Scalable Algorithm for Probabilistic Overlapping Community Detection

Kento Nozawa, Kei Wakabayashi
University of Tsukuba

WSDM 2017 workshop on SWM
It's hard to analyze a large graph.

Examples:
- Citation networks
- Co-author relationships
- Social networks
- Hyperlinks on web pages

Needs: Decomposition a large graph into some smaller subgraphs
Community Structures in Graph

In the same community,
- nodes are densely connected internally

- nodes resemble the others
  - Same affiliation
  - Same interest
  - Related research area
Overlapping Community

Each node belongs to multiple communities.

Many graphs have overlapping communities
  – Ex. Related Research areas in co-author graph

• Blue : Data mining
• Red : Machine learning

A has published in both areas
Bag-of-nodes Representation

Bag-of-words for graph
- A node corresponds to one document
- The node and its adjacency list correspond to words in the document

<table>
<thead>
<tr>
<th>Node as doc</th>
<th>Nodes as words</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>A, B, C, D, E, x, y, z</td>
</tr>
<tr>
<td>B</td>
<td>B, A, E</td>
</tr>
<tr>
<td>C</td>
<td>C, D, E, A</td>
</tr>
<tr>
<td>D</td>
<td>D, E, C, A</td>
</tr>
<tr>
<td>E</td>
<td>E, B, A, C, D</td>
</tr>
<tr>
<td>x</td>
<td>x, y, A, w</td>
</tr>
<tr>
<td>y</td>
<td>y, A, x, z</td>
</tr>
<tr>
<td>z</td>
<td>z, y, A</td>
</tr>
<tr>
<td>w</td>
<td>w, x</td>
</tr>
</tbody>
</table>

Graph

Bag-of-nodes
Latent Dirichlet Allocation (LDA) [Blei+, 2003]

- Probabilistic generative model for bag-of-words
- Find topics from words co-occurrence
- Each topic defines a distribution over all words

**Documents (bag-of-words)**

<table>
<thead>
<tr>
<th>Coffee shop</th>
<th>Topics (distribution over all words)</th>
</tr>
</thead>
</table>
| drink drink coffee coffee beans beans espresso cafe | coffee 0.15  
drink 0.15  
beans 0.14  
cafe 0.13  
... | author 0.14  
cite 0.12  
citation 0.11  
review 0.11  
... |
LDA for Graph

A topic represents an overlapping community. Each community is an affiliation probability distribution over nodes.

<table>
<thead>
<tr>
<th>Node as doc</th>
<th>Nodes as words</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>A, B, C, D, E, x, y, z</td>
</tr>
<tr>
<td>B</td>
<td>B, A, E</td>
</tr>
<tr>
<td>C</td>
<td>C, D, E, A</td>
</tr>
<tr>
<td>D</td>
<td>D, E, C, A</td>
</tr>
<tr>
<td>E</td>
<td>E, B, A, C, D</td>
</tr>
<tr>
<td>x</td>
<td>x, y, A, w</td>
</tr>
<tr>
<td>y</td>
<td>y, A, x, z</td>
</tr>
<tr>
<td>z</td>
<td>z, y, A</td>
</tr>
<tr>
<td>w</td>
<td>w, x</td>
</tr>
</tbody>
</table>

**Communities** (distributions over nodes)

- E: 0.20
- C: 0.20
- D: 0.18
- B: 0.18
- A: 0.12

- x: 0.22
- y: 0.22
- z: 0.20
- A: 0.15
- w: 0.07
- C: 0.05
- D: 0.04
- B: 0.04
- E: 0.01

**Graph as documents**

- A, B, C, D, E, x, y, z
- B, A, E
- C, D, E, A
- D, E, C, A
- E, B, A, C, D
- x, y, A, w
- y, A, x, z
- z, y, A
- w, x
Stochastic Variational Inference [Mimno+, 2012]

Inference algorithms based on stochastic gradient descent
- Update parameters based on sampling nodes in each iteration
- mini-batch size: # sampling nodes as document

<table>
<thead>
<tr>
<th>Node as doc</th>
<th>Nodes as words</th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
<td>B, A, E</td>
</tr>
<tr>
<td>C</td>
<td>C, D, E, A</td>
</tr>
<tr>
<td>z</td>
<td>z, y, A</td>
</tr>
<tr>
<td>w</td>
<td>w, x</td>
</tr>
</tbody>
</table>

When mini-batch size is 4

<table>
<thead>
<tr>
<th>Node as doc</th>
<th>Nodes as words</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>A, B, C, D, E, x, y, z</td>
</tr>
<tr>
<td>B</td>
<td>B, A, E</td>
</tr>
<tr>
<td>C</td>
<td>C, D, E, A</td>
</tr>
<tr>
<td>D</td>
<td>D, E, C, A</td>
</tr>
<tr>
<td>E</td>
<td>E, B, A, C, D</td>
</tr>
<tr>
<td>x</td>
<td>x, y, A, w</td>
</tr>
<tr>
<td>y</td>
<td>y, A, x, z</td>
</tr>
<tr>
<td>z</td>
<td>z, y, A</td>
</tr>
<tr>
<td>w</td>
<td>w, x</td>
</tr>
</tbody>
</table>

Graph as documents
Experiment

Evaluation of scalability for the graph size
• Runtime for overlapping community detection

Quality metrics for overlapping communities
• Triangle participation ratio (TPR)
  – Ratio of #nodes that belong to a triangle
  – Higher is better
• Conductance
  – Ratio of #edges that link to an outer node
  – Lower is better
Experimental Datasets

<table>
<thead>
<tr>
<th>Name</th>
<th>#nodes</th>
<th>#edges</th>
</tr>
</thead>
<tbody>
<tr>
<td>DBLP</td>
<td>317,080</td>
<td>1,049,866</td>
</tr>
<tr>
<td>Orkut</td>
<td>3,072,441</td>
<td>117,185,083</td>
</tr>
<tr>
<td>Friendster</td>
<td>65,608,366</td>
<td>1,806,067,135</td>
</tr>
</tbody>
</table>

*From SNAP Datasets*

Only Friendster:

- Store into MySQL
- Sample mini-batch size records from the table
Comparison of Runtime

#communities: 4,000
#iterations: 1,000
Mini-batch size: 2,000

Dataset
- DBLP
- Orkut
- Friendster

Runtime (hour)

Methods
- CFinder
- SVNET
- SVNET ITE5
- SVNET ITE10
- BIGCLAM
- SGLDA
- SGLDA_train
- SVBLDA
- SVBLDA_train

2 hours
7 min.
The Metrics of DBLP Communities

TPR: the median of SVBLDA is the third best

Conductance: the median of SVBLDA is the third worst

#communities: 4,000
#iterations: 1,000
Mini-batch size: 2,000
Parameter Sensitivity in DBLP

- Varying mini-batch size or # iterations when fixing the other parameter
- No significantly improvement of TPR/Conductance when mini-batch size > 3000 or # iterations > 2000

#iterations: 1,000
Mini-batch size: 2,000
Conclusion

• Scalable community detection algorithm based on LDA for large graph

• About 2 hours to detect communities from the large graph

• It’s unnecessary to set large mini-batch size and #iteration for DBLP datasets