Community Detection: A Bayesian Approach and the Challenge of Evaluation

Jon Berry
Danny Dunlavy
Cynthia A. Phillips
Dave Robinson
(Sandia National Laboratories)

Jiqiang Guo
Dan Nordman
(Iowa State University)

Alyson Wilson (IDA Science & Technology)
Analysis of Massive Graphs

- Finding communities
  - Subgraphs where nodes are more connected to each other than to the rest of the graph
- Exploring relationships between individuals
- Finding patterns (normal/abnormal)

- Power law degree distribution common

[Graph showing power law distribution with log-log scale]

Twitter social network ($|V|\approx 200M$)

[Clauset, 2007]

[Akshay Java, 2007]
Community Detection

One approach: Treat as an optimization problem
Metrics: modularity, conductance
Lots of algorithms: CNM, wCNM, Bader-McCloskey, Louvain, Ruan and Zhang, .... (hundreds of papers in physics and CS)

Some complications:
- How can we tell if communities are significant?
  - Distinguish from random fluctuations
  - Not resolved enough? Too resolved?
- Hierarchy
- Overlapping

Relational Data Mining: Our Goal

Issues for finding structure in social and biological networks
- Lack of rigorous measures for optimization
  - No graph theoretic function seems to always capture what humans perceive are the best communities
- Inherent randomness
- Uncertainty in observations

Makes it difficult to
- Judge quality of a solution
- Compare solutions/methods

Goal:
- Statistically informed algorithms
- Benchmarks
- Means to compare algorithms
Bayesian Approach

- **Prior distribution**: model what we know about a specific application
  - Probabilistic model of community distribution
  - Probabilistic models for edges
    - How do the communities influence existence of edges?
    - How do we observe the data?
  - Goals:
    - Socially justified
    - Properties of real-world graphs: heavy-tailed degree distribution, clustering coefficients, ...
    - Closed form for conditional probabilities
- **Update the distribution** based on an observation (= graph)
  - Given the graph, what is the probability distribution of the communities given our observation? How likely is a particular community partition given the topology?
- With **posterior distribution** make inferences about the data
Community Distribution

- \( P(V_1, V_2, \ldots, V_k) \), \( k \) generally not fixed
- \( P(c_1, c_2, \ldots, c_n) \), \( c_i \) community label for node \( i \)
- \# of partitions is huge
  - Can’t use a discrete distribution
- Use a model that produces a partition and has a small number of parameters \( B \)

Bayesian Approach

- Prior distribution \( P(c_1, c_2, \ldots, c_n | B) \) and \( P(B) \)
- Posterior distribution \( P(c_1, c_2, \ldots, c_n, B | G) \) for data graph \( G \) (edges)
  - Not a point model (e.g., Hoffman and Wiggins (2008))

- Graph distributions based on properties or generative models
  usually don’t have closed forms for these conditional probabilities
Gibbs Sampling

- If can’t compute a posterior distribution, sometimes can **sample** it
- Markov Chain Monte Carlo
  - Start with an initial community assignment and parameters $B$
  - Iterate:
    - Update $B$ by sampling from $P(B \mid c_1, c_2, \ldots, c_n)$
    - Update communities from $P(c_1, c_2, \ldots, c_n \mid B)$
      - We update each $c_i$ serially
      - Proof of correctness for our choice of prior
  - Updates use the edges
- Run for a burn-in period to forget initial state
- Run for long enough to mix, then save a sample
  - Empirical testing (new results Ray, Pinar, Seshadhri)
- Gives a set of communities $C_1, C_2, C_3, \ldots, C_s$ sampled uniformly from the posterior distribution
What to do with samples from the Posterior

- Compute probability estimates of arbitrary properties
  - \( P(n_i \text{ and } n_j \text{ in same community}) \)
- Find a “consensus community” or “average community” as a suggested community partition
  - One possible “average” is community assignment that minimizes Lau and Green loss function: deviation from posterior expectation (\( X \) is set of edges in given \( G \)):

\[
E(L(\pi, \hat{\pi}) \mid X) = \sum_{(i,j) \in M} \left( aPr \{ c_i = c_j \mid X \} \cdot 1_{[c_i \neq c_j]} + bPr \{ c_i \neq c_j \mid X \} \cdot 1_{[c_i = c_j]} \right)
\]

- Could score community assignments from other algorithms based on posterior probabilities such as loss function
A Simple Prior: Chinese Restaurant Process

A model for generating partitions of vertices

• Restaurant, infinite number of tables
• First customer sits at first table
• The \((n+1)\)st customer’s table is probabilistic:

\[
\begin{align*}
\Pr(\text{table } k \mid \text{previous } n \text{ customers}) &= \frac{n_k}{\alpha + n} \\
\Pr(\text{new table} \mid \text{previous } n \text{ customers}) &= \frac{\alpha}{\alpha + n}
\end{align*}
\]

Where \(n_k\) is number of customers already seated at table \(k\).

\[
\Pr(c_1, c_2, \ldots, c_n) = \frac{\alpha^{K-1} \prod_{k=1}^{K} (n_k - 1)!}{\prod_{i=1}^{n-1} (\alpha + i)}
\]

when \(K\) tables opened

• Note: CRP is moderately flexible. Prior influence will fade in sampling if it is not a good match
Edges from partitions

• Edges between two nodes in the same community, probability $p_{in}$
• Edges across communities, probability $p_{out}$
• $p_{in} > p_{out}$
• Generalizations: $p_{in}$ per community and $p_{out}$ per community pair

• A simple, tractable start. Other things influence edges.
Priors Distributions for Parameters

\[ p_{in} \sim \text{Beta}(2,1) \quad p_{out} \sim \text{Beta}(1,2) \quad \alpha \sim \Gamma(1,1) \]

- This starts \( p_{in} \) with mean 2/3 and \( p_{out} \) with mean 1/3. Range (0,1)
- Beta frequently represents lack of knowledge
  - Spreads probability (mushy with a peak)
- \( \Gamma(1,1) \) has most probability low (mean 1), low, steady prob to \( \infty \)
CRP MCMC

- Update $p_{\text{in}}$ and $p_{\text{out}}$ with altered beta distributions
- Compute probability of each community for a vertex to move to based on current communities and (intra- inter-) edge counts
  - Break into pieces related to current node and the rest
  - Uses exchangeability
- Update $\alpha$ using beta and gamma distributions

- Proof that this converges to the posterior distribution
Example Data Set: Football Schedule

- NCAA Division 1-A, 115 teams, 613 games, Fall 2000
Football Data: CRP algorithm
Modified Model for Football Data

• Model had trouble with the independent schools
• Add an “anti-social” table
  - Members not in any community (act ER)
  - Constant probability of joining this table

• Extended procedures and proofs

• Could we have a celebrity table?
  - High degree, no community
Football example with Antisocial Table
Experimental Data

- Moving beyond Karate club and Girvan-Newman

- There are many generative models for social graphs
  - Preferential attachment
  - Kronecker products
    - RMat
  - Forest fire
  - Hierarchical scale-free
  - LFR
  - BTER
LFR Graph Generator

- Parameter $\mu$, a goal for fraction of inter-community edges
- Draw community sizes iid from a distribution (usually heavy tail)
- Draw a degree for each node (usually heavy tail)
- Assign nodes to communities
  - Put node in communities large enough to absorb internal edges:
    for node of degree $d_v$, community size at least $(1 - \mu)d_v + 1$
- Rewire to achieve desired intra-community density on average.

BTER Generator

- Accepts a degree distribution
- Group nodes into communities approximately $d$ nodes of degree $d$
  (degree-1 and high-degree somewhat different)
- Connect community $k$ in Erdos-Renyi graph with

$$\rho_k = \rho \left[ 1 - \eta \left( \frac{\log(\bar{d}_k + 1)}{\log(d_{\text{max}} + 1)} \right)^2 \right]$$

$$\bar{d}_k = \text{min degree of a node in community } k \text{ and } d_{\text{max}} \text{ is max overall}$$

- Connect communities Chung-Lu style using remaining expected degree


Slide 19
Modularity

- A measure of community strength

\[ Q = \sum_s e_{ss} - a_s^2 \]

- Measures how much denser a community is compared to a random graph with same degree distribution
- Modularity of a whole partition is sum of individual modularities
Some Community Detection Algorithms

• CNM (Clauset, Newman, Moore): agglomerative
  - Start with nodes isolated
  - Merge to maximize modularity. Stop when no merge helps
• wCNM (Berry et al):
  - weight edges based on evidence endpoints belong in same community, then CNM
  - Overcomes resolution limit
• Louvain (Blondel et al):
  - Start with nodes in own community.
  - Each node in turn moves to a neighbor’s community to maximize change in modularity (stay if no improvement).
  - Iterate to local max. We do one major round.
• Didn’t consider Bayesian from Hofman and Wiggins (2008) which requires # of communities as a parameter
• A real horse race would require larger field, but Louvain popular
Comparing community assignments

- Jaccard index: fraction of conserved intra-community edges
- Edit distance
- Normalized mutual information:
  - Mutual information $I$, Entropy $H$
  - $NMI = I/H$
  - $X, Y$ are two partitions
  - How far are the two distributions from being independent
- $H$-index.
  - $x(v)$ is community for vertex $v$ in one partition, $y(v)$ is community for $v$ in other partition

\[
I = \sum_{y \in Y} \sum_{x \in X} p(x, y) \log \left( \frac{p(x, y)}{p(x) p(y)} \right)
\]

\[
H = \sum_{y \in Y} \sum_{x \in X} p(x, y) \log(p(x, y))
\]

\[
\frac{|x(v) \cap y(v)|}{\max(|x(v)|, |y(v)|)}
\]
First Experiments

- Quality before scalability
- Reasonably easy instance
- LFR generator: 1000 vertices, uniform degree of 8, community sizes between 10 and 15. So ~80 communities.
  - 10 instances for each value of \( \mu \) from .1 to .7
- BTER: varied \( \rho \) to give different densities
  - When smallest about .9 dense, ~67% of edges intra-community
  - When smallest about .5 dense, ~35% of edges intra-community
  - 1000 vertices, degree distribution from LFR with exponent 2
- CRP: 100 iterations for mixing (settles sooner)
Normalized Mutual Information

![Normalized Mutual Information Graph](attachment:image.png)

- CNM
- wCNM
- LOUVAIN
- SINGLETON
- CLUMP
- CRP
Normalized Mutual Information one-sided

\[ I = \sum_{y \in Y} \sum_{x \in X} p(x,y) \log \left( \frac{p(x,y)}{p(x)p(y)} \right) \]

Easier to think with counts: \( C_x = \) nodes in community \( x \), \( n_x = |C_x| \).

\[ p(x,y) = \frac{|C_x \cap C_y|}{n} \quad p(x) = \frac{n_x}{n} \quad \frac{p(x,y)}{p(x)p(y)} = \frac{n|C_x \cap C_y|}{n_x n_y} \]

- If all nodes in a single clump (\(|Y|=1\)):
  \[ \frac{n|C_x \cap C_y|}{n_x n_y} = \frac{n(n_x)}{n_x n} = 1, \text{ so NMI is 0 (log1 = 0)} \]

- If all nodes are in separate communities (\(|Y| = n\))
  \[ \frac{n|C_x \cap C_y|}{n_x n_y} = \frac{n(1)}{n_x (1)} = \frac{n}{n_x}, \text{ so less penalty for over-resolution} \]
NMI for singletons can approach 1

- Dunbar number (~150): meaningful human communities constant size
- If ground truth is \( k \) communities of size \( n/k \)

\[
I = \sum_{y \in Y} \sum_{x \in X} p(x, y) \log \left( \frac{p(x, y)}{p(x)p(y)} \right) = \sum_{y \in Y} \sum_{x \in X} \left( \frac{1}{n} \log \frac{n}{k} \right) = \log n - \log k
\]

\[
H = -\sum_{y \in Y} \sum_{x \in X} p(x, y) \log (p(x, y)) = \sum_{x \in X} \sum_{y \in Y} \left( \frac{1}{n} \log \frac{1}{n} \right) = -\log n
\]

So \( \text{NMI} = 1 - \frac{\log k}{\log n} \)
H-index

H-index values for LFR $\mu=0.5$

- Green: CRP
- Blue: Louvain

Vertex rank vs. H-value plot.
H-Index: Combined (not statistically justified)

Slide 29
Normalized Mutual Information for BTER Instances

Note: LFR with same degree sequence and similar community size distribution does *not* distinguish algorithms
H-Index for BTER Instances
Variation of Information (Meila, 2007)

\[ VI(X;Y) = H(X) + H(Y) - 2I(X,Y) \]
Concluding remarks

- Parallel Gibbs sampling: promising initial results for simple problems
- H-index is not the right answer
  - Investigating other community comparison measures that satisfy useful axioms