generation from $L_3$ to $C_4$
  • $L_3 = \{abc, abd, acd, ace, bcd\}$
  • Self-joining: $L_3 \times L_3$
    • $abcd$ from $abc$ and $abd$
    • $acde$ from $acd$ and $ace$
  • Pruning:
    • $acde$ is removed because $ade$ is not in $L_3$
  • $C_4 = \{abcd\}$
The Apriori Algorithm—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>

Sup$_{min}$ = 2

$C_1$

<table>
<thead>
<tr>
<th>Itemset</th>
<th>sup</th>
</tr>
</thead>
<tbody>
<tr>
<td>{A}</td>
<td>2</td>
</tr>
<tr>
<td>{B}</td>
<td>3</td>
</tr>
<tr>
<td>{C}</td>
<td>3</td>
</tr>
<tr>
<td>{D}</td>
<td>1</td>
</tr>
<tr>
<td>{E}</td>
<td>3</td>
</tr>
</tbody>
</table>

$L_1$

<table>
<thead>
<tr>
<th>Itemset</th>
<th>sup</th>
</tr>
</thead>
<tbody>
<tr>
<td>{A}</td>
<td>2</td>
</tr>
<tr>
<td>{B}</td>
<td>3</td>
</tr>
<tr>
<td>{C}</td>
<td>3</td>
</tr>
<tr>
<td>{E}</td>
<td>3</td>
</tr>
</tbody>
</table>

$C_2$

<table>
<thead>
<tr>
<th>Itemset</th>
<th>sup</th>
</tr>
</thead>
<tbody>
<tr>
<td>{A, B}</td>
<td>1</td>
</tr>
<tr>
<td>{A, C}</td>
<td>2</td>
</tr>
<tr>
<td>{A, E}</td>
<td>1</td>
</tr>
<tr>
<td>{B, C}</td>
<td>2</td>
</tr>
<tr>
<td>{B, E}</td>
<td>3</td>
</tr>
<tr>
<td>{C, E}</td>
<td>2</td>
</tr>
</tbody>
</table>

$L_2$

<table>
<thead>
<tr>
<th>Itemset</th>
<th>sup</th>
</tr>
</thead>
<tbody>
<tr>
<td>{A, C}</td>
<td>2</td>
</tr>
<tr>
<td>{B, C}</td>
<td>2</td>
</tr>
<tr>
<td>{B, E}</td>
<td>3</td>
</tr>
<tr>
<td>{C, E}</td>
<td>2</td>
</tr>
</tbody>
</table>

$C_3$

<table>
<thead>
<tr>
<th>Itemset</th>
<th>sup</th>
</tr>
</thead>
<tbody>
<tr>
<td>{B, C, E}</td>
<td>2</td>
</tr>
</tbody>
</table>

$L_3$

<table>
<thead>
<tr>
<th>Itemset</th>
<th>sup</th>
</tr>
</thead>
<tbody>
<tr>
<td>{B, C, E}</td>
<td>2</td>
</tr>
</tbody>
</table>
The Apriori Algorithm (Pseudo-Code)

\(C_k\): Candidate itemset of size k  
\(L_k\): frequent itemset of size k

\(L_1 = \{\text{frequent items}\};\)

\textbf{for} \ (k = 2; L_{k-1} \neq \emptyset; k++) \ \textbf{do begin}

\hspace{1em} \(C_k\) = candidates generated from \(L_{k-1}\);

\hspace{1em} \textbf{for each} transaction \(t\) in database do

\hspace{2em} \text{increment the count of all candidates in } C_{k+1} \text{ that are contained in } t

\hspace{1em} \(L_{k+1} = \text{candidates in } C_{k+1} \text{ with min\_support}\)

\textbf{end}

\textbf{return} \(\bigcup_k L_k\);
Questions

• How many scans on DB are needed for Apriori algorithm?
• When (k = ?) does Apriori algorithm generate the biggest number of candidate itemsets?
  • Is support counting for candidates expensive?
Further Improvement of the Apriori Method

• Major computational challenges
  • Multiple scans of transaction database
  • Huge number of candidates
  • Tedious workload of support counting for candidates

• Improving Apriori: general ideas
  • Reduce passes of transaction database scans
  • Shrink number of candidates
  • Facilitate support counting of candidates
*Partition: Scan Database Only Twice*

- Any itemset that is potentially frequent in DB must be frequent in at least one of the partitions of DB
  - Scan 1: partition database and find local frequent patterns
  - Scan 2: consolidate global frequent patterns

- A. Savasere, E. Omiecinski and S. Navathe,

\[ DB_1 + DB_2 + \cdots + DB_k = DB \]

\[ \text{sup}_1(i) < \sigma DB_1 \]
\[ \text{sup}_2(i) < \sigma DB_2 \]
\[ \cdots \]
\[ \text{sup}_k(i) < \sigma DB_k \]
\[ \text{sup}(i) < \sigma DB \]
**Hash-based Technique: Reduce the Number of Candidates**

- A \( k \)-itemset whose corresponding hashing bucket count is below the threshold cannot be frequent

  - Candidates: a, b, c, d, e
  - Hash entries
    - \{ab, ad, ae\}
    - \{bd, be, de\}
    - ...
  - Frequent 1-itemset: a, b, d, e

  - \( ab \) is not a candidate 2-itemset if the sum of count of \{ab, ad, ae\} is below support threshold

- J. Park, M. Chen, and P. Yu. *An effective hash-based algorithm for mining association rules*. *SIGMOD’95*
*Sampling for Frequent Patterns*

- Select a sample of original database, mine frequent patterns within sample using Apriori
- Scan database once to verify frequent itemsets found in sample, only borders of closure of frequent patterns are checked
  - Example: check $abcd$ instead of $ab$, $ac$, ..., etc.
- Scan database again to find missed frequent patterns
- H. Toivonen. *Sampling large databases for association rules.* In *VLDB’96*
Scalable Frequent Itemset Mining Methods

- Apriori: A Candidate Generation-and-Test Approach
  - Improving the Efficiency of Apriori
- FPGrowth: A Frequent Pattern-Growth Approach
- ECLAT: Frequent Pattern Mining with Vertical Data Format
- Generating Association Rules
Pattern-Growth Approach: Mining Frequent Patterns Without Candidate Generation

- Bottlenecks of the Apriori approach
  - Breadth-first (i.e., level-wise) search
    - Scan DB multiple times
  - Candidate generation and test
    - Often generates a huge number of candidates
- The FPGrowth Approach (J. Han, J. Pei, and Y. Yin, SIGMOD’ 00)
  - Depth-first search
  - Avoid explicit candidate generation
Major philosophy

- Grow long patterns from short ones using local frequent items only
  - “abc” is a frequent pattern
  - Get all transactions having “abc”, i.e., project DB on abc:
    \[ DB | abc \]
  - “d” is a local frequent item in \[ DB | abc \] \( \rightarrow \) abcd is a frequent pattern
FP-Growth Algorithm Sketch

• Construct FP-tree (frequent pattern-tree)
  • Compress the DB into a tree
• Recursively mine FP-tree by FP-Growth
  • Construct conditional pattern base from FP-tree
  • Construct conditional FP-tree from conditional pattern base
• Until the tree has a single path or empty
Construct FP-tree from a Transaction Database

1. **Scan** DB once, find frequent 1-itemset (single item pattern)

2. **Sort** frequent items in frequency descending order, **f-list**

3. **Scan** DB again, construct FP-tree

**F-list** = f-c-a-b-m-p

**Header Table**

<table>
<thead>
<tr>
<th>Item</th>
<th>Frequency</th>
<th>Head</th>
</tr>
</thead>
<tbody>
<tr>
<td>f</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>c</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>a</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>b</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>m</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>p</td>
<td>3</td>
<td></td>
</tr>
</tbody>
</table>

**min_support** = 3

```plaintext
TID | Items bought (ordered) frequent items
---|----------------------------------------
100 | {f, a, c, d, g, i, m, p}               | {f, c, a, m, p} |
200 | {a, b, c, f, l, m, o}                  | {f, c, a, b, m} |
300 | {b, f, h, j, o, w}                    | {f, b}          |
400 | {b, c, k, s, p}                       | {c, b, p}       |
500 | {a, f, c, e, l, p, m, n}               | {f, c, a, m, p} |
```
Frequent patterns can be partitioned into subsets according to f-list

- F-list = f-c-a-b-m-p
- Patterns containing p
- Patterns having m but no p
- ...
- Patterns having c but no a nor b, m, p
- Pattern f

Completeness and non-redundency
Find Patterns Having P From P-conditional Database

- Starting at the frequent item header table in the FP-tree
- Traverse the FP-tree by following the link of each frequent item \( p \)
- Accumulate all of transformed prefix paths of item \( p \) to form \( p \)'s conditional pattern base

### Header Table

<table>
<thead>
<tr>
<th>Item</th>
<th>Frequency</th>
<th>Head</th>
</tr>
</thead>
<tbody>
<tr>
<td>( f )</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>( c )</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>( a )</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>( b )</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>( m )</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>( p )</td>
<td>3</td>
<td></td>
</tr>
</tbody>
</table>

### Conditional pattern bases

<table>
<thead>
<tr>
<th>Item</th>
<th>Cond. Pattern Base</th>
</tr>
</thead>
<tbody>
<tr>
<td>( c )</td>
<td>( f:3 )</td>
</tr>
<tr>
<td>( a )</td>
<td>( fc:3 )</td>
</tr>
<tr>
<td>( b )</td>
<td>( fca:1, f:1, c:1 )</td>
</tr>
<tr>
<td>( m )</td>
<td>( fca:2, fcab:1 )</td>
</tr>
<tr>
<td>( p )</td>
<td>( fcam:2, cb:1 )</td>
</tr>
</tbody>
</table>
From Conditional Pattern-bases to Conditional FP-trees

- For each pattern-base
  - Accumulate the count for each item in the base
  - Construct the FP-tree for the frequent items of the pattern base

**Header Table**

<table>
<thead>
<tr>
<th>Item</th>
<th>frequency</th>
<th>head</th>
</tr>
</thead>
<tbody>
<tr>
<td>f</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>c</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>a</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>b</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>m</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>p</td>
<td>3</td>
<td></td>
</tr>
</tbody>
</table>

**m-conditional pattern base:**

- \{fca:2, fcab:1\}

**All frequent patterns relate to m**

- \{\}  
  - \{\}  
    - \{\}  
      - \{\}  
        - \{\}  
          - \{\}  

**m-conditional FP-tree**

Don’t forget to add back \( m \)!
Recursion: Mining Each Conditional FP-tree

Cond. pattern base of “am”: (fc:3)

m-conditional FP-tree

Cond. pattern base of “cm”: (f:3)

am-conditional FP-tree

Cond. pattern base of “cam”: (f:3)

cam-conditional FP-tree
Another Example: FP-Tree Construction

<table>
<thead>
<tr>
<th>TID</th>
<th>Items</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>{a,b}</td>
</tr>
<tr>
<td>2</td>
<td>{b,c,d}</td>
</tr>
<tr>
<td>3</td>
<td>{a,c,d,e}</td>
</tr>
<tr>
<td>4</td>
<td>{a,d,e}</td>
</tr>
<tr>
<td>5</td>
<td>{a,b,c}</td>
</tr>
<tr>
<td>6</td>
<td>{a,b,c,d}</td>
</tr>
<tr>
<td>7</td>
<td>{a}</td>
</tr>
<tr>
<td>8</td>
<td>{a,b,c}</td>
</tr>
<tr>
<td>9</td>
<td>{a,b,d}</td>
</tr>
<tr>
<td>10</td>
<td>{b,c,e}</td>
</tr>
</tbody>
</table>

Transaction Database

Header table

<table>
<thead>
<tr>
<th>Item</th>
<th>Pointer</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td></td>
</tr>
<tr>
<td>b</td>
<td></td>
</tr>
<tr>
<td>c</td>
<td></td>
</tr>
<tr>
<td>d</td>
<td></td>
</tr>
<tr>
<td>e</td>
<td></td>
</tr>
</tbody>
</table>

Pointers are used to assist frequent itemset generation
Mining Sub-tree Ending with e

• Conditional pattern base for e: \{acd:1; ad:1; bc:1\}
• Conditional FP-tree for e:

  ![Conditional FP-tree for e](image)

  • Conditional pattern base for de: \{ac:1; a:1\}
  • Conditional FP-tree for de:
  • Frequent patterns for de: \{ade:2, de:2\}
  • Conditional pattern base for ce: \{a:1\}
  • Conditional FP-tree for ce: empty
  • Frequent patterns for ce: \{ce:2\}
  • Conditional pattern base for ae: \{\emptyset\}
  • Conditional FP-tree for ae: empty
  • Frequent patterns for ae: \{ae:2\}

• Therefore, all frequent patterns with e are: \{ade:2, de:2, ce:2, ae:2, e:3\}
A Special Case: Single Prefix Path in FP-tree

• Suppose a (conditional) FP-tree \( T \) has a shared single prefix-path \( P \)

• Mining can be decomposed into two parts

\[
\begin{align*}
\{\} \\
\text{a}_1:n_1 \\
\text{a}_2:n_2 \\
\text{a}_3:n_3
\end{align*}
\]

• Reduction of the single prefix path into one node

• Concatenation of the mining results of the two parts

\[
\begin{align*}
\{\} \\
r_1 = \\
\text{a}_1:n_1 \\
\text{a}_2:n_2 \\
\text{a}_3:n_3 \\
\text{b}_1:m_1 \\
\text{c}_1:k_1 \\
\text{c}_2:k_2 \\
\text{c}_3:k_3
\end{align*}
\]

\[
\begin{align*}
\text{r}_1 = \\
\text{b}_1:m_1 \\
\text{c}_1:k_1 \\
\text{c}_2:k_2 \\
\text{c}_3:k_3
\end{align*}
\]
Benefits of the FP-tree Structure

• Completeness
  • Preserve complete information for frequent pattern mining
  • Never break a long pattern of any transaction

• Compactness
  • Reduce irrelevant info—infrequent items are gone
  • Items in frequency descending order: the more frequently occurring, the more likely to be shared
  • Never be larger than the original database (not count node-links and the count field)
Scaling FP-growth by Database Projection

• What about if FP-tree cannot fit in memory?
  • DB projection

• First partition a database into a set of projected DBs
• Then construct and mine FP-tree for each projected DB
• Parallel projection vs. partition projection techniques
  • Parallel projection
    • Project the DB in parallel for each frequent item
    • Parallel projection is space costly
    • All the partitions can be processed in parallel
  • Partition projection
    • Partition the DB based on the ordered frequent items
    • Passing the unprocessed parts to the subsequent partitions
FP-Growth vs. Apriori: Scalability With the Support Threshold

Data set T25I20D10K

Run time (sec.)

Support threshold (%)

---

D1 FP-growth runtime
D1 Apriori runtime
Advantages of the Pattern Growth Approach

- Divide-and-conquer:
  - Decompose both the mining task and DB according to the frequent patterns obtained so far
  - Lead to focused search of smaller databases

- Other factors
  - No candidate generation, no candidate test
  - Compressed database: FP-tree structure
  - No repeated scan of entire database
  - Basic ops: counting local freq items and building sub FP-tree, no pattern search and matching
*Further Improvements of Mining Methods*

- AFOPT (Liu, et al. @ KDD’03)
  - A “push-right” method for mining condensed frequent pattern (CFP) tree
- Carpenter (Pan, et al. @ KDD’03)
  - Mine data sets with small rows but numerous columns
  - Construct a row-enumeration tree for efficient mining
- FPgrowth+ (Grahne and Zhu, FIMI’03)
  - Efficiently Using Prefix-Trees in Mining Frequent Itemsets, Proc. ICDM'03 Int. Workshop on Frequent Itemset Mining Implementations (FIMI'03), Melbourne, FL, Nov. 2003
- TD-Close (Liu, et al, SDM’06)
Extension of Pattern Growth Mining Methodology

- Mining closed frequent itemsets and max-patterns
  - CLOSET (DMKD’00), FPclose, and FPMax (Grahne & Zhu, Fimi’03)
- Mining sequential patterns
  - PrefixSpan (ICDE’01), CloSpan (SDM’03), BIDE (ICDE’04)
- Mining graph patterns
  - gSpan (ICDM’02), CloseGraph (KDD’03)
- Constraint-based mining of frequent patterns
  -Convertible constraints (ICDE’01), gPrune (PAKDD’03)
- Computing iceberg data cubes with complex measures
  - H-tree, H-cubing, and Star-cubing (SIGMOD’01, VLDB’03)
- Pattern-growth-based Clustering
  - MaPle (Pei, et al., ICDM’03)
- Pattern-Growth-Based Classification
  - Mining frequent and discriminative patterns (Cheng, et al, ICDE’07)
Scalable Frequent Itemset Mining Methods

- Apriori: A Candidate Generation-and-Test Approach
  - Improving the Efficiency of Apriori
- FPGrowth: A Frequent Pattern-Growth Approach
- ECLAT: Frequent Pattern Mining with Vertical Data Format
- Generating Association Rules
ECLAT: Mining by Exploring Vertical Data Format

- Vertical format: $t(AB) = \{T_{11}, T_{25}, \ldots\}$
  - tid-list: list of trans.-ids containing an itemset
- Deriving frequent patterns based on vertical intersections
  - $t(X) = t(Y)$: $X$ and $Y$ always happen together
  - $t(X) \subset t(Y)$: transaction having $X$ always has $Y$
- Using diffset to accelerate mining
  - Only keep track of differences of tids
  - $t(X) = \{T_1, T_2, T_3\}$, $t(XY) = \{T_1, T_3\}$
  - Diffset $(XY, X) = \{T_2\}$
- Eclat (Zaki et al. @KDD’97)

Similar idea for inverted index in storing text
Scalable Frequent Itemset Mining Methods

• Apriori: A Candidate Generation-and-Test Approach
  • Improving the Efficiency of Apriori
• FPGrowth: A Frequent Pattern-Growth Approach
• ECLAT: Frequent Pattern Mining with Vertical Data Format
• Generating Association Rules
Generating Association Rules

• Strong association rules
  • Satisfying minimum support and minimum confidence
  • Recall: \( \text{Confidence}(A \Rightarrow B) = P(B|A) = \frac{\text{support}(A \cup B)}{\text{support}(A)} \)

• Steps of generating association rules from frequent pattern \( l \):
  • Step 1: generate all nonempty subsets of \( l \)
  • Step 2: for every nonempty subset \( s \), calculate the confidence for rule \( s \Rightarrow (l - s) \)
Example

• $X = \{I_1, I_2, I_5\}:2$
  • Nonempty subsets of $X$ are:
    $\{I_1, I_2\}: 4$, $\{I_1, I_5\}: 2$, $\{I_2, I_5\}: 2$, $\{I_1\}: 6$, $\{I_2\}: 7$, and $\{I_5\}: 2$
  • Association rules are:

\[
\begin{align*}
\{I_1, I_2\} & \Rightarrow I_5, & \text{confidence} = \frac{2}{4} = 50\% \\
\{I_1, I_5\} & \Rightarrow I_2, & \text{confidence} = \frac{2}{2} = 100\% \\
\{I_2, I_5\} & \Rightarrow I_1, & \text{confidence} = \frac{2}{2} = 100\% \\
I_1 & \Rightarrow \{I_2, I_5\}, & \text{confidence} = \frac{2}{6} = 33\% \\
I_2 & \Rightarrow \{I_1, I_5\}, & \text{confidence} = \frac{2}{7} = 29\% \\
I_5 & \Rightarrow \{I_1, I_2\}, & \text{confidence} = \frac{2}{2} = 100\% 
\end{align*}
\]
Chapter 6: Mining Frequent Patterns, Association and Correlations

• Basic Concepts

• Frequent Itemset Mining Methods

• Pattern Evaluation Methods

• Summary
Misleading Strong Association Rules

- Not all strong association rules are interesting

<table>
<thead>
<tr>
<th></th>
<th>Basketball</th>
<th>Not basketball</th>
<th>Sum (row)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cereal</td>
<td>2000</td>
<td>1750</td>
<td>3750</td>
</tr>
<tr>
<td>Not cereal</td>
<td>1000</td>
<td>250</td>
<td>1250</td>
</tr>
<tr>
<td>Sum(col.)</td>
<td>3000</td>
<td>2000</td>
<td>5000</td>
</tr>
</tbody>
</table>

- Shall we target people who play basketball for cereal ads? *play basketball ⇒ eat cereal* [40%, 66.7%]
- Hint: What is the overall probability of people who eat cereal?
  - 3750/5000 = 75% > 66.7%!
- Confidence measure of a rule could be misleading
Other Measures

- From association to correlation
  - Lift
  - $\chi^2$
  - All_confidence
  - Max_confidence
  - Kulczynski
  - Cosine
Interestingness Measure: Correlations
(Lift)

• play basketball ⇒ eat cereal [40%, 66.7%] is misleading
  • The overall % of students eating cereal is 75% > 66.7%.

• play basketball ⇒ not eat cereal [20%, 33.3%] is more accurate, although with lower support and confidence

• Measure of dependent/correlated events: lift

\[
lift = \frac{P(A \cup B)}{P(A)P(B)}
\]

\[
lift(B,C) = \frac{2000/5000}{3000/5000 \times 3750/5000} = 0.89
\]

\[
lift(B, \neg C) = \frac{1000/5000}{3000/5000 \times 1250/5000} = 1.33
\]

1: independent
>1: positively correlated
<1: negatively correlated
Correlation Analysis (Nominal Data)

• $\chi^2$ (chi-square) test

$$\chi^2 = \sum \frac{(Observed - Expected)^2}{Expected}$$

• Independency test between two attributes
  • The larger the $\chi^2$ value, the more likely the variables are related

• The cells that contribute the most to the $\chi^2$ value are those whose actual count is very different from the expected count under independence assumption

• Correlation does not imply causality
  • # of hospitals and # of car-theft in a city are correlated
  • Both are causally linked to the third variable: population
When Do We Need Chi-Square Test?

• Considering two attributes A and B
  • A: a nominal attribute with c distinct values, $a_1, \ldots, a_c$
    • E.g., Grades of Math
  • B: a nominal attribute with r distinct values, $b_1, \ldots, b_r$
    • E.g., Grades of Science
• Question: Are A and B related?
How Can We Run Chi-Square Test?

• Constructing contingency table
  • Observed frequency $o_{ij}$: number of data objects taking value $b_i$ for attribute B and taking value $a_j$ for attribute A

<table>
<thead>
<tr>
<th></th>
<th>$a_1$</th>
<th>$a_2$</th>
<th>...</th>
<th>$a_c$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$b_1$</td>
<td>$o_{11}$</td>
<td>$o_{12}$</td>
<td>...</td>
<td>$o_{1c}$</td>
</tr>
<tr>
<td>$b_2$</td>
<td>$o_{21}$</td>
<td>$o_{22}$</td>
<td>...</td>
<td>$o_{2c}$</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>$b_r$</td>
<td>$o_{r1}$</td>
<td>$o_{r2}$</td>
<td>...</td>
<td>$o_{rc}$</td>
</tr>
</tbody>
</table>

• Calculate expected frequency $e_{ij} = \frac{\text{count}(B=b_i) \times \text{count}(A=a_j)}{n}$
  • Null hypothesis: A and B are independent
The Pearson $\chi^2$ statistic is computed as:

$$X^2 = \sum_{i=1}^{r} \sum_{j=1}^{c} \frac{(o_{ij} - e_{ij})^2}{e_{ij}}$$

It follows a Chi-squared distribution with degrees of freedom $(r - 1) \times (c - 1)$. 

![Chi-squared distribution graphs](image)
Chi-Square Calculation: An Example

<table>
<thead>
<tr>
<th></th>
<th>Play chess</th>
<th>Not play chess</th>
<th>Sum (row)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Like science fiction</td>
<td>250(90)</td>
<td>200(360)</td>
<td>450</td>
</tr>
<tr>
<td>Not like science fiction</td>
<td>50(210)</td>
<td>1000(840)</td>
<td>1050</td>
</tr>
<tr>
<td>Sum(col.)</td>
<td>300</td>
<td>1200</td>
<td>1500</td>
</tr>
</tbody>
</table>

• $\chi^2$ (chi-square) calculation (numbers in parenthesis are expected counts calculated based on the data distribution in the two categories)

$$\chi^2 = \frac{(250-90)^2}{90} + \frac{(50-210)^2}{210} + \frac{(200-360)^2}{360} + \frac{(1000-840)^2}{840} = 507.93$$

• It shows that like_science_fiction and play_chess are correlated in the group
  • Degree of freedom = $(2-1)(2-1) = 1$
  • P-value = $P(\chi^2 > 507.93) = 0.0$
    • Reject the null hypothesis => A and B are dependent
Are \textit{lift} and $\chi^2$ Good Measures of Correlation?

- Lift and $\chi^2$ are affected by null-transaction
  - E.g., number of transactions that do not contain milk nor coffee
- All\_confidence
  - $\text{all}\_\text{conf}(A,B) = \min\{P(A|B), P(B|A)\}$
- Max\_confidence
  - $\text{max}\_\text{conf}(A, B) = \max\{P(A|B), P(B|A)\}$
- Kulczynski
  - $Kulc(A, B) = \frac{1}{2} (P(A|B) + P(B|A))$
- Cosine
  - $\text{cosine}(A, B) = \sqrt{P(A|B) \times P(B|A)}$
Comparison of Interestingness Measures

- Null-(transaction) invariance is crucial for correlation analysis
- Lift and $\chi^2$ are not null-invariant
- 5 null-invariant measures

<table>
<thead>
<tr>
<th>Measure</th>
<th>Definition</th>
<th>Range</th>
<th>Null-Invariant</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\chi^2(a, b)$</td>
<td>$\sum_{i,j=0,1} \frac{(e(a_i, b_j) - o(a_i, b_j))^2}{e(a_i, b_j)}$</td>
<td>$[0, \infty]$</td>
<td>No</td>
</tr>
<tr>
<td>Lift($a, b$)</td>
<td>$\frac{P(ab)}{P(a)P(b)}$</td>
<td>$[0, \infty]$</td>
<td>No</td>
</tr>
<tr>
<td>AllConf($a, b$)</td>
<td>$\frac{\sup(ab)}{\max{{\sup(a),\sup(b)}}}$</td>
<td>$[0, 1]$</td>
<td>Yes</td>
</tr>
<tr>
<td>Coherence($a, b$)</td>
<td>$\frac{\sup(ab)}{\sup(a)+\sup(b)-\sup(ab)}$</td>
<td>$[0, 1]$</td>
<td>Yes</td>
</tr>
<tr>
<td>Cosine($a, b$)</td>
<td>$\frac{\sup(ab)}{\sqrt{\sup(a)\sup(b)}}$</td>
<td>$[0, 1]$</td>
<td>Yes</td>
</tr>
<tr>
<td>Kulc($a, b$)</td>
<td>$\frac{\sup(ab)}{2} \left( \frac{1}{\sup(a)} + \frac{1}{\sup(b)} \right)$</td>
<td>$[0, 1]$</td>
<td>Yes</td>
</tr>
<tr>
<td>MaxConf($a, b$)</td>
<td>$\max{\frac{\sup(ab)}{\sup(a)}, \frac{\sup(ab)}{\sup(b)}}$</td>
<td>$[0, 1]$</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Null-transactions w.r.t. m and c

Kulczynski measure (1927)

Null-invariant

Table 2. Example data sets.

Table 3. Interestingness measure definitions.
Recent DB conferences, removing balanced associations, low sup, etc.

<table>
<thead>
<tr>
<th>ID</th>
<th>Author a</th>
<th>Author b</th>
<th>sup(ab)</th>
<th>sup(a)</th>
<th>sup(b)</th>
<th>Coherence</th>
<th>Cosine</th>
<th>Kulc</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Hans-Peter Kriegel</td>
<td>Martin Ester</td>
<td>28</td>
<td>146</td>
<td>54</td>
<td>0.163 (2)</td>
<td>0.315 (7)</td>
<td>0.355 (9)</td>
</tr>
<tr>
<td>2</td>
<td>Michael Carey</td>
<td>Miron Livny</td>
<td>26</td>
<td>104</td>
<td>58</td>
<td>0.191 (1)</td>
<td>0.335 (4)</td>
<td>0.349 (10)</td>
</tr>
<tr>
<td>3</td>
<td>Hans-Peter Kriegel</td>
<td>Joerg Sander</td>
<td>24</td>
<td>146</td>
<td>36</td>
<td>0.152 (3)</td>
<td>0.331 (5)</td>
<td>0.416 (8)</td>
</tr>
<tr>
<td>4</td>
<td>Christos Faloutsos</td>
<td>Spiros Papadimitriou</td>
<td>20</td>
<td>162</td>
<td>26</td>
<td>0.119 (7)</td>
<td>0.308 (10)</td>
<td>0.446 (7)</td>
</tr>
<tr>
<td>5</td>
<td>Hans-Peter Kriegel</td>
<td>Martin Pfeifle</td>
<td>18</td>
<td>146</td>
<td>18</td>
<td>0.123 (6)</td>
<td>0.351 (2)</td>
<td>0.562 (2)</td>
</tr>
<tr>
<td>6</td>
<td>Hector Garcia-Molina</td>
<td>Wilbur Labio</td>
<td>16</td>
<td>144</td>
<td>18</td>
<td>0.110 (9)</td>
<td>0.314 (8)</td>
<td>0.500 (4)</td>
</tr>
<tr>
<td>7</td>
<td>Divyakant Agrawal</td>
<td>Wang Hsiung</td>
<td>16</td>
<td>120</td>
<td>16</td>
<td>0.133 (5)</td>
<td>0.365 (1)</td>
<td>0.567 (1)</td>
</tr>
<tr>
<td>8</td>
<td>Elke Rundensteiner</td>
<td>Murali Mani</td>
<td>16</td>
<td>104</td>
<td>20</td>
<td>0.148 (4)</td>
<td>0.351 (3)</td>
<td>0.477 (6)</td>
</tr>
<tr>
<td>9</td>
<td>Divyakant Agrawal</td>
<td>Oliver Po</td>
<td>12</td>
<td>120</td>
<td>12</td>
<td>0.100 (10)</td>
<td>0.316 (6)</td>
<td>0.550 (3)</td>
</tr>
<tr>
<td>10</td>
<td>Gerhard Weikum</td>
<td>Martin Theobald</td>
<td>12</td>
<td>106</td>
<td>14</td>
<td>0.111 (8)</td>
<td>0.312 (9)</td>
<td>0.485 (5)</td>
</tr>
</tbody>
</table>

Table 5. Experiment on DBLP data set.

- Advisor-advisee relation: Kulc: high, coherence: low, cosine: middle

- Tianyi Wu, Yuguo Chen and Jiawei Han, “Association Mining in Large Databases: A Re-Examination of Its Measures”, Proc. 2007 Int. Conf. Principles and Practice of Knowledge Discovery in Databases (PKDD'07), Sept. 2007
**Which Null-Invariant Measure Is Better?**

- IR (Imbalance Ratio): measure the imbalance of two itemsets A and B in rule implications

\[
IR(A, B) = \frac{|sup(A) - sup(B)|}{sup(A) + sup(B) - sup(A \cup B)}
\]

- Kulczynski and Imbalance Ratio (IR) together present a clear picture for all the three datasets D\(_4\) through D\(_6\)
  - D\(_4\) is balanced & neutral
  - D\(_5\) is imbalanced & neutral
  - D\(_6\) is very imbalanced & neutral

<table>
<thead>
<tr>
<th>Data</th>
<th>mc</th>
<th>(\bar{mc})</th>
<th>(mc)</th>
<th>(\bar{mc})</th>
<th>all_conf.</th>
<th>max_conf.</th>
<th>Kulc.</th>
<th>cosine</th>
<th>IR</th>
</tr>
</thead>
<tbody>
<tr>
<td>D(_1)</td>
<td>10,000</td>
<td>1,000</td>
<td>1,000</td>
<td>100,000</td>
<td>0.91</td>
<td>0.91</td>
<td>0.91</td>
<td>0.91</td>
<td>0.0</td>
</tr>
<tr>
<td>D(_2)</td>
<td>10,000</td>
<td>1,000</td>
<td>1,000</td>
<td>100</td>
<td>0.91</td>
<td>0.91</td>
<td>0.91</td>
<td>0.91</td>
<td>0.0</td>
</tr>
<tr>
<td>D(_3)</td>
<td>100</td>
<td>1,000</td>
<td>1,000</td>
<td>100,000</td>
<td>0.09</td>
<td>0.09</td>
<td>0.09</td>
<td>0.09</td>
<td>0.0</td>
</tr>
<tr>
<td>D(_4)</td>
<td>1,000</td>
<td>1,000</td>
<td>1,000</td>
<td>100,000</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
<td>0.0</td>
</tr>
<tr>
<td>D(_5)</td>
<td>1,000</td>
<td>100</td>
<td>10,000</td>
<td>100,000</td>
<td>0.09</td>
<td>0.91</td>
<td>0.5</td>
<td>0.29</td>
<td>0.89</td>
</tr>
<tr>
<td>D(_6)</td>
<td>1,000</td>
<td>10</td>
<td>100,000</td>
<td>100,000</td>
<td>0.01</td>
<td>0.99</td>
<td>0.5</td>
<td>0.10</td>
<td>0.99</td>
</tr>
</tbody>
</table>
Chapter 6: Mining Frequent Patterns, Association and Correlations

• Basic Concepts

• Frequent Itemset Mining Methods

• Pattern Evaluation Methods

• Summary
Summary

- Basic concepts
  - Frequent pattern, association rules, support-confident framework, closed and max-patterns
- Scalable frequent pattern mining methods
  - Apriori
  - FPgrowth
  - Vertical format approach (ECLAT)
- Which patterns are interesting?
  - Pattern evaluation methods
Ref: Basic Concepts of Frequent Pattern Mining

- **Association Rules** R. Agrawal, T. Imielinski, and A. Swami. Mining association rules between sets of items in large databases. SIGMOD'93.

- **Max-pattern** R. J. Bayardo. Efficiently mining long patterns from databases. SIGMOD'98.


- **Sequential pattern** R. Agrawal and R. Srikant. Mining sequential patterns. ICDE'95
Ref: Apriori and Its Improvements

Ref: Depth-First, Projection-Based FP Mining

- J. Han, J. Pei, and Y. Yin. Mining frequent patterns without candidate generation. SIGMOD’ 00.
- J. Liu, Y. Pan, K. Wang, and J. Han. Mining Frequent Item Sets by Opportunistic Projection. KDD'02.
- J. Han, J. Wang, Y. Lu, and P. Tzvetkov. Mining Top-K Frequent Closed Patterns without Minimum Support. ICDM'02.
- J. Wang, J. Han, and J. Pei. CLOSET+: Searching for the Best Strategies for Mining Frequent Closed Itemsets. KDD'03.
- G. Grahne and J. Zhu, Efficiently Using Prefix-Trees in Mining Frequent Itemsets, Proc. ICDM'03 Int. Workshop on Frequent Itemset Mining Implementations (FIMI'03), Melbourne, FL, Nov. 2003
Ref: Mining Correlations and Interesting Rules

- P.-N. Tan, V. Kumar, and J. Srivastava. Selecting the Right Interestingness Measure for Association Patterns. KDD'02.
- E. Omiecinski. Alternative Interest Measures for Mining Associations. TKDE’03.
- T. Wu, Y. Chen and J. Han, “Association Mining in Large Databases: A Re-Examination of Its Measures”, PKDD'07
Ref: Freq. Pattern Mining Applications


• T. Dasu, T. Johnson, S. Muthukrishnan, and V. Shkapenyuk. Mining Database Structure; or How to Build a Data Quality Browser. SIGMOD'02.

• K. Wang, S. Zhou, J. Han. Profit Mining: From Patterns to Actions. EDBT’02.