Unfolding network communities by combining defensive and offensive label propagation

Lovro Šubelj and Marko Bajec

Faculty of Computer and Information Science, University of Ljubljana

September 20, 2010

1Workshop on the Analysis of Complex Networks (ACNE ’10)
Outline

1. Network communities

2. Detecting communities by label propagation
   - Label propagation algorithm
   - Issues with label propagation
   - Label hop attenuation

3. Defensive & offensive label propagation
   - Defensive preservation & offensive expansion
   - Combining the two strategies

4. Empirical evaluation

5. Conclusion
Network communities

- Intuitively, *communities* (or *modules*) are cohesive groups of nodes densely connected within, and only loosely connected between.
- Formally, e.g., notions of *weak* and *strong communities* [39], etc.

Play an important role in many real-world systems [15, 37].
Outline

1 Network communities

2 Detecting communities by label propagation
   - Label propagation algorithm
   - Issues with label propagation
   - Label hop attenuation

3 Defensive & offensive label propagation
   - Defensive preservation & offensive expansion
   - Combining the two strategies

4 Empirical evaluation

5 Conclusion
Label propagation algorithm

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

Label propagation algorithm [40] (LPA):

1. initialize nodes with unique labels, i.e., $\forall n \in N : c_n = l_n$,
2. set each node’s label to the label shared by most of its neighbors\(^2\), i.e., $\forall n \in N : c_n = \arg\max_l \sum_{m \in N_n} w_{nm}$,
3. if not converged, continue to 2.

Near linear time complexity [40, 28, 46].

\(^2\)Nodes are updated sequentially. Ties are broken uniformly at random.
Issues with label propagation

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

→ Nodes are updated sequentially (asynchronous), in a random order [40].

Convergence issues for, e.g., overlapping communities.

→ Node’s label is retained, when among most frequent [40].
Label hop attenuation

Emergence of a *major community* (in large networks).

Label *hop attenuation* [28]: each label $l_n$ has associated a score $s_n$ (initialized to 1) that decreases by $\delta \in [0, 1]$ after each step. Then,

$$\forall n \in N : c_n = \arg\max_l \sum_{m \in \mathcal{N}_n^l} s_m w_{nm} \text{ and } s_n = \left( \max_{m \in \mathcal{N}_n^{c_n}} s_m \right) - \delta.$$ 

Actually, $s_n = 1 - \delta d_n$, where $d_n = \left( \min_{m \in \mathcal{N}_n^{c_n}} d_m \right) + 1$. 

Some issues not discussed (e.g., oscillation of labels [40], stability [47]).
Outline

1. Network communities

2. Detecting communities by label propagation
   - Label propagation algorithm
   - Issues with label propagation
   - Label hop attenuation

3. Defensive & offensive label propagation
   - Defensive preservation & offensive expansion
   - Combining the two strategies

4. Empirical evaluation

5. Conclusion
Node propagation preference

Applying *node preference* [28] (i.e., propagation strength) can improve the algorithm. Thus,

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

for some preference $f_n$ and parameter $\alpha$.

(c) Zachary’s karate club [50]

However, static measures for $f_n$ do not work in general (see paper).
**dDaLPA & oDaLPA algorithms**

Estimate *diffusion* within (current) communities, i.e.,

\[ p_n = \sum_{m \in \mathcal{N}_n^{c_n}} \frac{p_m}{\text{deg}_m^{c_n}}, \]

using a random walker.

Apply preference to:
- the *core* of each (current) community, i.e.,
  \[ f_n^\alpha = p_n, \]
- the *border* of each (current) community, i.e.,
  \[ f_n^\alpha = 1 - p_n. \]

We get *defensive and offensive diffusion and label propagation algorithm* (dDaLPA and oDaLPA respectively.)
\textbf{dDaLPA \& oDaLPA algorithms, cont.}

\textbf{Algorithm (dDaLPA)}

\begin{algorithmic}
\STATE \{\textit{Initialization.}\}
\WHILE{\textbf{not} converged}
\STATE shuffle($N$)
\FOR{$n \in N$}
\STATE $c_n \leftarrow \text{argmax}_l \sum_{m \in N_n^l} p_m (1 - \delta d_m) w_{nm}$ \{1 - $p_m$ for oDaLPA.\}
\STATE $p_n \leftarrow \sum_{m \in N_n^{c_n}} p_m / \text{deg}_m^{c_n}$ \{deg$_m$ for oDaLPA.\}
\IF{$c_n$ has changed}
\STATE $d_n \leftarrow (\min_{m \in N_n^{c_n}} d_m) + 1$
\ENDIF
\ENDFOR
\{\textit{Re-estimation of $\delta$ (see paper).}\}
\ENDWHILE
\end{algorithmic}
Defensive & offensive label propagation

Defensive preservation & offensive expansion

- *dDaLPA defensively preserves* the communities – high “recall”.
- *oDaLPA offensively expands* the communities – high “precision”.

(d) American college football league [14]. (e) Nematode Caenorhabditis elegans [21].
Combining the two strategies

Find initial communities with $dDaLPA$, and refine them with $oDaLPA$ – high “recall” and “precision”.

However, simply running the algorithms successively does not work. Thus, relabel some of the nodes, e.g., a half.

We get $K$-Cores algorithm.
$K$-Cores algorithm

Algorithm (K-Cores)

\[
C \leftarrow dD\text{a}LPA(G, W) \quad \{\text{Defensive propagation.}\}
\]

while $|C|$ decreases do

for $c \in C$ do

\[
m_c \leftarrow \text{median}\{p_n \mid n \in N \land c_n = c\}
\]

\[
\{\text{Relabel nodes with } c_n = c \text{ and } p_n \leq m_c \text{ (i.e. retain cores).}\}
\]

end for

\[
C \leftarrow oD\text{a}LPA(G, W) \quad \{\text{Offensive propagation.}\}
\]

end while
Outline

1 Network communities

2 Detecting communities by label propagation
   - Label propagation algorithm
   - Issues with label propagation
   - Label hop attenuation

3 Defensive & offensive label propagation
   - Defensive preservation & offensive expansion
   - Combining the two strategies

4 Empirical evaluation

5 Conclusion
Experimental testbed

Experimental testbed:

- Lancichinetti et al. [22] benchmark networks (see paper),
- random graph à la Erdös-Rényi [10] (see paper),
- 22 real-world networks (moderate size),
- 9 large real-world networks (over $10^6$ edges).

Results are assessed in terms of *modularity* $Q$, i.e.,

$$Q = \frac{1}{2|E|} \sum_{n,m \in N} \left( A_{nm} - \frac{\deg_n \deg_m}{2|E|} \right) \delta(c_n, c_m).$$

and *Normalized Mutual Information*, i.e.,

$$NMI = \frac{2 I(C, P)}{H(C) + H(P)}, \text{ where } I(C, P) = H(C) - H(C|P).$$
Empirical evaluation

Lancichinetti et al. benchmark

(a) Lancichinetti et al. benchmark \( n = 1000, C = [10,50] \)

(b) Lancichinetti et al. benchmark \( n = 1000, C = [20,100] \)

(c) Lancichinetti et al. benchmark \( n = 5000, C = [10,50] \)

(d) Lancichinetti et al. benchmark \( n = 5000, C = [20,100] \)
Empirical evaluation

Erdős-Rényi random graph

Lovro Šubelj (University of Ljubljana)
## Real-world networks

<table>
<thead>
<tr>
<th>Type</th>
<th>Network</th>
<th>Nodes</th>
<th>Edges</th>
<th>LPA</th>
<th>dDaLPA</th>
<th>oDaLPA</th>
<th>K-Cores</th>
</tr>
</thead>
<tbody>
<tr>
<td>Communication</td>
<td>uni</td>
<td>1133</td>
<td>5451</td>
<td>0.364</td>
<td>0.481</td>
<td>0.389</td>
<td>0.518</td>
</tr>
<tr>
<td></td>
<td>enron</td>
<td>36692</td>
<td>36762</td>
<td>0.355</td>
<td>0.514</td>
<td>0.380</td>
<td>0.516</td>
</tr>
<tr>
<td></td>
<td>football</td>
<td>115</td>
<td>616</td>
<td>0.592</td>
<td>0.593</td>
<td>0.595</td>
<td>0.600</td>
</tr>
<tr>
<td></td>
<td>jazz</td>
<td>198</td>
<td>2742</td>
<td>0.346</td>
<td>0.418</td>
<td>0.377</td>
<td>0.418</td>
</tr>
<tr>
<td></td>
<td>wiki</td>
<td>7115</td>
<td>103689</td>
<td>0.056</td>
<td>0.195</td>
<td>0.046</td>
<td>0.202</td>
</tr>
<tr>
<td></td>
<td>epinions</td>
<td>75879</td>
<td>508837</td>
<td>0.106</td>
<td>0.288</td>
<td>0.111</td>
<td>0.291</td>
</tr>
<tr>
<td>Social</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>enron</td>
<td>36692</td>
<td>36762</td>
<td>0.355</td>
<td>0.514</td>
<td>0.380</td>
<td>0.516</td>
</tr>
<tr>
<td></td>
<td>football</td>
<td>115</td>
<td>616</td>
<td>0.592</td>
<td>0.593</td>
<td>0.595</td>
<td>0.600</td>
</tr>
<tr>
<td></td>
<td>jazz</td>
<td>198</td>
<td>2742</td>
<td>0.346</td>
<td>0.418</td>
<td>0.377</td>
<td>0.418</td>
</tr>
<tr>
<td></td>
<td>wiki</td>
<td>7115</td>
<td>103689</td>
<td>0.056</td>
<td>0.195</td>
<td>0.046</td>
<td>0.202</td>
</tr>
<tr>
<td></td>
<td>epinions</td>
<td>75879</td>
<td>508837</td>
<td>0.106</td>
<td>0.288</td>
<td>0.111</td>
<td>0.291</td>
</tr>
<tr>
<td>Protein</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>yeast</td>
<td>2114</td>
<td>4480</td>
<td>0.665</td>
<td>0.733</td>
<td>0.720</td>
<td>0.793</td>
</tr>
<tr>
<td>Metabolic</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>elegans</td>
<td>453</td>
<td>2025</td>
<td>0.122</td>
<td>0.172</td>
<td>0.131</td>
<td>0.173</td>
</tr>
<tr>
<td>Peer-to-peer</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>gnutella</td>
<td>62586</td>
<td>147892</td>
<td>0.338</td>
<td>0.412</td>
<td>0.387</td>
<td>0.447</td>
</tr>
<tr>
<td>Web</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>blogs</td>
<td>1490</td>
<td>16718</td>
<td>0.400</td>
<td>0.424</td>
<td>0.424</td>
<td>0.426</td>
</tr>
<tr>
<td></td>
<td>genrelat</td>
<td>5242</td>
<td>28980</td>
<td>0.737</td>
<td>0.769</td>
<td>0.779</td>
<td>0.820</td>
</tr>
<tr>
<td></td>
<td>codmat</td>
<td>27519</td>
<td>11618</td>
<td>0.596</td>
<td>0.611</td>
<td>0.627</td>
<td>0.687</td>
</tr>
<tr>
<td>Collaboration</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>codmat</td>
<td>36458</td>
<td>17136</td>
<td>0.548</td>
<td>0.575</td>
<td>0.590</td>
<td>0.648</td>
</tr>
<tr>
<td></td>
<td>hep</td>
<td>12008</td>
<td>237010</td>
<td>0.484</td>
<td>0.585</td>
<td>0.518</td>
<td>0.585</td>
</tr>
<tr>
<td></td>
<td>astro</td>
<td>18772</td>
<td>396160</td>
<td>0.326</td>
<td>0.538</td>
<td>0.337</td>
<td>0.538</td>
</tr>
<tr>
<td>Software</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>engine</td>
<td>139</td>
<td>243</td>
<td>0.689</td>
<td>0.724</td>
<td>0.726</td>
<td>0.747</td>
</tr>
<tr>
<td></td>
<td>jung</td>
<td>436</td>
<td>1303</td>
<td>0.611</td>
<td>0.587</td>
<td>0.623</td>
<td>0.631</td>
</tr>
<tr>
<td></td>
<td>javax</td>
<td>2089</td>
<td>7934</td>
<td>0.723</td>
<td>0.687</td>
<td>0.725</td>
<td>0.768</td>
</tr>
<tr>
<td>Power</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>power</td>
<td>4941</td>
<td>6594</td>
<td>0.595</td>
<td>0.690</td>
<td>0.698</td>
<td>0.820</td>
</tr>
<tr>
<td>Internet</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>oregon</td>
<td>767</td>
<td>3591</td>
<td>0.302</td>
<td>0.210</td>
<td>0.354</td>
<td>0.210</td>
</tr>
<tr>
<td></td>
<td>oregon</td>
<td>22963</td>
<td>48436</td>
<td>0.498</td>
<td>0.347</td>
<td>0.541</td>
<td>0.347</td>
</tr>
<tr>
<td></td>
<td>nec</td>
<td>75885</td>
<td>357317</td>
<td>0.683</td>
<td>0.628</td>
<td>0.688</td>
<td>0.736</td>
</tr>
</tbody>
</table>

**Tabela:** Mean modularities $Q$ (100 to 100000 runs).

Lovro Šubelj (University of Ljubljana)
Large real-world networks

\textbf{DPA} – faster alternative for \textit{K-Cores}.

\textbf{DPA}^+ – \textbf{DPA} with simple hierarchical investigation.

\textbf{DPA}^* – \textbf{DPA} with hierarchical \textit{core extraction} technique.

For more see [46].

<table>
<thead>
<tr>
<th>Network</th>
<th>Nodes</th>
<th>Edges</th>
<th>LPA</th>
<th>K-Cores</th>
<th>DPA</th>
<th>DPA^+</th>
<th>DPA^*</th>
</tr>
</thead>
<tbody>
<tr>
<td>amazon</td>
<td>0.3M</td>
<td>1.2M</td>
<td>0.681/15</td>
<td>0.783/273</td>
<td>0.700/34</td>
<td>0.883/65</td>
<td>0.856/78</td>
</tr>
<tr>
<td>ndedu</td>
<td>0.3M</td>
<td>1.5M</td>
<td>0.838/53</td>
<td>0.891/471</td>
<td>0.860/50</td>
<td>0.897/37</td>
<td>0.901/58</td>
</tr>
<tr>
<td>road</td>
<td>1.1M</td>
<td>3.1M</td>
<td>0.552/10</td>
<td>0.847/895</td>
<td>0.626/82</td>
<td>\textbf{0.985/136}</td>
<td>0.883/142</td>
</tr>
<tr>
<td>google</td>
<td>0.9M</td>
<td>4.3M</td>
<td>0.801/15</td>
<td>0.889/444</td>
<td>0.820/59</td>
<td>0.962/45</td>
<td>\textbf{0.967/48}</td>
</tr>
<tr>
<td>skitter</td>
<td>1.7M</td>
<td>11.1M</td>
<td>0.746/25</td>
<td>-</td>
<td>0.755/126</td>
<td>0.680/52</td>
<td>\textbf{0.801/76}</td>
</tr>
<tr>
<td>movie</td>
<td>0.4M</td>
<td>15.0M</td>
<td>0.524/21</td>
<td>-</td>
<td>0.533/147</td>
<td>0.474/39</td>
<td>\textbf{0.606/71}</td>
</tr>
<tr>
<td>nber</td>
<td>3.8M</td>
<td>16.5M</td>
<td>0.576/109</td>
<td>-</td>
<td>0.582/336</td>
<td>0.707/112</td>
<td>\textbf{0.739/308}</td>
</tr>
<tr>
<td>live</td>
<td>4.8M</td>
<td>69.0M</td>
<td>0.673/100</td>
<td>-</td>
<td>0.548/206</td>
<td>0.683/73</td>
<td>\textbf{0.688/125}</td>
</tr>
<tr>
<td>webbase</td>
<td>14.5M</td>
<td>101.0M</td>
<td>0.894/38</td>
<td>-</td>
<td>0.923/114</td>
<td>0.942/43</td>
<td>\textbf{0.954/39}</td>
</tr>
</tbody>
</table>

\textbf{Tabela:} Peak modularities \textit{Q} and \# iterations (1 to 10 runs).
Conclusion

- Different advanced label propagation algorithms.
- Two unique strategies of community formation – *different types of networks favor different formation strategies*.
- Extensions and improvements for large networks.

For more see [46].

For material see http://www.lovre.appspot.com/?navigation=research_main.
Thank you.


Leskovec, J., Kleinberg, J., Faloutsos, C.: Graph evolution: Densification and shrinking diameters. ACM Transactions on Knowledge Discovery from Data 1(1) (2007)


