Overlapping Community Detection
Using Seed Set Expansion

Joyce Jiyoung Whang\textsuperscript{1}  David F. Gleich\textsuperscript{2}  Inderjit S. Dhillon\textsuperscript{1}

\textsuperscript{1}The University of Texas at Austin
\textsuperscript{2}Purdue University

International Conference on Information and Knowledge Management
Contents

- Introduction
  - Overlapping Communities in Real-world Networks
  - Measures of Cluster Quality
  - Graph Clustering and Weighted Kernel $k$-Means

- The Proposed Algorithm
  - Filtering Phase
  - Seeding Phase
  - Seed Set Expansion Phase
  - Propagation Phase

- Experimental Results
  - Conductance
  - Ground-truth Accuracy
  - Runtime

- Conclusions
Overlapping Communities

- Community (cluster) in a graph $G = (\mathcal{V}, \mathcal{E})$
  - Set of cohesive vertices
  - Communities naturally overlap (e.g. social circles)
- Graph Clustering (Partitioning)
  - $k$ disjoint clusters $C_1, \cdots, C_k$ such that $\mathcal{V} = C_1 \cup \cdots \cup C_k$
- Overlapping Community Detection
  - $k$ overlapping clusters such that $C_1 \cup \cdots \cup C_k \subseteq \mathcal{V}$
### Real-world Networks

- **Collaboration networks:** co-authorship
- **Social networks:** friendship
- **Product network:** co-purchasing information

<table>
<thead>
<tr>
<th>Graph</th>
<th>No. of vertices</th>
<th>No. of edges</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Collaboration networks</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HepPh</td>
<td>11,204</td>
<td>117,619</td>
</tr>
<tr>
<td>AstroPh</td>
<td>17,903</td>
<td>196,972</td>
</tr>
<tr>
<td>CondMat</td>
<td>21,363</td>
<td>91,286</td>
</tr>
<tr>
<td>DBLP</td>
<td>317,080</td>
<td>1,049,866</td>
</tr>
<tr>
<td><strong>Social networks</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Flickr</td>
<td>1,994,422</td>
<td>21,445,057</td>
</tr>
<tr>
<td>Myspace</td>
<td>2,086,141</td>
<td>45,459,079</td>
</tr>
<tr>
<td>LiveJournal</td>
<td>1,757,326</td>
<td>42,183,338</td>
</tr>
<tr>
<td><strong>Product network</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Amazon</td>
<td>334,863</td>
<td>925,872</td>
</tr>
</tbody>
</table>
Measures of cluster quality

- Normalized Cut of a cluster

\[ \text{ncut}(C_i) = \frac{\text{links}(C_i, V \setminus C_i)}{\text{links}(C_i, V)} . \]

- Conductance

\[ \text{conductance}(C_i) = \frac{\text{links}(C_i, V \setminus C_i)}{\min \left( \text{links}(C_i, V), \text{links}(V \setminus C_i, V) \right)} . \]

\[ \text{links}(C_1, V \setminus C_1) = 2, \text{links}(C_1, V) = 10, \text{links}(V \setminus C_1, V) = 9 \]
A general weighted kernel \( k \)-means objective is equivalent to a weighted graph clustering objective (Dhillon et al. 2007).

**Objective**

\[
J = \sum_{c=1}^{k} \sum_{x_i \in \pi_c} w_i \| \varphi(x_i) - m_c \|^2, \text{ where } m_c = \frac{\sum_{x_i \in \pi_c} w_i \varphi(x_i)}{\sum_{x_i \in \pi_c} w_i}.
\]

**Distance between a vertex \( v \in C_i \) and cluster \( C_i \)**

\[
\text{dist}(v, C_i) = -\frac{2 \text{links}(v, C_i)}{\deg(v) \deg(C_i)} + \frac{\text{links}(C_i, C_i)}{\deg(C_i)^2} + \frac{\sigma}{\deg(v)} - \frac{\sigma}{\deg(C_i)}
\]
The Proposed Algorithm
Proposed Algorithm

- **Seed Set Expansion**
  - Carefully select seeds
  - Greedily expand communities around the seed sets

- The algorithm
  - Filtering Phase
  - Seeding Phase
  - Seed Set Expansion Phase
  - Propagation Phase

![Diagram showing the phases of the proposed algorithm](image-url)
Filtering Phase

Original Graph

Biconnected Core

Filtering
Filtering Phase

- Remove unimportant regions of the graph
  - Trivially separable from the rest of the graph
  - Do not participate in overlapping clustering

- Our filtering procedure
  - Remove all single-edge biconnected components (remain connected after removing any vertex and its adjacent edges)
  - Compute the largest connected component (LCC)
Filtering Phase
Filtering Phase

Joyce Jiyoung Whang, The University of Texas at Austin
Filtering Phase

Biconnected Core

Bridges

Whiskers
### Filtering Phase

The biconnected core – substantial portion of the edges

Detached graph – likely to be disconnected

Whiskers – separable from each other, no significant size

<table>
<thead>
<tr>
<th></th>
<th>Biconnected core</th>
<th>Detached graph</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>No. of vertices (%)</td>
<td>No. of edges (%)</td>
</tr>
<tr>
<td>HepPh</td>
<td>9,945 (88.8%)</td>
<td>116,099 (98.7%)</td>
</tr>
<tr>
<td>AstroPh</td>
<td>16,829 (94.0%)</td>
<td>195,835 (99.4%)</td>
</tr>
<tr>
<td>CondMat</td>
<td>19,378 (90.7%)</td>
<td>89,128 (97.6%)</td>
</tr>
<tr>
<td>DBLP</td>
<td>264,341 (83.4%)</td>
<td>991,125 (94.4%)</td>
</tr>
<tr>
<td>Flickr</td>
<td>954,672 (47.9%)</td>
<td>20,390,649 (95.1%)</td>
</tr>
<tr>
<td>Myspace</td>
<td>1,724,184 (82.7%)</td>
<td>45,096,696 (99.2%)</td>
</tr>
<tr>
<td>LiveJournal</td>
<td>1,650,851 (93.9%)</td>
<td>42,071,541 (99.7%)</td>
</tr>
<tr>
<td>Amazon</td>
<td>291,449 (87.0%)</td>
<td>862,836 (93.2%)</td>
</tr>
</tbody>
</table>
Seeding Phase

Biconnected Core

Graph with Seeds

Seeding
Seeding Phase

- Graclus centers
  - Graclus: a high quality and efficient graph partitioning scheme

**Algorithm 1** Seeding by Graclus Centers

**Input:** graph $G$, the number of seeds $k$.

**Output:** the seed set $S$.

1. Compute exhaustive and non-overlapping clusters $C_i$ ($i=1, \ldots, k$) on $G$.
2. Initialize $S = \emptyset$.
3. for each cluster $C_i$ do
4.     for each vertex $v \in C_i$ do
5.         Compute $\text{dist}(v, C_i)$.
6.     end for
7.     $S = \{\text{argmin}_v \text{dist}(v, C_i)\} \cup S$.
8. end for

Find the most central vertex in cluster $C_i$
Seeding Phase
Seeding Phase
Algorithm 1 Seeding by Spread Hubs

Input: graph $G = (V, E)$, the number of seeds $k$.
Output: the seed set $S$.

1: Initialize $S = \emptyset$.
2: All vertices in $V$ are unmarked.
3: while $|S| < k$ do
4: Let $T$ be the set of unmarked vertices with max degree.
5: for each $t \in T$ do
6: if $t$ is unmarked then
7: $S = \{t\} \cup S$.
8: Mark $t$ and its neighbors.
9: end if
10: end for
11: end while
Seeding Phase
Seeding Phase
Seeding Phase

- Other seeding strategies
  - **Local Optimal Egonets.** (Gleich and Seshadhri 2012)
    - ego(s): the egonet of vertex s.
    - Select a seed s such that
      \[
      \text{conductance}(\text{ego}(s)) \leq \text{conductance}(\text{ego}(v))
      \]
      for all v adjacent to s.

  - **Random Seeds.** (Andersen and Lang 2006)
    - Randomly select k seeds.
Seed Set Expansion Phase

Graph with Seeds

Expanded Clusters

Expansion
Seed Set Expansion Phase

- **Personalized PageRank clustering scheme** (Andersen et al. 2006)
  1. Given a seed node, compute an approximation of the stationary distribution of a random walk.
  2. Divide the stationary distribution scores by the degree of each node (technical detail needed to remove bias towards high-degree nodes).
  3. Sort the vector, and examine nodes in order of highest to lowest score and compute the conductance score for each threshold cut.

- Returns a good conductance cluster
- Remarkably efficient when combined with appropriate data structures
- For each seed, we use the entire vertex neighborhood as the restart for the personalized PageRank routine.
Seed Set Expansion Phase

Joyce Jiyoung Whang, The University of Texas at Austin
Conference on Information and Knowledge Management (27/44)
Propagation Phase

Expanded Clusters

Final Communities

Propagation
Propagation Phase

- Each community is further expanded.
- Add whiskers to communities via bridge.

**Algorithm 2** Propagation Module

**Input:** graph $G = (\mathcal{V}, \mathcal{E})$, biconnected core $G_C = (\mathcal{V}_C, \mathcal{E}_C)$, communities of $G_C : C_i$ ($i = 1, ..., k$) $\in C$.

**Output:** communities of $G$.

1: for each $C_i \in C$ do
2:   Detect bridges $\mathcal{E}_{B_i}$ attached to $C_i$.
3:   for each $b_j \in \mathcal{E}_{B_i}$ do
4:     Detect the whisker $w_j = (\mathcal{V}_j, \mathcal{E}_j)$ which is attached to $b_j$.
5:     $C_i = C_i \cup \mathcal{V}_j$.
6:   end for
7: end for
Propagation Phase

Joyce Jiyoung Whang, The University of Texas at Austin
This process does not increase the cut of each cluster.

Normalized cut of the expanded cluster is always smaller than equal to that of original cluster.
Experimental Results
Experiments

- Comparison with other state-of-the-art methods
  - **Demon** (Coscia et al. 2012)
    - Extracts and computes clustering of ego networks
  - **Bigclam** (Yang and Leskovec 2013)
    - Low-rank non-negative matrix factorization based modeling

- Seed set expansion methods with different seeding strategies
  - **Graclus centers**
  - **Spread hubs**
  - **Local Optimal Egonets** (Gleich and Seshadhri 2012)
Community Quality using Conductance

- arXiv CondMat collaboration network (21,363 nodes)

![Graph showing Maximum Conductance vs Coverage (percentage) for different methods: egonet, graclus centers, spread hubs, random, demon, bigclam.](image)
Community Quality using Conductance

- Flickr (1,994,422 nodes)
  - Demon fails on Flickr.

![Graph showing comparison of different algorithms like egonet, graclus centers, spread hubs, random, and bigclam in terms of coverage and maximum conductance.](image)
Community Quality using Conductance

- LiveJournal (1,757,326 nodes)
- Demon fails on LiveJournal.

![Graph showing maximum conductance vs coverage for different methods: egonet, graclus centers, spread hubs, random, and bigclam.](image-url)
Community Quality using Conductance

- Myspace (2,086,141 nodes)
  - Demon fails on Myspace.
  - Bigclam does not finish after running for one week.
Community Quality via Ground Truth

- **Precision**
  - how many vertices are actually in the same ground truth community

- **Recall**
  - how many vertices are predicted to be in the same community in a retrieved community

- **Compute** $F_1$, and $F_2$ measures
  - The ground truth communities are partially annotated.
  - $F_2$ measure puts more emphasis on recall than precision
Community Quality via Ground Truth

![Bar chart showing the community quality via ground truth for different algorithms in the DBLP dataset.](chart.png)

- **DBLP**
- Demon
- BigClam
- Graclus Centers
- Spread Hubs
- Random
- Egonet

Joyce Jiyoung Whang, The University of Texas at Austin
Comparison of Running Times

![Comparison of Running Times](image)

Joyce Jiyoung Whang, The University of Texas at Austin

Conference on Information and Knowledge Management (41/44)
Conclusions
Conclusions

- Efficient overlapping community detection algorithm
  - Uses a seed set expansion
- Two seed finding strategies
  - Graclus centers
  - Spread hubs
- Our new seeding strategies are better than other strategies, and are thus effective in finding good overlapping clusters in a graph.
- The seed set expansion approach significantly outperforms other state-of-the-art methods.
References