Community detection on block models with geometric kernels

B R Vinay Kumar
Joint work with
Konstantin Avrachenkov and Lasse Leskelä

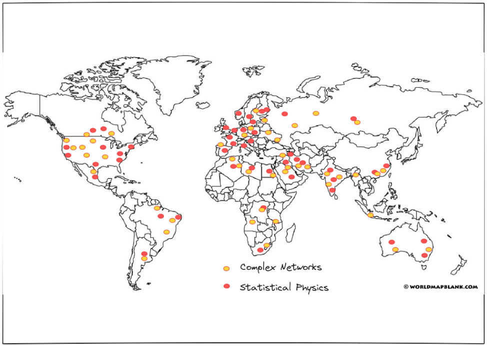
Workshop on Modelling and Mining Networks (WAW 2024)
June 6, 2024
Warsaw, Poland

Motivation

- Networks exhibiting geometric structure.
- Social networks: friends of friends are friends

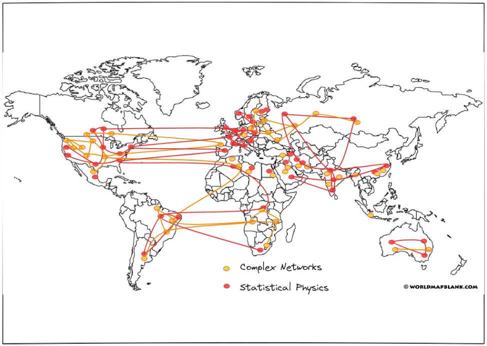
Motivation

- Networks exhibiting geometric structure.
- Social networks: friends of friends are friends
- Collaboration networks



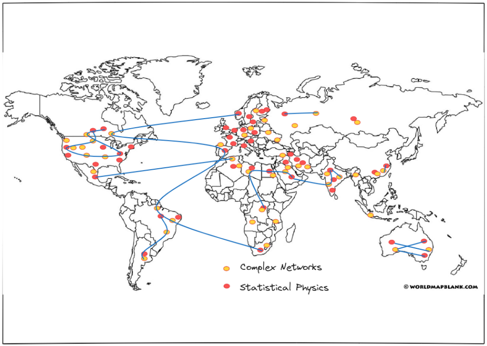
Motivation

- Networks exhibiting geometric structure.
- Social networks: friends of friends are friends
- Collaboration networks



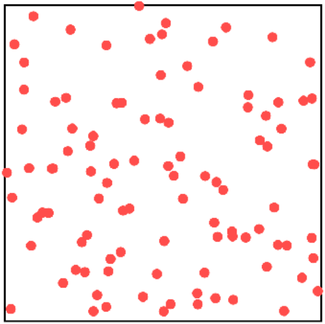
Motivation

- Networks exhibiting geometric structure.
- Social networks: friends of friends are friends
- Collaboration networks



Model

$$\mathbf{S} = \left(\frac{-1}{2}, \frac{1}{2}\right]^d$$

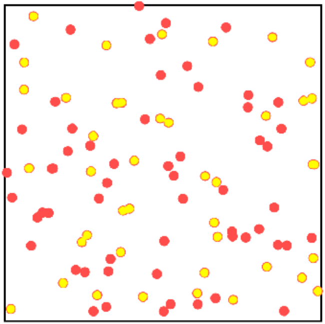


$N \sim \mathrm{Poi}(\lambda n)$

- Poisson point process $\mathbf{X} = (X_u)_{u=1}^N$ of intensity λn.

Model

$$\mathbf{S} = \left(\frac{-1}{2}, \frac{1}{2}\right]^d$$



$$N \sim \text{Poi}(\lambda n)$$

- Poisson point process $\mathbf{X} = (X_u)_{u=1}^N$ of intensity λn.
- Two communities: $\boldsymbol{\sigma} = (\sigma(1), \cdots, \sigma(N))$

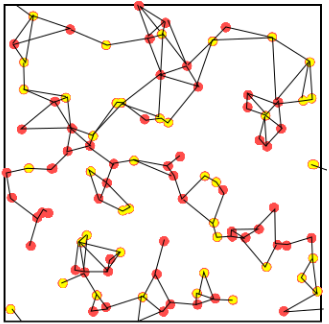
$$\mathbb{P}(\sigma(u) = +1) = \mathbb{P}(\sigma(u) = -1) = \frac{1}{2}$$

Given locations $\mathbf{X}$ and communities $\boldsymbol{\sigma}$

$$A_{uv} = 1 \begin{cases} \text{w.p. } f_{\text{in}}(X_u, X_v) & \text{if } \sigma(u) = \sigma(v) \\ \text{w.p. } f_{\text{out}}(X_u, X_v) & \text{if } \sigma(u) \neq \sigma(v) \end{cases}$$

Model

$$\mathbf{S} = \left(\frac{-1}{2}, \frac{1}{2}\right]^d$$



$$N \sim \text{Poi}(\lambda n)$$

- Poisson point process $\mathbf{X} = (X_u)_{u=1}^N$ of intensity λn.
- Two communities: $\boldsymbol{\sigma} = (\sigma(1), \cdots, \sigma(N))$

$$\mathbb{P}(\sigma(u) = +1) = \mathbb{P}(\sigma(u) = -1) = \frac{1}{2}$$

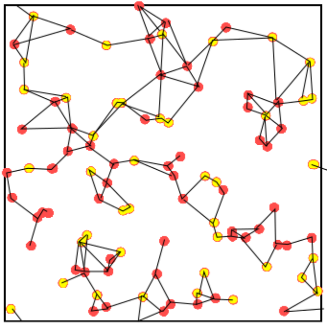
- Connection functions $f_{\text{in}}(\cdot), f_{\text{out}}(\cdot) : \mathbf{S} \times \mathbf{S} \to [0, 1]$

Given locations $\mathbf{X}$ and communities $\boldsymbol{\sigma}$

$$A_{uv} = 1 \begin{cases} \text{w.p. } f_{\text{in}}(X_u, X_v) & \text{if } \sigma(u) = \sigma(v) \\ \text{w.p. } f_{\text{out}}(X_u, X_v) & \text{if } \sigma(u) \neq \sigma(v) \end{cases}$$

Model

$$\mathbf{S} = \left(\frac{-1}{2}, \frac{1}{2} \right]^d$$



$N \sim \mathrm{Poi}(\lambda n)$

- Poisson point process $\mathbf{X} = (X_u)_{u=1}^N$ of intensity λn.
- Two communities: $\boldsymbol{\sigma} = (\sigma(1), \cdots, \sigma(N))$

$$\mathbb{P}(\sigma(u) = +1) = \mathbb{P}(\sigma(u) = -1) = \frac{1}{2}$$

- Connection functions $f_{\text{in}}(\cdot), f_{\text{out}}(\cdot) : \mathbf{S} \times \mathbf{S} \to [0, 1]$

Given locations $\mathbf{X}$ and communities $\boldsymbol{\sigma}$

$$A_{uv} = 1 \begin{cases} \text{w.p. } f_{\text{in}}(X_u, X_v) & \text{if } \sigma(u) = \sigma(v) \\ \text{w.p. } f_{\text{out}}(X_u, X_v) & \text{if } \sigma(u) \neq \sigma(v) \end{cases}$$

Here, f_{in} and f_{out} are functions of the distance $d(X_u, X_v)$.

Model: 1d case

Torus: $\mathbf{S} = \left(\frac{-1}{2}, \frac{1}{2}\right]$

$d(x, y) := \min\{|x - y|, 1 - |x - y|\}$

- Poisson point process $\mathbf{X} = (X_u)_{u=1}^N$ of intensity λn.
- Two communities: $\boldsymbol{\sigma} = (\sigma(1), \cdots, \sigma(N))$

$$\mathbb{P}(\sigma(u) = +1) = \mathbb{P}(\sigma(u) = -1) = \frac{1}{2}$$

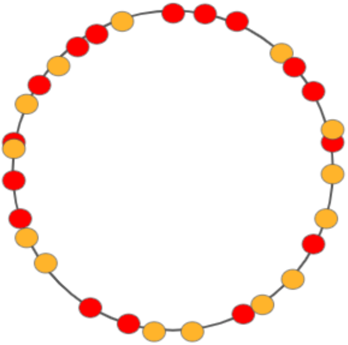
- Connection functions $f_{\text{in}}(\cdot), f_{\text{out}}(\cdot) : \mathbf{S} \times \mathbf{S} \to [0, 1]$

Given locations $\mathbf{X}$ and communities $\boldsymbol{\sigma}$

$$A_{uv} = 1 \begin{cases} \text{w.p. } f_{\text{in}}(d(X_u, X_v)) & \text{if } \sigma(u) = \sigma(v) \\ \text{w.p. } f_{\text{out}}(d(X_u, X_v)) & \text{if } \sigma(u) \neq \sigma(v) \end{cases}$$

Model: 1d case

Torus: $\mathbf{S} = \left(\frac{-1}{2}, \frac{1}{2}\right]$
$d(x,y) := \min\{|x-y|, 1-|x-y|\}$



- Poisson point process $\mathbf{X} = (X_u)_{u=1}^{N}$ of intensity λn.
- Two communities: $\boldsymbol{\sigma} = (\sigma(1), \cdots, \sigma(N))$

$$\mathbb{P}(\sigma(u) = +1) = \mathbb{P}(\sigma(u) = -1) = \frac{1}{2}$$

- Connection functions $f_{\text{in}}(\cdot), f_{\text{out}}(\cdot) : \mathbf{S} \times \mathbf{S} \to [0,1]$

Given locations $\mathbf{X}$ and communities $\boldsymbol{\sigma}$

$$A_{uv} = 1 \begin{cases} \text{w.p. } f_{\text{in}}(d(X_u, X_v)) & \text{if } \sigma(u) = \sigma(v) \\ \text{w.p. } f_{\text{out}}(d(X_u, X_v)) & \text{if } \sigma(u) \neq \sigma(v) \end{cases}$$

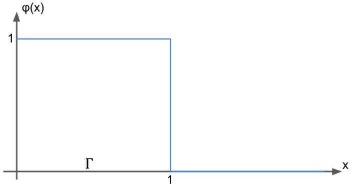
Geometric kernel

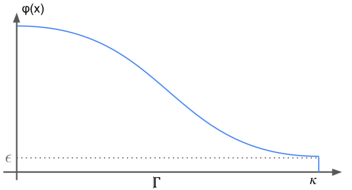
- Geometric kernel
  A measurable function
  $\phi : \mathbb{R}^+ \rightarrow [0, 1]$

Geometric kernel

- Geometric kernel
  A measurable function
  $\phi : \mathbb{R}^+ \to [0, 1]$

- Examples:
  1. $\phi(x) = \mathbf{1}\{x \leq 1\}$
  2. A general kernel



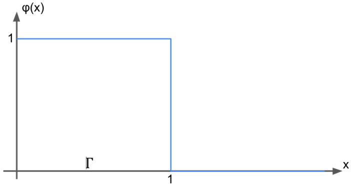


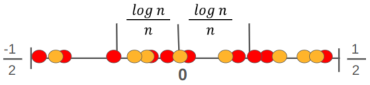
Geometric kernel

- Geometric kernel
  A measurable function
  $\phi : \mathbb{R}^+ \to [0, 1]$

- Examples:
  1. $\phi(x) = \mathbf{1}\{x \leq 1\}$
  2. A general kernel

- $f_{\text{in}}(X_u, X_v) = p\phi\left(\frac{d(X_u, X_v)}{\frac{\log n}{n}}\right)$ and
  $f_{\text{out}}(X_u, X_v) = q\phi\left(\frac{d(X_u, X_v)}{\frac{\log n}{n}}\right)$,
  where $p > q$.





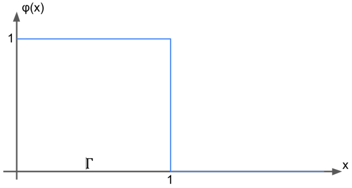
Abbe, E., Baccelli, F., and Sankararaman, A. (2021). Community detection on Euclidean random graphs. Information and Inference: A Journal of the IMA, 10(1), 109-160.

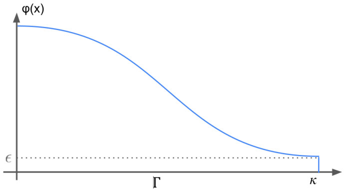
Geometric kernel

- Geometric kernel
  A measurable function
  $\phi : \mathbb{R}^+ \to [0,1]$

- Examples:
  1. $\phi(x) = \mathbf{1}\{x \le 1\}$
  2. A general kernel

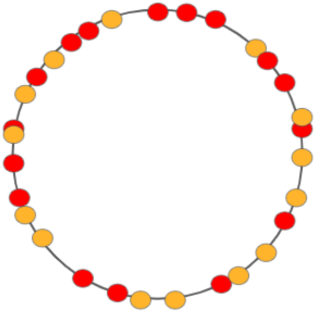
- $f_{\text{in}}(X_u, X_v) = p\phi\left(\frac{d(X_u, X_v)}{\frac{\log n}{n}}\right)$ and
  $f_{\text{out}}(X_u, X_v) = q\phi\left(\frac{d(X_u, X_v)}{\frac{\log n}{n}}\right)$,
  where $p > q$.





Geometric kernel block model

- Locations: $\mathbf{X} \sim$ PPP(λn) on $\mathbf{S}$
- Communities: $\boldsymbol{\sigma} : \sigma(u) \sim$ Unif $(\{-1, +1\})$
- Probabilities $p, q \in [0, 1]$ with $p > q$
- Geometric kernel: ϕ



Given locations $\mathbf{X}$ and communities $\boldsymbol{\sigma}$

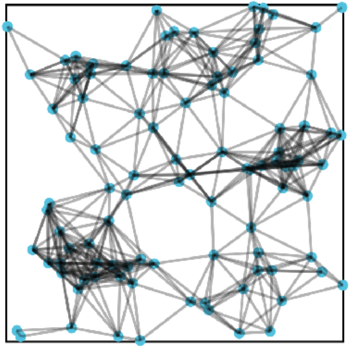
$$A_{uv} = 1 \begin{cases} \text{with prob. } p\phi\left(\frac{d(X_u, X_v)}{\frac{\log n}{n}}\right) & \text{if } \sigma(u) = \sigma(v) \\ \text{with prob. } q\phi\left(\frac{d(X_u, X_v)}{\frac{\log n}{n}}\right) & \text{if } \sigma(u) \neq \sigma(v) \end{cases}$$

$$\mathbf{A} = (A_{uv})_{u,v=1}^{N} \sim GKBM(\lambda n, p, q, \phi)$$

Problem formulation

$$\mathbf{A} \sim GKBM(\lambda n, p, q, \phi)$$

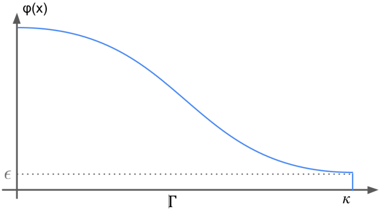
Problem: Given the locations $\mathbf{X}$ and the adjacency matrix $\mathbf{A}$, recover $\boldsymbol{\sigma}$ exactly.



- An estimate $\hat{\boldsymbol{\sigma}}_n$ of $\boldsymbol{\sigma}_n$ recovers the communities exactly if

$$\lim_{n \to \infty} \mathbb{P}\left(\hat{\boldsymbol{\sigma}}_n \in \{\pm \boldsymbol{\sigma}_n\}\right) = 1$$

Main results



Define $\kappa = \sup_{x \in \Gamma} x$, $0 < \kappa < \infty$ and

$$I_\phi(p,q) := 2 \int_{\mathbb{R}_+} \left[1 - \sqrt{pq}\phi(x) - \sqrt{(1 - p\phi(x))(1 - q\phi(x))} \right] dx$$

Converse: If $\lambda\kappa < 1$ or $\lambda I_\phi(p,q) < 1$, exact recovery is not possible using any algorithm.

Achievability: If $\lambda\kappa > 1$ and $\lambda I_\phi(p,q) > 1$, then there exists a linear time algorithm (in the number of edges) achieving exact-recovery.

Impossibility: Idea

If $\lambda\kappa < 1$ or $\lambda I_\phi(p,q) < 1$, exact recovery is not possible

$$I_\phi(p,q) := 2\int_{\mathbb{R}_+} \left[1 - \sqrt{pq}\phi(x) - \sqrt{(1-p\phi(x))(1-q\phi(x))}\,\right] dx$$

Impossibility: Idea

If $\lambda\kappa < 1$ or $\lambda I_\phi(p, q) < 1$, exact recovery is not possible

$$I_\phi(p,q) := 2\int_{\mathbb{R}_+} \left[1 - \sqrt{pq}\phi(x) - \sqrt{(1-p\phi(x))(1-q\phi(x))}\,\right] dx$$

- Genie based estimator: Log likelihood $\mathcal{L}(\mathbf{A}, \boldsymbol{\sigma}, \mathbf{X})$

$$\sum_{\substack{v\sim 0\\ \sigma_v=\sigma_0}} \log\left(p\phi_{v0}\right) + \sum_{\substack{v\sim 0\\ \sigma_v\neq\sigma_0}} \log\left(q\phi_{v0}\right) + \sum_{\substack{v\nsim 0\\ \sigma_v=\sigma_0}} \log\left(1-p\phi_{v0}\right) + \sum_{\substack{v\nsim 0\\ \sigma_v\neq\sigma_0}} \log\left(1-q\phi_{v0}\right)$$

Impossibility: Idea

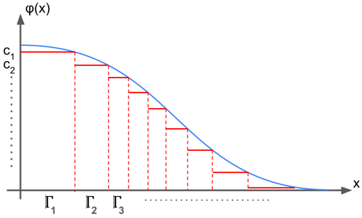
If $\lambda\kappa < 1$ or $\lambda I_\phi(p,q) < 1$, exact recovery is not possible

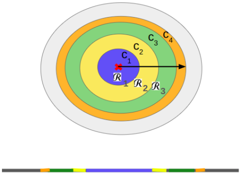
$$I_\phi(p,q) := 2\int_{\mathbb{R}_+}\left[1 - \sqrt{pq}\phi(x) - \sqrt{(1-p\phi(x))(1-q\phi(x))}\,\right]dx$$

- Genie based estimator: Log likelihood $\mathcal{L}(\mathbf{A}, \boldsymbol{\sigma}, \mathbf{X})$

$$\sum_{\substack{v\sim 0\\ \sigma_v=\sigma_0}} \log\left(p\phi_{v0}\right) + \sum_{\substack{v\sim 0\\ \sigma_v\neq\sigma_0}} \log\left(q\phi_{v0}\right) + \sum_{\substack{v\nsim 0\\ \sigma_v=\sigma_0}} \log\left(1-p\phi_{v0}\right) + \sum_{\substack{v\nsim 0\\ \sigma_v\neq\sigma_0}} \log\left(1-q\phi_{v0}\right)$$

- Approximate by simple functions





Impossibility: Idea

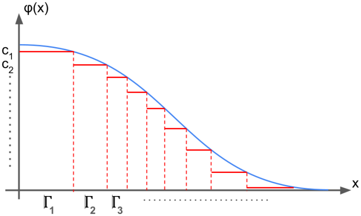
If $\lambda\kappa < 1$ or $\lambda I_\phi(p,q) < 1$, exact recovery is not possible

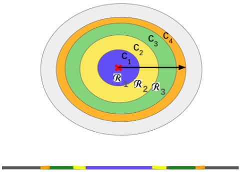
$$I_\phi(p,q) := 2\int_{\mathbb{R}_+}\left[1 - \sqrt{pq}\phi(x) - \sqrt{(1-p\phi(x))(1-q\phi(x))}\,\right]dx$$

- Genie based estimator: Log-likelihood function: $\mathcal{L}(\mathbf{A}, \boldsymbol{\sigma}, \mathbf{X})$

$$\sum_{s=1}^{\ell}\sum_{v\in\mathcal{R}_s}\sum_{\substack{v\sim u\\ \sigma_v=\sigma_u}}\log(pc_s) + \sum_{\substack{v\sim u\\ \sigma_v\neq\sigma_u}}\log(qc_s) + \sum_{\substack{v\nsim u\\ \sigma_v=\sigma_u}}\log(1-pc_s) + \sum_{\substack{v\nsim u\\ \sigma_v\neq\sigma_u}}\log(1-qc_s)$$

- Approximate by simple functions





Impossibility: Idea

If $\lambda\kappa < 1$ or $\lambda I_\phi(p,q) < 1$, exact recovery is not possible

$$I_\phi(p,q) := 2\int_{\mathbb{R}_+}\left[1 - \sqrt{pq}\,\phi(x) - \sqrt{(1-p\phi(x))(1-q\phi(x))}\,\right]dx$$

- Genie based estimator: Log-likelihood function: $\mathcal{L}(\mathbf{A}, \boldsymbol{\sigma}, \mathbf{X})$

$$\sum_{s=1}^{\ell}\sum_{v\in\mathcal{R}_s}\sum_{\substack{v\sim u\\ \sigma_v=\sigma_u}}\log(pc_s)+\sum_{\substack{v\sim u\\ \sigma_v\neq\sigma_u}}\log(qc_s)+\sum_{\substack{v\nsim u\\ \sigma_v=\sigma_u}}\log(1-pc_s)+\sum_{\substack{v\nsim u\\ \sigma_v\neq\sigma_u}}\log(1-qc_s)$$

- Testing Poisson vectors

In $\mathcal{R}_s$	Neighbours	Non-neighbours
Same	$\text{Poi}\left(\frac{\lambda \log n}{2} pc_s \text{vol}(\Gamma_s)\right)$	$\text{Poi}\left(\frac{\lambda \log n}{2}(1-pc_s)\text{vol}(\Gamma_s)\right)$
Different	$\text{Poi}\left(\frac{\lambda \log n}{2} qc_s \text{vol}(\Gamma_s)\right)$	$\text{Poi}\left(\frac{\lambda \log n}{2}(1-qc_s)\text{vol}(\Gamma_s)\right)$

- Hypothesis testing error $\to \exp\left(-\log n\ \lambda I_\phi(p,q)\right) = n^{-\lambda I_\phi(p,q)}$
- Total number of errors $\approx n^{1-\lambda I_\phi(p,q)} \to \infty$ when $\lambda I_\phi(p,q) < 1$.

Achievability

Q: Can we recover the communities exactly when $\lambda I_\phi(p, q) > 1$?

Achievability

Q: Can we recover the communities exactly when $\lambda I_\phi(p,q) > 1$?
YES !! We provide next a two phase algorithm.
Define $\kappa = \max_{x \in \Gamma} x$.

Achievability

Q: Can we recover the communities exactly when $\lambda I_{\phi}(p, q) > 1$?

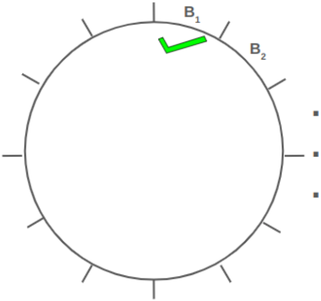
YES !! We provide next a two phase algorithm.

Define $\kappa = \max_{x \in \Gamma} x$.

Phase 1: Almost-exact recovery

- Divide into blocks of size $\kappa \frac{\log n}{n}$
- Recover exactly in an initial block
- Propagate from a recovered block to adjacent block and so on

Phase 2: Refinement step



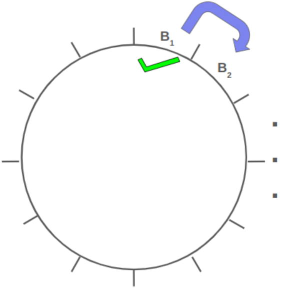
Achievability

Q: Can we recover the communities exactly when $\lambda I_\phi(p, q) > 1$?
YES !! We provide next a two phase algorithm.
Define $\kappa = \max_{x \in \Gamma} x$.

Phase 1: Almost-exact recovery

- Divide into blocks of size $\kappa \frac{\log n}{n}$
- Recover exactly in an initial block
- Propagate from a recovered block to adjacent block and so on

Phase 2: Refinement step



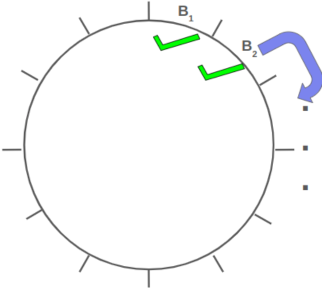
Achievability

Q: Can we recover the communities exactly when $\lambda I_{\phi}(p, q) > 1$?
YES !! We provide next a two phase algorithm.
Define $\kappa = \max_{x \in \Gamma} x$.

Phase 1: Almost-exact recovery

- Divide into blocks of size $\kappa \frac{\log n}{n}$
- Recover exactly in an initial block
- Propagate from a recovered block to adjacent block and so on

Phase 2: Refinement step



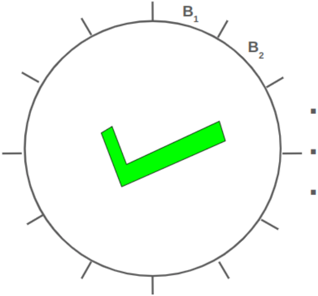
Achievability

Q: Can we recover the communities exactly when $\lambda I_\phi(p, q) > 1$?
YES !! We provide next a two phase algorithm.
Define $\kappa = \max_{x \in \Gamma} x$.

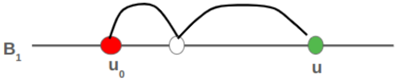
Phase 1: Almost-exact recovery

- Divide into blocks of size $\kappa \frac{\log n}{n}$
- Recover exactly in an initial block
- Propagate from a recovered block to adjacent block and so on

Phase 2: Refinement step

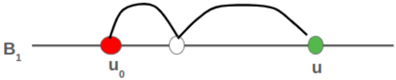


Recovering the initial block



- Dense graph within the block.
- Off-the-shelf algorithms for e.g., spectral.
- Choose $u_0 \in V_1$ and set $\hat{\sigma}(u_0) = +1$
- Cluster using number of common neighbours of u and u_0

Recovering the initial block



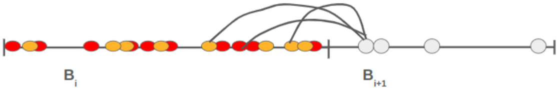
- Dense graph within the block.
- Off-the-shelf algorithms for e.g., spectral.
- Choose $u_0 \in V_1$ and set $\hat{\sigma}(u_0) = +1$
- Cluster using number of common neighbours of u and u_0

Lemma

For any $p > q$ and any $\Delta > 0$, communities of nodes in the initial block B_1 are recovered w.h.p., i.e.,

$$\mathbb{P}\left(\bigcap_{u \in V_1} \{\hat{\sigma}(u) = \sigma(u)\}\right) \geq 1 - \Delta n^{-c_1} \log n.$$

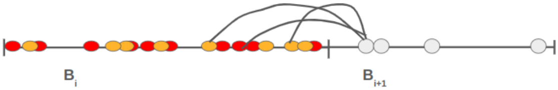
Label propagation



- Assume that the estimated communities in block B_i are the true communities.
- Evaluate the likelihood for every $u \in B_{i+1}$

$$\sum_{v \in V_i} \hat{\sigma}(v) \left[A_{uv} \log \frac{p\,(1 - q\psi_n(X_u, X_v))}{q\,(1 - p\psi_n(X_u, X_v))} + \log \frac{(1 - p\psi_n(X_u, X_v))}{(1 - q\psi_n(X_u, X_v))} \right]$$

Label propagation



- Assume that the estimated communities in block B_i are the true communities.
- Evaluate the likelihood for every $u \in B_{i+1}$

$$\sum_{v \in V_i} \hat{\sigma}(v) \left[A_{uv} \log \frac{p\left(1 - q\psi_n(X_u, X_v)\right)}{q\left(1 - p\psi_n(X_u, X_v)\right)} + \log \frac{\left(1 - p\psi_n(X_u, X_v)\right)}{\left(1 - q\psi_n(X_u, X_v)\right)} \right]$$

Lemma

For $G \sim GKBM(\lambda n, p, q, \phi)$, there exists an $M \equiv M(p, q, \phi) > 0$ such that

$$\mathbb{P}\left(\bigcap_{i=1}^{n/\kappa \log n} \{\# \text{ of mistakes in } B_i \leq M\} \right) \geq 1 - o(1).$$

Crucial idea

$$\mathcal{A}_i = \{\text{at most } M \text{ mistakes within block } B_i\}$$

Sacrifice on probability but have constant number of mistakes

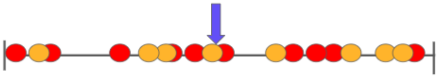
$$\mathbb{P}\left(\bigcap_{i=1}^{n/\kappa \log n} \mathcal{A}_i\right) = \mathbb{P}(\mathcal{A}_1) \prod_{i=2}^{n/\kappa \log n} \mathbb{P}\left(\mathcal{A}_i \middle| \mathcal{A}_{i-1}\right)$$

Lemma

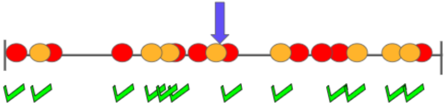
Fix $\eta > 0$. For $G \sim GKBM(\lambda n, p, q, \phi)$, we have that

$$\mathbb{P}\left(\textit{Total \# of mistakes} \leq \frac{\eta n}{3\kappa}\right) = 1 - o(1).$$

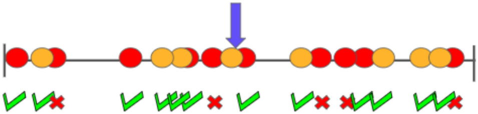
Refinement step



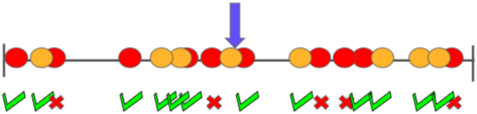
Refinement step



Refinement step



Refinement step



- Evaluate $g(u, \hat{\boldsymbol{\sigma}})$ to be

$$\sum_{v \in V(u)} \hat{\sigma}(v) \left[A_{uv} \log \frac{p\,(1 - q\psi_n(X_u, X_v))}{q\,(1 - p\psi_n(X_u, X_v))} + \log \frac{1 - p\psi_n(X_u, X_v)}{1 - q\psi_n(X_u, X_v)} \right]$$

- Bound the worst case error vector $|g(u, \hat{\boldsymbol{\sigma}}) - g(u, \boldsymbol{\sigma})| \leq \beta\eta \log n$ for some $\beta \equiv \beta(p, q, \phi)$.
- Use simple function approximation

$$\mathbb{P}(g(u, \hat{\boldsymbol{\sigma}}) > 0 | \sigma(u) = -1) \leq n^{\frac{\beta\eta}{2} - \lambda n \sum_{s=1}^{\ell'} \mathrm{vol}(\mathcal{R}_s) \left[1 - \sqrt{pq}c_s - \sqrt{(1-pc_s)(1-qc_s)}\right]}$$

- Take $\eta = \frac{\lambda I_\phi(p,q) - 1}{\beta} > 0$ and using union bound

$$\mathbb{P}(\exists u : \tilde{\sigma}(u) \neq \sigma(u)) = o(1)$$

Conclusions and Future Work

- Introduced block models with geometric kernels.
- Information metric $I_\phi(p, q)$ governs community recovery.
- $\lambda I_\phi(p, q) < 1$ or $\lambda\kappa < 1$: exact recovery not possible
- $\lambda I_\phi(p, q) > 1$ and $\lambda\kappa > 1$: linear time algorithm for community recovery
- Multiple communities
- Higher dimensions

Conclusions and Future Work

- Introduced block models with geometric kernels.
- Information metric $I_\phi(p, q)$ governs community recovery.
- $\lambda I_\phi(p, q) < 1$ or $\lambda\kappa < 1$: exact recovery not possible
- $\lambda I_\phi(p, q) > 1$ and $\lambda\kappa > 1$: linear time algorithm for community recovery
- Multiple communities
- Higher dimensions

Thank you !!

Community Detection on Block Models with Geometric Kernels

arxiv.org/abs/2403.02802

Thank you !!