Generalized network community detection

Lovro Šubelj and Marko Bajec

Faculty of Computer and Information Science, University of Ljubljana
http://lovro.lpt.fri.uni-lj.si/
lovro.subelj@fri.uni-lj.si

September 9, 2011

---

1 ECML PKDD Workshop on Finding Patterns of Human Behaviours in Network ... (NEMO '11)
Outline

1 Motivation

2 Classical community detection
   - Label propagation algorithm
   - Balanced propagation algorithm
   - Defensive propagation

3 Generalized community detection
   - General propagation algorithm
   - Model-based propagation algorithm

4 Empirical evaluation
   - Synthetic networks
   - Real-world networks

5 Conclusions & future work
Motivation

- Community structure is regarded as an intrinsic property of complex real-world—social and information—networks.
- Intuitively, communities correspond to groups of nodes densely connected within, and loosely connected between.
- They provide an insight into not only structural organization but also functional behavior of various real-world systems.

Still, the majority of past work was limited to cohesive modules of nodes—*link-density communities*. Recent work suggests more general structures may exist in real-world networks—*link-pattern communities*.

1) GN bench. 2) Karate club 3) South. women 4) JUNG communities
Outline

1 Motivation

2 Classical community detection
   - Label propagation algorithm
   - Balanced propagation algorithm
   - Defensive propagation

3 Generalized community detection
   - General propagation algorithm
   - Model-based propagation algorithm

4 Empirical evaluation
   - Synthetic networks
   - Real-world networks

5 Conclusions & future work
Label propagation algorithm

Undirected graph $G(N, L)$ with weights $W$ and communities $C$.

Label propagation algorithm (**LPA**) [Raghavan et al., 2007]:

1. initialize nodes with unique labels:

   $$\forall n \in N : c_n = l_n,$$

2. set node’s label to the label shared by most of its neighbors:

   $$\forall n \in N : c_n = \arg\max_l \sum_{m \in \Gamma_n^l} w_{nm},$$

3. repeat step 2. until convergence.

Algorithm has near linear time complexity $O(|L|) = O(k|N|)$. 
Balanced propagation algorithm

Oscillation of labels in, e.g., two-mode networks.

→ Labels are updated in a random order [Raghavan et al., 2007].

The above severely hampers the robustness of the algorithm.

→ Balanced propagation algorithm (BPA) [ˇSubelj & Bajec, 2011c]:

\[ \forall n \in N : c_n = \arg \max \sum_{m \in \Gamma'_n} b_m w_{nm} \]

where

\[ b_n = \frac{1}{1 + e^{-\mu(i_n - \lambda)}} \quad (\text{or } b_n = i_n). \]

\( i_n \) is a normalized position of node \( n \in N \) in a random order, \( i_n \in (0,1] \), while \( \lambda \) is fixed to \( \frac{1}{2} \) and \( \mu \) is set to 2.
Defensive propagation

Algorithm is further improved through defensive prop. [Šubelj & Bajec, 2011e]:

$$\forall n \in N : c_n = \arg\max_l \sum_{m \in \Gamma'_l} d_m b_m w_{nm}$$

where

$$d_n = \sum_{m \in \Gamma'_{c_n}} \frac{d_m}{k_{c_m}^m}.$$ 

Thus, higher and lower preferences are given to core and border nodes of each current community, respectively (estimated using a random walker).
Outline

1 Motivation

2 Classical community detection
   - Label propagation algorithm
   - Balanced propagation algorithm
   - Defensive propagation

3 Generalized community detection
   - General propagation algorithm
   - Model-based propagation algorithm

4 Empirical evaluation
   - Synthetic networks
   - Real-world networks

5 Conclusions & future work
General propagation algorithm

Label propagation cannot be directly applied for detection of link-pattern communities—prop. requires connected and cohesive modules of nodes.

Still, labels can be propagated through nodes’ neighbors.

\[ \forall n \in N : \arg \max_l \left( \delta_l \sum_{m \in \Gamma'_n} b_m d_m w_{nm} + (1 - \delta_l) \sum_{m \in \Gamma'_s \setminus \Gamma_n | s \in \Gamma_n} b_m \tilde{d}_m w_{nm}^s \right), \]

where \( \delta_l \in [0, 1] \) is close to 1 and 0 for link-density and link-pattern communities, respectively.

\[ w_{nm}^s = \frac{w_{ns} w_{sm}}{\sum_{m \in \Gamma_n} w_{nm}} \quad \text{and} \quad \tilde{d}_n = \sum_{m \in \Gamma'_s \setminus \Gamma_n | s \in \Gamma_n} \frac{\tilde{d}_m}{\sum_{s \in \Gamma_m} k_s^c_n}. \]
Community modeling

The core of GPA is in fact represented by community parameters $\delta_l!$

In GPA the type of each community is estimated by means of conductance $\Phi$ [Bollobas, 1998]. Hence,

$$\delta_c = 1 - \Phi(c) = \frac{\sum_{n \in N_c} k_n^c}{\sum_{n \in N_c} k_n}.$$

All $\delta_c$ are initially set to $\frac{1}{2}$. 
Model-based propagation algorithm

Weakness of GPA—each community is treated independently of others.

In an ideal case, *link-density* and *link-pattern communities* would link to other *link-density* and *link-pattern communities*, respectively.

→ Model-based propagation algorithm (MPA):

\[
\delta_c = \frac{1}{|N_c|} \sum_{m \in \Gamma_n | n \in N^c} \frac{\delta_{cm}}{k_n}.
\]

Initialization of \( \delta_c \) is of vital importance!
Model-based propagation algorithm—initialization

For initialization, the hypothesis is refined: node’s neighbors should not only reside in the same type of community, but in the same community.

Thus, $\delta_{c_n}$ could be initialized to clustering coefficient $C_n$ [Watts & Strogatz, 1998]. However, in many real-world networks $C_n \sim k_n^{-1}$.

Hence, we initialize $\delta_{c_n}$ as:

$$\delta_{c_n} = \begin{cases} 1 & \text{for } C_n > \alpha k_n^{-1} + \beta, \\ \rho & \text{otherwise,} \end{cases}$$

where $\alpha$, $\beta$ are estimated using ordinary least squares and $\rho$ is fixed to $\frac{1}{4}$. 
Model-based propagation algorithm—pseudo-code

### Algorithm (MPA)

**Input:** Graph $G(N, L)$ and parameters $\lambda$, $\mu$, $\rho$

**Output:** Communities $C$

\{Initialization.\}

\begin{algorithmic}
  \While {not converged}
    \State shuffle($N$)
    \For {$n \in N$}
      \State $b_n \leftarrow 1/(1 + e^{-\mu(i_n - \lambda)})$
      \State $c_n \leftarrow \arg\max_l \left( \delta_l \sum_{m \in \Gamma_n^l} b_m d_m + (1 - \delta_l) \sum_{m \in \Gamma_n^l \setminus \Gamma_n | s \in \Gamma_n} b_m \tilde{d}_m \right)$
      \State $d_n \leftarrow \sum_{m \in \Gamma_n^{c_n}} d_m / k_m^{c_n}$ and $\tilde{d}_n \leftarrow \sum_{m \in \Gamma_n^{c_n} | s \in \Gamma_n} \tilde{d}_m / \sum_{s \in \Gamma_m} k_s^{c_n}$
    \EndFor
    \For {$c \in C$}
      \State $\delta_c \leftarrow 1/|N^c| \sum_{m \in \Gamma_n | n \in \Gamma^c} \delta_{c_m} / k_n$
    \EndFor
  \EndWhile
\end{algorithmic}
Some properties of $MPA$:

- same algorithm for link-density and link-pattern communities,
- **no prior knowledge** is required (e.g., number of communities),
- algorithm uses only local information (straightforward parallelization),
- relatively simple to extend (e.g., prior knowledge),
- time complexity near $O(k|L|) = O(k^2|N|)$,
- relatively simple to implement,
- etc.
Outline

1. Motivation

2. Classical community detection
   - Label propagation algorithm
   - Balanced propagation algorithm
   - Defensive propagation

3. Generalized community detection
   - General propagation algorithm
   - Model-based propagation algorithm

4. Empirical evaluation
   - Synthetic networks
   - Real-world networks

5. Conclusions & future work
Experimental testbed:

- classical, fully link-pattern and generalized community detection,
- synthetic, real-world and random networks,
- predictive data clustering (see paper).

Adopted algorithms:

- **MPA** Model-based propagation algorithm
- **MPA(D)** MPA with $\delta_c = 1$ (only classical communities)
- **MPA(P)** MPA with $\delta_c = 0$ (only link-pattern communities)
- **GPA** General propagation algorithm [Šubelj & Bajec, 2011d]
- **MM(EM)** Mixture model with EM algorithm [Newman & Leicht, 2007]
- **MO(G)** Greedy modularity optimization [Clauset et al., 2004]

Quality measures:

$$NMI = \frac{2I(C,P)}{H(C) + H(P)} \quad \text{and} \quad NVOI = \frac{H(C|P) + H(P|C)}{\log |N|}$$
Synthetic networks (I)

Classical community detection—Lancichinetti et al. benchmark:

Erdős-Rényi random graphs, and resolution limit networks:
Synthetic networks (II)

Gen. community detection—generalized Girvan-Newman benchmark:

Generalized community detection—Šubelj-Bajec benchmark:
**Synthetic networks (III)**

Generalized community detection—hierarchical networks:

![Diagram of hierarchical network]

Community modeling strategy in *MPA*:

![Diagram of community modeling strategy]

![Graph showing community parameter over iterations]
# Real-world networks (I)

<table>
<thead>
<tr>
<th>Network</th>
<th>$N$</th>
<th>$L$</th>
<th>$C$</th>
<th>$MO(G)$</th>
<th>GPA</th>
<th>$MM(EM)$</th>
<th>MPA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Zachary's karate club</td>
<td>34</td>
<td>78</td>
<td>2</td>
<td>0.6925</td>
<td>0.7155</td>
<td>0.7870</td>
<td>0.8949</td>
</tr>
<tr>
<td>American college football</td>
<td>115</td>
<td>616</td>
<td>12</td>
<td>0.7547</td>
<td>0.8769</td>
<td>0.8049</td>
<td>0.8919</td>
</tr>
<tr>
<td>Davis's southern women</td>
<td>32</td>
<td>89</td>
<td>4</td>
<td>0.7338</td>
<td>0.8332</td>
<td></td>
<td>0.8084</td>
</tr>
<tr>
<td>Scottish corpor. interlocks</td>
<td>217</td>
<td>348</td>
<td>8</td>
<td>0.6634</td>
<td>0.5988</td>
<td></td>
<td>0.6411</td>
</tr>
</tbody>
</table>

Table: Analysis subject to NMI
Real-world networks (II)

| Network                                      | |N| | |L| | |C| |MO(G)  | GPA    | MM(EM) | MPA    |
|----------------------------------------------|---|---|---|---|---|---|---|---|---|---|---|---|
| Java (org namespace)                        | 709| 3571| 47| 0.5029| **0.5190**  | -  | **0.5187**  |
| Java (javax namespace)                      | 1595| 5287| 107| 0.7048| **0.7369**  | -  | **0.7386**  |

Table: Analysis subject to NMI

26) javax adj. matrix
27) javax blockmodel
28) javax communities (GPA)

(javax.swing, javax.management, javax.xml, javax.print, javax.naming, javax.lang...)

Šubelj & Bajec (University of Ljubljana) Generalized network community detection NEMO '11 21 / 25
Outline

1 Motivation

2 Classical community detection
   - Label propagation algorithm
   - Balanced propagation algorithm
   - Defensive propagation

3 Generalized community detection
   - General propagation algorithm
   - Model-based propagation algorithm

4 Empirical evaluation
   - Synthetic networks
   - Real-world networks

5 Conclusions & future work
Conclusions:

- algorithm for detection of arbitrary network modules,
- community modeling strategy based on network clustering,
- requires no prior knowledge about the true structure,
- comparable to current state-of-the-art.

Properties of real-world networks can be even further utilized within the algorithm (i.e., community model)!
Future work

Open questions:

- Where and why do link-pattern communities emerge?
- How do different types of communities link between each other?
- How do link-pattern communities coincide with known properties of real-world networks?
Thank you.

lovro.subelj@fri.uni-lj.si
http://lovro.lpt.fri.uni-lj.si/