Community Detection and Relational Clustering

Adam Phillippy
Samuel Huang
Community Detection
Community structure

- Identify a natural grouping
  - Many edges *within* groups
  - Few edges *between* groups

- Structure only

- No attributes

[Newman 2006]
Vs. Blockmodeling

- Blockmodels
  - Similar connections

- Communities
  - Interconnections

[Reichardt, Bornholdt 2006]
Betweenness

- Compute betweenness for each edge
  - All pairs shortest paths or random walk
  - Count traversals for each edge

- Remove edge with highest score

- Find components

- Repeat until desired number of communities

[Girvan, Newman 2002]
Modularity

- Difference from random model
- For a subset of nodes $s$ in $G$

  $Q_s = \sum_{ij} \left( A_{ij} - \frac{k_i k_j}{2m} \right)$

[Newman 2006]
Modularity

- For some binary grouping of nodes
  - Modularity of the grouping
    \[
    Q = \frac{1}{4m} \sum_{ij} \left( A_{ij} - \frac{k_i k_j}{2m} \right) (s_i s_j + 1)
    \]
  - Maximize the modularity \( Q \)
Forget ferromagnetism, spin, and all the other physics jargon. They are essentially the same thing.

\[ H = \sum_{ij} \left( A_{ij} - \gamma \frac{k_i k_j}{2m} \right) \delta(s_i, s_j) \]

\[ Q = \frac{1}{4m} \sum_{ij} \left( A_{ij} - \frac{k_i k_j}{2m} \right) (s_i s_j + 1) \]
Graph layouts
Spectral analysis

\[ G(n, m) \rightarrow A \]

\[ A v_i = \lambda_i I v_i \]

\[ \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n \]
Graph Laplacian

\[ L = D - A \]

\[
\begin{pmatrix}
2 & -1 & 0 & 0 & -1 & 0 \\
-1 & 3 & -1 & 0 & -1 & 0 \\
0 & -1 & 2 & -1 & 0 & 0 \\
0 & 0 & -1 & 3 & -1 & -1 \\
-1 & -1 & 0 & -1 & 3 & 0 \\
0 & 0 & 0 & -1 & 0 & 1
\end{pmatrix}
\]
Laplacian pros

- Symmetric, positive semidefinite
  - All eigenvectors are mutually orthogonal
  - All $n$ eigenvalues real and non-negative

- Nice for graphs
  - All rows sum to 0, $\therefore$ exists $\lambda_1=0$, $v_1=\{1,1,\ldots,1\}$
  - Multiplicity of $\lambda_1=0$ is number of components
  - $\lambda_2$ proportional to the graphs connectivity
  - $v_2$ Fiedler eigenvector used for spectral bisection

[Boccaletti et al 2006]
Laplacian eigenvectors
Modularity matrix

\[ B_{ij} = A_{ij} - \frac{k_i k_j}{2m} \]

- Reminiscent of Laplacian
  - Symmetric
    - All eigenvectors are mutually orthogonal
  - All rows sum to zero
    - Exists \( \lambda_1 = 0, \mathbf{v}_1 = \{1, 1, \ldots, 1\} \)
    - All other \( \mathbf{v} \) must contain both +/- elements

[Newman 2006]
Matrix form modularity

\[ Q = \frac{1}{4m} \sum_{ij} \left( A_{ij} - \frac{k_i k_j}{2m} \right) s_i s_j \]

\[ = \frac{1}{4m} s^T B s \]

\[ = \frac{1}{4m} \sum_{i=1}^{n} \left( u_i^T s \right)^2 \beta_i \]

[Newman 2006]
Maximizing the split

\[ Q \propto \sum_{i=1}^{n} (u_i^T \cdot s)^2 \beta_i \]

- Assume ordered eigenvalues
  \[ \beta_1 \geq \beta_2 \geq \cdots \geq \beta_n \]
- Aim to maximize
  \[ u_1^T \cdot s \]
  - Set \( s_i \) to +1 if \( u_{1,i} \) is positive.
  - Set \( s_i \) to −1 if \( u_{1,i} \) is negative.

[Newman 2006]
Finding $u_1$

- Power method
  - Iterative multiplication, normalization
  - Start with random $v$, stop on convergence
  
  \[ v_{k+1} = \frac{Bv_k}{\|Bv_k\|} \]

- Sparse matrix for speed

\[ Bv = Av - \frac{k(k^T \cdot v)}{2m} \]

[Newman 2006]
Karate club

[Newman 2006]
Recursive partitioning

- Split for greatest modularity gain
  - Recurse until modularity can’t be improved

[Girvan, Newman 2002]
Short list

- Spectral
  - Laplacian
  - Modularity

- $Q$ or $H$ Optimization
  - Simulated annealing
  - Extremal optimization

- Density
  - $(\alpha, \beta)$–clustering

- Edge removal
  - Betweenness
  - MinCut

- Greedy
  - Bottom up $Q$

- Cliques and Bicliques
  - Clique finding
Performance

- GN betweenness
  - $O(n^3)$
- CNM greedy
  - $O(n \log^2 n)$
- DA extremal
  - $O(n^2 \log^2 n)$
- Modularity matrix
  - $O(n^2 \log n)$

<table>
<thead>
<tr>
<th>Network</th>
<th>Size $n$</th>
<th>GN</th>
<th>CNM</th>
<th>DA</th>
<th>This article</th>
</tr>
</thead>
<tbody>
<tr>
<td>Karate</td>
<td>34</td>
<td>0.401</td>
<td>0.381</td>
<td>0.419</td>
<td>0.419</td>
</tr>
<tr>
<td>Jazz musicians</td>
<td>198</td>
<td>0.405</td>
<td>0.439</td>
<td>0.445</td>
<td>0.442</td>
</tr>
<tr>
<td>Metabolic</td>
<td>453</td>
<td>0.403</td>
<td>0.402</td>
<td>0.434</td>
<td>0.435</td>
</tr>
<tr>
<td>E-mail</td>
<td>1,133</td>
<td>0.532</td>
<td>0.494</td>
<td>0.574</td>
<td>0.572</td>
</tr>
<tr>
<td>Key signing</td>
<td>10,680</td>
<td>0.816</td>
<td>0.733</td>
<td>0.846</td>
<td>0.855</td>
</tr>
<tr>
<td>Physicists</td>
<td>27,519</td>
<td>—</td>
<td>0.668</td>
<td>0.679</td>
<td>0.723</td>
</tr>
</tbody>
</table>

[Newman 2006]
Fraction of correctly identified nodes

Proportion of out links $z_{out}/k$

[Danon et al 2005]
Overlapping and hierarchical communities

[Reichardt, Bornholdt 2006]
Community Reading List

Relational Clustering
What is Clustering?

- “A descriptive task that seeks to identify natural groupings in data.” (Neville et al. 2003)
- Partitioning data into non-overlapping groups
- Algorithms like k-means and hierarchical clustering are used
Problems?

- Data may be drawn from different distributions, and so normal clustering approaches fall short.
Problems?

- Relations between objects may give better insight into clusters
What about *Relational* Clustering?

- Data instances are not i.i.d. – relations of instances could affect clustering.
So what information should be used?

- Non-relational clustering uses object attributes, but not relations
- Relational clustering makes use of connections
- Neville et al. (2003) showed that using both attributes and links do better than either alone (except maybe in some cases where the linkage data is perturbed)
Some examples?

- Several examples of specific clustering situations
  - Semi-supervised Clustering
  - Co-clustering
  - Graph Clustering (Partitioning)
Semi-supervised Clustering

- Like unsupervised clustering, but with some pairwise constraints
  - Set $\mathcal{M}$ of “must link” constraints (same cluster), and set $\mathcal{C}$ of “cannot link” constraints (different clusters)
  - Pairwise constraints are more realistic than a partial set of class labels? (Basu et al. 2004)

- Many EM approaches

[Long et al, 2007]
Co-clustering

- aka Bi-clustering (Tri-clustering, …)
- Have two (or more) types of data, and want to cluster them all
- Potentially could use information about relations *between* data types
Graph Clustering (Partitioning)

- Joining nodes/Dividing graph
- Approaches mainly consist of edge cut objectives
  - Karger Min-Cut (Neville et al. 2003, Karger 1993)
  - Normalized cut (Shi, Malik 2000)
  - Ratio cut (Chan et al. 1993)
Are these related?

- Semi-supervised clustering: 1 type of data with multiple classes (clusters), and some constraint edges relating them

- Co-clustering: 2 (or more) types of data with multiple classes, with edges relating different types

- Graph clustering: 1 type of data, no features, set of edges relating data instances
How to formalize “general” relational data?

- Different types of data instances
  - Classes/Features for each type of data

- Different types of relationships
  - Homogeneous relations – between same type
  - Heterogeneous relations – between different types

[Long et al. 2007] Has all 3
Matrix Representation...

- ... of data objects
  \[ \mathcal{X}^{(1)} = \{ x_i^{(1)} \}_{i=1}^{n_1}, \ldots, \mathcal{X}^{(m)} = \{ x_i^{(m)} \}_{i=1}^{n_m} \]

- ... of latent clusters
  \[ \{ C^{(j)} \in \{0,1\}^{k_j \times n_j} \}_{j=1}^{m} \]

- ... of attributes
  \[ \{ F^{(j)} \in \mathbb{R}^{d_j \times n_j} \}_{j=1}^{m} \]

- ... of homogeneous relations
  \[ \{ S^{(j)} \in \mathbb{R}^{n_j \times n_j} \}_{j=1}^{m} \]

- ... of heterogeneous relations
  \[ \{ R^{(i,j)} \in \mathbb{R}^{n_i \times n_j} \}_{i,j=1}^{m} \]
Mixed Membership Relational Clustering (MMRC) \textit{from Long et al. (2007)}

- A Generative Model with...
  \[
  \Omega = \left\{ \{\Lambda^{(j)}\}_{j=1}^m, \{\Theta^{(j)}\}_{j=1}^m, \{\Gamma^{(j)}\}_{j=1}^m, \{Y^{(i,j)}\}_{i,j=1}^m \right\}
  \]
- ... membership parameters (clusters)
  \[
  \Lambda^{(j)} \in [0,1]^{k_j \times n_j}
  \]
- ... attribute distribution parameters
  \[
  \Theta^{(j)} \in \mathbb{R}^{d_j \times k_j}
  \]
- ... homogeneous relation dist. parameters
  \[
  \Gamma^{(j)} \in \mathbb{R}^{k_j \times k_j}
  \]
- ... heterogeneous relation dist. parameters
  \[
  Y^{(i,j)} \in \mathbb{R}^{k_i \times k_j}
  \]
Generating Samples for MMRC

- Sample from the distributions!

\[ C_{\cdot p}^{(j)} \sim \text{Multinomial}(\Lambda_{\cdot p}^{(j)}, 1) \]

\[ F_{\cdot p}^{(j)} \sim \Pr(F_{\cdot p}^{(j)} \mid \Theta^{(j)} C_{\cdot p}^{(j)}) \]

\[ S_{pq}^{(j)} \sim \Pr(S_{pq}^{(j)} \mid (C_{\cdot p}^{(j)})^T \Gamma_{\cdot p}^{(j)} C_{\cdot q}^{(j)}) \]

\[ R_{pq}^{(i,j)} \sim \Pr(R_{pq}^{(i,j)} \mid (C_{\cdot p}^{(i)})^T Y_{\cdot q}^{(i,j)} C_{\cdot q}^{(j)}) \]

- After sampling, we get

\[ \Psi = \{(C^{(j)})_{j=1}^m, (F^{(j)})_{j=1}^m, (S^{(j)})_{j=1}^m, (R^{(i,j)})_{i,j=1}^m\} \]
What’s going on here?

\[ \{ \Lambda \Theta \Gamma Y \} = \Omega \]

\[ = \Psi \]
Plate Notation

\[ \Lambda \rightarrow C \rightarrow \Theta \rightarrow F \]
\[ \Lambda \rightarrow C \rightarrow \gamma \rightarrow S \]
\[ \Lambda \rightarrow C \rightarrow \nu \rightarrow R \]

\[ N_i \]
Objective Function for MMRC

- “Just” \( \Pr(\Psi | \Omega) \):

\[
\Pr(\Psi | \Omega) = \prod_j \Pr(C^{(j)} | \Lambda^{(j)}) \times \prod_j \Pr(F^{(j)} | \Theta^{(j)} C^{(j)}) \\
\times \prod_j \Pr(S^{(j)} | (C^{(j)})^T \Gamma^{(j)} C^{(j)}) \times \prod_{i,j} \Pr(R^{(i,j)} | (C^{(i)})^T Y^{(i,j)} C^{(j)})
\]

- Use \( \mathcal{L}(\Omega | \Psi) = \Pr(\Psi | \Omega) \) as likelihood function
- Use log–likelihood for (relative) ease
MMRC Algorithms

- Two algorithms
  - Soft relational clustering (determine $\Lambda$)
  - Hard relational clustering (throw out the $\Lambda$ parameters, just determine $C$)

- Both use EM
  - E–step: Refine values of $C$
    - Soft: Posterior probability for $C$ (next slide)
    - Hard: Reassign clusters to maximize objective function
  - M–step: Refine values for parameters:
    - Soft: $\Omega = \{\Lambda(j)\}, \{\Theta(j)\}, \{\Gamma(j)\}, \{Y(i,j)\}$
    - Hard: $\Omega = \{\Theta(j)\}, \{\Gamma(j)\}, \{Y(i,j)\}$
Closer look: Monte Carlo E-step for Soft Algorithm

- Get posterior probability of $C$
  \[ \Pr\left(\{C^{(j)}\} \mid \{F^{(j)}\}, \{S^{(j)}\}, \{R^{(i,j)}\}, \Omega\right) \]
- ... hard to do if we have more than one class: joint distributions can make intractable
- Can use approximation techniques
  - Monte Carlo
  - Belief propagation
- Long et al. (2007) use Gibbs sampling (type of Monte Carlo approach)
Metropolis–Hastings algorithm: A general case of Gibbs sampling

Wikipedia:
http://en.wikipedia.org/wiki/Metropolis–Hastings_algorithm
A few comments about MMRC Algorithms

- Complexity is $O\left( t m n^2 k \right)$ …
  - … where $t$ is the # of iterations
  - … where $m$ is the # of data types
  - … where $n$ is the max # of objects of one data type
  - … where $k$ is the max # of classes for one data type

- Same as k-means (Long et al. 2007)
  - … although maybe not, since k-means was proven to be lower-bounded by $2^{\Omega(\sqrt{n})}$ (Arthur, Vassilvitski 2006) and Long et al. don’t go into details
Semi-supervised Clustering using MMRC

- Only one type of data
  - No R matrices
  - $\mathcal{M}$ – “must link” constraints
  - $\mathcal{C}$ – “cannot link” constraints

$$S_{pq} = \begin{cases} 
  f_M(x_p, x_q) & \text{if } (x_p, x_q) \in \mathcal{M} \\
  f_C(x_p, x_q) & \text{if } (x_p, x_q) \in \mathcal{C} \\
  0 & \text{otherwise}
\end{cases}$$

- $f_M$ and $f_C$ penalize violations of constraints
Co-clustering using MMRC

- Two types of data
  - One $R$ matrix
  - No $S$ matrices

- We really only care about $\Pr(R | (C^{(1)})^T Y C^{(2)})$
Graph Clustering (Partitioning) using MMRC

- Single type data
  - One homogeneous relation matrix $S$
  - No $R$ matrices
  - No feature matrices $F$

- Concerned with
  $$\Pr(S \mid (C^{(1)})^T \Gamma C^{(1)})$$
Relational Clustering References

- Probabilistic Clustering in Relational Data. B. Taskar, E. Segal, and D. Koller. Seventeenth International Joint Conference on Artificial Intelligence (IJCAI01), Seattle, Washington, August 2001.
- Clustering Relational Data. Batagelj, V. and Ferligoj Anuska.