EgoLP: Fast and Distributed Community Detection in Billion-node Social Networks

Nazar Buzun, Anton Korshunov, Valeriy Avanesov, Ilya Filonenko, Ilya Kozlov, Denis Turdakov, Hangkyu Kim

Institute for System Programming of RAS
Online social networks
Monthly active users [Kamber, 15 November 2013]
Online social networks

Facebook monthly users

<table>
<thead>
<tr>
<th>Year</th>
<th>Monthly Users</th>
</tr>
</thead>
<tbody>
<tr>
<td>2004</td>
<td>1m</td>
</tr>
<tr>
<td>2005</td>
<td>6m</td>
</tr>
<tr>
<td>2006</td>
<td>12m</td>
</tr>
<tr>
<td>2007</td>
<td>58m</td>
</tr>
<tr>
<td>2008</td>
<td>145m</td>
</tr>
<tr>
<td>2009</td>
<td>360m</td>
</tr>
<tr>
<td>2010</td>
<td>608m</td>
</tr>
<tr>
<td>2011</td>
<td>845m</td>
</tr>
<tr>
<td>2012</td>
<td>1,056m</td>
</tr>
<tr>
<td>2013</td>
<td>1,230m</td>
</tr>
</tbody>
</table>

Twitter monthly active users worldwide as of Q4 2013

(Source: Facebook)

http://gigaom.com/
Community [Leskovec et al, 2012]
– a group of social network users created on shared affiliation, role, activity, social circle, interest or function
User communities detection

Motivation

- Full list of communities is seldom disclosed and maintained by users
  - sexual/political groups
  - latent communities (geographical/organizational) not known to users
- Social graph contains enough data to uncover both explicit and latent community structure

Applications

- Attribute and link prediction
- Identifying spam communities
- Social recommendation
- Content categorization
- Enabling in-depth analysis of selected communities
Task definition

Input

Social network $G = (V, E)$
- social ties are essential for any network
- profile and content analysis is not necessary
- edges could be assigned weights (e.g., user-user similarity)

Output

Set of communities $C = \{Z_c\}_{c=1}^K$, $Z_c \subseteq V$
Cover

- a set of communities where any user could belong to $\geq 1$ communities
### Structural properties of communities

[Leskovec et al., 2012]

- **Separability**: communities are well-separated from the rest of the network.
- **Density**: communities are well connected.
- **Cohesiveness**: it should be relatively hard to split a community into two sub-communities.
Structural properties of communities [Leskovec et al, 2012]

- Dense community overlaps
- Number of edges increases superlinearly with the community size
- Power-law distribution of community size
- Power-law distribution of user-community memberships

<table>
<thead>
<tr>
<th>Dataset</th>
<th>N</th>
<th>E</th>
<th>C</th>
<th>S</th>
<th>A</th>
</tr>
</thead>
<tbody>
<tr>
<td>LiveJournal</td>
<td>4.0 M</td>
<td>34.9 M</td>
<td>310 k</td>
<td>40.06</td>
<td>3.09</td>
</tr>
<tr>
<td>Friendster</td>
<td>120 M</td>
<td>2,600 M</td>
<td>1.5 M</td>
<td>26.72</td>
<td>0.33</td>
</tr>
<tr>
<td>Orkut</td>
<td>3.1 M</td>
<td>120 M</td>
<td>8.5 M</td>
<td>34.86</td>
<td>95.93</td>
</tr>
<tr>
<td>Youtube</td>
<td>1.1 M</td>
<td>3.0 M</td>
<td>30 k</td>
<td>9.75</td>
<td>0.26</td>
</tr>
<tr>
<td>DBLP</td>
<td>0.43 M</td>
<td>1.3 M</td>
<td>2.5 k</td>
<td>429.79</td>
<td>2.57</td>
</tr>
<tr>
<td>Amazon</td>
<td>0.34 M</td>
<td>0.93 M</td>
<td>49 k</td>
<td>99.86</td>
<td>14.83</td>
</tr>
</tbody>
</table>

Data: Stanford Network Analysis Platform

- non-sampled social networks
- lists of members for public groups
- each connected component of a group is a community
Structural properties of communities
Dense community overlaps [Leskovec et al, 2012]
Requirements

1. High accuracy in discovering highly overlapping community structure
2. $O(|E|)$ complexity or better
3. Scalability:
   - near-linear by graph size and cluster size
   - ability to process graphs with $> 10^9$ users
Label propagation algorithm template

**Data:** Graph $(V, E)$

**Parameters:** $T$, $r$

**Result:** Cover $\{C_i\}$, $C_i \subseteq V$

```plaintext
for $v \in V$ do
  initialize $v$ with a community label
end

for $i = 1:T$ do
  for $v \in V$ do
    for $n \in \text{neighbours } v$ do
      SenderStrategy($v,n$)
    end
  end
  for $v \in V$ do
    ReceiverStrategy($v$)
  end
end

for $v \in V$ do
  remove all labels with frequency $< r$ from labels of $v$
end
call label sets to $\{C_i\}$
```
Speaker-listener Label Propagation Algorithm [Xie et al, 2011]
Advantages

1. Ability to discover (non-) overlapping community structure
2. Ability to discover both global communities and egomunities
3. High accuracy given small community overlaps
4. $O(T \cdot |E|)$ complexity
5. Ease of distributed implementation

Drawbacks

1. **Hubs effect**: labels of top-degree nodes propagate intensively and lead to huge communities
2. **Small number of communities for a node**: few communities dominate the others
3. Noise labels
4. Poor connectivity of communities
EgoLP

A distributed method for uncovering highly overlapping community structure based on distributed label propagation along graph edges

Main steps

1. Preprocessing:
   1. initialize each node with a unique community label
   2. remove hubs (top-degree nodes)

2. Egomunities retrieval

3. Label propagation

4. Communities post-processing

5. Converting label sets to user-community assignments
Egommunities retrieval

Input
Each user’s *ego-network*: a user + her friends + all links among them

Algorithm
SLPA($T, r, k$)

Output
Egommunities cover for each node (stored in the node’s memory)
Egomunities overlap with global communities

- Students at University X
- Employees of Business Y
- Residents of Neighborhood Z
Data: Graph \((V, E)\), egomunities \(\{E_{vk}\}\)

Parameters: \(T, ls, lx, lr, mx, T_2, r\)

Result: Cover \(\{C_i\}, C_i \subseteq V\)

\[
\text{for } i = 1:T \text{ do} \\
\quad \text{for } v \in V \text{ do} \\
\quad \\
\quad \quad \text{for } n \in \text{neighbours } v \text{ do} \\
\quad \quad \\
\quad \quad \quad \text{randomly pick } ls \text{ labels from } v\text{'s labels} \\
\quad \quad \quad \text{send the selected labels to } n \\
\quad \quad \text{end} \\
\quad \text{end} \\
\quad \text{for } v \in V \text{ do} \\
\quad \quad \mathcal{L}_{vi} := \text{labels received by } v \text{ from its neighbours} \\
\quad \quad \text{remove all labels from } \mathcal{L}_{vi} \text{ except for } lx \text{ most frequent} \\
\quad \quad \text{for } e_k \in E_{vk} \text{ do} \\
\quad \quad \\
\quad \quad \quad \mathcal{L}_{vik} := \text{labels from } \mathcal{L}_{vi} \text{ received by } v \text{ from neighbours from } e_k \\
\quad \quad \quad \text{pick } lr \text{ most frequent labels from } \mathcal{L}_{vik} \\
\quad \quad \quad \text{add the selected labels to } v\text{'s labels} \\
\quad \quad \text{end} \\
\quad \quad \text{remove all } v\text{'s labels except for } mx \text{ most frequent} \\
\quad \text{end} \\
\text{end}
**Data:** Graph \((V, E)\), egomunities \(\{E_{vk}\}\)

**Parameters:** \(T, ls, lx, lr, mx, T_2, r\)

**Result:** Cover \(\{C_i\}\), \(C_i \subseteq V\)

\[
V := V \cup \{hubs\};
\]

\[
\text{for } v \in V \text{ do}
\]

\[
\quad E_{vk} := \{\text{neighbours } v\};
\]

\[
\text{end}
\]

\[
\text{for } i = 1:T_2 \text{ do}
\]

\[
\quad \text{perform labels exchange exactly like in cycle with } T \text{ iterations}
\]

\[
\text{end}
\]

\[
\text{remove all labels with frequency } < r \text{ from labels of } v
\]

\[
\text{convert label sets to } \{C_i\}
\]
Data: Graph \((V, E)\), communities \(\{C_i\}\)

Parameters: \(cx, \lambda_x, k, T, r, minc\)

Result: Cover \(\{S_i\}\), \(\forall i \exists j : S_i \subseteq C_j\)

Remove communities with size \(> cx\);

for \(c \in \{C_i\}\) do
  find \(\lambda_{n-1}\) of Laplacian \(L_c = A_c - \text{diag}(\vec{d}_c)\)
  if \(\lambda_{n-1} < \lambda_x\) then
    \(\{C'_i\}\) := communities found by applying SLPA\((T, r, k)\) to \(c\)
    remove communities with size \(\leq minc\) from \(\{C'_i\}\)
    for \(c' \in \{C'_i\}\) do
      \(\{CC'_i\}\) := connected components of \(c'\)
      \(\{S_i\}\) := \(\{S_i\}\) \(\cup\) \(\{CC'_i\}\)
    end
  else
    \(\{S_i\}\) := \(\{S_i\}\) \(\cup\) connected components of \(c\)
  end
end
Identifying ground-truth covers
LFR benchmark [Lancichinetti et al, 2009]

\[ N = 120, \langle k \rangle = 20, k_{max} = 35, c_{min} = 12, c_{max} = 30, O_n = 10, O_m = 6 \]
Identifying ground-truth covers
LFR benchmark: NMI dependency on $O_m$

$$N = 2500, \ c_{min} = 15, \ c_{max} = 150, \ O_n = 0.5N,$$

$$\langle k \rangle = 15(1 + (O_m - 1)O_n/N), \ k_{max} = 2.5 \langle k \rangle$$
Identifying ground-truth covers
LFR benchmark: NMI dependency on $O_m$

$N = 2500, c_{min} = 15, c_{max} = 150, O_n = 0.8N,$
$\langle k \rangle = 15(1 + (O_m - 1)O_n/N), k_{max} = 2.5\langle k \rangle$
Identifying ground-truth covers
LFR benchmark: NMI dependency on $O_n$

$N = 2500$, $c_{min} = 15$, $c_{max} = 150$, $O_m = 4$, 
$\langle k \rangle = 15(1 + (O_m - 1)O_n/N)$, $k_{max} = 2.5\langle k \rangle$
Identifying ground-truth covers
CKB benchmark: NMI dependency on average membership [Chykhradze et al., 2014]

\[ N = 5000, \ c_{min} = 20, \ c_{max} = 300, \ x_{min} = 1, \ x_{max} = 300, \]
\[ \langle k \rangle = 100, \ \beta_{\text{communities}} = 2.5, \ \beta_{\text{memberships}} = 2.5 \]
Latent user attribute prediction

Dataset

Facebook100 dataset [Traud et al, 2011]
- comes directly from Facebook and is not sampled
- includes 100 college networks
- smallest network: 769 nodes and $17 \times 10^3$ edges
- largest network: $36 \times 10^3$ nodes and $1.6 \times 10^6$ edges
- includes node attribute information:
  - gender
  - year of graduation
  - dormitory
  - academic major
  - high school

Attributes associated with community structure
- year of graduation
- dormitory
Latent user attribute prediction
Evaluation of the discovered communities [Lee et al, 2013]
### Latent user attribute prediction

#### Accuracy

<table>
<thead>
<tr>
<th>Network/Attribute</th>
<th>EgoLP</th>
<th>GCE</th>
<th>OSLOM</th>
<th>MOSES</th>
<th>SLPA</th>
</tr>
</thead>
<tbody>
<tr>
<td>UChicago/year of graduation</td>
<td>0.64</td>
<td>0.56</td>
<td>0.62</td>
<td>0.72</td>
<td>0.56</td>
</tr>
<tr>
<td>UChicago/dormitory</td>
<td>0.57</td>
<td>0.54</td>
<td>0.55</td>
<td>0.66</td>
<td>0.41</td>
</tr>
<tr>
<td>Caltech/year of graduation</td>
<td>0.51</td>
<td>0.44</td>
<td>0.42</td>
<td>0.70</td>
<td>0.38</td>
</tr>
<tr>
<td>Caltech/dormitory</td>
<td>0.79</td>
<td>0.85</td>
<td>0.83</td>
<td>0.78</td>
<td>0.70</td>
</tr>
<tr>
<td>Wellesley/year of graduation</td>
<td>0.71</td>
<td>0.75</td>
<td>0.75</td>
<td>0.86</td>
<td>0.75</td>
</tr>
<tr>
<td>Princeton/year of graduation</td>
<td>0.82</td>
<td>0.77</td>
<td>0.78</td>
<td>0.88</td>
<td>0.68</td>
</tr>
<tr>
<td>Lehigh/year of graduation</td>
<td>0.74</td>
<td>0.71</td>
<td>0.71</td>
<td>0.87</td>
<td>0.70</td>
</tr>
<tr>
<td>Cal65/year of graduation</td>
<td>0.71</td>
<td>0.60</td>
<td>0.63</td>
<td>0.70</td>
<td>0.68</td>
</tr>
<tr>
<td><strong>AVERAGE</strong></td>
<td>0.69</td>
<td>0.65</td>
<td>0.66</td>
<td>0.77</td>
<td>0.61</td>
</tr>
</tbody>
</table>
Community structure likelihood

Facebook100 dataset

<table>
<thead>
<tr>
<th>Network</th>
<th>EgoLP</th>
<th>SLPA</th>
<th>GCE</th>
<th>OSLOM</th>
<th>MOSES</th>
</tr>
</thead>
<tbody>
<tr>
<td>Caltech</td>
<td>-113</td>
<td>-120</td>
<td>-119</td>
<td>-116</td>
<td>-106</td>
</tr>
<tr>
<td>Cal65</td>
<td>-206</td>
<td>-207</td>
<td>-225</td>
<td>-217</td>
<td>-177</td>
</tr>
<tr>
<td>Lehigh</td>
<td>-233</td>
<td>-233</td>
<td>-259</td>
<td>-257</td>
<td>-207</td>
</tr>
<tr>
<td>Princeton</td>
<td>-273</td>
<td>-315</td>
<td>-311</td>
<td>-318</td>
<td>-248</td>
</tr>
<tr>
<td>UChicago</td>
<td>-193</td>
<td>-211</td>
<td>-221</td>
<td>-214</td>
<td>-174</td>
</tr>
<tr>
<td>Wellesley</td>
<td>-194</td>
<td>-207</td>
<td>-218</td>
<td>-209</td>
<td>-176</td>
</tr>
</tbody>
</table>
Distributed implementation

Apache Spark + Bagel

- Nodes are distributed over the cluster nodes in \textit{RDD} datasets
- Undirected edges are treated as bi-directional
- At each iteration labels are sent between nodes as messages (BSP)
- Nodes accumulate labels in the memory
- Labels of each node are post-processed to obtain the cover
- Loosely connected communities are separated
Scalability by graph size

- **ego** – egomunities retrieval
- **lp** – label propagation
- **post** – communities post-processing
- **sum** – total running time
Scalability by Spark workers count

- **ego** – egomunities retrieval
- **lp** – label propagation
- **sum** – total performance
Running time: comparison with other methods

![Graph showing running time comparison with other methods]
Questions?
Identifying ground-truth covers
Cover comparison [Lancichinetti et al, 2009]

Normalized Mutual Information (NMI) of covers $C'$ and $C''$

\[
NMI(X : Y) = 1 - \frac{1}{2} \left[ H(X|Y)_{\text{norm}} + H(Y|X)_{\text{norm}} \right]
\]

\[
H(X|Y)_{\text{norm}} = \frac{1}{|C'|} \sum_{k} \frac{H(X_k|Y)}{H(X_k)}
\]

\[
H(X_k|Y) = \min_{l \in \{1, 2, \ldots, |C''|\}} H(X_k|Y_l)
\]

\[
H(X_k|Y_l) = H(X_k, Y_l) - H(Y_l)
\]

\[
P(X_k = 1) = \frac{|C'_k|}{|V|}
\]

\[
P(Y_l = 1) = \frac{|C''_l|}{|V|}
\]

$X_k, Y_l$ – random variables associated with communities $C'_k$ and $C''_l$