Overlapping Communities

Yangyang Hou, Mu Wang, Yongyang Yu

Purdue University
Department of Computer Science

April 25, 2013
Overview

Datasets

Algorithm I

Algorithm II

Algorithm III

Evaluation
Overview

Graph models of many real world applications exhibit an overlapping community structure, which is hard to grasp with the classical graph clustering methods where each vertex of the graph is assigned to exactly one community.

**GOAL:** Allow each vertex of the graph to belong to multiple communities at the same time.
Datasets

- Zachary’s karate club: social network of friendships between 34 members of a karate club at a US university in 1970s
- Karate dataset for result illustration
- American College football: network of American football games between Division IA colleges during regular season Fall 2000 (115 vertices, 615 edges)
Algorithm I: LA-IS2 (Baumes et al. 2005)
Link-Aggregate and Improved Iterative Scan (LA-IS2)

- Optimize density metric: $W_{ad}(C) = \frac{2|E(C)|}{|C|}$
- $E(C)$ – set of edges with both endpoints in $C$
- Intuition: average degree, the bigger the better for local communities
- Step 1: generate community candidates (LA step)
- Step 2: refine community candidates and remove duplicates (IS2 step)
LA-LS2 algorithm framework

1. LA step
   - Order the vertices based on PageRank values
   - Add $v_i$ to $C_j$ if $W_{ad}(C_j \cup v_i) > W_{ad}(C_j)$, else $C_k = \{v_i\}$

2. IS2 step
   - for each candidate $C_j$
   - $N \leftarrow C_j \cup adj(C_j)$
   - for all $v \in N$, if $v \in C_j$, $C' \leftarrow C_j \setminus \{v\}$, else $C' \leftarrow C_j \cup \{v\}$
   - $C_j \leftarrow C'$ if $W_{ad}(C') > W_{ad}(C_j)$
   - until $W_{ad}(C_j)$ no longer increases
Algorithm II (Zhang et al. 2007)
Spectral Mapping & Fuzzy C-means

- Minimize a modularity function:

\[
Q(U) = \sum_{c=1}^{k} \left[ \frac{A(V_c, V_c)}{A(V, V)} - \left( \frac{A(V_c, V)}{A(V, V)} \right)^2 \right]
\]

\[
A(V_c, V_c) = \sum_{i \in V_c, j \in V_c} \frac{(u_{i,c} + u_{j,c})}{2} w(i, j)
\]

\[
A(V_c, V) = A(V_c, V_c) + \sum_{i \in V_c, j \notin V_c} \frac{u_{i,c} + (1 - u_{j,c})}{2} w(i, j)
\]

\[
A(V, V) = \sum_{i \in V, j \in V} w(i, j)
\]
Algorithm II (Zhang et al. 2007)

- $u(i, c)$ describes the “possibility” that vertex $i$ belongs to cluster $c$
- $u(i, c)$ is solved by fuzzy $c$-means (a method of clustering which allows one piece of data to belong to two or more clusters)
- Using a given threshold to classify

$$J = \sum_{i=1}^{N} \sum_{k=1}^{C} u_{i,k} \| x_i - c_k \|$$
Algorithm II (Zhang et al. 2007)

1. Spectral mapping:
   - solve a generalized eigenvalue problem to get the top $K$ eigenvectors
     \[ Ax = \lambda Dx \]
   - where $A$ is the adjacency matrix, $D$ is the degree diagonal

2. Fuzzy $c$-means:
   - Normalize each row of the $K - 1$ eigenvectors (the largest one is ignored)
   - Cluster the row vectors using fuzzy $c$-means method to get $U$

3. Choose the largest result which maximizes $Q(U)$
Algorithm III (Nepusz et al., 2008)

Partition Matrix $U$ ($c \times |V|$):

- $u_{ik} \in [0, 1]$ for all $1 \leq i \leq c, 1 \leq k \leq N$
- $\sum_{i=1}^{c} u_{ik} = 1$ for all $1 \leq k \leq N$
- $0 \leq \sum_{k=1}^{N} u_{ik} \leq N$ for all $1 \leq i \leq c$

Derive the similarity matrix $S$ ($|V| \times |V|$) from $U$:

$s_{ij} = \sum_{k=1}^{c} u_{ik}u_{jk}$

or $S = U^T U$ in the matrix form.
Algorithm III (Nepusz et al., 2008)

Given the number of cluster $c$, the weight matrix $W$ and the desired similarities $\tilde{S}$, find $U$ that minimizes $D_G(U)$:

$$D_G(U) = \sum_{i=1}^{N} \sum_{j=1}^{N} w_{ij} (\tilde{s}_{ij} - s_{ij})^2$$

Usually, we set $\tilde{S} = A_G$. 
Algorithm III (Nepusz et al., 2008)

- Employ a gradient-based iterative optimization method.
- Initially, set \( c = 2 \). Keep on increasing the number of communities until the newly introduced community does not improve the overall community structure of the network.
Algorithm I for karate dataset
Algorithm I for karate dataset
Algorithm I for karate dataset
Algorithm II for karate dataset
Algorithm II for karate dataset
Algorithm II for karate dataset
Algorithm II for karate dataset
Algorithm II for karate dataset
Algorithm III for karate dataset
Algorithm III for karate dataset
Algorithm III for karate dataset
Algorithm III for karate dataset
Algorithm III for karate dataset
Measurements

- Normalized Cut

\[ N = \#\text{Cut-Edges} \times \left( \sum_{i=1}^{k} \frac{1}{d_i} \right) \]

- This also shows the number of shared nodes.

- Average Degree

\[ W_{ad}(C) = \frac{2|E(C)|}{|C|} \]

- Hope this will be significantly larger than the average degree.
# Result

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Algorithm I</th>
<th>Algorithm II</th>
<th>Algorithm III</th>
</tr>
</thead>
<tbody>
<tr>
<td>Karate</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>#Community</td>
<td>2</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>Normalized Cut</td>
<td>0</td>
<td>2.6184</td>
<td>3.6104</td>
</tr>
<tr>
<td>Shared nodes</td>
<td>2</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>Average Degree</td>
<td>3.9163</td>
<td>3.1607</td>
<td>2.9313</td>
</tr>
<tr>
<td>Football</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>#Community</td>
<td>10</td>
<td>5</td>
<td>12</td>
</tr>
<tr>
<td>Normalized Cut</td>
<td>12.6884</td>
<td>2.3668</td>
<td>16.3667</td>
</tr>
<tr>
<td>Shared nodes</td>
<td>46</td>
<td>18</td>
<td>43</td>
</tr>
<tr>
<td>Average Degree</td>
<td>7.8051</td>
<td>7.9167</td>
<td>6.3571</td>
</tr>
</tbody>
</table>
Currently working on..

▶ Testing three algorithms on large datasets.
▶ e.g. partial facebook network.
References


QUESTION?