Finding Representative Patterns for a Set of Subgraphs

Furkan Kocayusufoglu and Isaac Mackey
Fall 2016, Network Science Project
Network Processes

- Network processes occurring on one large network
  - Belief propagation, disease propagation
  - Users spreading and sharing multiple attributes
Communities

- Subsets of nodes, assumed to be connected, that participates together in network processes
Communities

- We want to identify communities using network processes.
Network processes

• Given a network and a set of processes, there is a subgraph associated with each process.

• A subgraph is the set of nodes participating in a certain process.

• Mining - Use these subgraphs (use these processes) to identify common patterns (communities)
Mining Network Processes

- Use these subgraphs (use these processes) to identify common patterns (communities)
Mining Network Processes

- Use these subgraphs (use these processes) to identify common patterns (communities)
Mining Network Processes

Subgraphs

#sports
#food
#nature
#music
Mining Network Processes

Find communities who participate often in same process

Subgraphs

#sports
#food
#nature
#music
Community Detection

- Often done using structural properties of network

- Here we use network processes
Community Detection with Frequent Subgraph Mining
Community Detection with FSM
Community Detection with Representative Sampling
How do we get them?

- Set of representative patterns which accurately represent set of processes (subgraphs)

Underlying graph

Potential patterns
How do we get them?

- Sample from underlying graph
Markov Chain Monte Carlo (MCMC) method for obtaining a sequence of random samples from a distribution that's difficult to directly sample
Use Partial Order Graph
Use Partial Order Graph
Estimating $P(x)$

- For distribution $P(x)$, we need function $f(x)$ such that

\[ \frac{f(x)}{f(x')} = \frac{P(x)}{P(x')} \]
Estimating $P(x)$

• We need function $f(x)$ that maximizes a pattern’s frequency and novelty in the set of subgraphs.
Matrix Factorization

Subgraphs

G1

A B C D

G2

A B C

G3

A B E

Pattern Set

P1

A B C

P2

A F

\[
\begin{align*}
\text{G1} & : \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 1 \end{bmatrix} \\
\text{G2} & : \begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 0 \end{bmatrix} \\
\text{G3} & : \begin{bmatrix} 1 & 1 & 0 & 0 & 1 & 1 \end{bmatrix}
\end{align*}
\]

\[
A \approx \begin{bmatrix} 1 & 1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}
\]
Gain

• Intuition about how representative set of patterns is
• Rooted in matrix factorization

\[
\begin{align*}
\text{A} & = \begin{bmatrix}
1 & 1 & 1 & 0 & 1 \\
1 & 1 & 0 & 1 & 0 \\
1 & 1 & 0 & 0 & 1
\end{bmatrix} \\
\text{B} & = \begin{bmatrix}
1 & 1 & 1 & 0 \\
1 & 0 & 0 & 0
\end{bmatrix} \\
\text{C} & = \begin{bmatrix}
1 & 1 & 1 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 1
\end{bmatrix}
\end{align*}
\]

• Assign value to similarity of A and B * C to measure how representative pattern set is
Gain

• We have $f(x)$.

• For potential patterns $x$ and $x'$, the ratio of the gain of $x$ to the gain of $x'$ is proportional to density of $P(x)$.

\[
\begin{align*}
\mathbf{f} (\begin{bmatrix}
1 & 1 & 1 & 1 & 0 & 1 \\
1 & 1 & 1 & 0 & 1 & 0 \\
1 & 1 & 0 & 0 & 1 & 1
\end{bmatrix} \text{ nodes}) & \approx \begin{bmatrix}
1 \\
1 \\
0
\end{bmatrix} \text{ patterns} \\
\mathbf{f} (\begin{bmatrix}
1 & 1 & 1 & 1 & 0 & 1 \\
1 & 1 & 1 & 0 & 1 & 0 \\
1 & 1 & 0 & 0 & 1 & 1
\end{bmatrix} \text{ nodes}) & \approx \begin{bmatrix}
1 \\
1 \\
0
\end{bmatrix} \text{ patterns}
\end{align*}
\]
\[
\begin{align*}
\text{nodes} & \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 \end{bmatrix} \quad \approx \quad \text{graphs} \quad \begin{bmatrix} 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 \end{bmatrix} \\
\text{patterns} & \begin{bmatrix} 1 \\ 1 & \mathbf{x} \\ 0 \end{bmatrix} \quad \begin{bmatrix} 1 \\ 1 & \mathbf{x}' \\ 0 \end{bmatrix} \\
\text{nodes} & \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \end{bmatrix} \quad \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ \mathbf{x} \\ \mathbf{x}' \end{bmatrix}
\end{align*}
\]
Transition

• Transition proportional to ratio of gain of new and current patterns

• Bias to get larger patterns because we start sampling from single nodes
Random Walk on POG for Sampling

1: procedure Build Representative Set($G, Subgraphs, k$)
2:     pattern_set $\leftarrow [ ]$
3:     while length(pattern_set) $< k$ do
4:         current $\leftarrow$ random_choice($G$.nodes())
5:         for number_iterations do
6:             neighbor $= \text{random_choice}($neighbors of current in POG of $G$)
7:             current_gain $\leftarrow$ gain($Subgraphs, pattern_set + current$)
8:             neighbor_gain $\leftarrow$ gain($Subgraphs, pattern_set + neighbor$)
9:             ratio $= \frac{neighbor\_gain}{current\_gain}$
10:            current $\leftarrow$ neighbor when ratio $> \text{threshold}$
11:     return pattern_set
Motivation from Sampling

- Keep algorithm analysis useful

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Nodes</th>
<th>Edges</th>
<th>Communities</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>com-LiveJournal</td>
<td>Undirected, Communities</td>
<td>3,997,962</td>
<td>34,681,189</td>
<td>287,512</td>
<td>LiveJournal online social network</td>
</tr>
<tr>
<td>com-Friendster</td>
<td>Undirected, Communities</td>
<td>65,608,366</td>
<td>1,806,067,135</td>
<td>957,154</td>
<td>Friendster online social network</td>
</tr>
<tr>
<td>com-Orkut</td>
<td>Undirected, Communities</td>
<td>3,072,441</td>
<td>117,185,083</td>
<td>6,288,363</td>
<td>Orkut online social network</td>
</tr>
<tr>
<td>com-Youtube</td>
<td>Undirected, Communities</td>
<td>1,134,890</td>
<td>2,987,624</td>
<td>8,385</td>
<td>Youtube online social network</td>
</tr>
<tr>
<td>com-DBLP</td>
<td>Undirected, Communities</td>
<td>317,080</td>
<td>1,049,866</td>
<td>13,477</td>
<td>DBLP collaboration network</td>
</tr>
<tr>
<td>com-Amazon</td>
<td>Undirected, Communities</td>
<td>334,863</td>
<td>925,872</td>
<td>75,149</td>
<td>Amazon product network</td>
</tr>
</tbody>
</table>
# Data Progress

<table>
<thead>
<tr>
<th>Days until Presentation</th>
<th>Underlying Graph</th>
<th>Subgraphs</th>
<th>Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>40</td>
<td>Real</td>
<td>Real</td>
<td>Twitter</td>
</tr>
<tr>
<td>39</td>
<td>Real</td>
<td>Synthetic</td>
<td>Flickr</td>
</tr>
<tr>
<td><strong>Exp. 1</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>Synthetic</td>
<td>Synthetic</td>
<td>Barabasi</td>
</tr>
<tr>
<td><strong>Exp. 2</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Real</td>
<td>Real</td>
<td>Amazon</td>
</tr>
</tbody>
</table>
Experiment 1

1: procedure Build Representative Set\((G, \text{Subgraphs}, k)\)

- Synthetic Barabasi-Albert
  \(n = 5000\), average degree = 10
- Synthetic subgraphs from communities
Noise

- Make data realistic

Synthetic Subgraphs $\xrightarrow{\text{Randomly adding and deleting nodes}}$ Noisy Subgraphs

Deleting nodes | Original | Adding nodes
Accuracy vs. Number of Patterns with 15 Communities

Accuracy

Number of Patterns

n = 5000
subgraphs = 500
noise = 5
Accuracy vs. Noise for Communities of Size 20

Similarity of Patterns and Ground Truth Communities

Noise

$n = 5000$
subgraphs = 500
## Real Data

### Stanford Large Network Dataset Collection

### Networks with ground-truth communities

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Nodes</th>
<th>Edges</th>
<th>Communities</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>com-LiveJournal</td>
<td>Undirected, Communities</td>
<td>3,997,962</td>
<td>34,681,189</td>
<td>287,512</td>
<td>LiveJournal online social network</td>
</tr>
<tr>
<td>com-Friendster</td>
<td>Undirected, Communities</td>
<td>65,608,366</td>
<td>1,806,067,135</td>
<td>957,154</td>
<td>Friendster online social network</td>
</tr>
<tr>
<td>com-Orkut</td>
<td>Undirected, Communities</td>
<td>3,072,441</td>
<td>117,185,083</td>
<td>6,288,363</td>
<td>Orkut online social network</td>
</tr>
<tr>
<td>com-Youtube</td>
<td>Undirected, Communities</td>
<td>1,134,890</td>
<td>2,987,624</td>
<td>8,385</td>
<td>Youtube online social network</td>
</tr>
<tr>
<td>com-DBLP</td>
<td>Undirected, Communities</td>
<td>317,980</td>
<td>1,040,866</td>
<td>13,477</td>
<td>DBLP collaboration network</td>
</tr>
<tr>
<td>com-Amazon</td>
<td>Undirected, Communities</td>
<td>334,863</td>
<td>925,872</td>
<td>75,149</td>
<td>Amazon product network</td>
</tr>
</tbody>
</table>
Data

Stanford Large Network Dataset Collection

- Crawl of Amazon product network using “Customers Who Bought This Item Also Bought” feature
- If a product i is frequently co-purchased with product j, the graph contains an edge (i,j)
Data

Stanford Large Network Dataset Collection

- Crawl of Amazon product network using “Customers Who Bought This Item Also Bought” feature
- If a product \( i \) is frequently co-purchased with product \( j \), the graph contains an edge \((i,j)\)
Our Data

**Stanford Large Network Dataset Collection**

- Product categories = communities
- 500 best communities (based on conductance and clustering) on connected graph
- These 500 communities have in total 6648 nodes, 22,029 edges
Experiment 2

1: **procedure** Build Representative Set \((G, Subgraphs, k)\)

Amazon data
- 334,862 nodes
- 925,872 edges

Reduced to
- 6,648 nodes
- 22,029 edges

\(\times 500\)
ASSO

• Uses association matrix

• $A(i,j) = 1$ if $c(i \rightarrow j) >$ threshold

• Chooses best $k$ rows of association matrix to maximize cover

• Ignored graph structure when applied to pattern mining

Average Pattern Size vs. Number of Patterns

- **Sampling**
- **ASSO**

**Axes:**
- **Y-axis:** Avg Pattern Size
- **X-axis:** Number of Patterns
ASSO vs. Sampling

One of the ASSO patterns with 42 nodes

One of our patterns with 40 nodes
ASSO vs. Sampling

• If ASSO has higher accuracy, why bother using a sampling algorithm?
Runtime Comparison

• **Our Sampling Algorithm**
  • $O(\text{subgraphs} \times \text{average}_\text{subgraph}_\text{degree} + \text{patterns}^2 \times \text{subgraphs} \times \text{nodes}) = O(k^2 n m)$
  • **Linear** in number of nodes

• **ASSO**
  • $O(\text{patterns} \times \text{subgraphs} \times \text{nodes}^2) = O(k n m^2)$
  • **Quadratic** in number of nodes
Future Work

- Application in social networks (human communities from Twitter, Facebook)
- Dynamic processes with dynamic patterns
References


