Community detection by pseudo-likelihood

Liza Levina

Department of Statistics, University of Michigan

Joint work with Arash Amini (U Michigan), Aiyou Chen (Google), and Peter Bickel (UC Berkeley)
Community detection

- Most common interpretation: many links within and few links between
- There are also disassortative communities, bipartite graphs, etc.
- A large number of methods have been proposed
  - Greedy heuristic algorithms
  - Global criteria to optimize over all partitions, often with spectral approximations
  - Fitting probabilistic models with communities: stochastic block model and extensions
The stochastic block model

Holland et al (1983)

- Each node $i$, $i = 1, \ldots, n$ is independently assigned a community label $c_i \in \{1, \ldots, K\}$, with probabilities $\pi = (\pi_1, \ldots, \pi_K)^T$.
- Given node labels $c = (c_1, \ldots, c_n)$, the edges $A_{ij}$ are independent Bernoulli (or Poisson) random variables with

$$EA_{ij} = P_{c_i c_j}$$

where $P = [P_{ab}]$ is a $K \times K$ symmetric matrix.
- $K = 1$: the Erdos-Renyi graph (all edges form independently with probability $p$)
Challenges in fitting the block model

- There are $K^n$ possible label assignments $(\hat{c}_1, \ldots, \hat{c}_n)$; optimization is in principle NP-hard.
- Estimation is hard for sparse networks.
  - **Weak consistency** of labels requires expected degree $\lambda_n \to \infty$:
    \[
    \frac{1}{n} \sum_{i=1}^{n} 1(\hat{c}_i = c_i) \to 1 \text{ as } n \to \infty
    \]
  - **Strong consistency** of labels typically requires $\lambda_n / \log n \to \infty$
    \[\mathbb{P}[\hat{c} = c] \to 1 \text{ as } n \to \infty\]
- However: given labels $c_i$, estimating parameters $\pi$ and $P$ is trivial.
Algorithms for fitting the block model

- **MCMC**: Snijders & Nowicki (1997, 2001), up to $\sim 10^2$ nodes.
- **Profile likelihood** (Bickel & Chen 2009): profile parameters and maximize over label assignments by greedy search, up to $10^3 – 10^4$ nodes.
- **Method of moments** (Bickel, Chen, L. 2012): counts patterns and matches to their expectations. General but slow.
- **Belief propagation** (Decelle et al, 2012)
- **Pseudo-likelihood**: (Amini, Chen, Bickel, L. 2012)
Pseudo-likelihood: original proposal


- Suppose the observations $x_1, x_2, \ldots, x_n$ are true pixel values in an image, $y_1, \ldots, y_n$ observed noisy versions
- $p(x|y)$ is extremely complicated and not tractable
- Instead look at $p(x_i|y_{N(i)})$ where $N(i)$ is a neighborhood of $i$ and assume a Markov random field model with
- $p(x_{N(i)}|x_i) = \prod_{j \in N(i)} p(x_j|x_i)$
Pseudo-likelihood: current usage

- Generally refers to simplifying an intractable likelihood by assuming some independencies
- No general theory; results have been obtained in specific cases.

**Examples:** instead of the true log-likelihood \( \log p(x_1, \ldots, x_n) \), maximize

- sum of marginal log-likelihoods
  \[ \sum_i \log p(x_i) \]
- sum of conditional log-likelihoods
  \[ \sum_i \log p(x_i | x_{\{\neg i\}}) \]
- composite log-likelihood
  \[ \sum_{i<j} \log p(x_i, x_j | x_{\{\neg i, \neg j\}}) \]
Pseudo-likelihood for fitting the block model

- Start with an initial value \( e \) for labels \( c \)
- Apply “block compression”:

\[
b_{ik} = b_{ik}(e) = \sum_{j=1}^{n} A_{ij} 1\{e_j = k\}
\]

which gives us a \( n \times K \) matrix \( B \) instead of a \( n \times n \) matrix \( A \).
- Example: \( n = 5, K = 2, e = \{1, 2, 1, 2, 2\} \)

\[
A = \begin{bmatrix}
0 & 1 & 0 & 1 & 0 \\
0 & 0 & 1 & 0 & 1 \\
0 & 1 & 0 & 0 & 1 \\
1 & 1 & 1 & 0 & 0 \\
1 & 0 & 1 & 1 & 1 \\
\end{bmatrix}
\]

\[
B = \begin{bmatrix}
0 & 2 \\
1 & 1 \\
0 & 2 \\
2 & 1 \\
2 & 2 \\
\end{bmatrix}
\]
Consider the rows of $B$, $b_i = (b_{i1}, \ldots, b_{iK})$.

Conditional on $c_i = \ell$, $(b_{i1}, \ldots, b_{iK})$ are mutually independent.

Assume further

1. $\{b_i\}$ are independent across different rows $i$, that is, ignore the symmetry of $A$.

2. $b_{ik}$ follows the Poisson distribution (a sum of independent Bernoulli variables).
The Pseudo-likelihood

- Let $\lambda_{\ell k} = E[b_{ik} \mid c_i = \ell]$ (a function of $e$, $c$, and $P$).
- The log of pseudo-likelihood is given by

$$
\ell_{PL}(\pi, \Lambda; b_{ik}) = \sum_{i=1}^{n} \log \left( \sum_{\ell=1}^{K} P(c_i = \ell) \frac{P(b_{i1}, \ldots, b_{iK} \mid c_i = \ell)}{\prod_{k=1}^{K} \lambda_{\ell k} b_{ik}^{\ell - 1}} \right),
$$

- This is the log-likelihood of a mixture of vectors with Poisson components, which can be fitted by a standard Expectation-Maximization (EM) algorithm.
- Once EM converges, use estimated $\hat{c}$ to estimate $\pi$, $P$ in the block model.
EM for mixture models

- A standard algorithm for maximizing likelihoods of partially observed data (here $b_{ik}$ is observed, $c_i$ is not observed) - see Dempster (1972).

- Given an initial value for parameters, iterate till convergence:
  1. E-step: compute the expected full likelihood (integrate out the unobserved data)
  2. M-step: update parameter estimates by maximizing the expected likelihood from the E-step

- In the context of networks, see Newman & Leicht (2007) (similar to setting our blocks size to 1).
The pseudo-likelihood algorithm for block models

Initialize labels \( e \), and parameters \( \pi, P, \lambda \). Then repeat \( T \) times:

1. Compute the block sums \( \{b_{i\ell}\} \).

2. Estimate probabilities for node labels:

\[
\hat{\pi}_{i\ell} = P(c_i = \ell \mid b_i) = \frac{\hat{\pi}_\ell \prod_{m=1}^K \exp(b_{im} \log \hat{\lambda}_{\ell m} - \hat{\lambda}_{\ell m})}{\sum_{k=1}^K \hat{\pi}_k \prod_{m=1}^K \exp(b_{im} \log \hat{\lambda}_{km} - \hat{\lambda}_{km})}.
\]

3. Update parameter values:

\[
\hat{\pi}_\ell = \frac{1}{n} \sum_{i=1}^n \hat{\pi}_{i\ell}, \quad \hat{\lambda}_{\ell k} = \frac{\sum_i \hat{\pi}_{i\ell} b_{ik}}{\sum_i \hat{\pi}_{i\ell}}.
\]

4. Return to step 2 unless the parameter estimates have converged.

5. Update labels by \( e_i = \arg\max_\ell \hat{\pi}_{i\ell} \) and return to step 1.

6. Update \( \hat{P} \) and \( \hat{\pi} \).

\[O(T_{outer}(|E| + nK^2 T_{inner}))\]
Consistency of Pseudo-likelihood

- Only shown for $K = 2$
- Condition on the true labels $c$ (treat them as deterministic)
- Assume the initial labeling $e$ matches $c$ on a certain fraction $\gamma_k$ of nodes in community $k$ ($\gamma_k$'s are unknown and values unimportant)
- Can then show weak consistency $\frac{1}{n} \sum_{i=1}^{n} 1(\hat{c}_i = c_i) \xrightarrow{P} 1$ as $n \to \infty$ under a set of conditions for which degree $\lambda_n \to \infty$ is sufficient.
- For example, for

$$P_n = \frac{\lambda_n}{n} \begin{bmatrix} a & b \\ b & a \end{bmatrix}$$

it is sufficient to have

$$\lambda_n \frac{(a - b)^2}{a + b} \to \infty$$

which holds if $a \neq b$ are constants, $\lambda_n \to \infty$. 

- Addresses a limitation of the block model: expected degree within one community is the same for all nodes, which does not allow for "hubs"
- Each node is associated with a degree parameter $\theta_i$, and

$$E(A_{ij}) = \theta_i \theta_j P_{c_i c_j}.$$ 

- $K = 1$: the expected degree random graph, a.k.a. configuration model or the Chung-Lu model. All edges form independently with $P(A_{ij} = 1) \propto \theta_i \theta_j$. 
The political blogs example

- node size in the plot is proportional to log degree

Block model

Degree-corrected block model
Profile likelihood (Karrer & Newman 2010): profile parameters, maximize over label assignments by greedy search

Recent improvements, including spectral methods

Can we apply pseudo-likelihood? Sort of: condition on node degrees $d_i = \sum_{j=1}^{n} A_{ij}$.

Still work with block sums $b_i = (b_{i1}, \ldots, b_{iK})$, conditioning on $\sum_{k=1}^{K} b_{ik} = d_i$. 

Fitting the degree-corrected model
Conditional pseudo-likelihood (CPL)

- Assume $b_{ik} \sim \text{Poisson}(\lambda_{\ell k})$ and define
  \[ \theta_{\ell k} = \frac{\lambda_{\ell k}}{\sum_j \lambda_{\ell j}} \]

- Conditional on $c_i = \ell$ and $\sum_{k=1}^{K} b_{ik} = d_i$, $b_i$ follows the multinomial distribution with parameters $d_i$, $(\theta_{\ell 1}, \ldots, \theta_{\ell K})$.

- The pseudo-likelihood (ignoring dependence among rows again):
  \[ \ell_{\text{CPL}}(\pi, \theta; \{b_i\}) = \sum_{i=1}^{n} \log \left( \sum_{\ell=1}^{K} \pi_{\ell} \prod_{k=1}^{K} \theta_{\ell k}^{b_{ik}} \right) \]
  which is the likelihood of a mixture of multinomials – fit by EM.
Initialization

Initial label assignments matter: compare initialization choices.

- **Spectral clustering (SC)** based on the normalized Laplacian $D^{-1/2} AD^{-1/2}$

- **Spectral clustering with perturbations (SCP):** a new algorithm of independent interest which we found to outperform regular spectral clustering for sparse networks.

- **Degree clustering (DC):** a “naive” approach clustering nodes based on the pair (degree, 2-degree).
Spectral clustering with perturbations

- SC tends to fail when there are many small components
- Regularize by replacing $A$ with

$$A + \alpha \frac{\lambda}{np} E$$

where $E$ is an Erdos-Renyi graph with edge probability $p$, $\alpha$ a small constant, $\lambda$ expected degree of $A$

- Empirical results are essentially the same for all $p > 0.1$
- For computational efficiency, set $p = 1$, $\alpha = 0.25$, i.e. add a small constant to all entries of $A$
- Then perform regular SC on the “perturbed” matrix
Simulations: settings

- Fix $K = 3$, $\pi = (1/3, 1/3, 1/3)$.
- Degree-corrected block model: $P(A_{ij} = 1) \propto \theta_i\theta_j P_{c_i c_j}$
- $\theta_i$ iid, $\rho = 0$ or $0.9$,

$$\theta_i = \begin{cases} 1 & \text{w/prob } \rho \\ 5 & \text{w/prob } 1 - \rho \end{cases}$$

- Fix out-in-ratio $\beta$ and weight vector $w = (w_1, w_2, w_3)$, set

$$P_0 = \begin{bmatrix} w_1 & \beta & \beta \\ \beta & w_2 & \beta \\ \beta & \beta & w_3 \end{bmatrix}$$

- Fix expected degree $\lambda$ and rescale

$$P = \lambda \left[ n(\pi^T P^{(0)} \pi)(E\theta)^2 \right]^{-1} P_0$$
Simulations: varying the expected degree $\lambda$

$\lambda = \begin{cases} \lambda = (1, 1, 1) & \rho = 0.0 \\ \lambda = (1, 5, 10) & \rho = 0.0 \\ \lambda = (1, 1, 1) & \rho = 0.9 \\ \lambda = (1, 5, 10) & \rho = 0.9 \end{cases}$

- $\beta = 0.05$
- $n = 1200$
- $K = 3$
- $N_{rea} = 1000$

$\text{DC}$, $\text{CPL [DC]}$, $\text{UPL [DC]}$, $\text{SC}$, $\text{SCP}$, $\text{CPL [SCP]}$, $\text{UPL [SCP]}$
Simulations: varying the out-in-ratio $\beta$

- $w = (1, 1, 1)$
- $\rho = 0.0$
- $\lambda = 7$
- $n = 1200$
- $K = 3$
- $N_{rea} = 1000$

- $w = (1, 5, 10)$
- $\rho = 0.0$

- $w = (1, 1, 1)$
- $\rho = 0.9$

- $w = (1, 5, 10)$
- $\rho = 0.9$
Computational complexity

$T$ is running time (seconds)

$\lambda = 7.00$, $\rho = 0.0$, $\beta = 0.05$, $K = 3$, $N_{\text{rea}} = 8$, $w = [1 \ 1 \ 1]$
Political blogs data
Pseudo-likelihood is a fast, accurate, and asymptotically consistent method for fitting block models, and can handle variation in node degrees by conditioning.

In general, simplifying likelihoods can bring large computational gains.

Small perturbations are a good way to deal with very sparse networks.