Attributed Networks: Social Circles, Summarization, Comparison

Leman Akoglu

Joint work with Bryan Perozzi (Google Research NYC), Rashmi Raghunandan, Shruti Sridhar, Upasna Suman (CMU), Aria Rezaei (Stony Brook University).

NetSci 2018 Satellite on Machine Learning In Network Science

June 12, 2018
Attributed graphs

Attributed graph: each node has 1+ properties
Attributed networks

- Social networks
  - demographics, lifestyles, likes, ...
- PPI networks
  - Gene encodings
- Gene interaction networks
  - ontological properties
- Web
  - page properties
- ...

Carnegie Mellon
Motivating question:

How can we make sense of node-attributed networks?

- subgraphs
- summaries
- comparisons

220 nodes, 6215 edges
Attributed networks

Idea is “description-by-parts”: identifying & characterizing the subgraphs.
SOCIAL CIRCLES
How offline relationships influence online behavior and what it means for design and marketing

Paul Adams
Research questions:

① How to characterize & measure the **quality** of …
② How to **summarize** & interactively explore …
③ How to **characterize** differences between **classes** of …

… attributed subgraphs?

1) Scalable Anomaly Ranking of Attributed Neighborhoods  
   SIAM SDM 2016
2) Discovering Communities and Anomalies in Attributed Graphs:  
   Interactive Visual Exploration and Summarization  
   ACM TKDD, 2018
   *Bryan Perozzi and Leman Akoglu*
3) Ties That Bind - Characterizing Classes by Attributes and Social Ties  
   *Aria Rezaei, Bryan Perozzi, Leman Akoglu*  
   WWW 2017 Companion
This talk

- Attributed (sub)graphs*

  Subgraphs [SIAM SDM’16]
  - Summarization [ACM TKDD’18]
  - Comparisons [WWW ’17]

* social circles, communities, egonetworks, …
What’s a “good” subgraph anyway?

- Given an attributed subgraph, how to quantify its quality?
  - Structure-only
    - Internal-only
      - average degree
    - Boundary-only
      - cut edges
    - Internal + Boundary
      - conductance
  - Structure + Attributes

Scalable Anomaly Ranking of Attributed Neighborhoods
Bryan Perozzi and Leman Akoglu
SIAM SDM 2016.
Normality (intuition)

- **Given** an attributed subgraph
  - how to **quantify** quality?
  - Internal
    - structural density
Normality (intuition)

- Given an attributed subgraph, how to quantify quality?
  - Internal
    - structural density AND
    - attribute coherence
      - neighborhood “focus”

Carnegie Mellon
Normality (intuition)

- **Given** an attributed subgraph, how to **quantify** quality?

  - **Internal**
    - structural density AND
    - attribute coherence
      - *neighborhood “focus”*
  
  - **Boundary**
    - structural sparsity, OR
    - external separation
      - *“exoneration”*
Normality (intuition)

- “exoneration”: by (a) null model, (b) attributes edges expected, not surprising separable by different “focus”

- Motivation:
  - no good cuts in real-world graphs [Leskovec+ ‘08]
  - social circles overlap [McAuley+ ‘14]
The measure of Normality

Null model

\[ N = I + E = \sum_{i \in C, j \in C} \left( A_{ij} - \frac{k_i k_j}{2m} \right) s(x_i, x_j | w) \]

Internal consistency

dot-product, or Kronecker's δ

“Focus” vector

wine    cheese
The measure of Normality

\[
N = I + E = \sum_{i \in C, j \in C} (A_{ij} - \frac{k_i k_j}{2m}) s(x_i, x_j | w)
\]

- \[
\sum_{i \in C, b \in B} (1 - \min(1, \frac{k_i k_b}{2m})) s(x_i, x_b | w)
\]

external separability
The measure of **Normality**

- **Given an attributed subgraph, can we find the attribute weights?**

\[
N(C') = \sum_{i \in C, j \in C, \atop i \neq j} \left( A_{ij} - \frac{k_i k_j}{2m} \right) \text{sim}_w(x_i, x_j)
\]

\[ - \sum_{i \in C, b \in B \atop (i, b) \in E} \left( 1 - \min(1, \frac{k_i k_b}{2m}) \right) \text{sim}_w(x_i, x_b) \]

\[ \arg \max_w \mathbf{w}^T \left[ \sum_{i \in C, j \in C, \atop i \neq j} \left( A_{ij} - \frac{k_i k_j}{2m} \right) (x_i \ominus x_j) \right. \]

\[ - \sum_{i \in C, b \in B \atop (i, b) \in E} \left( 1 - \min(1, \frac{k_i k_b}{2m}) \right) (x_i \ominus x_b) \]
Optimizing Normality

\[
N = I + E = \sum_{i \in C, j \in C} (A_{ij} - \frac{k_i k_j}{2m}) s(x_i, x_j | w) - \sum_{i \in C, b \in B, (i, b) \in E} (1 - \min(1, \frac{k_i k_b}{2m})) s(x_i, x_b | w)
\]

\[
\max_{w_C} \quad w_C^T \cdot \left[ \sum_{i \in C, j \in C} (A_{ij} - \frac{k_i k_j}{2m}) s(x_i, x_j) \right. \\
\left. - \sum_{i \in C, b \in B, (i, b) \in E} (1 - \min(1, \frac{k_i k_b}{2m})) s(x_i, x_b) \right]
\]

\[
\max_{w_C} \quad w_C^T \cdot (\hat{x}_I + \hat{x}_E)
\]

\[
\text{s.t.} \quad \|w_C\|_p = 1, \quad w_C(f) \geq 0, \quad \forall f = 1 \ldots d
\]
Optimizing Normality

$$\max_{w_C} w_C^T \cdot \left( \frac{\hat{x}_I + \hat{x}_E}{x} \right)$$

s.t. $$\|w_C\|_p = 1, \ w_C(f) \geq 0, \ \forall f = 1 \ldots d$$

$p = 1$: $w_C(f) = 1$ one attribute $f$ with largest $x$

$p = 2$: $w_C(f) = \frac{x(f)}{\sqrt{\sum_{x(i) > 0} x(i)^2}}$ all $f$ with positive $x$

Normality becomes $N = w_C^T \cdot x = \|x_+\|_2$

Linear in number of attributes!

when $p = 1$, $N \in [-1, 1]$ $N \in [-1, \|x_+\|_2]$ when $p = 2$. 

Carnegie Mellon
Illustrative examples

telescopic op-amps

split-radix FFT

Jaime Ramírez-Angulo
Milind S. Sawant
Ramón González Carvajal
Antonio J. López-Martín

M. N. Shanmukha Swamy
Saad Bouguezel
M. Omair Ahmad

reciprocal
reserve
split
Example neighborhoods

DBLP
$L_1 = 0.979$, $L_2 = 2.17$

Twitter
$L_1 = 0.724$, $L_2 = 1.10$

Google+
$L_1 = L_2 = -0.873$

Citeseer
$L_1 = L_2 = -0.956$
Anomaly detection: Perturbed data

Citeseer Structure Perturbation

Citeseer Attribute Perturbation

Average Precision AUC

Perturbation Level

Carnegie Mellon
Normality vs Conductance, DBLP
Attribute distribution, DBLP
Summary

A new quality measure for attributed subgraphs called *normality* considers:
- internal + boundary
- structure + attributes
- subgraph focus
- "exoneration"

Automatic inference of focus via *normality* maximization
- unsupervised
- linear in #attributes
Anomoly Ranking of Attributed Neighborhoods

Bryan Perozzi, Leman Akoglu
May 9, 2016

Awards: Best Paper Runner-up, SDM’16!

Overview

Given a graph with node attributes, what neighborhoods are anomalous? To answer this question, one needs a quality score that utilizes both structure and attributes. Popular existing measures either quantify the structure only and ignore the attributes (e.g., closeness centrality) or quantifies the attributes only and ignore the structure (e.g., density) and ignore the cross-edges at the boundary (e.g., density).
This talk

- Attributed (sub)graphs*
  - Subgraphs  [SIAM SDM’16]
  - Summarization  [ACM TKDD’18]
  - Comparisons  [WWW ’17]

* social circles, communities, egonetworks, …
Overview

Interactive exploration

Summarization

Social circle extraction

Input graph
Overview

1. **Summarization**

   - **Algorithm Summary**
     - Normality: 1.0084
     - Coverage: 1.0096
     - Diversity: 1.71429

2. **Filter Panel**

   - Attribute: Student/Faculty, Gender, Major, Minor, Donor, Grad Year, High School

3. **Interactive exploration**

   - **Community Exploration Panel**
   - **Algorithm Input**
     - K: 1
     - Weight - Normality: 1.8
     - Weight - Coverage: 1.8
     - Weight - Diversity: 1.8

Input graph

Social circle extraction

---

**Fig. 1.** Sensemaking of attributed graphs is challenging with traditional means. (a) Table view for edge relations and node attributes—does not provide much focus attribute and circle size is proportional to community size and characterizing attributes. Hovering over each circle displays the output from our summarization algorithm. To do this, we deconstruct the network into its building blocks, i.e. communities, and devise alternative summaries.

---

**Exploratory and Interactive Summarization:**

1. **Filtering:**

   - Words and future research.

2. **Summarization:**

   - Each circle is colored by its focus attribute and circle size is proportional to community size. We define the distance between the pair of corresponding circles as the distance between the community terms. A weighted combination of circle quality, network normality, coverage, and diversity is presented to a user. It contains three main panels, the filtering panel, the community exploration panel, and the algorithm summary panel. The filtering panel allows users to focus on specific aspects of the network, such as attributes, size, and other characteristics. The community exploration panel displays the communities found by the algorithm, along with their properties, such as size and centrality. The algorithm summary panel shows the results of the summarization process, including the selected communities, their quality, and other relevant metrics.

3. **Interactive exploration:**

   - The user interface that enables users to efficiently explore, characterize, and build various alternative summaries of the network via a few essential communities that make up its backbone. Our work differs from prior work in multiple key aspects. It attempts to build a single framework that unify all three aspects (filtering, summarization, and interactive visualization), none of those works attempt to build a single framework that unify all three aspects. We further introduce an interactive system for end-users to analyze the communities, explore and build summaries. While the emphasis is on communities and yet we do not solely propose a concise list and summary of communities, and help devise alternative summaries.

---

**Fig. 2.** Sensemaking of attributed graphs is challenging with traditional means. (a) Table view for edge relations and node attributes—does not provide much focus attribute and circle size is proportional to community size and characterizing attributes. Hovering over each circle displays the output from our summarization algorithm. To do this, we deconstruct the network into its building blocks, i.e. communities, and devise alternative summaries.
Extracting Social Circles

- a GRASP (Greedy Randomized Adaptive Search Procedure) approach [Feo & Resende ‘95]

```
Algorithm 1 EXTRACTATTRIBUTEDSOCIALCIRCLES
Input:  \( G = (\mathcal{V}, \mathcal{E}, \mathcal{A}) \), node attribute vectors \( x_{u \in \mathcal{V}} \), \( T_{max}, \alpha \)
Output: set of extracted communities \( \mathcal{C} \)
1:  \( \mathcal{C} := \emptyset \)
2:  for each \( u \in \mathcal{V} \) do
3:      for \( t = 1 : T_{max} \) do
4:         \( S := \text{CONSTRUCTION}(u, G, \alpha) \)
5:         \( \mathcal{C} := \mathcal{C} \cup \text{LOCALSEARCH}(S, G') \)
6:      end for
7:  end for
8:  return \( \mathcal{C} \)
```

- note: one focus attribute per circle
Overview

Interactive Visual Analysis

Summarization

1) Exploratory and Interactive Summarization: The last selection; which, in the figure, increased (bottom left). Red vertical lines show the

2) Social circle extraction

3) Input graph

4) Overview

Carnegie Mellon
Summarization

- Social circles: what size, quality and focus?
  - Attempt: visual summary
    - size \( \propto \) #nodes
    - color: 'focus'
    - text: normality

125 circles!

- does not reflect overlap between circles!
Summarization

- Want a summary (a few circles):
  - high normality
  - well-“cover” the graph
  - diverse in ‘focus’

\[
\max_{S \subseteq C, |S| = K} f(S) = \alpha \ \text{avgnorm}(S) + \beta \ \text{cov}(S) + (1 - \alpha - \beta) \ \text{div}(S)
\]

\[
= \alpha \sum_{C \in S} \frac{N(C)}{K} + \beta \frac{|\bigcup_{C \in S} C|}{n} + (1 - \alpha - \beta) \frac{|\bigcup_{C \in S} A(C)|}{d}
\]

\[0 \leq \alpha, \beta \leq 1\] can be interactively adjusted by users
**Summarization**

\[
\max_{s \subseteq C \atop |s| = K} f(S) = \alpha \sum_{c \in s} \frac{N(C)}{K} + \beta \frac{\bigcup_{c \in s} C}{n} + (1 - \alpha - \beta) \frac{\bigcup_{c \in s} A(C)}{d}
\]

- Provided \( K, n, d \) (denominators) fixed, easy to show that \( f : 2^C \rightarrow \mathbb{R}_+ \) is
  - non-negative
  - monotonic: \( A \subseteq B \subseteq C \), \( f(A) \leq f(B) \)
  - submodular: for every \( A \subseteq B \subseteq C \) and \( C \in C \setminus B \),
    \[
    f(A \cup \{C\}) - f(A) \geq f(B \cup \{C\}) - f(B)
    \]
  - The “next-best” greedy algorithm: at least 63% of the objective value \( f(\cdot) \) of the optimum set.
Summarization

- surface formed by various parameter combinations \((\alpha, \beta, 1-\alpha-\beta)\) (blue dots)
  - (green) square around the “knee”: a good trade-off between quality, coverage, and diversity
Overview

Interactive exploration

Summarization

Input graph

Social circle extraction
Interactive Visual Exploration & Summarization

**Sensemaking of Attributed Social Networks**

**Filter Panel**
- **Attribute**
  - Student/Faculty
  - Gender
  - Major
  - Minor
  - Dorm
  - Grad Year
  - High School
- **Normality**
  - 0.0562
  - 1.0000
- **Size**
  - (All)
  - 3
  - 4
  - 5
  - 6
  - 7
  - 8
  - 9
  - 10
  - 11
  - 12
  - 13
  - 14
  - 15

**Community Exploration Panel**
- **Size**
  - 17
  - Members: 2 4 8 10 12 15 17 21 23
  - Members: 24 31 33 38 40 48 51 52
  - Normality: 0.7495

**Algorithm Summary**
- **Algorithm Input**
  - K: 5
  - Weight - Normality: 0.4
  - Weight - Coverage: 0.4
  - Weight - Diversity: 0.2

**Algorithm Output**
- **Algorithm Normality**: 0.83942
- **Algorithm Coverage**: 0.53636
- **Algorithm Diversity**: 0.71429

**User Output**
- **Normality**
- **Coverage**
- **Diversity**

---

This helps the user to quickly identify overlapping/redundant ones, as well as observe the distribution of social circles by diversity of the summary is displayed (bottom right). The user can also devise their own summary by selecting (through a click) the circles they would like to include in the summary, or selecting a subset of representative circles as selected by our summarization algorithm. This enables users to explore and build alternative summaries.

In this work we adopt a measure of quality for attributed communities, called "surprise", the higher the better. In this document, we denote by $\text{Sim}(G, B)$ the similarity between graph $G$ and its normalized version $B$, which enables us (1) to extract communities from the input graph independently in parallel using local search algorithms, and (2) rank attributes by their contribution to the building blocks are the communities, or the social circles. For an attributed graph $G = (V, E, X)$, we define $\text{Sim}(G, B)$ as the weighted dot product similarity between two nodes. Intuitively, for a given community $C$ we have at least one edge to some node in $A$ (i.e., the set of boundary nodes, which reside outside the circle but within the community or circle (subset of nodes) in $G$). The term (1) captures internal connectivity and cover the input graph well; whereas (2) rank attributes by their contribution from the input graph independently in parallel using local normality, egonet coverage, and attribute diversity is maximized.

For an attributed graph $G = (V, E, X)$, we define $\text{Sim}(G, B)$ as the expected number of edges between two nodes of the building blocks is the communities, or the social circles. Our goal is to follow a divide-and-conquer approach to summarizing large graphs. As such, we will aim to decompose the graph into its building blocks. For social networks, the building blocks are the communities, or the social circles. To do so, we use a divide-and-conquer approach to decompose the graph into communities. We start by finding the communities, or the social circles. Then, we find the communities, or the social circles. Finally, we find the communities, or the social circles. This helps the user to quickly identify overlapping/redundant ones, as well as observe the distribution of social circles by diversity of the summary is displayed (bottom right). The user can also devise their own summary by selecting (through a click) the circles they would like to include in the summary, or selecting a subset of representative circles as selected by our summarization algorithm. This enables users to explore and build alternative summaries.
Circle embedding

2-D MDS embedding preserving:

$$\text{dist}(C_k, C_l) = 1 - \frac{|C_k \cap C_l|}{\min(|C_k|, |C_l|)}$$

size $\propto$ #nodes
color: focus
Interaction: Filtering

Filter Panel

Attribute
- Student/Faculty
- Gender
- Major
- Dorm
- Grad Year

Normality

Size

[ ] (All)
- 3
- 4
- 6
- 7
- 8
- 9
- 10
- 13
- 14

While searching for circles to select for their own summary, the user may want to focus on communities with certain properties. As such, we introduce a panel for filtering communities by focus attribute, normality, and size, as shown in Fig. 5.

Algorithm-Guided Human-in-the-loop Summarization

While creating the summary, users can view, select, and de-select normality, coverage, and diversity—are displayed in (blue) bar plots as in Fig. 7.

To this end, our idea is to visualize each community as a nested community. Multi-Dimensional Scaling (MDS) is used to find a 2-d embedding of the communities such that the pairwise distances as defined above are preserved in the Euclidean space as much as possible. As such, largely overlapping communities are clustered in the display. The user can then aim to select large circles that are spread out in the space as much as possible. As such, largely overlapping as well as nested communities should help the user quickly grasp the size, normality, and the diversity of their current summary (best in color).
Interaction: Circle summarization
**Evaluation**

**Q1) Summarization by visual exploration.** *Does interactive visualization help users construct effective summaries, as compared to strawman baselines?*

![Objective Value Chart]

<table>
<thead>
<tr>
<th>Objective Value</th>
<th>Avg User (No Guidance)</th>
<th>Avg User (w/ Guidance)</th>
<th>Algo.</th>
<th>Baseline (TopS)</th>
<th>Baseline (TopN)</th>
</tr>
</thead>
<tbody>
<tr>
<td>D1 (5)</td>
<td>0.8</td>
<td>0.9</td>
<td>0.9</td>
<td>0.8</td>
<td>0.8</td>
</tr>
<tr>
<td>D1 (10)</td>
<td>0.8</td>
<td>0.9</td>
<td>0.9</td>
<td>0.8</td>
<td>0.8</td>
</tr>
<tr>
<td>D2 (5)</td>
<td>0.8</td>
<td>0.9</td>
<td>0.9</td>
<td>0.8</td>
<td>0.8</td>
</tr>
<tr>
<td>D2 (10)</td>
<td>0.8</td>
<td>0.9</td>
<td>0.9</td>
<td>0.8</td>
<td>0.8</td>
</tr>
<tr>
<td>D3 (5)</td>
<td>0.8</td>
<td>0.9</td>
<td>0.9</td>
<td>0.8</td>
<td>0.8</td>
</tr>
<tr>
<td>D3 (10)</td>
<td>0.8</td>
<td>0.9</td>
<td>0.9</td>
<td>0.8</td>
<td>0.8</td>
</tr>
<tr>
<td>D4 (5)</td>
<td>0.8</td>
<td>0.9</td>
<td>0.9</td>
<td>0.8</td>
<td>0.8</td>
</tr>
<tr>
<td>D4 (10)</td>
<td>0.8</td>
<td>0.9</td>
<td>0.9</td>
<td>0.8</td>
<td>0.8</td>
</tr>
<tr>
<td>D5 (5)</td>
<td>0.8</td>
<td>0.9</td>
<td>0.9</td>
<td>0.8</td>
<td>0.8</td>
</tr>
<tr>
<td>D5 (10)</td>
<td>0.8</td>
<td>0.9</td>
<td>0.9</td>
<td>0.8</td>
<td>0.8</td>
</tr>
<tr>
<td>AVG</td>
<td>0.8</td>
<td>0.9</td>
<td>0.9</td>
<td>0.8</td>
<td>0.8</td>
</tr>
</tbody>
</table>
Evaluation

Q2) How close do the summaries by users without guidance get to the algorithm results (in terms of normality, coverage, diversity, and overall objective value)?

[Bar chart showing the percentage (%) of algorithm's N, C, D, O achieved by each user for normality, coverage, diversity, and objective value.]

User 1 - User 5 and AVG User are compared. The chart shows that the users tend to achieve high normality and coverage, with diversity and objective value varying across users.
Q3) Alternative summarization by algorithmic guidance. How much guidance does our summarization algorithm provide users to derive alternative summaries and improve over their earlier results?

$$100 \frac{O(user\ after)}{O(user\ before)}$$

Percent % improvement in objective value by each user on each data/task after algorithmic guidance.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>User 1</td>
<td>112.59</td>
<td>156.44</td>
<td>99.53</td>
<td>114.31</td>
<td>129.89</td>
<td>130.58</td>
<td>92.20</td>
<td>106.17</td>
<td>170.86</td>
<td>121.08</td>
</tr>
<tr>
<td>User 2</td>
<td>91.79</td>
<td>118.14</td>
<td>87.56</td>
<td>102.86</td>
<td>99.19</td>
<td>112.31</td>
<td>92.66</td>
<td>100.00</td>
<td>107.39</td>
<td>117.97</td>
</tr>
<tr>
<td>User 3</td>
<td>101.60</td>
<td>112.95</td>
<td>101.30</td>
<td>120.73</td>
<td>140.15</td>
<td>101.75</td>
<td>85.78</td>
<td>96.60</td>
<td>199.57</td>
<td>142.96</td>
</tr>
<tr>
<td>User 4</td>
<td>103.98</td>
<td>104.18</td>
<td>100.85</td>
<td>140.65</td>
<td>103.76</td>
<td>105.94</td>
<td>116.86</td>
<td>124.73</td>
<td>110.13</td>
<td>109.13</td>
</tr>
<tr>
<td>User 5</td>
<td>117.61</td>
<td>124.02</td>
<td>102.70</td>
<td>129.06</td>
<td>169.17</td>
<td>117.77</td>
<td>105.06</td>
<td>106.17</td>
<td>113.34</td>
<td>109.65</td>
</tr>
<tr>
<td>Avg User</td>
<td>105.51</td>
<td>123.15</td>
<td>98.39</td>
<td>121.52</td>
<td>128.43</td>
<td>113.67</td>
<td>98.51</td>
<td>106.73</td>
<td>140.26</td>
<td>120.16</td>
</tr>
</tbody>
</table>
Evaluation

Q4) Efficiency. How long does it take per user on average to construct (i) a summary without guidance, and (ii) an alternative summary with guidance?

<table>
<thead>
<tr>
<th>Dataset Dataset</th>
<th>Without algo. guidance</th>
<th>With algo. guidance</th>
</tr>
</thead>
<tbody>
<tr>
<td>D1 (5)</td>
<td>313.2</td>
<td>183.4</td>
</tr>
<tr>
<td>D1 (10)</td>
<td>234.4</td>
<td>230.2</td>
</tr>
<tr>
<td>D2 (5)</td>
<td>263.8</td>
<td>215.8</td>
</tr>
<tr>
<td>D2 (10)</td>
<td>210.6</td>
<td>206.4</td>
</tr>
<tr>
<td>D3 (5)</td>
<td>276.6</td>
<td>218</td>
</tr>
<tr>
<td>D3 (10)</td>
<td>269.6</td>
<td>177.2</td>
</tr>
<tr>
<td>D4 (5)</td>
<td>175.4</td>
<td>166.6</td>
</tr>
<tr>
<td>D4 (10)</td>
<td>171</td>
<td>171</td>
</tr>
<tr>
<td>D5 (5)</td>
<td>212.6</td>
<td>162.4</td>
</tr>
<tr>
<td>D5 (10)</td>
<td>171.6</td>
<td>141.6</td>
</tr>
<tr>
<td>AVG</td>
<td>229.44</td>
<td>182.36</td>
</tr>
</tbody>
</table>

When algorithm results per task are also shown to the users, the avg. users spend anywhere from 2.5 to 5 minutes to build a summary without guidance. The longest time users take to summarize is 3.8 minutes on average. This is mainly for two reasons. Obviously, the imperfections of a simple baseline are apparent when the algorithm results are presented to each user, which can be seen as the warm-up period. Overall (AVG) is 3.8 minutes across tasks.
Summary

An end-to-end system for sensemaking of node-attributed networks

1. Circle extraction based on normality
2. Summarization wrt - quality, - coverage, and - diversity
3. Interactive interface for - exploration.

Discovering Communities and Anomalies in Attributed Graphs: Interactive Visual Exploration and Summarization  
Bryan Perozzi and Leman Akoglu  
ACM TKDD, 2018
This talk

- Attributed (sub)graphs*
  - Subgraphs [SIAM SDM’16]
  - Summarization [ACM TKDD’18]

Comparisons [WWW ’17]

* social circles, communities, egonetworks, …
Comparing attributed (sub)graphs

- **Motivating question:**
  Given a collection of attributed subgraphs from different classes, how can we discover the attributes that characterize their differences?

- **Hypothesis:** subgraphs from different classes exhibit *different* focus attributes

Crosswords  
Tea parties  
Gardening  

Class A subgraphs  
Selfies  
Partying  
Video Games  

Class B subgraphs
Problem Sketch

(a) Attributed graph

- Class A
  - $a_3$
  - $a_4$
  - $a_1$
  - $a_2$

- Class B
  - $a_5$
  - $a_6$

(b) Class A subgraphs

- $g_{A1}$
- $g_{A2}$
- $g_{A3}$

(c) Class B subgraphs

- $g_{B1}$
- $g_{B2}$

(d) Assignment & ranking

- Characterizing subspaces

- $g_{A1}$
- $g_{A2}$
- $g_{A3}$
- $g_{B1}$
- $g_{B2}$

- Attributes:
  - $a_1$, $a_2$, $a_3$, $a_4$, $a_5$, $a_6$
Characterization Problem: Formal

Given

- \( p \) attributed subgraphs \( g_1^+, g_2^+, \ldots, g_p^+ \) from class 1, \( S^+ \)
- \( n \) attributed subgraphs \( g_1^-, g_2^-, \ldots, g_n^- \) from class 2, \( S^- \)

from graph \( G \), and attribute vector \( \mathbf{a} \in \mathbb{R}^d \) for each node;

Find

- a partitioning of attributes to classes as \( A^+ \) and \( A^- \),
  where \( A^+ \cup A^- = A \) and \( A^+ \cap A^- = \emptyset \),
- focus attributes \( A_i^+ \subseteq A^+ \) (and respective weights \( \mathbf{w}_i^+ \)) for each subgraph \( g_i^+ \), \( \forall i \), and
- focus attributes \( A_j^- \subseteq A^- \) (and respective weights \( \mathbf{w}_j^- \)) for each subgraph \( g_j^- \), \( \forall j \);

such that

- total quality \( Q \) of all subgraphs is maximized, where
  \[ Q = \sum_{i=1}^p q(g_i^+ | A^+) + \sum_{j=1}^n q(g_j^- | A^-); \]

**Rank** attributes within \( A^+ \) and \( A^- \).
Reminder: Normality

- Normality as subgraph quality q:

\[ N = \mathbf{w}_c^T \cdot (\hat{x}_I + \hat{x}_X) \]

\[ \max N \quad \text{s.t.} \quad ||\mathbf{w}_c||_p = 1, \mathbf{w}_c(a) \geq 0, \forall a = 1, \ldots, d \]

- \( \mathbf{w}_c \):
  - **L_1 norm**: \( \mathbf{w}_c(a) = 1 \), one attribute with largest \( x \)
  - **L_2 norm**: \( \mathbf{w}_c(a) = \frac{x(a)}{\sqrt{\sum_{x(i) > 0} x(i)^2}} \), all attributes with positive \( x \)
Splitting attributes by class: intuition

Class A

Common Focus Attributes

Subgraphs

Attributes

Class B

Common Focus Attributes

Subgraphs

Attributes
Splitting attributes by class: intuition

- We don’t want attributes that are:
  - Relevant or irrelevant to both classes

Highly relevant to both. Not distinguishing.

Irrelevant to both. Not Interesting.
Splitting attributes by class: intuition

- We want attributes that are:
  - Relevant to **one** class & irrelevant to other(s)

A good attribute for class B

A good attribute for class A
Setting up the objective

Given a subset of attributes $S$, normality of subgraph $g$ is

$$N(g|S) = \sqrt{\sum_{a \in S} x(a)^2} = \|x[S]\|_2$$

- 2-norm of $x$ induced on the attribute subspace
- attribute weight vector of $g$
Setting up the objective

Quality of an attribute split is:

\[
\max_{A^+ \subseteq A, A^- \subseteq A} \frac{1}{p} \sum_{i \in S^+} ||x_i[A^+]||_2 + \frac{1}{n} \sum_{j \in S^-} ||x_j[A^-]||_2
\]

Such that \( A^+ \cap A^- = \emptyset \)

- \( p \) = number of subgraphs in class +
- \( n \) = number of subgraphs in class -
Setting up the objective

- **Quality of an attribute split is:**

  \[
  \max_{A^+ \subseteq A, A^- \subseteq A} \sum_{i \in S^+} \frac{1}{p} ||x_i[A^+]||_2 + \sum_{j \in S^-} \frac{1}{n} ||x_j[A^-]||_2
  \]

  Such that \( A^+ \cap A^- = \emptyset \)

- **Rank attributes by**

  \[
  rc(a) = \frac{1}{p} \sum_{i \in S^+} x_i(a) - \frac{1}{n} \sum_{j \in S^-} x_j(a)
  \]

  - Normalized contribution of \( a \) to Class +
  - Normalized contribution of \( a \) to Class -

  \( p = \) number of subgraphs in class +
  \( n = \) number of subgraphs in class -
Submodular Welfare Problem

- **Definition:**
  
  Given *d* items and *m* players having a *monotone* and *submodular* utility function \(w_i\) over subsets of items. Partition the *d* items into *m* disjoint sets \((I_1, I_2, \ldots, I_m)\) in order to maximize:
  
  \[
  \sum_{i=1}^{m} w_i(I_i)
  \]

- Our quality function \(N(g|S)\) is a *monotone* and *submodular* set function.

  \[
  w_c(I_c) = N(S^{(c)}|A^{(c)}) = \frac{1}{n^{(c)}} \sum_{k \in S^{(c)}} \| x_k[A^{(c)}] \|_2
  \]
Attribute splitting as SWP

- SWP is NP-hard
- First approx. factor is $\frac{1}{2}$ [Lehmann+, 2001]
- Improved to $(1 - \frac{1}{e})$ [Vondrák+, 2008]
- No better approximation unless
  - $P = NP$ [Khot+, 2008]
  - Using exponentially-many value queries [Mirrokni+, 2008]

$\Rightarrow$ [Vondrák+, 2008] is optimal approximation
Experiments

- Datasets
  - Congress Co-sponsorship Network
  - Amazon Co-purchase Network
  - DBLP Co-authorship Network

- Baseline (LASSO): L1-Regularized Logistic Regression
  - Positive weights are assigned to class A
  - Negative weights are assigned to class B
Congress Co-sponsorship

- Bills in Congress
  - each bill has a set of sponsors & policy area tag

- Attributed Graph:
  - Nodes: congressmen
  - Edges: co-sponsoring a bill
  - Attributes: policy areas of bills they sponsored:
    - National Security and Armed Forces
    - Environmental Protection
    - Foreign Affairs
    - ...

- Classes: party affiliation of congressmen
Liberal and Conservative Ideals

Democrats focus mostly on **social** programs

Republicans focus mostly on **governance** and **finance**
Focus Over Time

- 13 consecutive congress two-year cycles:

PARTY FOCUS ON ARMED FORCES

1993 1995 1997 1999 2001 2003 2005 2007

- Bombing of Iraq
- War in Afghanistan
- War in Iraq

Democratic: Blue
Republican: Red
Amazon.com Co-purchases

Attributed Graph:

- **Nodes**: Amazon videos
- **Edges**: being co-purchased together
- **Attributes**:
  - Product genre (Drama, Comedy, etc.)
  - Audience age range (e.g., 10-12 years)
  - Creators (e.g. Warner Video)
  - ...


Classes: Animation vs. Classic

- **Age range**
  - Kids & Family
  - 3-6 Years
  - 7-9 Years
  - 10-12 Years
- **Creator**
  - Warner Home
  - Warner Video
  - Cartoon Network
  - Bible
  - Christian Video
  - Bible Stories

**Ours**

**LASSO**

- **Franchise names**
  - Kids & Family
  - Educational
  - Dr. Seuss
  - 7-9 Years
  - Holidays
  - Dragon Tales
  - Infantil y familiar
  - Nickelodeon
  - Warner Video
  - Franklin

**Content**

**Animation**
- Performing Arts
- Comedy
- Musicals
- Drama
- Suspense
- Mystery
- Classic Comedy
- Ma & Pa Kettle
- Detectives
- Romance

**Classics**
- Drama
- Comedy
- Performing Arts
- Action
- Westerns
- Musicals
- Mystery
- French
- Sesame Street
- Puppets
Classes: Under 13 vs. Over 13

Attribute weight goes down as quality decreases

Ours

LASSO

Not much differentiation
Characterization vs. Classification

- Regularized linear classifiers (e.g. LASSO) can find
  - a sparse attribute subspace
  - coefficients for ranking
- How is our work different?

Classifiers focus on confidence while we focus on support
Characterization vs. Classification

Confidence

\[ Cf_d(c, a) = \Pr(c|a) = \frac{#(c, a)}{#(a)} \]

Prob. of belonging to class \( c \) if \( a \) is observed

Support

\[ Sup(c, a) = \frac{#(c, a)}{#(c)} \]

Portion of nodes in class \( c \) exhibiting \( a \)
Characterization vs. Classification

Class Confidence

\[ CC(c^+, a) = \Pr(c^+ | a) - \Pr(c^- | a) \]

Class Support

\[ CS(c^+, a) = Sup(c^+, a) - Sup(c^-, a) \]

Classifiers focus on **confidence** while we focus on **support**
Characterization vs. Classification

Characterizing Classes by Attributes and Social Ties

Aria Rezaei

Proposed Method vs. LR
Characterizing Class Differences in Attributed Graphs

Aria Rezaei, Bryan Perozzi, Leman Akoglu

Overview

We find differences in a set of co-occurrence matrices. Given the graph pair (two classes) to be compared, we find the attributes that characterize them the most. These attributes can be used to reason on the differences in behavior among the two groups.

Ties That Bind - Characterizing Classes by Attributes and Social Ties. Aria Rezaei, Bryan Perozzi, Leman Akoglu.
WWW 2017 Companion
This talk

- Attributed (sub)graphs*
  - Subgraphs [SIAM SDM’16]
  - Summarization [ACM TKDD’18]
  - Comparisons [WWW ’17]

* social circles, communities, egonetworks, …
References, Links to Code&Data:

- **Scalable Anomaly Ranking of Attributed Neighborhoods.** *Bryan Perozzi and Leman Akoglu.* SIAM SDM 2016  
  [https://github.com/phanein/amen](https://github.com/phanein/amen)

- **Discovering Communities and Anomalies in Attributed Graphs: Interactive Visual Exploration and Summarization.** *Bryan Perozzi and Leman Akoglu.* ACM TKDD, 2018  
  [https://www.dropbox.com/home/Public/iSCAN](https://www.dropbox.com/home/Public/iSCAN)

- **Ties That Bind - Characterizing Classes by Attributes and Social Ties.** *Aria Rezaei, Bryan Perozzi, Leman Akoglu.* WWW 2017 Companion  
  [https://github.com/rezaeia/AmenChar](https://github.com/rezaeia/AmenChar)
Subgraphs

Summarization

Comparisons

Democrats

- health
- families
- education
- commerce
- housing
- employment
- emergencies
- foreign trade
- environment
- criminal law

Republicans

- government
- taxation
- law
- employment
- public works
- natural resources
- congress
- finance
- commerce
- immigration

Thanks!

Contact: lakoglu@andrew.cmu.edu
www.andrew.cmu.edu/~lakoglu