A short introduction to software LocalGraphClustering

Kimon Fountoulakis, M. Liu, D. Gleich and M. Mahoney

PASC 2019
Classification of clustering algorithms

Global Algorithm (Fiedler vector)  Locally biased algorithm

Strongly local algorithm

What is Local Graph Clustering?

Given a seed node, our goal is to find a target cluster that is around the seed node.

What is a target cluster?

- There are numerous theoretical perspectives, two popular ones are:
  
  - Theoretical Computer Science (worst-case): find a target cluster with low external connectivity and large volume.
  
  - Statistical (average): find a target cluster that has been generated by a random model.

- All algorithms in our API are principled algorithms, meaning that they come with worst-case or average clustering guarantees.

D. Wang, K. Fountoulakis, M. Henzinger, M. Mahoney, S. Rao, Capacity Releasing Diffusion, ICML, 2017
W. Ha, K. Fountoulakis, M. Mahoney, Statistical Guarantees for Local Graph Clustering, ArXiv, 2019
What is the LocalGraphClustering API?

- A collection of principled local graph clustering algorithms.
- Pipelines based on local graph clustering, e.g., Network Community Profiles, Improve Plots, Graph Partitioning, Higher-Order Network Analysis, Multi-label Classification, Vertex Neighborhood Metrics, Visualization Tools.
- C++ code with Python 3 interface, Linux and OS X (no Windows, any help?)
Our objective for LocalGraphClustering

• Easy to use interface

• Scalable - compute clusters in parallel (shared memory) using local graph clustering methods.

• Build-up pipelines based on local graph clustering
Where to find LocalGraphClustering?

• > pip install localgraphclustering

• > clone from github.com/kfoynt/LocalGraphClustering
We measure cluster quality using

\[
\text{Conductance} := \frac{\text{number of edges leaving cluster}}{\text{sum of degrees of vertices in cluster}}
\]

- Conductance({A,B}) = 2/(2 + 2) = 1/2
- Conductance({A,B,C}) = 1/(2 + 2 + 3) = 1/7

• The smaller the conductance value the better
We measure cluster quality using

\[
\text{Conductance} := \frac{\text{number of edges leaving cluster}}{\text{sum of degrees of vertices in cluster}}
\]

For the set \{A, B\}:

\[
\text{Conductance}(\{A, B\}) = \frac{2}{2 + 2} = \frac{1}{2}
\]

For the set \{A, B, C\}:

\[
\text{Conductance}(\{A, B, C\}) = \frac{1}{2 + 2 + 3} = \frac{1}{7}
\]

The smaller the conductance value the better.
Conductance({A,B}) = \frac{2}{2 + 2} = \frac{1}{2}

We measure cluster quality using

\[ \text{Conductance} := \frac{\text{number of edges leaving cluster}}{\text{sum of degrees of vertices in cluster}} \]

The smaller the conductance value the better

Conductance({A,B,C}) = \frac{1}{2 + 2 + 3} = \frac{1}{7}

Metrics: conductance
Metrics: conductance

We measure cluster quality using

Conductance := \frac{\text{number of edges leaving cluster}}{\text{sum of degrees of vertices in cluster}}

Conductance(\{A,B\}) = \frac{2}{(2 + 2)} = \frac{1}{2}

Conductance(\{A,B,C\}) = \frac{1}{(2 + 2 + 3)} = \frac{1}{7}

• The smaller the conductance value the better
We measure cluster quality using the **Conductance** metric:

$$\text{Conductance} := \frac{\text{number of edges leaving cluster}}{\text{sum of degrees of vertices in cluster}}$$

- Conductance({A,B}) = \( \frac{2}{2 + 2} = \frac{1}{2} \)
- Conductance({A,B,C}) = \( \frac{1}{2 + 2 + 3} = \frac{1}{7} \)

- The smaller the conductance value the better
ML/Statistics metrics for given “ground-truth”

Methods in LocalGraphClustering

- Local spectral methods
  - L1-regularized PageRank (Fountoulakis, Shun, Cheng, Khorasani, Mahoney, 2017)
  - ACL (Andersen, Chung, Lang, 2006)
  - MOV (Mahoney, Orecchia, Vishnoi, 2012)
  - Spectral MQI (Chung, 2007)

- Local flow methods
  - MQI (Lang and Rao, 2004)
  - FlowImprove (Andersen and Lang, 2008)
  - Local FlowImprove (Orecchia, Zhu, 2014)
  - SimpleLocal (Veldt, Gleich, Mahoney, 2015)
  - CRD (Wang, Fountoulakis, Henzinger, Mahoney, Rao, 2017)
Local spectral methods

**Global Spectral**

\[
\text{minimize} \quad \frac{\|Bx\|_2^2}{\|D^{1/2}x\|_2^2}
\]

subject to:
\[
\|x\|_2^2 = 1 \quad x^T De = 0
\]

- B: incidence matrix: two nnz per row, i.e., (1,-1) for each edge.
- D: diagonal degree matrix
- s: seed vector (input by the user that indicates preference of nodes)

**Local Spectral (MOV)**

\[
\text{minimize} \quad \frac{\|Bx\|_2^2}{\|D^{1/2}x\|_2^2}
\]

subject to:
\[
x^T De = 0 \quad x^T s \geq \kappa
\]

M. Mahoney, L. Orecchia, N. Vishnoi, A Local Spectral Method for Graphs, JMLR, 2012
L1-Regularized PageRank

\[
\text{minimize} \quad \frac{1 + \alpha}{4} \| D^{1/2} x \|_2^2 + \frac{1 - \alpha}{4} \| B x \|_2^2 - \alpha s^T x + \rho \alpha \| D x \|_1
\]

- The volume of nonzero nodes of the optimal solution is bounded by $1/\rho$.

- Randomized coordinate descent obtains an epsilon accurate solution $O(1/\rho)$ (independent of the size of the graph).

K. Fountoulakis, J. Shun, F. Khorasani, X. Cheng, M. Mahoney, Variational Perspective on Local Graph Clustering, Mathematical Programming, 2017
L1-Regularized PageRank

The same method can be applied to any graph. I chose images because they are easy to visualize.

K. Fountoulakis, J. Shun, F. Khorasani, X. Cheng, M. Mahoney, Variational Perspective on Local Graph Clustering, Mathematical Programming, 2017
FlowImprove Methods

• Give as input a cluster that has many false positives or few true positives. Then solve:

\[
\begin{align*}
\text{MQI} & \quad \text{minimize} \quad \frac{\|Bx\|_1}{x^T d} \\
\text{subject to:} \quad & x_i = 0 \quad \forall i \in R^c \\
\end{align*}
\]

\[
\begin{align*}
\text{FlowImprove} & \quad \text{minimize} \quad \frac{\|Bx\|_1}{x^T d_R - \delta x^T d_{R^c}} \\
\end{align*}
\]
Capacity Releasing Diffusion

\[
\text{maximize } \| f \|_\infty
\]

subject to: \( B^T f + s \leq d \)

- \( s \): seed vector (initial mass on nodes)
- \( d \): degree vector

• Solve the Linear Program using a modified push-relabel algorithm
• Round the output

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Cluster</th>
<th>Precision/Recall</th>
<th>L1-reg. PageRank</th>
<th>CRD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Facebook100</td>
<td>2008</td>
<td>0.64/0.95</td>
<td>0.94/0.96</td>
<td></td>
</tr>
<tr>
<td>Colgate University</td>
<td>2009</td>
<td>0.93/0.98</td>
<td>0.97/0.98</td>
<td></td>
</tr>
</tbody>
</table>
Network Community Profiles

- Study structure of a graph at different scales

Time evolving graph with downward slopping NCP

Social network with constant NCP. Most clusters have conductance $O(0.1)$

- Parallelized for shared memory architectures

https://github.com/kfoynt/LocalGraphClustering/blob/master/notebooks/examples.ipynb
Interactive Network Community Profiles

- Click on each node to see the index of that experiment.

https://github.com/kfoynt/LocalGraphClustering/blob/master/notebooks/examples.ipynb
Improvement plots

- Study structure of a graph based on clustering improvement.
- Parallelized for shared memory architectures

https://github.com/kfoynt/LocalGraphClustering/blob/master/notebooks/Demos-SFData-MQI-NCP.ipynb
Finding small clusters

• Find clusters using the NCP pipeline and filter based on conductance.

https://github.com/kfoynt/LocalGraphClustering/blob/master/notebooks/find_very_small_clusters_in_images.ipynb
Graph Partitioning

- Compute in parallel local spectral embeddings (one l1-reg. PageRank vector for each node)
- Normalize the solutions
- Use hierarchical clustering

https://github.com/kfoynt/LocalGraphClustering/blob/master/notebooks/find_clusters_social_network.ipynb
Graph Partitioning

- Compute in parallel local spectral embeddings (one l1-reg. PageRank vector for each node)
- Normalize the solutions
- Use hierarchical clustering

Marked boundaries Local Graph Clustering + MQI  
Label2rgb Local Graph Clustering + MQI  
Original

https://github.com/kfoynt/LocalGraphClustering/blob/master/notebooks/image_segmentation_using_local_graph_clustering_and_gPb.ipynb
Multi-label Classification

- Use a fraction of the given labelled nodes.
- Find the remaining labels using flow methods

https://github.com/kfoynt/LocalGraphClustering/blob/master/notebooks/flow_paper_experiments/semisupervised_learning.ipynb
Scalability of Local Spectral

- Find thousands of clusters fast using l1-regularized PageRank

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Number of clusters</th>
<th>Time in Seconds</th>
</tr>
</thead>
<tbody>
<tr>
<td>Orkut</td>
<td>153622</td>
<td>1129</td>
</tr>
<tr>
<td>LiveJournal</td>
<td>242378</td>
<td>1072</td>
</tr>
</tbody>
</table>

2x Intel E5-2670 (8 cores) CPU with 128 GB RAM

Orkut: 3072441 nodes and 117185083 edges, LiveJournal: 4847571 nodes and 68993773 edges
Scalability of Flow Improve Methods

- Improve thousands of clusters fast using MQI and SimpleLocal

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Number of clusters</th>
<th>Time in Seconds for MQI</th>
<th>Time in Seconds for SimpleLocal</th>
</tr>
</thead>
<tbody>
<tr>
<td>Orkut</td>
<td>153622</td>
<td>391</td>
<td>1042</td>
</tr>
<tr>
<td>LiveJournal</td>
<td>242378</td>
<td>1327</td>
<td>2710</td>
</tr>
</tbody>
</table>

Orkut: 3072441 nodes and 117185083 edges, LiveJournal: 4847571 nodes and 68993773 edges
Thank You!
Target clusters in practice

- There are numerous papers that claim state-of-the-art practical performance for methods in LocalGraphClustering.


D. Wang, K. Fountoulakis, M. Henzinger, M. Mahoney, S. Rao, Capacity Releasing Diffusion, ICML, 2017