Joint Estimation and Inference for Data Integration Problems based on Multiple Multi-layered Gaussian Graphical Models

Subhabrata Majumdar and George Michailidis

Introduction

Objective

We propose a general statistical framework based on Gaussian graphical models for horizontal (i.e. across conditions or subtypes) and vertical (i.e. across different layers containing data on molecular compartments) integration of information across heterogeneous biological Omics datasets, e.g. genomic, proteomic or metabolomic data.

Contributions

We borrow information across multiple similar multi-layer networks (see figure) to simultaneously perform inference on all model parameters.

- **Estimation:** Incorporate structured sparsity to estimate undirected within-layer edges and directed between-layer edges simultaneously across all networks.
- **Testing:** Develop a debiasing technique and asymptotic distributions of inter-layer directed edge weights, and establish global and simultaneous testing procedures for them.

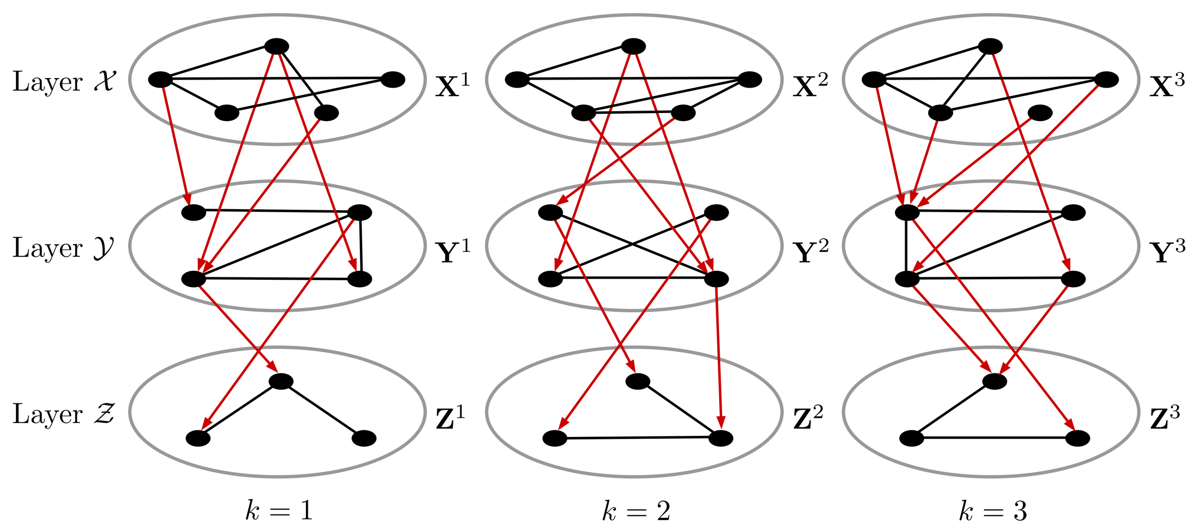

Figure 1: Multiple multi-layer graphical models: $(\mathbf{X}^k, \mathbf{Y}^k, \mathbf{Z}^k), k = 1,2,3$ indicate data for each layer and category k. Within-layer connections (black lines) are undirected, while between-layer connections (red lines) go from an upper layer to the successive lower layer. For each type of edges (i.e. within $\mathcal{X}, \mathcal{Y}, \mathcal{Z}$ and $\mathcal{X} \to \mathcal{Y}, \mathcal{Y} \to \mathcal{Z}$), there are common edges across some or all k.

Statistical model

There are K independent datasets, each pertaining to an M-layered Gaussian Graphical Model (GGM).

Layer 1- $\mathbb{D}_1^k = (D_{11}^k, \ldots, D_{1p_1}^k) \sim \mathcal{N}(0, \Sigma_1^k); \quad k \in \mathcal{I}_K,$

Layer m $(1 < m \leq M)$- $\mathbb{D}_m^k = \mathbb{D}_{m-1}^k \mathbf{B}_m^k + \mathbb{E}_m^k, \mathbf{B}_m^k \in \mathbb{R}^{p_{m-1} \times p_m}$ and $\mathbb{E}_m^k = (E_{m1}^k, \ldots, E_{mp_m}^k) \sim \mathcal{N}(0, \Sigma_m^k)$.

for $k = 1, \ldots, K$. The parameters of interest are the precision matrices $\Omega_m^k := (\Sigma_m^k)^{-1}$ and the regression coefficient matrices $\mathbf{B}_m^k$.

We focus our discussion on jointly estimating $\Omega_y = \{\Omega_y^k\} = \{(\Sigma_y^k)^{-1}\}$ and $\mathcal{B} := \{\mathbf{B}^k\}$ from a two-layer model:

$$\mathbb{X}^k = (X_1^k, \ldots, X_p^k)^T \sim \mathcal{N}(0, \Sigma_x^k); \quad (1)$$

$$\mathbb{Y}^k = \mathbb{X}^k \mathbf{B}^k + \mathbb{E}^k; \quad \mathbb{E}^k = (E_1^k, \ldots, E_p^k)^T \sim \mathcal{N}(0, \Sigma_y^k), \quad (2)$$

using data $\{(\mathbf{Y}^k, \mathbf{X}^k); \mathbf{Y}^k \in \mathbb{R}^{n \times q}, \mathbf{X}^k \in \mathbb{R}^{n \times p}, k = 1, \ldots, K\}$ in presence of known grouping structures $\mathcal{G}_x, \mathcal{G}_y, \mathcal{H}$ respectively. This is because

(a) For $M > 2$, within-layer undirected edges of any m^{th} layer $(m > 1)$ and between-layer directed edges from the $(m-1)^{\text{th}}$ layer to the m^{th} layer can be estimated using the same method (see details in **?**).

(b) Parameters in first layer are analogous to $\{\Omega_x^k\} = \{(\Sigma_x^k)^{-1}\}$, and can be estimated from $\{\mathbf{X}^k\}$ using any method for joint estimation of multiple graphical models (e.g. **??**).

Estimation algorithm

Denote the neighborhood coefficients of the j^{th} variable in the lower layer by $\boldsymbol{\theta}_j^k$, and $\Theta_j := (\boldsymbol{\theta}_j^1, \ldots, \boldsymbol{\theta}_j^K), \Theta = \{\Theta_j\}$. We obtain sparse estimates of $\mathcal{B}, \Theta$, and subsequently Ω_y, by solving the following group-penalized least square minimization problem that has the tuning parameters γ_n and λ_n and then refitting:

$$\{\widehat{\mathcal{B}}, \widehat{\Theta}\} = \underset{\mathcal{B}, \Theta}{\arg\min} \left\{ \frac{1}{n} \sum_{j=1}^{q} \sum_{k=1}^{K} \| \mathbf{Y}_j^k - (\mathbf{Y}_{-j}^k - \mathbf{X}^k \mathbf{B}_{-j}^k) \boldsymbol{\theta}_j^k - \mathbf{X}^k \mathbf{B}_j^k \|^2 \right.$$
$$\left. + \gamma_n P(\Theta) + \lambda_n Q(\mathcal{B}) \right\}, \quad (3)$$

$$\widehat{E}_y^k = \{(j, j') : 1 \leq j < j' \leq q, \hat{\theta}_{jj'}^k \neq 0 \text{ OR } \hat{\theta}_{j'j}^k \neq 0\},$$

$$\widehat{\Omega}_y^k = \underset{\Omega_y^k \in \mathbb{S}_+(\widehat{E}_y^k)}{\arg\min} \left\{ \text{Tr}(\widehat{\mathbf{S}}_y^k \Omega_y^k) - \log\det(\Omega_y^k) \right\}. \quad (4)$$

where $P(\cdot), Q(\cdot)$ are group lasso penalties encoding structured sparsity patterns for respective parameters.

The objective function is bi-convex, and we apply an alternating block algorithm to compute solutions. A high-dimensional BIC is used to select tuning parameters (γ_n, λ_n).

Testing in multi-layer models

Debiased estimator

We define debiased estimates for individual rows of $\mathbf{B}^k$ as

$$\widehat{\mathbf{c}}_i^k = \widehat{\mathbf{b}}_i^k + \frac{1}{n t_i^k} \left(\mathbf{X}_i^k - \mathbf{X}_{-i}^k \widehat{\boldsymbol{\zeta}}_i^k \right)^T (\mathbf{Y}^k - \mathbf{X}^k \widehat{\mathbf{B}}^k); \quad 1 \leq i \leq p \quad (5)$$

where $\widehat{\boldsymbol{\zeta}}_i^k, \widehat{\mathbf{B}}^k$ are *generic estimators* of the neighborhood coefficient matrices in the X-layer and $\mathbf{B}^k$, respectively, $\widehat{\mathbf{b}}_i^k$ is the i^{th} row of $\widehat{\mathbf{B}}^k$, and $t_i^k = (\mathbf{X}_i^k - \mathbf{X}_{-i}^k \widehat{\boldsymbol{\zeta}}_i^k)^T \mathbf{X}_i^k / n$.

Under mild conditions, a centered and scaled $\text{vec}(\widehat{\mathbf{c}}_i^1, \ldots, \widehat{\mathbf{c}}_i^K)$ achieves asymptotic normality. Our estimators conform to these conditions when total sparsity in true values of $\mathcal{B}$ and Ω_y are $o(\sqrt{n}/\log(pq))$ and $o(n/\log(pq))$, respectively.

Pairwise testing

We set K = 2, and are interested in testing whether there are overall and elementwise differences between individual rows of the true coefficient matrices, say $\mathbf{b}_{0i}^1$ and $\mathbf{b}_{0i}^2$.

Algorithm 1. (Global test for $H_0^i : \mathbf{b}_{0i}^1 = \mathbf{b}_{0i}^2$ at level $\alpha \in (0,1)$)

1. Obtain the debiased estimators $\widehat{\mathbf{c}}_i^1, \widehat{\mathbf{c}}_i^2$ using (5).
2. Calculate the test statistic

$$D_i = (\widehat{\mathbf{c}}_i^1 - \widehat{\mathbf{c}}_i^2)^T \left(\frac{\widehat{\Sigma}_y^1}{(m_i^1)^2} + \frac{\widehat{\Sigma}_y^2}{(m_i^2)^2} \right)^{-1} (\widehat{\mathbf{c}}_i^1 - \widehat{\mathbf{c}}_i^2)$$

where $\widehat{\Sigma}_y^k = (\widehat{\Omega}_y^k)^{-1}$, and $m_i^k = \sqrt{n} t_i^k / \hat{s}_i^k$ with $\hat{s}_i^k = \sqrt{\| \mathbf{X}_i^k - \mathbf{X}_{-i}^k \widehat{\boldsymbol{\zeta}}_i^k \|^2 / n}$ for $k = 1, 2$.

3. Reject H_0^i if $D_i \geq \chi^2_{q, 1-\alpha}$.

The above testing procedure maintains rate optimal power, i.e. P(rejection)$\to 1$ when $\|\mathbf{b}_{0i}^1 - \mathbf{b}_{0i}^2\| > O(n^{-1/2})$.

Algorithm 2. (Simultaneous tests for $H_0^{ij} : b_{0ij}^1 = b_{0ij}^2$ at FDR level $\beta \in (0,1)$)

1. Calculate the pairwise test statistics

$$d_{ij} = \frac{\widehat{c}_{ij}^1 - \widehat{c}_{ij}^2}{\sqrt{\hat{\sigma}_{jj}^1 / (m_i^1)^2 + \hat{\sigma}_{jj}^2 / (m_i^2)^2}},$$

with $\hat{\sigma}_{jj}^k$ being the j^{th} diagonal element of $\widehat{\Sigma}_y^k, k = 1, 2$, for $j = 1, \ldots, q$.

2. Obtain the threshold

$$\hat{\tau} = \inf \left\{ \tau \in \mathbb{R} : 1 - \Phi(\tau) \leq \frac{\beta.1 \vee \{\#j : |d_{ij}| \geq \tau)\}}{2q} \right\}.$$

3. For $j = 1, \ldots, q$, reject H_0^{ij} if $|d_{ij}| \geq \hat{\tau}$.

Performance evaluation

Simulation 1: Estimation

- Fix $K = 5$. Take shared groups across k for X- and Y-precision matrices in the setup of **?** and set elements in a group non-zero w.p. π_x. For each element in $\mathcal{B}^k$, entries across k are set to all zero or all non-zero w.p. π_y;
- Non-zero entries in $\mathcal{B}, \Omega_x, \Omega_y$ are drawn from Unif$\{[-1, -0.5] \cup [0.5, 1]\}$ independently.
- 50 replications in each setup of (π_x, π_y, p, q, n).
- Tuning parameters-

$$\gamma_n \in \{0.3, 0.4, ..., 1\} \sqrt{\frac{\log q}{n}};$$

$$\lambda_n \in \{0.4, 0.6, ..., 1.8\} \sqrt{\frac{\log p}{n}}$$

- Compare with separate estimation method of **?**.

(π_x, π_y)	(p,q,n)	Method	TPR($\widehat{\mathcal{B}}$)	TNR($\widehat{\mathcal{B}}$)	MCC($\widehat{\mathcal{B}}$)	RF($\widehat{\mathcal{B}}$)
$(5/p, 5/q)$	(200,200,150)	JMMLE	0.98(0.011)	1.0(0)	0.99(0.005)	0.16(0.025)
		Separate	0.99(0.001)	0.99 (0.001)	0.88(0.009)	0.18(0.007)
	(300,300,150)	JMMLE	1.0(0.001)	1.0(0)	0.99(0.001)	0.14 (0.015)
		Separate	1.0(0.001)	0.99(0.001)	0.84(0.01)	0.21(0.007)
$(30/p, 30/q)$	(200,200,100)	JMMLE	0.97(0.017)	1.0(0)	0.98(0.008)	0.21(0.032)
		Separate	0.32(0.01)	0.99(0.001)	0.49(0.009)	0.85(0.06)
	(200,200,200)	JMMLE	0.99(0.006)	1.0(0)	0.99(0.007)	0.13(0.016)
		Separate	0.97(0.004)	0.98(0.001)	0.93(0.002)	0.19(0.07)
(π_x, π_y)	(p,q,n)	Method	TPR($\widehat{\Theta}$)	TNR($\widehat{\Theta}$)	MCC($\widehat{\Theta}$)	RF($\widehat{\Theta}$)
$(5/p, 5/q)$	(200,200,150)	JMMLE	0.68(0.017)	0.98(0)	0.48(0.013)	0.26(0.002)
		Separate	0.78(0.019)	0.97(0.001)	0.55(0.012)	0.6(0.007)
	(300,300,150)	JMMLE	0.71(0.014)	0.98(0)	0.44(0.008)	0.25(0.002)
		Separate	0.71(0.017)	0.98(0.001)	0.51(0.011)	0.59(0.005)
$(30/p, 30/q)$	(200,200,100)	JMMLE	0.77(0.016)	0.98(0)	0.46(0.013)	0.31(0.003)
		Separate	0.57(0.027)	0.44(0.007)	0.04(0.008)	0.84(0.002)
	(200,200,200)	JMMLE	0.76(0.018)	0.98(0)	0.55(0.015)	0.27(0.004)
		Separate	0.73(0.023)	0.94(0.003)	0.39(0.017)	0.62(0.011)

Table 1: Table of estimation outputs, giving empirical mean and standard deviation (in brackets) of each evaluation metric over 50 replications. TPR = True Positive Rate, TNR = True Negative Rate, MCC = Matthews Correlation Coef., RF = Relative Frobenius Norm Error.

Effect of heterogeneity

- While generating data, individual elements inside a non-zero group are set as 0 with probability 0.2;
- Calculated estimates $\widehat{\mathbf{B}}^k$ are passed through the FDR controlling thresholds (at $\beta = 0.2$):

$$\hat{\tau}_i^k = \inf \left\{ \tau : 1 - \Phi(\tau) \leq \frac{\beta.1 \vee \{\#j : |(\hat{\omega}_{jj}^k)^{1/2} m_i^k \hat{c}_{ij}^k| \geq \tau\}}{2q} \right\},$$

$$\hat{b}_{ij}^{k,\text{thr}} = \hat{b}_{ij}^k \mathbb{I}\left(|(\hat{\omega}_{jj}^k)^{1/2} m_i^k \hat{c}_{ij}^k| \geq \hat{\tau}_i^k \right).$$

- For $\widehat{\mathcal{B}}$, performance is very close to the correctly specified counterparts, but for $\widehat{\Omega}_y$ it is slightly worse.

(π_x, π_y)	(p,q,n)	TPR($\widehat{\mathcal{B}}$)	TNR($\widehat{\mathcal{B}}$)	MCC($\widehat{\mathcal{B}}$)	RF($\widehat{\mathcal{B}}$)
$(5/p, 5/q)$	(200,200,150)	0.99 (0.002)	0.99 (0)	0.98 (0.004)	0.17 (0.007)
	(300,300,150)	0.99 (0.001)	1 (0)	0.99 (0.002)	0.15 (0.006)
$(30/p, 30/q)$	(200,200,100)	0.99 (0.006)	1 (0)	0.98 (0.005)	0.2 (0.014)
	(200,200,200)	0.99 (0.009)	1 (0)	0.98 (0.005)	0.15 (0.017)
(π_x, π_y)	(p,q,n)	TPR($\widehat{\Omega}_y$)	TNR($\widehat{\Omega}_y$)	MCC($\widehat{\Omega}_y$)	RF($\widehat{\Omega}_y$)
$(5/p, 5/q)$	(200,200,150)	0.62 (0.012)	0.98 (0)	0.43 (0.009)	0.27 (0.003)
	(300,300,150)	0.69 (0.013)	0.98 (0)	0.39 (0.008)	0.26 (0.02)
$(30/p, 30/q)$	(200,200,100)	0.78 (0.024)	0.98 (0)	0.43 (0.012)	0.31 0.003)
	(200,200,200)	0.69 (0.026)	0.98 (0.001)	0.5 (0.02)	0.29 (0.004)

Table 2: Table of outputs for joint estimation in presence of group misspecification

Simulation 2: Testing

- Set $K = 2$, then randomly assign each element of $\mathbf{B}_0^1$ as non-zero w.p. π, then draw their values from Unif$\{[-1, -0.5] \cup [0.5, 1]\}$ independently. Then generate a matrix of differences $\mathbf{D}$, where $(\mathbf{D})_{ij}, i \in \mathcal{I}_p, j \in \mathcal{I}_q$ takes values –1, 1, 0 w.p. 0.1, 0.1 and 0.8, respectively. Finally set $\mathbf{B}_0^2 = \mathbf{B}_0^1 + \mathbf{D}$.
- Identical sparsity structures for the pairs of X- and Y-precision matrices.
- Type-I error set at $\alpha = 0.05$, FDR controlled at $\beta = 0.2$.
- Empirical sizes of global tests are calculated from estimators obtained from a separate set of data generated by setting all elements of $\mathbf{D}$ to 0.

(π_x, π_y)	(p,q)	n	Global test		Simultaneous tests	
			Power	Size	Power	FDR
$(5/p, 5/q)$	(60,30)	100	0.977 (0.018)	0.058 (0.035)	0.937 (0.021)	0.237 (0.028)
		200	0.987 (0.016)	0.046 (0.032)	0.968 (0.013)	0.218 (0.032)
	(30,60)	100	0.985 (0.018)	0.097 (0.069)	0.925 (0.022)	0.24 (0.034)
		200	0.990 (0.02)	0.119 (0.059)	0.958 (0.024)	0.245 (0.041)
	(200,200)	150	0.987 (0.005)	0.004 (0.004)	0.841 (0.13)	0.213 (0.007)
	(300,300)	150	0.988 (0.002)	0.002 (0.003)	0.546 (0.035)	0.347 (0.017)
		300	0.998 (0.003)	0.000 (0.001)	0.989 (0.003)	0.117 (0.006)
$(30/p, 30/q)$	(200,200)	100	0.994 (0.005)	0.262 (0.06)	0.479 (0.01)	0.557 (0.006)
		200	0.998 (0.004)	0.020 (0.01)	0.962 (0.003)	0.266 (0.007)
		300	0.999 (0.002)	0.011 (0.008)	0.990 (0.004)	0.185 (0.009)

Table 3: Table of outputs for hypothesis testing.

Computation

- **Initial estimates**: Separate lasso regressions are performed to initialize columns of $\mathbf{B}^k$. Residuals from these initializers are then used in the method of **?** to get initial values of $\widehat{\Theta}$.
- **One-step algorithm**: The original iterative algorithm alternately updates $\widehat{\mathcal{B}}$ and $\widehat{\Theta}$ in each iteration. This becomes too costly for high data dimensions. When p, q become high, we instead use the one-step algorithm: let $\widehat{\mathcal{B}}$ converge completely after initialization, then update $\widehat{\Theta}$ only once. This saves computation time significantly without impacting performance.
- **Update of $\widehat{\mathcal{B}}$**: When updating $\widehat{\mathcal{B}}$ in each iteration, updating columns of the solution matrices sequentially, as well as refitting an OLS estimate on the support set of the penalized solution speeds up convergence.

Discussion

References: