Fundamentals of Data Mining Algorithms
Graph Pattern Mining (Chapter 13)

Loïc Cerf
September, 21st 2011
UFMG – ICEx – DCC
Outline

1. Association Rule Mining (in Chapter 10)
2. Frequent Subgraph
3. Frequent Subgraph Mining
4. gSpan’s Enumeration
5. gSpan’s Graph Isomorphism Test
6. Conclusion
Outline

1. Association Rule Mining (in Chapter 10)

2. Frequent Subgraph

3. Frequent Subgraph Mining

4. gSpan’s Enumeration

5. gSpan’s Graph Isomorphism Test

6. Conclusion
Association rule: definition

**Definition**

Given a set of Boolean attributes $\mathcal{A}$, an *association rule* is a couple of itemsets $(X, Y) \in 2^\mathcal{A} \times 2^\mathcal{A}$.
Association rule: definition

Definition

Given a set of Boolean attributes $\mathcal{A}$, an association rule is a couple of itemsets $(X, Y) \in 2^\mathcal{A} \times 2^\mathcal{A}$.

Frequency

Given a set of objects $\mathcal{O}$ and a set of Boolean attributes $\mathcal{A}$, the frequency of an association rule $(X, Y) \in 2^\mathcal{A} \times 2^\mathcal{A}$ in a dataset $D \in \mathcal{O} \times \mathcal{A}$ is the frequency of the itemset $X \cup Y$ in $D$. 

Confidence

The confidence is a conditional probability.
Association rule: definition

**Definition**

Given a set of Boolean attributes $\mathcal{A}$, an association rule is a couple of itemsets $(X, Y) \in 2^\mathcal{A} \times 2^\mathcal{A}$.

**Frequency**

Given a set of objects $\mathcal{O}$ and a set of Boolean attributes $\mathcal{A}$, the frequency of an association rule $(X, Y) \in 2^\mathcal{A} \times 2^\mathcal{A}$ in a dataset $\mathcal{D} \in \mathcal{O} \times \mathcal{A}$ is the $f(X \cup Y, \mathcal{D})$. 
Association rule: definition

**Definition**

Given a set of Boolean attributes $\mathcal{A}$, an *association rule* is a couple of itemsets $(X, Y) \in 2^\mathcal{A} \times 2^\mathcal{A}$.

**Frequency**

Given a set of objects $\mathcal{O}$ and a set of Boolean attributes $\mathcal{A}$, the *frequency* of an association rule $(X, Y) \in 2^\mathcal{A} \times 2^\mathcal{A}$ in a dataset $\mathcal{D} \in \mathcal{O} \times \mathcal{A}$ is the $f(X \cup Y, \mathcal{D})$.

**Confidence**

Given a set of objects $\mathcal{O}$ and a set of Boolean attributes $\mathcal{A}$, the *confidence* of an association rule $(X, Y) \in 2^\mathcal{A} \times 2^\mathcal{A}$ in a dataset $\mathcal{D} \in \mathcal{O} \times \mathcal{A}$ is $\frac{f(X \cup Y, \mathcal{D})}{f(X, \mathcal{D})}$. The confidence is a conditional probability.
Association rule: definition

**Definition**

Given a set of Boolean attributes $\mathcal{A}$, an *association rule* is a couple of itemsets $(X, Y) \in 2^\mathcal{A} \times 2^\mathcal{A}$.

**Frequency**

Given a set of objects $\mathcal{O}$ and a set of Boolean attributes $\mathcal{A}$, the *frequency* of an association rule $(X, Y) \in 2^\mathcal{A} \times 2^\mathcal{A}$ in a dataset $\mathcal{D} \in \mathcal{O} \times \mathcal{A}$ is the $f(X \cup Y, \mathcal{D})$.

**Confidence**

Given a set of objects $\mathcal{O}$ and a set of Boolean attributes $\mathcal{A}$, the *confidence* of an association rule $(X, Y) \in 2^\mathcal{A} \times 2^\mathcal{A}$ in a dataset $\mathcal{D} \in \mathcal{O} \times \mathcal{A}$ is $\frac{f(X \cup Y, \mathcal{D})}{f(X, \mathcal{D})}$.

The confidence is a conditional probability.
Association rule: definition

Definition

Given a set of Boolean attributes $\mathcal{A}$, an association rule is a couple of itemsets $(X, Y) \in 2^\mathcal{A} \times 2^\mathcal{A}$ such that $X \cap Y = \emptyset$.

Frequency

Given a set of objects $\mathcal{O}$ and a set of Boolean attributes $\mathcal{A}$, the frequency of an association rule $(X, Y) \in 2^\mathcal{A} \times 2^\mathcal{A}$ in a dataset $\mathcal{D} \in \mathcal{O} \times \mathcal{A}$ is the $f(X \cup Y, \mathcal{D})$.

Confidence

Given a set of objects $\mathcal{O}$ and a set of Boolean attributes $\mathcal{A}$, the confidence of an association rule $(X, Y) \in 2^\mathcal{A} \times 2^\mathcal{A}$ in a dataset $\mathcal{D} \in \mathcal{O} \times \mathcal{A}$ is $\frac{f(X \cup Y, \mathcal{D})}{f(X, \mathcal{D})}$.

The confidence is a conditional probability.
Definition

Given the objects in $\mathcal{O}$ described with the Boolean attributes in $\mathcal{A}$, listing every association rule having a frequency above a given threshold $\mu \in \mathbb{N}$ and a confidence above a given threshold $\nu \in [0, 1]$.

Input:

<table>
<thead>
<tr>
<th></th>
<th>$a_1$</th>
<th>$a_2$</th>
<th>\ldots</th>
<th>$a_n$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$o_1$</td>
<td>$d_{1,1}$</td>
<td>$d_{1,2}$</td>
<td>\ldots</td>
<td>$d_{1,n}$</td>
</tr>
<tr>
<td>$o_2$</td>
<td>$d_{2,1}$</td>
<td>$d_{2,2}$</td>
<td>\ldots</td>
<td>$d_{2,n}$</td>
</tr>
<tr>
<td>\vdots</td>
<td>\vdots</td>
<td>\vdots</td>
<td>\ddots</td>
<td>\vdots</td>
</tr>
<tr>
<td>$o_m$</td>
<td>$d_{m,1}$</td>
<td>$d_{m,2}$</td>
<td>\ldots</td>
<td>$d_{m,n}$</td>
</tr>
</tbody>
</table>

where $d_{i,j} \in \{\text{true}, \text{false}\}$

and a minimal frequency $\mu \in \mathbb{N}$

and a minimal confidence $\nu \in [0, 1]$. 
Association rule mining

**Definition**

Given the objects in \( \mathcal{O} \) described with the Boolean attributes in \( \mathcal{A} \), listing every association rule having a frequency above a given threshold \( \mu \in \mathbb{N} \) and a confidence above a given threshold \( \nu \in [0, 1] \).

Output: *every* \( (X, Y) \in 2^\mathcal{A} \times 2^\mathcal{A} \) such that:

- \( X \cap Y = \emptyset \);
- there are at least \( \mu \) objects having all attributes in \( X \cup Y \)...
- ... and at most \( \frac{1}{\nu} \) times more objects having all attributes in \( X \).
Specifying a minimal absolute frequency $\mu = 2$ objects (or, equivalently, a minimal relative frequency of 50%) and a minimal confidence of $\nu = \frac{2}{3}$.

The association rules are:

<table>
<thead>
<tr>
<th></th>
<th>$a_1$</th>
<th>$a_2$</th>
<th>$a_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$o_1$</td>
<td>×</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>$o_2$</td>
<td>×</td>
<td>×</td>
<td></td>
</tr>
<tr>
<td>$o_3$</td>
<td></td>
<td>×</td>
<td></td>
</tr>
<tr>
<td>$o_4$</td>
<td></td>
<td></td>
<td>×</td>
</tr>
</tbody>
</table>
Association rule mining: illustration

Specifying a minimal absolute frequency $\mu = 2$ objects (or, equivalently, a minimal relative frequency of 50%) and a minimal confidence of $\nu = \frac{2}{3}$.

The association rules are:

- the frequent itemsets implying the empty set: $\emptyset \rightarrow \emptyset$ (4, 1), $\{a_1\} \rightarrow \emptyset$ (2, 1), $\{a_2\} \rightarrow \emptyset$ (3, 1), $\{a_3\} \rightarrow \emptyset$ (2, 1) and $\{a_1, a_2\} \rightarrow \emptyset$ (2, 1);
Specifying a minimal absolute frequency $\mu = 2$ objects (or, equivalently, a minimal relative frequency of 50%) and a minimal confidence of $\nu = \frac{2}{3}$.

The association rules are:

- the frequent itemsets implying the empty set: $\emptyset \rightarrow \emptyset$ (4, 1), $\{a_1\} \rightarrow \emptyset$ (2, 1), $\{a_2\} \rightarrow \emptyset$ (3, 1), $\{a_3\} \rightarrow \emptyset$ (2, 1) and $\{a_1, a_2\} \rightarrow \emptyset$ (2, 1);
- but also $\emptyset \rightarrow \{a_2\}$ (3, 3/4), $\{a_1\} \rightarrow \{a_2\}$ (2, 1) and $\{a_2\} \rightarrow \{a_1\}$ (2, 2/3).
Post-processing the frequent itemsets

The frequent and confident association rules can be listed by post-processing the frequent itemsets (same minimal frequency) associated with the objects supporting them.
The frequent and confident association rules can be listed by post-processing the frequent itemsets (same minimal frequency) associated with the objects supporting them.

Notice that, given a set of objects \( O \), a set of Boolean attributes \( A \), a dataset \( D \subseteq O \times A \) and two association rules \( X \rightarrow Y \) and \( X' \rightarrow Y' \), the following implication holds and can be used to prune the enumeration of the association rules:

\[
\begin{align*}
X \cup Y &= X' \cup Y' \\
X &\subseteq X' \\
\Rightarrow \quad & \frac{f(X \cup Y, D)}{f(X, D)} \geq \frac{f(X' \cup Y', D)}{f(X', D)}.
\end{align*}
\]
Association Rule Mining (in Chapter 10)

Lift

**Definition**

Given a set of objects $\mathcal{O}$ and a set of Boolean attributes $\mathcal{A}$, the *lift* of an association rule $X \rightarrow Y$ in a dataset $\mathcal{D} \subseteq \mathcal{O} \times \mathcal{A}$ is

$$
\frac{\frac{f(X \cup Y, \mathcal{D})}{|\mathcal{O}|}}{\frac{f(X, \mathcal{D})}{|\mathcal{O}|} \cdot \frac{f(Y, \mathcal{D})}{|\mathcal{O}|}} = \frac{|\mathcal{O}| f(X \cup Y, \mathcal{D})}{f(X, \mathcal{D}) f(Y, \mathcal{D})}.
$$

The lift of:

<table>
<thead>
<tr>
<th></th>
<th>$a_1$</th>
<th>$a_2$</th>
<th>$a_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$o_1$</td>
<td>×</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>$o_2$</td>
<td>×</td>
<td>×</td>
<td></td>
</tr>
<tr>
<td>$o_3$</td>
<td></td>
<td>×</td>
<td></td>
</tr>
<tr>
<td>$o_4$</td>
<td></td>
<td></td>
<td>×</td>
</tr>
</tbody>
</table>

- $\emptyset \rightarrow \{a_2\}$ is
- $\{a_1\} \rightarrow \{a_2\}$ is
- $\{a_2\} \rightarrow \{a_1\}$ is
- $X \rightarrow \emptyset$ is
### Lift

**Definition**

Given a set of objects \( \mathcal{O} \) and a set of Boolean attributes \( \mathcal{A} \), the *lift* of an association rule \( X \rightarrow Y \) in a dataset \( \mathcal{D} \subseteq \mathcal{O} \times \mathcal{A} \) is

\[
\text{lift}(X \cup Y, \mathcal{D}) = \frac{|O| \cdot |f(X \cup Y, \mathcal{D})|}{|f(X, \mathcal{D})| \cdot |f(Y, \mathcal{D})|} = \frac{|O| f(X \cup Y, \mathcal{D})}{f(X, \mathcal{D}) f(Y, \mathcal{D})}.
\]

The lift of:

<table>
<thead>
<tr>
<th></th>
<th>( a_1 )</th>
<th>( a_2 )</th>
<th>( a_3 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( o_1 )</td>
<td>( \times )</td>
<td>( \times )</td>
<td>( \times )</td>
</tr>
<tr>
<td>( o_2 )</td>
<td>( \times )</td>
<td>( \times )</td>
<td>\   \</td>
</tr>
<tr>
<td>( o_3 )</td>
<td>\</td>
<td>( \times )</td>
<td>\</td>
</tr>
<tr>
<td>( o_4 )</td>
<td>\</td>
<td>\</td>
<td>( \times )</td>
</tr>
</tbody>
</table>

- \( \emptyset \rightarrow \{ a_2 \} \) is 1;
- \( \{ a_1 \} \rightarrow \{ a_2 \} \) is
- \( \{ a_2 \} \rightarrow \{ a_1 \} \) is
- \( X \rightarrow \emptyset \) is
Lift

Definition

Given a set of objects \( \mathcal{O} \) and a set of Boolean attributes \( \mathcal{A} \), the \textit{lift} of an association rule \( X \rightarrow Y \) in a dataset \( \mathcal{D} \subseteq \mathcal{O} \times \mathcal{A} \) is

\[
\frac{f(X \cup Y, \mathcal{D})}{|\mathcal{O}|} \cdot \frac{|\mathcal{O}|}{f(X, \mathcal{D})} \cdot \frac{f(Y, \mathcal{D})}{f(Y, \mathcal{D})} = \frac{|\mathcal{O}|f(X \cup Y, \mathcal{D})}{f(X, \mathcal{D})f(Y, \mathcal{D})}.
\]

The lift of:

<table>
<thead>
<tr>
<th></th>
<th>( a_1 )</th>
<th>( a_2 )</th>
<th>( a_3 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( o_1 )</td>
<td>( \times )</td>
<td>( \times )</td>
<td>( \times )</td>
</tr>
<tr>
<td>( o_2 )</td>
<td>( \times )</td>
<td>( \times )</td>
<td></td>
</tr>
<tr>
<td>( o_3 )</td>
<td></td>
<td>( \times )</td>
<td></td>
</tr>
<tr>
<td>( o_4 )</td>
<td></td>
<td></td>
<td>( \times )</td>
</tr>
</tbody>
</table>

- \( \emptyset \rightarrow \{a_2\} \) is 1;
- \( \{a_1\} \rightarrow \{a_2\} \) is 1.33;
- \( \{a_2\} \rightarrow \{a_1\} \) is
- \( X \rightarrow \emptyset \) is
Association Rule Mining (in Chapter 10)

Lift

**Definition**

Given a set of objects $\mathcal{O}$ and a set of Boolean attributes $\mathcal{A}$, the lift of an association rule $X \rightarrow Y$ in a dataset $\mathcal{D} \subseteq \mathcal{O} \times \mathcal{A}$ is

$$\frac{f(X \cup Y, \mathcal{D})}{|\mathcal{O}|} \frac{|\mathcal{O}|}{f(X, \mathcal{D}) f(Y, \mathcal{D})} = \frac{|\mathcal{O}| f(X \cup Y, \mathcal{D})}{f(X, \mathcal{D}) f(Y, \mathcal{D})}.$$

The lift of:

<table>
<thead>
<tr>
<th></th>
<th>$a_1$</th>
<th>$a_2$</th>
<th>$a_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$o_1$</td>
<td>$\times$</td>
<td>$\times$</td>
<td>$\times$</td>
</tr>
<tr>
<td>$o_2$</td>
<td>$\times$</td>
<td>$\times$</td>
<td></td>
</tr>
<tr>
<td>$o_3$</td>
<td></td>
<td>$\times$</td>
<td></td>
</tr>
<tr>
<td>$o_4$</td>
<td></td>
<td></td>
<td>$\times$</td>
</tr>
</tbody>
</table>

- $\emptyset \rightarrow \{a_2\}$ is 1;
- $\{a_1\} \rightarrow \{a_2\}$ is 1.33;
- $\{a_2\} \rightarrow \{a_1\}$ is 1.33;
- $X \rightarrow \emptyset$ is 1.8.
Association Rule Mining (in Chapter 10)

Lift

Definition

Given a set of objects $O$ and a set of Boolean attributes $A$, the \textit{lift} of an association rule $X \rightarrow Y$ in a dataset $D \subseteq O \times A$ is

\[
\frac{|O| f(X \cup Y, D)}{f(X, D) f(Y, D)} = \frac{|O| f(X \cup Y, D)}{f(X, D) f(Y, D)}.\]

The lift of:

<table>
<thead>
<tr>
<th></th>
<th>$a_1$</th>
<th>$a_2$</th>
<th>$a_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$O_1$</td>
<td>$\times$</td>
<td>$\times$</td>
<td>$\times$</td>
</tr>
<tr>
<td>$O_2$</td>
<td>$\times$</td>
<td>$\times$</td>
<td></td>
</tr>
<tr>
<td>$O_3$</td>
<td></td>
<td>$\times$</td>
<td></td>
</tr>
<tr>
<td>$O_4$</td>
<td></td>
<td></td>
<td>$\times$</td>
</tr>
</tbody>
</table>

- $\emptyset \rightarrow \{a_2\}$ is 1;
- $\{a_1\} \rightarrow \{a_2\}$ is 1.33;
- $\{a_2\} \rightarrow \{a_1\}$ is 1.33;
- $X \rightarrow \emptyset$ is 1.
Theorem

Given a set of objects $\mathcal{O}$, a set of Boolean attributes $\mathcal{A}$ and a dataset $\mathcal{D} \subseteq \mathcal{O} \times \mathcal{A}$, an association rule $X \rightarrow Y$ has the same frequency, the same confidence and at least the same lift as the association rule $X \rightarrow c(X \cup Y) \setminus X$ (where $c$ is the closure operator).
Association rules and closure

**Theorem**

Given a set of objects $\mathcal{O}$, a set of Boolean attributes $\mathcal{A}$ and a dataset $\mathcal{D} \subseteq \mathcal{O} \times \mathcal{A}$, an association rule $X \rightarrow Y$ has the same frequency, the same confidence and at least the same lift as the association rule $X \rightarrow c(X \cup Y) \setminus X$ (where $c$ is the closure operator).

By extensibility of $c$, $X \cup Y \subseteq c(X \cup Y)$ and the obtained rule is “more interesting” because it concludes on more attributes the associated quality measures are at least the same.
Theorem

Given a set of objects $\mathcal{O}$, a set of Boolean attributes $\mathcal{A}$ and a dataset $\mathcal{D} \subseteq \mathcal{O} \times \mathcal{A}$, an association rule $X \rightarrow Y$ has the same frequency, the same confidence and at least the same lift as the association rule $X \rightarrow c(X \cup Y) \setminus X$ (where $c$ is the closure operator).

By extensibility of $c$, $X \cup Y \subseteq c(X \cup Y)$ and the obtained rule is “more interesting” because it concludes on more attributes the associated quality measures are at least the same.

Therefore, the “most interesting” association rules can be obtained by post-processing the closed itemsets only.
Additionally, it can be shown that the association rule \( X \rightarrow c(X \cup Y) \setminus X \) is “more interesting” if \( X \) is at one of the smallest itemsets among those having the same supporting objects because it requires less attributes to apply and the quality measures are at least the same. Such an itemset \( X \) is called generator. All generators are a condensed representation of the frequent itemsets.
Additionally, it can be shown that the association rule \( X \rightarrow c(X \cup Y) \setminus X \) is “more interesting” if \( X \) is at one of the smallest itemsets among those having the same supporting objects because it requires less attributes to apply and the quality measures are at least the same. Such an itemset \( X \) is called generator. All generators are a condensed representation of the frequent itemsets.

Therefore, the “most interesting” association rules can be obtained by post-processing the closed itemsets and only considering generators for the left-hand side.
Association Rule Mining (in Chapter 10)

Support equivalence classes
Outline

1 Association Rule Mining (in Chapter 10)

2 Frequent Subgraph

3 Frequent Subgraph Mining

4 gSpan’s Enumeration

5 gSpan’s Graph Isomorphism Test

6 Conclusion
Frequent Subgraph

Labeled graph

**Definition**

Given a set of $m \in \mathbb{N}$ vertices $V = (v_i)_{i=1..m}$ and a set of labels $\Sigma$, a *labeled graph* is a 3-tuple $(V, E, L)$, where $E \subseteq \{(v_i, v_j) \in V^2 \mid i < j\}$ and $L \in \Sigma^{V \cup E}$.
Number of unlabeled graphs

Question

Given a set of $m \in \mathbb{N}$ vertices $V = (v_i)_{i=1..m}$, how many valid edges are there?
Question

Given a set of \( m \in \mathbb{N} \) vertices \( V = (v_i)_{i=1..m} \), how many valid edges are there? \( \binom{m}{2} = O(m^2) \).
Question
Given a set of \( m \in \mathbb{N} \) vertices \( V = (v_i)_{i=1..m} \), how many valid edges are there? \( \binom{m}{2} = O(m^2) \).
**Number of unlabeled graphs**

**Question**

Given a set of \( m \in \mathbb{N} \) vertices \( V = (v_i)_{i=1..m} \), how many valid edges are there? \( \binom{m}{2} = O(m^2) \).

**Question**

Given a set of \( m \in \mathbb{N} \) vertices \( V = (v_i)_{i=1..m} \), how many valid sets of edges are there? \( 2^{O(m^2)} \).
**Question**

Given a set of $m \in \mathbb{N}$ vertices $V = (v_i)_{i=1..m}$ and a set $\Sigma_V$ of vertex labels, how many valid vertex labelings are there?
Question

Given a set of $m \in \mathbb{N}$ vertices $V = (v_i)_{i=1..m}$ and a set $\Sigma_V$ of vertex labels, how many valid vertex labelings are there? $|\Sigma_V|^m$. 
Frequent Subgraph

Number of labelings

**Question**

Given a set of $m \in \mathbb{N}$ vertices $V = (v_i)_{i=1..m}$ and a set $\Sigma_V$ of vertex labels, how many valid vertex labelings are there? $|\Sigma_V|^m$.

**Question**

Given a set of $m \in \mathbb{N}$ vertices $V = (v_i)_{i=1..m}$ and a set $\Sigma_E$ of edge labels, how many valid labeled edges are there?
Frequent Subgraph

Number of labelings

**Question**

Given a set of \( m \in \mathbb{N} \) vertices \( V = (v_i)_{i=1..m} \) and a set \( \Sigma_V \) of vertex labels, how many valid vertex labelings are there? \( |\Sigma_V|^m \).

**Question**

Given a set of \( m \in \mathbb{N} \) vertices \( V = (v_i)_{i=1..m} \) and a set \( \Sigma_E \) of edge labels, how many valid labeled edges are there? \( |\Sigma_E|^{O(m^2)} \).
Frequent Subgraph

Number of labeled graphs

**Question**

Given a set of \( m \in \mathbb{N} \) vertices \( V = (v_i)_{i=1}^m \), a set \( \Sigma_V \) of vertex labels and a set \( \Sigma_E \) of edge labels, how many valid labeled graphs are there?

\[ O\left( m^2 \right) \]
Question

Given a set of $m \in \mathbb{N}$ vertices $V = (v_i)_{i=1..m}$, a set $\Sigma_V$ of vertex labels and a set $\Sigma_E$ of edge labels, how many valid labeled graphs are there? $2^{O(m^2)}|\Sigma_V|^m|\Sigma_E|^{O(m^2)}$. 
A labeled graph \((V_1, E_1, L_1)\) is a subgraph of a labeled graph \((V_2, E_2, L_2)\) if and only if

\[
\begin{align*}
V_1 & \subseteq V_2 \\
E_1 & \subseteq E_2 \\
L_1 & \subseteq L_2
\end{align*}
\]

NB: These labeled graphs only have one edge label, which is not indicated.
**Definition**

Two labeled graphs \((V_1, E_1, L_1)\) and \((V_2, E_2, L_2)\) are **isomorphic**

\[
\Leftrightarrow \exists \phi \in V_2^{V_1} \mid \begin{cases} 
(v_i, v_j) \in E_1 \Leftrightarrow (\phi(v_i), \phi(v_j)) \in E_2 \\
\forall v \in V_1, L_1(v) = L_2(\phi(v)) \\
\forall (v_i, v_j) \in E_1, L_1(v_i, v_j) = L_2(\phi(v_i), \phi(v_j))
\end{cases}
\]

These two graphs are isomorphic:

![Graph 1](image1.png)

![Graph 2](image2.png)
**Definition**

Two labeled graphs \((V_1, E_1, L_1)\) and \((V_2, E_2, L_2)\) are isomorphic

\[
\iff \exists \phi \in V_2^{V_1} | \begin{cases} 
(v_i, v_j) \in E_1 \iff (\phi(v_i), \phi(v_j)) \in E_2 \\
\forall v \in V_1, L_1(v) = L_2(\phi(v)) \\
\forall (v_i, v_j) \in E_1, L_1(v_i, v_j) = L_2(\phi(v_i), \phi(v_j))
\end{cases}
\]

These two graphs are isomorphic:

\[
\phi(v_1) = v_3 \\
\phi(v_2) = v_1 \\
\phi(v_3) = v_4 \\
\phi(v_4) = v_2
\]
Given two (labeled) graphs, deciding whether they are isomorphic is an NP problem... and that is the best we know so far!
Given two (labeled) graphs, deciding whether they are isomorphic is an NP problem... and that is the best we know so far!

The best known algorithm so far has a time complexity in $2^{O(\sqrt{|V| \log(|V|))}}$. 
Frequent Subgraph

Subgraph isomorphism

**Definition**

A labeled graph \((V_1, E_1, L_1)\) is **subgraph isomorphic** to a labeled graph \((V_2, E_2, L_2)\)

\[
\iff \exists \phi \in V_2^{V_1} \mid \begin{cases}
(v_i, v_j) \in E_1 \Rightarrow (\phi(v_i), \phi(v_j)) \in E_2 \\
\forall v \in V_1, L_1(v) = L_2(\phi(v)) \\
\forall(v_i, v_j) \in E_1, L_1(v_i, v_j) = L_2(\phi(v_i), \phi(v_j))
\end{cases}
\]

The rightmost graph is subgraph isomorphic to the leftmost one:
Subgraph isomorphism

Definition

A labeled graph \((V_1, E_1, L_1)\) is **subgraph isomorphic** to a labeled graph \((V_2, E_2, L_2)\)

\[
\begin{align*}
\text{IF} & \exists \phi \in V_2^{V_1} \mid \begin{cases} 
(v_i, v_j) \in E_1 \Rightarrow (\phi(v_i), \phi(v_j)) \in E_2 \\
\forall v \in V_1, L_1(v) = L_2(\phi(v)) \\
\forall (v_i, v_j) \in E_1, L_1(v_i, v_j) = L_2(\phi(v_i), \phi(v_j))
\end{cases}
\end{align*}
\]

The rightmost graph is subgraph isomorphic to the leftmost one:

\[
\begin{align*}
\phi(v_1) &= v_1 \\
\phi(v_2) &= v_3 \\
\phi(v_3) &= v_2
\end{align*}
\]
The subgraph isomorphism problem is NP-complete.
**Definition**

A graph dataset is a finite set of labeled graphs.
Frequent Subgraph

Frequency

**Definition (absolute frequency)**

Given a graph dataset $\mathcal{D}$, the absolute *frequency* of a labeled graph $G$ in $\mathcal{D}$ is $|\{S \in \mathcal{D} \mid G \text{ is subgraph isomorphic to } S\}|$. 
**Frequent Subgraph Frequency**

**Definition (absolute frequency)**

Given a graph dataset $D$, the absolute frequency of a labeled graph $G$ in $D$ is $|\{S \in D \mid G \text{ is subgraph isomorphic to } S\}|$.

**Definition (relative frequency)**

Given a graph dataset $D$, the relative frequency of a labeled graph $G$ in $D$ is $\frac{|\{S \in D \mid G \text{ is subgraph isomorphic to } S\}|}{|D|}$. 
Frequent Subgraph

Frequency

**Definition (absolute frequency)**
Given a graph dataset \( \mathcal{D} \), the absolute frequency of a labeled graph \( G \) in \( \mathcal{D} \) is \( f(G, \mathcal{D}) \).

**Definition (relative frequency)**
Given a graph dataset \( \mathcal{D} \), the relative frequency of a labeled graph \( G \) in \( \mathcal{D} \) is \( \frac{|\{S \in \mathcal{D} \mid G \text{ is subgraph isomorphic to } S\}|}{|\mathcal{D}|} \).
Outline

1. Association Rule Mining (in Chapter 10)
2. Frequent Subgraph
3. Frequent Subgraph Mining
4. gSpan’s Enumeration
5. gSpan’s Graph Isomorphism Test
6. Conclusion
Frequent Subgraph Mining

Frequent subgraph mining

Definition

Given a graph dataset \( \mathcal{D} \), listing every labeled graph having a frequency above a given threshold \( \mu \in \mathbb{N} \).
Frequent Subgraph Mining

Frequent connected subgraph mining

**Definition**

A labeled graph \((V, E, L)\) is **connected** if:

\[
\exists (p_i)_{i \in \mathbb{N}} \in V^\mathbb{N} \mid \begin{cases} 
\forall i \in \mathbb{N} \mid (p_i, p_{i+1}) \in E \lor (p_{i+1}, p_i) \in E \\
\forall v \in V, \exists i \in \mathbb{N} \mid v = p_i
\end{cases}
\]

Given a graph dataset \(D\), listing every connected labeled graph having a frequency above a given threshold \(\mu \in \mathbb{N}\).
Frequent Subgraph Mining

Frequent connected subgraph mining

**Definition**

A labeled graph \((V, E, L)\) is connected

\[
\Leftrightarrow \exists (p_i)_{i \in \mathbb{N}} \in V^{\mathbb{N}} \mid \begin{cases} 
\forall i \in \mathbb{N} \mid (p_i, p_{i+1}) \in E \lor (p_{i+1}, p_i) \in E \\
\forall v \in V, \exists i \in \mathbb{N} \mid v = p_i 
\end{cases}
\]

**Definition**

Given a graph dataset \(D\), listing every connected labeled graph having a frequency above a given threshold \(\mu \in \mathbb{N}\).
Example of applicative problem

**Frequent molecular sub-structure**

In a dataset of molecules performing a same function, are there common sub-structures shared by many of them?
gSpan’s Enumeration

Outline

1. Association Rule Mining (in Chapter 10)
2. Frequent Subgraph
3. Frequent Subgraph Mining
4. gSpan’s Enumeration
5. gSpan’s Graph Isomorphism Test
6. Conclusion
Definition

A spanning tree of a connected labeled graph $G = (V_G, E_G, L_G)$ is a labeled tree $T = (V_G, E_T, L_T)$ that is a subgraph of $G$. 
Definition

A *spanning tree* of a connected labeled graph $G = (V_G, E_G, L_G)$ is a labeled tree $T = (V_G, E_T, L_T)$ that is a subgraph of $G$.

Definition

The *depth-first spanning tree* of a connected labeled graph $G = (V_G, E_G, L_G)$ with numbered vertices (i.e., $V_G = (v_i)_{i=1..|V_G|}$) is the spanning tree of $G$ obtained by depth-first traversal of $G$ choosing the next vertex to access by respecting the precedence $v_i \prec v_j \iff i < j$. 
Question

What is the depth-first spanning tree of this labeled graph:

```
  v1  a   v2
 C  a   b  v3
  v4  c   v5
    d   c   a
    v6   v5  v7
             v8
```
Question

What is the depth-first spanning tree of this labeled graph:

\[\begin{align*}
\text{\(v_1\)} & \quad \text{\(a\)} & \quad \text{\(a\)} & \quad \text{\(v_2\)} & \quad \text{\(b\)} & \quad \text{\(v_3\)} \\
\text{\(v_4\)} & \quad \text{\(c\)} & \quad \text{\(c\)} & \quad \text{\(v_5\)} & \quad \text{\(a\)} & \quad \text{\(v_7\)} \\
\text{\(v_6\)} & \quad \text{\(d\)} & \quad \text{\(v_5\)} & \quad \text{\(a\)} & \quad \text{\(v_8\)} & \quad
\end{align*}\]
gSpan's Enumeration

Rightmost path

**Definition**

The *rightmost path* of a connected labeled graph $G = (V_G, E_G, L_G)$ with numbered vertices (i.e., $V_G = (v_i)_{i=1}^{\vert V_G \vert}$) is the path that is a subgraph of its depth-first spanning tree and connects $v_1$ to $v_{\vert V_G \vert}$.
In this example, there are six ordered children to be multiplied by all possible labels for the added edge and, if a vertex is added, all possible labels for the added vertex.
gSpan’s Enumeration

Extensions of the rightmost path

In this example, there are six ordered children to be multiplied by all possible labels for the added edge and, if a vertex is added, all possible labels for the added vertex.
In this example, there are six ordered children to be multiplied by all possible labels for the added edge and, if a vertex is added, all possible labels for the added vertex.
gSpan’s Enumeration

Rightmost path extension

The rightmost path extension enumerates connected labeled graphs.
The rightmost path extension enumerates connected labeled graphs.

“Being graph isomorphic” is an equivalency relation.
Rightmost path extension

The rightmost path extension enumerates connected labeled graphs.

“Being graph isomorphic” is an equivalency relation.

Theorem

The rightmost path extension enumerates at least one representative of every equivalence class (completeness).
gSpan’s Enumeration

Rightmost path extension

The rightmost path extension enumerates connected labeled graphs.

“Being graph isomorphic” is an equivalency relation.

**Theorem**

The rightmost path extension enumerates at least one representative of every equivalence class (completeness).

**Theorem**

If a labeled graph is isomorphic to a previously enumerated labeled graph, every labeled graph in the enumeration subtree it roots is isomorphic to a previously enumerated labeled graph (pruning).
gSpan’s Enumeration

Anti-monotonicity of the frequency

The rightmost path extension enumerates, depth-first, growing connected labeled graphs.

Theorem

Given a graph dataset $D$ and two labeled graphs $G_1$ and $G_2$:

$G_1$ subgraph isomorphic to $G_2$ $\Rightarrow f(G_1, D) \geq f(G_2, D)$.

Whenever a labeled graph is shown infrequent, the extensions of it need not be considered (pruning).
gSpan’s Enumeration

Anti-monotonicity of the frequency

The rightmost path extension enumerates, depth-first, growing connected labeled graphs.

**Theorem**

Given a graph dataset $\mathcal{D}$ and two labeled graphs $G_1$ and $G_2$:

$$G_1 \text{ subgraph isomorphic to } G_2 \Rightarrow f(G_1, \mathcal{D}) \geq f(G_2, \mathcal{D}).$$
The rightmost path extension enumerates, depth-first, growing connected labeled graphs.

**Corollary**

Given a graph dataset $\mathcal{D}$, a minimal frequency $\mu \in \mathbb{N}$ and two labeled graphs $G_1$ and $G_2$:

$$G_1 \text{ subgraph iso. to } G_2 \Rightarrow \left( f(G_1, \mathcal{D}) < \mu \Rightarrow f(G_2, \mathcal{D}) < \mu \right).$$
gSpan’s Enumeration

Anti-monotonicity of the frequency

The rightmost path extension enumerates, depth-first, growing connected labeled graphs.

**Corollary**

Given a graph dataset $\mathcal{D}$, a minimal frequency $\mu \in \mathbb{N}$ and two labeled graphs $G_1$ and $G_2$:

$$G_1 \text{ subgraph iso. to } G_2 \Rightarrow \left( f(G_1, \mathcal{D}) < \mu \Rightarrow f(G_2, \mathcal{D}) < \mu \right).$$

Whenever a labeled graph is shown infrequent, the extensions of it need not be considered (pruning).
gSpan's Enumeration

Pruning: illustration

$G_{12}$ a

$G_{13}$ a

$G_{14}$ a

$G_{15}$ a

$G_{16}$ a

$G_{17}$ a

$G_{18}$ a

$G_{19}$ a

$G_{20}$ a

$G_{21}$ a

$G_{22}$ a

Fundamentals of Data Mining Algorithms
“Simple” enumeration

The labeled graphs obtained by rightmost path extension of a parent graph are enumerated breadth-first (all labeled graphs on a same level have the same number of edges).
gSpan’s Enumeration

“Simple” enumeration

The labeled graphs obtained by rightmost path extension of a parent graph are enumerated breadth-first (all labeled graphs on a same level have the same number of edges).

A labeled graph is added to the children level unless:

- it is subgraph isomorphic to strictly less than $\gamma$ subgraphs in the dataset (the labeled graphs that are not in the support of the parent need not be tested);
gSpan’s Enumeration

“Simple” enumeration

The labeled graphs obtained by rightmost path extension of a parent graph are enumerated breadth-first (all labeled graphs on a same level have the same number of edges).

A labeled graph is added to the children level unless:

- it is subgraph isomorphic to strictly less than $\gamma$ subgraphs in the dataset (the labeled graphs that are not in the support of the parent need not be tested);
- it is graph isomorphic to at least one labeled graph previously added to the children level (tests against every labeled graph previously added to the children level).
gSpan’s Enumeration

“Simple” enumeration

The labeled graphs obtained by rightmost path extension of a parent graph are enumerated breadth-first (all labeled graphs on a same level have the same number of edges).

A labeled graph is added to the children level unless:

- it is subgraph isomorphic to strictly less than $\gamma$ subgraphs in the dataset (the labeled graphs that are not in the support of the parent need not be tested);

- it is graph isomorphic to at least one labeled graph previously added to the children level (tests against every labeled graph previously added to the children level).
gSpan’s Enumeration

“Simple” enumeration

The labeled graphs obtained by rightmost path extension of a parent graph are enumerated breadth-first (all labeled graphs on a same level have the same number of edges).

A labeled graph is added to the children level unless:

- it is subgraph isomorphic to strictly less than $\gamma$ subgraphs in the dataset (the labeled graphs that are not in the support of the parent need not be tested);

- it is graph isomorphic to at least one labeled graph previously added to the children level (tests against every labeled graph previously added to the children level).
Outline

1. Association Rule Mining (in Chapter 10)
2. Frequent Subgraph
3. Frequent Subgraph Mining
4. gSpan’s Enumeration
5. gSpan’s Graph Isomorphism Test
6. Conclusion
The labeled graphs obtained by rightmost path extension of a parent graph are enumerated depth-first.

A labeled graph is added to the children level unless:

- it is subgraph isomorphic to strictly less than $\gamma$ subgraphs in the dataset (the labeled graphs that are not in the support of the parent need not be tested);
- it is graph isomorphic to at least one graph previously added to the children level (test whether it is the first one among all isomorphic labeled graph potentially generated).
gSpan’s Graph Isomorphism Test

**gSpan enumeration**

The labeled graphs obtained by rightmost path extension of a parent graph are enumerated **depth-first**.

A labeled graph is added to the children level unless:

- it is subgraph isomorphic to strictly less than $\gamma$ subgraphs in the dataset (the labeled graphs that are not in the support of the parent need not be tested);

- it is graph isomorphic to at least one graph previously added to the children level (test whether it is the first one among all isomorphic labeled graph potentially generated).
The labeled graphs obtained by rightmost path extension of a parent graph are enumerated depth-first.

A labeled graph is added to the children level unless:

- it is subgraph isomorphic to strictly less than $\gamma$ subgraphs in the dataset (the labeled graphs that are not in the support of the parent need not be tested);
- it is graph isomorphic to at least one graph previously added to the children level (test whether it is the first one among all isomorphic labeled graph potentially generated).
gSpan’s Graph Isomorphism Test

Canonical code

“Being graph isomorphic” is an equivalency relation.
gSpan’s Graph Isomorphism Test

Canonical code

“Being graph isomorphic” is an equivalency relation.

A function of graphs can be built so that it returned a same code if and only if it is given, at input, isomorphic graphs.
gSpan’s Graph Isomorphism Test

Canonical code

“Being graph isomorphic” is an equivalency relation.

A function of graphs can be built so that it returned a same code if and only if it is given, at input, isomorphic graphs.

For instance, each graph can be represented as the concatenation of the rows of its adjacency matrix and the canonical code could be the lexicographically smallest string for all permutations of the vertex numbering.
The following theorem prunes the search space and imposes a canonical code to solve graph isomorphism:

**Theorem**

If a labeled graph is isomorphic to a previously enumerated labeled graph, every labeled graph in the enumeration subtree it roots is isomorphic to a previously enumerated labeled graph.
The following theorem prunes the search space and imposes a canonical code to solve graph isomorphism:

**Theorem**

If a labeled graph is isomorphic to a previously enumerated labeled graph, every labeled graph in the enumeration subtree it roots is isomorphic to a previously enumerated labeled graph.

The canonical code must identify the first isomorphic graph to be enumerated by rightmost path extension.
gSpan’s Graph Isomorphism Test

DFS code

The labeled graphs are represented by a the list of edges (with the associated vertex and edge labeling) in which they were appended by rightmost path extension.

\[
\begin{align*}
(v_1, v_2, aaq) \\
(v_2, v_3, abr) \\
(v_2, v_4, aar) \\
(v_4, v_1, aar)
\end{align*}
\]
The enumeration order of the graph relates to a total order of their DFS codes that depends on:

- their first different edges in the order in which they were appended;

- if the numbering of the linked vertices is different, rules on these numberings that depend on the “nature” of the two edges (“backward”, i.e., linking two previously added vertices; or “forward”, i.e., involving the addition of a new vertex);

- if only the labelings differ, the lexicographic order on the string they form.
gSpan's Graph Isomorphism Test

Order on the DFS code: illustration

\[(v_1, v_2, aaq) \succ (v_2, v_3, abr) \succ (v_2, v_4, aar) \succ (v_4, v_1, aar)\]
The spanning trees of a child graph are enumerated. For each tree, a new numbering of the vertices is obtained so that this tree is a DFS spanning tree. The DFS code relating to this numbering is compared to that of the child graph. If smaller, a graph that is isomorphic to the child graph was previously enumerated.
gSpan’s Graph Isomorphism Test

Graph isomorphism test

The spanning trees of a child graph are enumerated. For each tree, a new numbering of the vertices is obtained so that this tree is a DFS spanning tree. The DFS code relating to this numbering is compared to that of the child graph. If smaller, a graph that is isomorphic to the child graph was previously enumerated.

Since the order on the DFS codes depends on the first different edge in the rightmost path extension order, the construction of a spanning tree can stop as soon as an edge differs from that of the child graph. In this way, all spanning trees sharing this “base” are tested at the same time.
Conclusion

Outline

1. Association Rule Mining (in Chapter 10)
2. Frequent Subgraph
3. Frequent Subgraph Mining
4. gSpan’s Enumeration
5. gSpan’s Graph Isomorphism Test
6. Conclusion
Condensed representations of the frequent subgraphs are defined in the same way than those of frequent itemsets;
Condensed representations of the frequent subgraphs are defined in the same way than those of frequent itemsets;

Graphs can be labeled with itemsets, sequences or even graphs;
Advanced topics

- Condensed representations of the frequent subgraphs are defined in the same way than those of frequent itemsets;

- Graphs can be labeled with itemsets, sequences or even graphs;

- Mining subgraphs that are frequent in one large labeled graph is more difficult because the chosen definition of frequency usually is not anti-monotone.
Don’t be afraid of simpler patterns

Mining arbitrary graphs is harder than mining specific ones (e.g., planar graphs, trees or sequences). Always wonder whether this complexity is useful.
These slides are licensed under the Creative Commons Attribution-ShareAlike 3.0 Unported License.