On the Separability of Structural Classes of Communities

Bruno Abrahao
Sucheta Soundarajan
John Hopcroft
Robert Kleinberg

Cornell University
Community Structure

[Newman-Girvan, 2004]
How do their structures differ?

- Metis
- ‘Real’ Community
- Random Walk
- Infomap
- Newman-Modularity
- Louvain
Community Detection

- Community structure is not well defined
  - different people have different notions of community structure

- Traditional strategy
  - (1) start with an expectation of what a community should look like
    - e.g., a set of nodes that interact more within the set than with the outside
  - (2) define an optimization problem
  - (3) design heuristic
  - (4) the solution gives the desired communities
Key questions

• A multitude of algorithms
  – different objective functions
  – different heuristics

How dissimilar are their outputs?

• Communities may differ from the proposed mathematical constructs
  – e.g., preponderance of links to the outside

Which algorithms extract communities that most closely resemble the structure of real communities?
Obstacles to answering the questions

• We don't know what properties communities possess

• We can't characterize communities in the absence of negative examples
  – Look at real communities and determine their structure
  – do other sets that are not communities have these properties?
  – every other connected set could be a negative example - intractable
  – sets that are not annotated could also be communities

• We don't know what metrics we should use
  – modularity, conductance, clustering coefficient...
Our plan

• Propose a methodology to **analyze** structural community properties by **comparing** different notions of communities

  – **key idea**: analyze community structure without requiring negative examples of communities

• Scalable and comprehensive, simultaneously considering
  – multiple notions of communities
  – diverse domains of application
  – a broad spectrum of community metrics

• Assess the structural dissimilarities between
  – the output of different community detection algorithms
  – the output of algorithms and real communities
Building structural classes

```c
class Port {
    protected:
        uint8_t portNum;
    public:
        Port (uint8_t num);
        // 320 pin
        void code(const uint8_t value) const;
        uint8_t digRead() const;
        void digWrite(const uint8_t val) const;
        uint32_t_pulse(const uint8_t state, uint32_t timeout = 1000000) const;

        // 320 pin
        void node2(const uint8_t value) const;
        uint8_t anaRead() const;
        uint8_t digRead2() const;
        void digWrite2(const uint8_t value) const;
        uint32_t_pulse2(const uint8_t state, uint32_t timeout = 1000000) const;

        // 180 pin (INT1, shared across all ports)
        static void node3(const uint8_t value);
        static uint8_t digRead3();
        static void digWrite3(const uint8_t value);

        // both pins: data on OSO, clock on A10
        void shift(const uint8_t bitOrder, uint8_t value) const;
        void shiftWrite(const uint8_t bitOrder, uint8_t count = 0) const;
    }
```
Building structural classes

Algorithm 1
1111111111
1111

Algorithm 2

Algorithm 3

Algorithm 4

Algorithm k

Class 1

Class 2

Class 3

Class 4

Class k
Building a feature space

Labeled Example

<table>
<thead>
<tr>
<th>#</th>
<th>Feature</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$n$</td>
</tr>
<tr>
<td>2</td>
<td>$m$</td>
</tr>
<tr>
<td>3</td>
<td>Diameter</td>
</tr>
<tr>
<td>4</td>
<td>Edge Density</td>
</tr>
<tr>
<td>5</td>
<td>Conductance</td>
</tr>
<tr>
<td>6</td>
<td>Transitivity</td>
</tr>
<tr>
<td>7</td>
<td>Triangle Density</td>
</tr>
<tr>
<td>8-11</td>
<td>Shortest Path</td>
</tr>
<tr>
<td>12-15</td>
<td>Edge Betweenness</td>
</tr>
<tr>
<td>16-20</td>
<td>Node Betweenness</td>
</tr>
<tr>
<td>21-25</td>
<td>$\alpha$</td>
</tr>
<tr>
<td>26-30</td>
<td>$\beta$</td>
</tr>
<tr>
<td>31</td>
<td>Treesum</td>
</tr>
<tr>
<td>32-36</td>
<td>Information Centrality</td>
</tr>
</tbody>
</table>
Building a feature space

Feature Space
Inter-class separability

Are the classes separable?

Class Separability Measure

Separability = Distinct structures

Feature Space
Large-scale network datasets

- **Social**
  - LiveJournal
  - Facebook
  - Rice University

- **Commercial**
  - Amazon.com

- **Biological**
  - Human evolution
  - Insect patterns

Facebook+Rice with permission of Mislove et al.. Other datasets publicly available.
Community detection algorithms

- BFS (Random connected subgraphs)
- Random-Walk-based (with and without restart)
- \((\alpha, \beta)\)-communities
- InfoMap
- Markov Clustering
- Metis
- Louvain
- Newman-Clauset-Moore
- Link Communities

```cpp
class Port {
  protected:
    uint8_t portNum;
  public:
    Port (uint8_t num);
    // DIO pin
    void node(uint8_t value) const;
    uint8_t digiRead() const;
    void digiWrite(uint8_t value) const;
    void enWrite(uint8_t val) const;
    uint32_t pulse(uint8_t state, uint32_t timeout =1000000L) const;
    // AIO pin
    void node2(uint8_t value) const;
    uint16_t anaRead() const;
    uint8_t digiRead2() const;
    void digiWrite2(uint8_t value) const;
    uint32_t pulse2(uint8_t state, uint32_t timeout =1000000L) const;
    // IRQ pin (INT1, shared across all ports)
    static void node3(uint8_t value);
    static uint8_t digiRead3();
    static void digiWrite3(uint8_t value);
    // both pins: data on DIO, clock on AIO
    void shift(uint8_t bitOrder, uint8_t value) const;
    uint16_t shiftRead(uint8_t bitOrder, uint8_t count =8) const;
    void shiftWrite(uint8_t bitOrder, uint16_t value, uint8_t count =8) const;
};
```
Annotated communities

Metadata included in the datasets identifies exemplar communities that form in these domains.
To what extent are the classes separable?
Separability measures

- Traditional methods for measuring class separability give a single score, e.g., scatter matrices

<table>
<thead>
<tr>
<th>Network</th>
<th>Grad</th>
<th>Ugrad</th>
<th>HS</th>
<th>SC</th>
<th>Fly</th>
<th>DBLP</th>
<th>Amaz</th>
<th>LJ1</th>
<th>LJ2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Global</td>
<td>19.2</td>
<td>22.3</td>
<td>26.0</td>
<td>27.4</td>
<td>6.3</td>
<td>22.7</td>
<td>16.4</td>
<td>19.7</td>
<td>21.9</td>
</tr>
<tr>
<td>Ref.</td>
<td>14.7</td>
<td>13.1</td>
<td>13.0</td>
<td>13.0</td>
<td>12.9</td>
<td>13.0</td>
<td>12.9</td>
<td>12.9</td>
<td>12.9</td>
</tr>
</tbody>
</table>

- Reference: the same data with shuffled labels

- This is a global measure. We need more fine-grained separability information of each class!
Probabilistic multi-class learners

Train

Probabilistic k-way classifier (SVM, k-NN)

Algorithm 1

Algorithm 2

Annotated communities
Probabilistic multi-class learners

Classify (cross-validation)

Probabilistic k-way classifier (SVM, k-NN)

\[ \text{Pr(Algorithm 1)} = 0.05 \]
\[ \text{Pr(Algorithm 2)} = 0.08 \]
\[ \text{...} \]
\[ \text{Pr(Annotated)} = 0.48 \]
Cross-validation performance

Probabilistic-SVM cross-validation outcome with 11 structural classes.
Data: DBLP network.
Matching annotated communities

Which algorithms extract communities that most closely resemble the structure of annotated communities?
Probabilistic multi-class learners

Learn

Probabilistic k-way classifier

Algorithm 1

Algorithm 2

Algorithm N
Probabilistic multi-class learners

Classify

Probabilistic k-way classifier

Pr(Algorithm 1) = 0.02
Pr(Algorithm 2) = 0.19
...
Pr(Algorithm k) = 0.12
Probabilistic-SVM classification of annotated communities into 11 structural classes structural class for 9 different networks.
Can we reveal latent similarities among community detection algorithms?

Our framework enables one to cluster algorithms that behave similarly
Step 1: identifying the most important features

<table>
<thead>
<tr>
<th>Rank</th>
<th>Feature</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Conductance</td>
</tr>
<tr>
<td>1</td>
<td>Diameter</td>
</tr>
<tr>
<td>3</td>
<td>Info Centrality*</td>
</tr>
<tr>
<td>4</td>
<td>Node Betweenness*</td>
</tr>
<tr>
<td>5</td>
<td>Shortest Path*</td>
</tr>
<tr>
<td>6</td>
<td>$\beta^*$</td>
</tr>
<tr>
<td>7</td>
<td>$\alpha^*$</td>
</tr>
</tbody>
</table>

7 features out of 36 retain the discriminative power of the full set
Tendencies of algorithms with respect to most discriminative features
Conclusion

- We present a methodology to address the complexity of analyzing community structure, which simultaneously considers
  - large number of algorithms
  - multiples domains of application
  - a broad spectrum of metrics to characterize community structure
- A scalable framework that enables
  - researchers to compare and understand biases of new and existing community detection algorithms
  - practitioners to decide on the most suitable algorithm for particular purpose and network
Conclusion

- Our experimental analysis, which include 10 community detection algorithms and 9 different networks analyzed with 36 properties reveals:
  - High variability among the output of community detection methods
  - Annotated communities have a distinct structure from what we expect
    - their structure is closer to the output of baseline procedures than to that of popular algorithms
  - A small set of features explain the biases produced by different algorithms
  - We can organize the tapestry of available community detection algorithms by grouping them with respect to similarities in behavior
Final remarks

- Traditional methods are **unsupervised**
  - they find a particular type of community
  - little sensitivity to different purposes, structures of interest and domains of application

- Our approach suggests a **supervised** approach to community detection
  - user specifies what they intended to find through examples (real or synthetic)
  - algorithm learns from those examples and retrieves similar structures in the network
Thank you!

On the Separability of Structural Classes of Communities

Bruno Abrahao
Sucheta Soundarajan
John Hopcroft
Robert Kleinberg

Cornell University
My comments

Chi Wang
Sep 11 2012
What this paper is about

• High-level idea
  – Which algorithms extract communities that most closely resemble the structure of real communities?

• Specific setting
  – Homogeneous network
  – Comparing different communities by their structural features
  – Comparing communities extracted by different algorithms.
What we can do - 1

• High-level idea
  – Which algorithms extract communities that most closely resemble the structure of real communities?

• Specific setting
  – Homogeneous network ➔ heterogeneous
  – Comparing different communities by their structural features ➔ meta-path based features
  – Comparing communities extracted by different algorithms ➔ (NetClus, MetaSim, homogeneous net clustering algorithms)
What we can do – 2 (ongoing)

• High-level idea
  – Which algorithms extract communities that most closely resemble the structure of real communities?
  
  ➡️
  – Which measures can be used to detect groupings that are most similar to a particular target grouping?