Data Integration: Network-Guided Influential Covariates Recovery

Wanjie Wang

Department of Statistics and Data Science
National University of Singapore

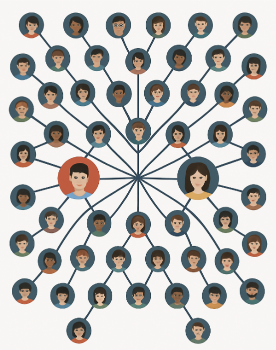
SNAB 2025 Workshop

June 3rd, 2025

Data Integration: Different Formats

One user, with connections and microblogs:

Follower-Followee Network



User micro-blogs:

Low-Dimensional Embedding

Weibo data:

- Each user has an *intrinsic response* $Y_i \in \mathcal{R}^K$: interests and opinions in topics
- Y_i's decide the connections and micro-blogs
- Estimate of Y_i is difficult, without precise interpretation

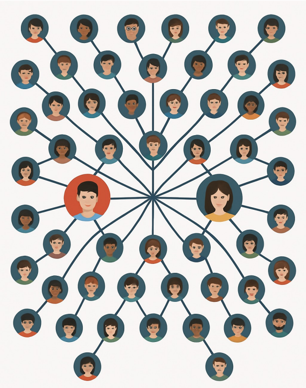
Our goal:

- find the influential covariates in microblogs on Y_i
- Use them for estimation/prediction

Collaborate with Mr. Tao Shen, DSDS, NUS.

Network and Covariates

On a social platform with N users, we collect:



Network data $A \in \mathcal{R}^{N \times N}$

$$A_{ij} = \begin{cases} 1, & \text{users } i, j \text{ are connected} \\ 0, & \text{otherwise.} \end{cases}$$

Hidden information:

- Intrinsic response $Y_i \in \mathcal{R}^K$ for user i, $K = O(1)$.

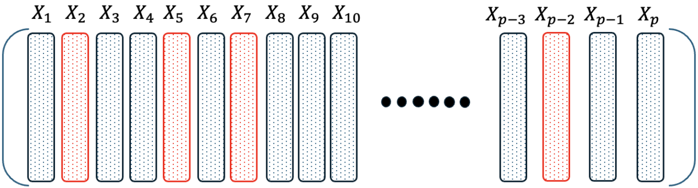
Popular models

- Stochastic Blockmodel and its variants: clustering
- Latent position model

High-Dimensional Covariates X

Covariates $x_i \in \mathcal{R}^p$ captures the user's information

- Basic information: age, gender, location, etc
- Behavior: posts, tags, favourite movies, etc



- High-dimensional covariates (large p)
- Sparse influential covariates related to the intrinsic response $Y_i \in \mathcal{R}^K$

Goal

Data from two sources:

$$A \in \mathcal{R}^{N \times N}, \quad X \in \mathcal{R}^{N \times p}$$

Assumptions:

- large N and p
- Intrinsic response $Y_i \in \mathcal{R}^K$ for user i, $K = O(1)$
- Sparse influential covariates related to Y_i

$$\mathcal{S} = \{j \in p; \quad X_j \text{ depends on } Y\},$$

and $|\mathcal{S}|/p \to 0$ when $p \to \infty$.

Goal

Data from two sources:

$$A \in \mathcal{R}^{N \times N}, \quad X \in \mathcal{R}^{N \times p}$$

Assumptions:

- large N and p
- Intrinsic response $Y_i \in \mathcal{R}^K$ for user i, $K = O(1)$
- Sparse influential covariates related to Y_i

$$\mathcal{S} = \{j \in p; \quad X_j \text{ depends on } Y\},$$

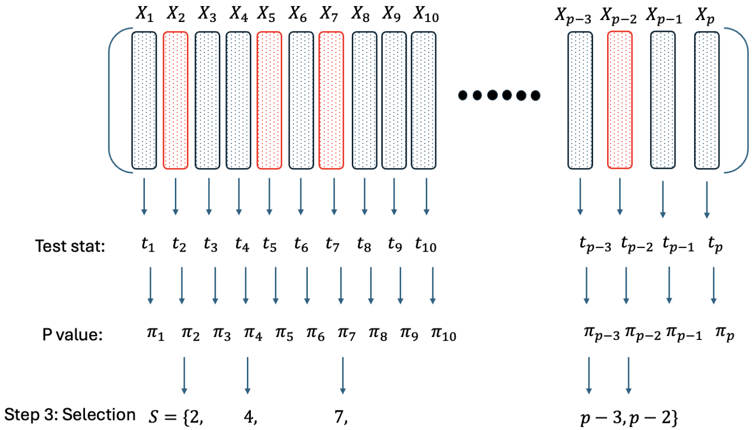
  and $|\mathcal{S}|/p \to 0$ when $p \to \infty$.

Goal:

- Part I: Recover $\mathcal{S}$ based on A and X
- Part II: Estimate and predict Y_i based on $\mathcal{S}$

Goal I: Network-Guided Influential Covariate Selection

Review: Covariate-wise Screening Statistics



Review: Covariate-wise Screening Statistics

Example: High-dimensional clustering problem, no network info.

- Assumption: $X_j \sim N(0,1)$ for $j \notin \mathcal{S}$
- Step 1: test statistic when labels are unknown

$$t_j = \sum_{i=1}^{N} X_{ij}^2 \sim \chi_N^2, \qquad j \notin \mathcal{S}$$

- Step 2: p-value

$$\pi_j = P(\chi_N^2 \geq t_j), \quad j \in [p]$$

- Step 3: select the influential covariates $\mathcal{S}$

$$\hat{\mathcal{S}} = \{j; \pi_j \leq \text{given threshold } \pi_{thre}\}$$

e.g. Jin and **Wang**, 2016

Review: Covariate-wise Screening Statistics

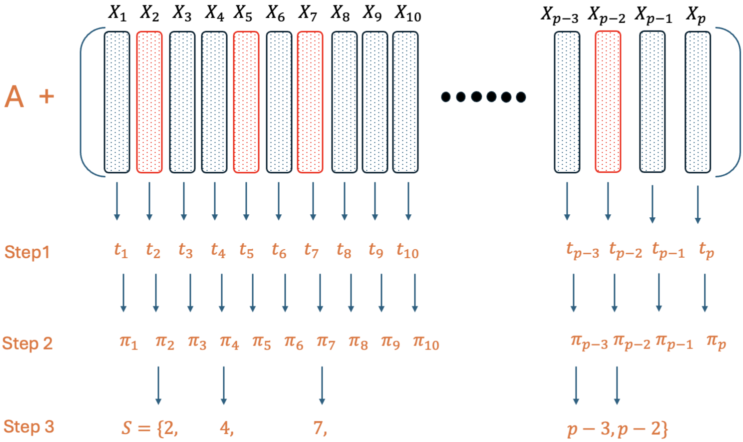
Pros:

- Computationally efficient
  - test stat is based on one column, not the whole matrix
- Flexible
  - Adjust the test statistic to adapt to complex dist and data

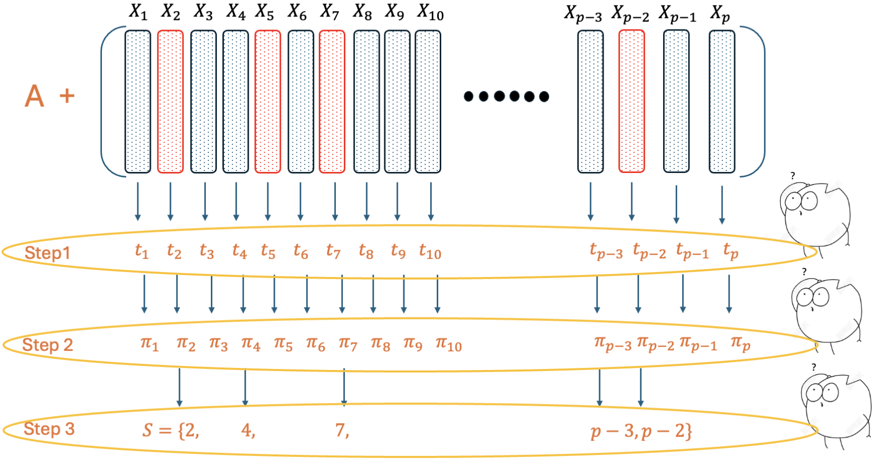
Cons:

- The p-value calculation requires the null dist. of test stat
- Deciding a proper t_{thre} is complicated

Network is in....



Network is in....



Step 1. Network-Guided Statistic

Goal: $\mathcal{S} = \{j \in p; \quad X_j \text{ depends on } Y\}$, where Y is the intrinsic response

- Only X: no info about Y; unsupervised
- X and A: A has partial info about Y

Step 1. Network-Guided Statistic

Goal: $\mathcal{S} = \{j \in p; \quad X_j \text{ depends on } Y\}$, where Y is the intrinsic response

- Only X: no info about Y; unsupervised
- X and A: A has partial info about Y

Network-Guided test stat:

Input: Network A, Covariate X_j, tuning parameter $\hat{K}$

(i) (Extract partial info. by the spectral analysis)
Let ξ_k be the k-th leading eigenvector of A

(ii) (Construct the stat based on ξ_k and X_j)

$$t_j = t_j(A, X_j; \hat{K}) = \sum_{k=1}^{\hat{K}} (\xi_k^T X_j)^2$$

Step 2: Null Distribution

Assumption: $X_j \sim N(0, I_n)$ for $j \notin \mathcal{S}$

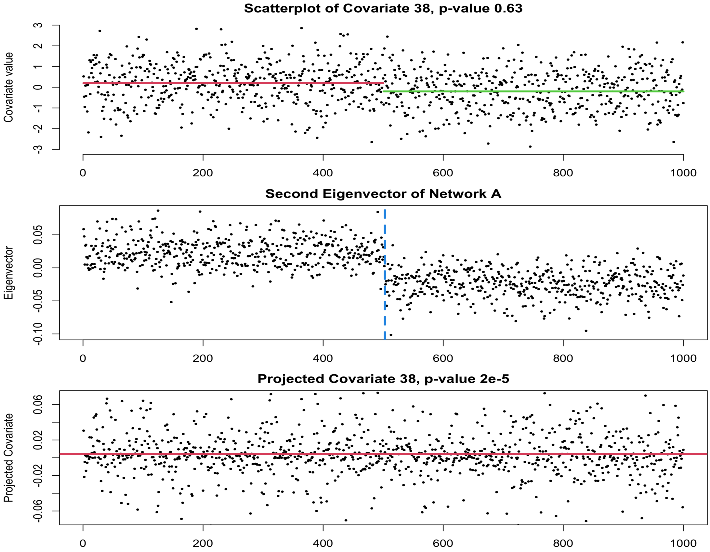
$$t_j = t_j(A, X_j; \hat{K}) = \sum_{k=1}^{\hat{K}} (\xi_k^T X_j)^2$$

- Since ξ_k is an eigenvector with norm 1, $\xi_k^T X_j \sim N(0, 1)$
- Since $\xi_k^T \xi_l = 0$, $\xi_k^T X_j$ and $\xi_l^T X_j$ are indep.
- As a conclusion, the null dist.

$$t_j = t_j(A, X_j; \hat{K}) \sim \chi^2_{\hat{K}},$$

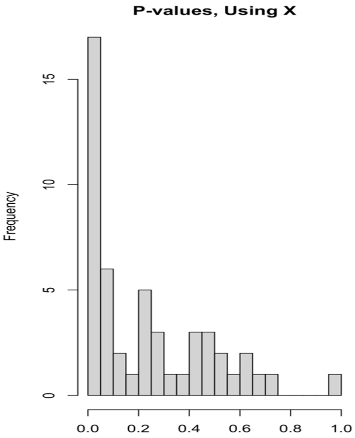
p-values: $\pi_j = P(\chi^2_{\hat{K}} > t_j)$.

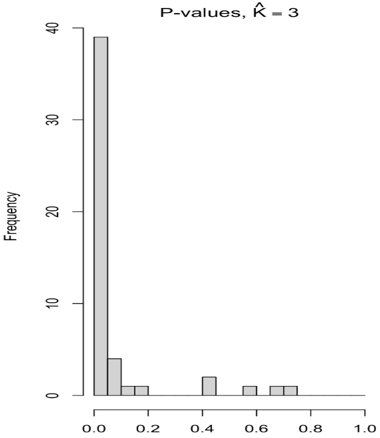
$Y_i \in \{1,2\}$, $N = 1000$, Influential covariate 38



Network Spectral Info Guide the Tests

Histograms of influential covariates p-values, $K = 3$:





- p-values for non-influential covariates follow uniform distribution
- Significant power gain from network info.

Step 3: Selection

$$\hat{\mathcal{S}} = \{j; \pi_j \leq \text{given threshold } \pi_{thre}\}$$

Deciding π_{thre} is challenging

Step 3: Selection

$$\hat{\mathcal{S}} = \{j; \pi_j \leq \text{given threshold } \pi_{thre}\}$$

Deciding π_{thre} is challenging

- For p tests, there will always be $\sim p * \pi_{thre}$ covariates selected even if there are no signals
- Data-driven Higher Criticism Threshold:
  - Original idea goes back to John Tukey, at a given level
  - Donoho and Jin extends the stat to a function

Step 3: Higher Criticism Thresholding (HCT)

Input: the p-value of each covariate, say π_j, $1 \leq j \leq p$

1. *(Ordering)* Order them as $\pi_{(1)} \leq \pi_{(2)} \leq \cdots \leq \pi_{(p)}$
2. *(Decide the cut-off)* Calculate the Higher Criticism score

$$HC(j) = \sqrt{p}\frac{j/p - \pi_{(j)}}{\sqrt{\pi_{(j)}(1-\pi_{(j)})}}$$

3. Let $\hat{s} = \max_{1 \leq j \leq p/2} HC(j)$
4. The threshold is $\pi_{thre} = \pi_{(\hat{j})}$. The selected covariates are

$$\hat{\mathcal{S}} = \{j : \pi_j \leq \pi_{thre}\} = \{j : \pi_j \leq \pi_{(\hat{j})}\},$$

with the cardinality $\hat{s}$.

Algorithm: Network-Guided Covariate Selection

Network-Guided Covariate Selection (NGCS) algorithm

Input: Network A, covariates X, tuning parameter $\hat{K}$

Step 1 Construct the test statistic

1. Find the top $\hat{K}$ eigenvectors of A (or the Laplacian L), denoted as $\xi_1, \cdots, \xi_{\hat{K}}$
2. Define the test stat $t_j = \sum_{k=1}^{\hat{K}} (\xi_k^T x_j)^2$

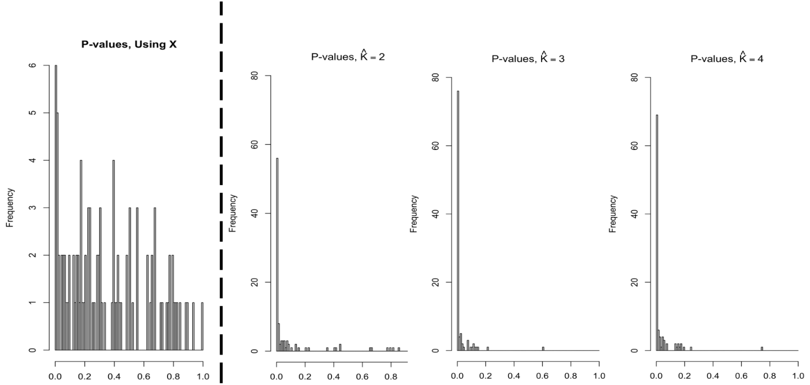
Step 2 Find p-values that $\pi_j = P(\chi^2_{\hat{K}} > t_j)$

Step 3 Higher Criticism Thresholding to decide $\hat{\mathcal{S}}$, using π_js.

Output: The set of selected influential covariates $\hat{\mathcal{S}}$

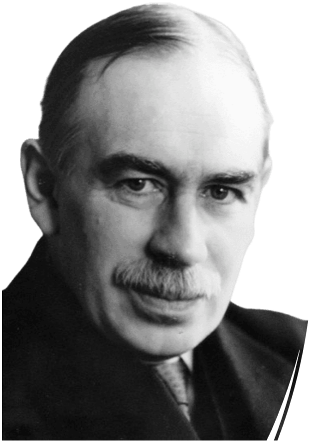
How many eigenvectors?

Histograms of influential covariates p-values, $K = 3$:



- Less eigenvectors $\hat{K} < K$: suffers a power loss, still better than using X
- More eigenvectors $\hat{K} > K$: not significant power loss

Intuition of NGCS



It is better to be approximately right than precisely wrong

-- John Maynard Keynes

Model and Theoretical Guarantee

Sparse and Weak Influential Covariates

$$X \in \mathcal{R}^{N \times p}, \quad A \in \mathcal{R}^{N \times N}$$

Assumptions:

- Covariates j: $X_j \sim N(YM_j, I_n)$
- Influential covariates $\mathcal{S} = \{j : \|M_j\| \neq 0\}$
- Sparsity: $|\mathcal{S}| = p^{1-\beta}$, $\beta > 0$
- Weakness: $\|M_j\| \to 0$.

Define the network-guided signal strength

$$\kappa_j = \sum_{k=1}^{\hat{K}} (\xi_k^T E[X_j])^2, \qquad \kappa_A = \min_{j \in \mathcal{S}} \kappa_j.$$

It doesn't have network model assumptions.

Consistency

The network-guided signal strength

$$\kappa_A = \min_{j \in \mathcal{S}} \kappa_j.$$

Theorem (Consistency)

Suppose the assumptions hold and $\kappa_A \geq \max\{16(1-\beta), 14\} \log p$,

(i) *[Sure screening property] with a high prob., the network-guided p-values satisfy that*

$$\max_{i \in \mathcal{S}} \pi_i < \min_{i \notin \mathcal{S}} \pi_i.$$

(ii) *[Exact recovery] Furthermore, the NGCS algorithm with HCT satisfies that*

$$\mathcal{S} \subset \hat{\mathcal{S}}, \qquad |\hat{\mathcal{S}} \backslash \mathcal{S}| \leq C \log^2 p \ll |\mathcal{S}|.$$

Network Models

Requirement on network:

$$\kappa_A \geq \max\{16(1-\beta), 14\} \log p$$

Corollaries under popular models:

- Degree-Corrected SBM
  - expected degree $\geq c \log n$
  - $\|M_j\|^2 \geq C \log p/n$, and $\hat{K} \geq K$
- Random Dot Product Graph
  - expected degree $\geq c \log n$
  - $\|M_j\|^2 \geq C \log p/n$ and $\hat{K} \geq K$
- More possibilities...

Network Models

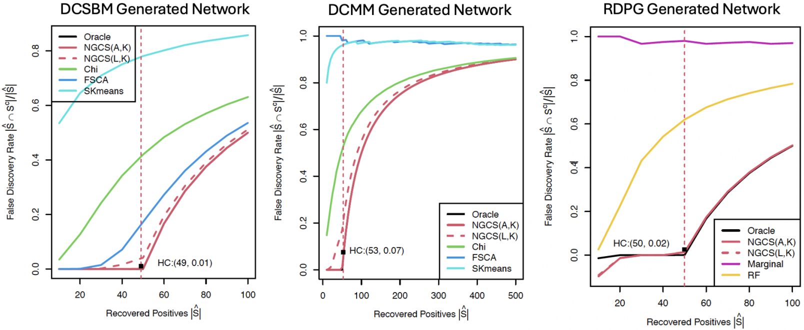
Requirement on network:

$$\kappa_A \geq \max\{16(1-\beta), 14\} \log p$$

Summary:

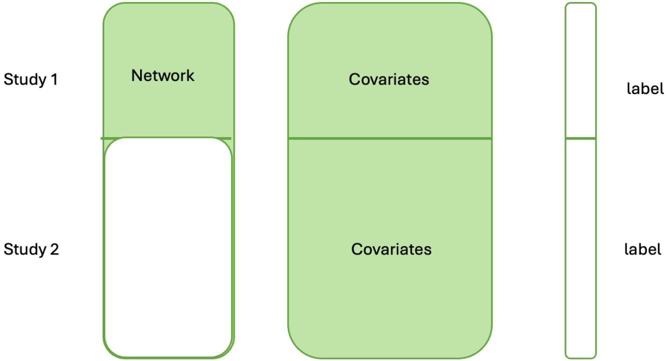
- the NGCS algorithm and theorem doesn't need network assumptions
- Under popular network models, with rich network info, NGCS achieves *the same rate* as the supervised learning case!

- 3 network models with different underlying K
- 50 repetitions
- Network-guided test stat outperforms other methods, and HCT achieves almost perfect selection



Goal II. Estimation and Prediction with Selected Influential Covariates

Data Integration: Clustering on Two Datasets



- Clustering of two datasets: partial network info.
- Recover the complete label vector $Y = \begin{pmatrix} Y^{(1)} \\ Y^{(2)} \end{pmatrix}$

NGCS Clustering

NG-Clu Algorithm

Input: $A^{(1)}$ and $X^{(1)}$ for Dataset 1, $X^{(2)}$ for Dataset 2, $\hat{K}$

Step 1. Apply NGCS to $A^{(1)}$, $X^{(1)}$, with $\hat{K}$ as tuning parameter. Let $\hat{\mathcal{S}}$ be the selected influential covariates

Step 2. Construct $X = \begin{pmatrix} X^{(1),\hat{\mathcal{S}}} \\ X^{(2),\hat{\mathcal{S}}} \end{pmatrix}$, where $X^{(1),\hat{\mathcal{S}}}$ and $X^{(2),\hat{\mathcal{S}}}$ are the submatrix of $X^{(1)}$ and $X^{(2)}$ restricted on $\hat{\mathcal{S}}$.

Step 3. Let $\Lambda_{\hat{K}}$ be the diagonal matrix of leading $\hat{K}$ singular values of X and $U_{\hat{K}}$ containing the left singular vectors.

Step 4. Apply k-means to $U_{\hat{K}}\Lambda_{\hat{K}}$.

Output: The label vector from k-means.

Consistency of Clustering

- Since the intrinsic responses are labels, we consider the DCSBM network model
- Under DCSBM, κ_A is simplified as $\kappa = \min_{j \in \mathcal{S}} \|M_j\|$

Theorem (Consistency of Clustering)

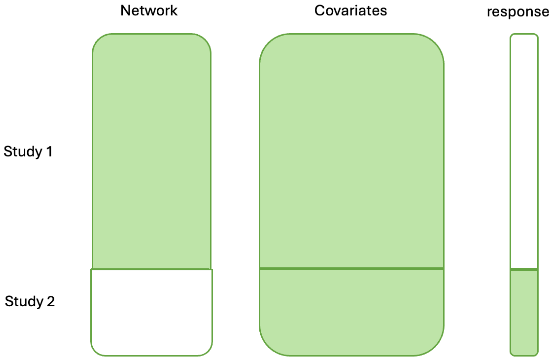
Under DCSBM and regular conditions on the distance between rows of M, there is

$$Err = \frac{misclassified}{N+n} \leq \frac{\hat{s}+N+n}{2(N+n)s\kappa^2}$$

In particular, if $\kappa^2 > (\hat{s}+N+n)/(N+n)s$, then there are no misclassified nodes.

- Error is the same with $N+n$ samples and s covariates.

Data Integration: Regression on Two Datasets



- Partial network and partial response vector z
- Goal: $z^{(1)}$ for Study 1 and z_{new} for x_{new}
- $n \ll N$

Network-Guided Regression

NG-Reg Algorithm

Input: $A^{(1)}$ and $X^{(1)}$ for Dataset 1, $X^{(2)}$ and z for Dataset 2, $\hat{K}$

Step 1. Apply NGCS to $A^{(1)}$, $X^{(1)}$, with $\hat{K}$ as tuning parameter. Let $\hat{\mathcal{S}}$ be the selected influential covariates

Step 2. Let $X^{(2),\hat{\mathcal{S}}}$ be the submatrix of $X^{(2)}$ restricted on $\hat{\mathcal{S}}$.

Step 3. Let $X^{(2),\hat{\mathcal{S}}} = U \Lambda V^T$. Define $U_{\hat{K}}$ and $V_{\hat{K}}$ be the matrices of U and V containing the leading $\hat{K}$ columns.

Step 4. Estimate coefficient vector $\hat{\gamma} = V_{\hat{K}} \Lambda_{\hat{K}} U_{\hat{K}}^T z \in \mathcal{R}^{|\hat{\mathcal{S}}|}$.

Output: The estimate is $X^{(1),\hat{\mathcal{S}}} \hat{\gamma}$

Consistency of Regression

- We consider the RDPG network model
- Under RDPG, κ_A is simplified as $\kappa = \min_{j \in \mathcal{S}} \|M_j\|$

Theorem (Consistency of Regression)

Under RDPG and further condition that rank$(M) = K$, $\lambda_K(M) \geq c\|M\|$ and $\kappa > 3\frac{\sqrt{n}+\sqrt{\hat{s}}}{\sqrt{ns}}$, there is

$$|\hat{\gamma}^T X_{n+1}^{\hat{S}} - \alpha^T Y_{n+1}| \leq \frac{\sqrt{n}+\sqrt{\hat{s}}}{\kappa\sqrt{ns}} + C\sigma_\epsilon(\frac{1}{\sqrt{n}} + \frac{1}{\kappa\sqrt{s}})$$

- Error is at the same order with the ordinary linear regression on n samples and s covariates in Data 2.

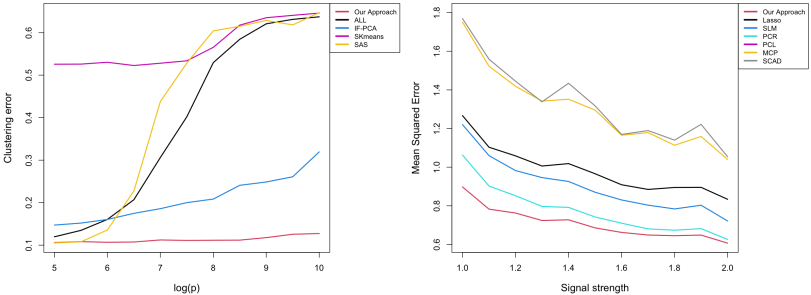
Clustering and Regression

Simulation settings:

- Study 1: $N = 800$ samples, $A^{(1)}$ and $X^{(1)}$
- Study 2: $n = 200$ samples, $X^{(2)}$ and possibly z
- $p = 1000$ covariates, among them $s = 50$ contribute to the clustering

- Clustering: DCSBM, $K = 3$
- Regression: RDPG, $K = 10$

Clustering and Regression Errors

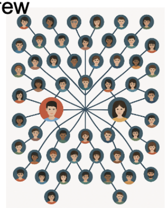


- Left panel: clustering result vs the number of covariates p
- Right panel: regression result vs the signal strength in M

Sina Weibo Data Analysis

Network

1. Start from 100 VIP users
2. Include their follower/followee
3. Include the follower/followee in previous step
4. Include the follower/followee in previous step
5. Remove those with few microblogs



10 Topics

Finance and economics
Literature and arts
Fashion and vogue
Current events and politics
Sports
Science and technology
Entertainment
Parenting and education
Public welfare
Etc.

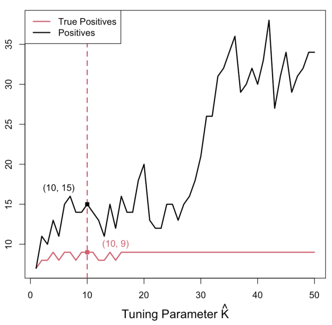
Jia et al. (2017) Node attribute-enhanced community detection in complex networks.

Sina Weibo Data

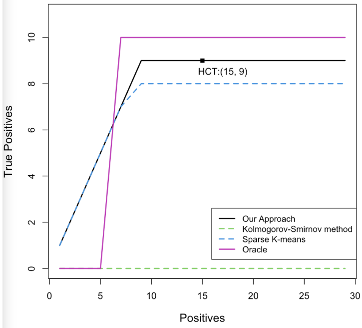
- Dataset 1: $N = 2000$ users, $p = 3000$ covariates
  - Network A: $A_{ij} = 1$ if i follows j
  - Covariates: 10 covariates from topic modelling; 2990 generated "fake" covariates
- Goal 1: Recover the 10 influential covariates

Recovery of the 10 influential covariates

Effects of Tuning Parameter $\hat{K}$



Methods Comparison when $\hat{K} = 10$



- Goal 2: Predict the position on a given topic
- Select one given topic, and define the position as

$$z_i = 1 - \sum_{j \in \mathcal{S}} X_{ij} + \epsilon_i, \quad \epsilon_i \sim N(0, 0.5^2)$$

- Dataset 2 sample sizes $n = 100, 150, \cdots, 500$

Regression

- Recover $\hat{\mathcal{S}}$ using Dataset 1
  - the number of total influential covariates $|\hat{\mathcal{S}}|$
  - the number of recovered true influential covariates $|\hat{\mathcal{S}} \cap \mathcal{S}|$
- Determine the linear coefficients using Dataset 2
- Estimate z_i for users in Dataset 1
  - MSE for $N = 2000$ users in Dataset 1

	New			Lasso			MCP			SCAD		
n_2	RMSE	$\lvert\hat{\mathcal{S}}\rvert$	$\lvert\hat{\mathcal{S}} \cap \mathcal{S}\rvert$	RMSE	$\lvert\hat{\mathcal{S}}\rvert$	$\lvert\hat{\mathcal{S}} \cap \mathcal{S}\rvert$	RMSE	$\lvert\hat{\mathcal{S}}\rvert$	$\lvert\hat{\mathcal{S}} \cap \mathcal{S}\rvert$	RMSE	$\lvert\hat{\mathcal{S}}\rvert$	$\lvert\hat{\mathcal{S}} \cap \mathcal{S}\rvert$
100	0.25	11.9 (1.79)	8 (0.00)	0.23	12.7 (23.30)	0.2 (0.42)	0.27	4.5 (7.21)	0.2 (0.42)	0.23	10.5 (16.39)	0.2 (0.42)
150	0.21	12.5 (4.01)	8 (0.00)	0.20	14.0 (21.75)	0.3 (0.67)	0.21	5.9 (6.37)	0.3 (0.67)	0.20	9.3 (16.91)	0.5 (0.85)
200	0.17	12.2 (2.25)	8 (0.00)	0.20	2.4 (5.15)	0.1 (0.32)	0.20	1.4 (2.17)	0.1 (0.32)	0.19	3.7 (4.99)	0.3 (0.48)
250	0.14	10.3 (1.16)	8 (0.00)	0.22	43.2 (51.47)	0.2 (0.42)	0.19	3.2 (5.79)	0.0 (0.00)	0.20	20.7 (28.23)	0.0 (0.00)
300	0.13	11.0 (2.10)	8 (0.00)	0.19	11.5 (22.18)	0.6 (0.70)	0.19	5.4 (8.42)	0.5 (0.53)	0.20	13.9 (21.89)	0.5 (0.53)
350	0.13	11.3 (1.64)	8 (0.00)	0.19	3.0 (7.15)	0.2 (0.42)	0.19	2.6 (4.86)	0.2 (0.42)	0.19	9.5 (16.66)	0.4 (0.52)
400	0.11	12.5 (2.84)	8 (0.00)	0.19	12.9 (19.02)	0.5 (0.97)	0.19	4.6 (7.49)	0.4 (0.70)	0.19	5.1 (8.71)	0.4 (0.70)
450	0.11	11.3 (1.77)	8 (0.00)	0.19	6.6 (7.82)	0.7 (0.95)	0.19	3.1 (4.63)	0.5 (0.71)	0.19	5.6 (6.69)	0.6 (0.84)
500	0.11	12.2 (2.53)	8 (0.00)	0.19	10.5 (10.83)	0.8 (0.79)	0.19	6.7 (6.88)	0.7 (0.82)	0.19	13.5 (13.00)	1.0 (1.05)

Take-Away Messages

- High-dimensional covariates and Network
  - General NGCS algorithm
  - Robust to network model mis-specification and K
  - Achieves the same rate of the supervised learning setting
  - Consistency analysis for clustering and regression
- Generalization to other integration problem
  - Spectral info is useful. e.g. manifold data

Main paper:

- Optimal Network-Guided Covariate Selection for High-Dimensional Data Integration. arXiv: 2504.04866

Appendix

Future Directions

Network + Covariates gains more and more interests:

- Many literature from various viewpoints:
  - Gene network and gene-expression data: Li and Li (2008) on linear regression; Wu, Zhu and Feng (2018) constructs a Markov chain on ranking statistics and network; Wang and Chen (2021) on Kendall's tau statistic
  - Network autoregression model: Zhu et al. (2019)
  - Dimension reduction of covariates: Gu and Han (2011); Zhao et al. (2022)
  - Community detection on sparse networks with covariates: Newman and Clauset (2016), Yang et al. (2013), Yan et al. (2019); Yan and Sarkar (2021), Zhang, Levina and Zhu (2016); Binkiewicz et al. (2017), Abbe et al. (2022); Xu et al. (2022)
  - Community detection with covariates bounds: Deshpande et al. (2018), Ma and Nandy (2023); Abbe et al. (2022)

Random Dot Product Graph

Consider the a special case in the latent position model

$$A_{i,j} \sim Bernoulli(\rho_n Y_i^T Y_j), \qquad Y_i \overset{i.i.d.}{\sim} F$$

- Y_i: the latent position of sample i
- ρ_n: the network density parameter

- The domain of F is a subset in the unit ball in $\mathcal{R}^K$ and $Y_i^T Y_j \geq 0$.
- $Cov(Y_i) \in \mathcal{R}^{K \times K}$ has a full rank
- For any realization Y_i, $Y_i^T E[Y_j] \geq c > 0$
- $n\rho_n \geq c_d \log n$ for a constant $c_d > 0$.

Consistency under RDPG

Corollary (Consistency under RDPG)

Under RDPG with $n\rho_n \geq c_d \log n$. Let $K \leq \hat{K} = O(1)$, then there is a constant c, so that

$$\kappa_A \geq cn \min_{j \in \mathcal{S}} \|M_j\|^2.$$

Therefore, $\mathcal{S}$ can be almost exactly recovered when

$$\min_{j \in \mathcal{S}} \|M_j\| \geq c\sqrt{\log p/n}.$$

- The minimum signal strength is $\|M_j\| = O(\sqrt{\log p/n})$

Degree-Corrected SBM

Degree-Corrected SBM:

$$A_{i,j} \sim Bernoulli(\theta_i \theta_j Y_i^T B Y_j), \qquad Y_i \in \{0,1\}^K.$$

- Y_i is the community membership vector
- $B \in \mathcal{R}^{K \times K}$ is the community by community matrix
- θ_i denotes the heterogeneity among samples

- B has a rank of K
- $n_k/n \geq c > 0$ for each community k
- there is $C > 0$, so that $C\theta_i \geq \max_i \theta_i$ for $i \in [n]$

Consistency under DCSBM

Corollary (Consistency under DCSBM)

Consider DCSBM where $n \max_i \theta_i^2 \geq C \log n$ *for a constant* $C > 0$*. Let* $K \leq \hat{K} = O(1)$*, then there is a constant* c*, so that*

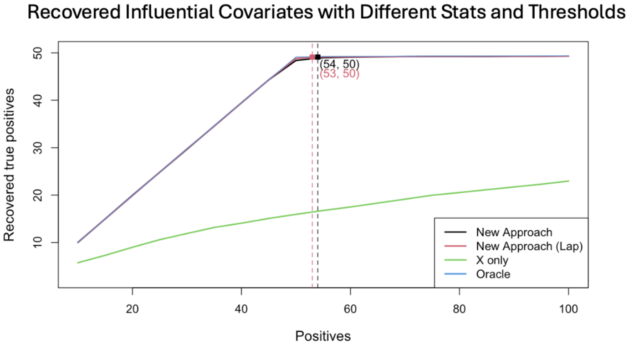
$$\kappa_A \geq cn \min_{j \in \mathcal{S}} \|M_j\|^2.$$

Therefore, $\mathcal{S}$ *can be almost exactly recovered when*

$$\min_{j \in \mathcal{S}} \|M_j\| \geq c\sqrt{\log p/n}.$$

Optimal Threshold: HCT

Set $N = 1000$, $p = 1200$, $K = 3$, $|\mathcal{S}| = 50$ influential covariates.



- Network-guided test stat largely improves the power
- Data-driven HCT is almost optimal