Fast Block-Sparse Estimation for Vector Networks

Zuogong Yue[1]

Padmavathi Sundaram[2], Victor Solo[1,2]

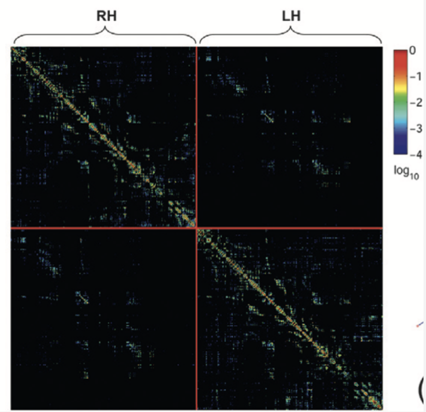
[1] University of New South Wales, Australia
[2] Martinos Center for Biomedical Imaging,
Dept. Radiology, Harvard Medical School, USA

4-8 May · ICASSP 2020 · Virtual Conference

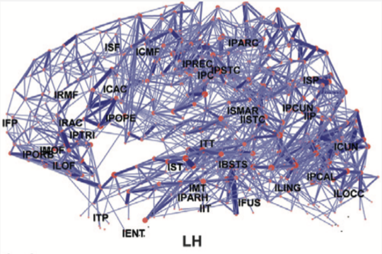
Outline

Motivation

Application: brain networks from human MRI



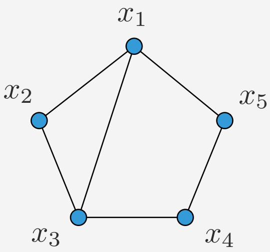
(a) MRI images Fornito et al. (2016)



(b) brain network Fornito et al. (2016)

Gaussian Graphical Models and (scalar) networks:

Given $[x_1, ..., x_5]^T \sim \mathcal{N}(\mu, \Sigma)$,

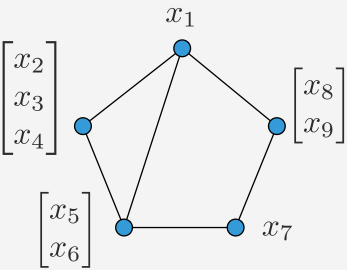


conditional independence graph

$$\Longleftrightarrow \qquad \Sigma^{-1} = \begin{bmatrix} * & * & * & 0 & * \\ * & * & * & 0 & 0 \\ * & * & * & * & 0 \\ 0 & 0 & * & * & * \\ * & 0 & 0 & * & * \end{bmatrix}$$

Introduction (cont.)

"Vector" networks: Kolar, Liu, and Xing (2013), Marjanovic and Solo (2018)



conditional independence graph

$$\Longleftrightarrow \quad \Sigma^{-1} = \left[\begin{array}{c|ccc|cc|c|cc}
* & * & * & * & * & * & 0 & * & * \\ \hline
* & * & * & * & * & * & 0 & 0 & 0 \\
* & * & * & * & * & * & 0 & 0 & 0 \\
* & * & * & * & * & * & 0 & 0 & 0 \\ \hline
* & * & * & * & * & * & * & 0 & 0 \\
* & * & * & * & * & * & * & 0 & 0 \\ \hline
0 & 0 & 0 & 0 & * & * & * & * & * \\ \hline
* & 0 & 0 & 0 & 0 & 0 & * & * & * \\
* & 0 & 0 & 0 & 0 & 0 & * & * & *
\end{array}\right]$$

Problem Description

Data: samples $\{x_1, \ldots, x_T\}$ with $x_t \sim \mathcal{N}(\mu, \Sigma)$.

To estimate a block-sparse inverse covariance matrix, minimize

$$F(\Omega) = -\log\det(\Omega) + \operatorname{tr}(S\Omega) + \lambda \sum_{i \neq j} I(\Omega_{ij} \neq \mathbf{0}) \qquad (*)$$

where

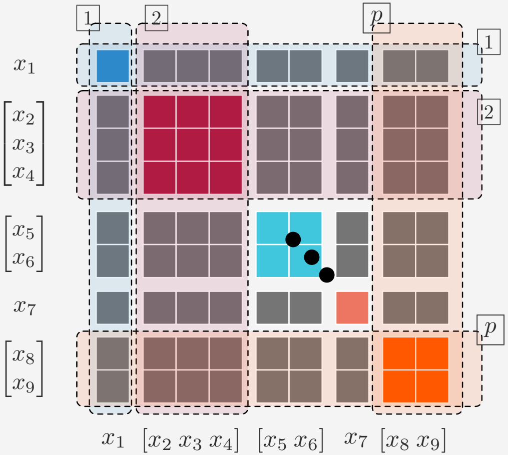
- $S = (1/T)\sum_{t=1}^{T} x_t x_t^T$ sample covariance;
- $I(\Omega_{ij} \neq \mathbf{0}) = \begin{cases} 0 & \text{if } \Omega_{ij} = \mathbf{0}, \\ 1 & \text{otherwise.} \end{cases}$

Outline

Iterative Update

The way to apply *coordinate descent* method:

Ω :

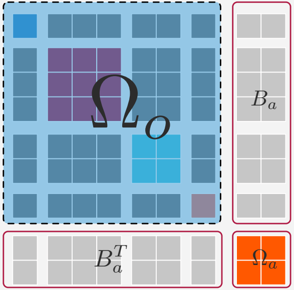


Iterative Update (cont.)

$$\Omega = \begin{bmatrix} \Omega_o & B_a \\ B_a^T & \Omega_a \end{bmatrix}$$

$$S = \begin{bmatrix} S_o & S_{oa} \\ S_{oa}^T & S_a \end{bmatrix}$$

$$\Sigma = \Omega^{-1} = \begin{bmatrix} \Sigma_o & G_a \\ G_a^T & \Sigma_a \end{bmatrix}$$



By omitting the fixed terms, minimize

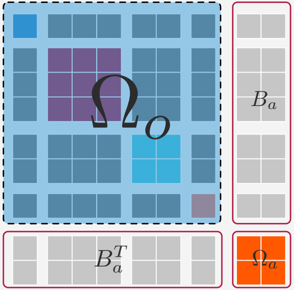
$$F_a(\Omega_a, B_a) = -\log\det(\Omega - B_a^T \Omega_o^{-1} B_a) + 2\,\mathrm{tr}(S_{oa}^T B_a) + \mathrm{tr}(S_a \Omega_a) + 2\lambda \textstyle\sum_{i=1}^{p-1} I(B_{a,i} \neq 0). \qquad (**)$$

Iterative Update (cont.)

$$\Omega = \begin{bmatrix} \Omega_o & B_a \\ B_a^T & \Omega_a \end{bmatrix}$$

$$S = \begin{bmatrix} S_o & S_{oa} \\ S_{oa}^T & S_a \end{bmatrix}$$

$$\Sigma = \Omega^{-1} = \begin{bmatrix} \Sigma_o & G_a \\ G_a^T & \Sigma_a \end{bmatrix}$$



Taking derivative w.r.t. Ω_a and setting to zero, yields

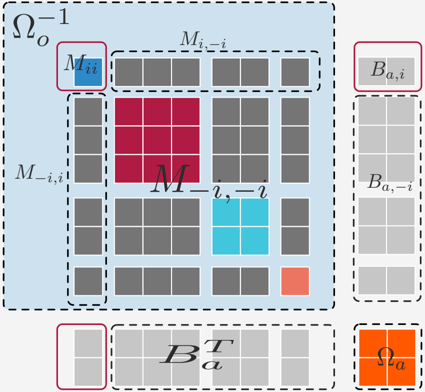
$$F_a(B_a) = \operatorname{tr}(S_a B_a^T \Omega_o^{-1} B_a) + 2\operatorname{tr}(S_{oa}^T B_a) + 2\lambda \textstyle\sum_i I(B_{a,i} \neq 0). \qquad (***)$$

Iterative Update (cont.)

$$B_a = \begin{bmatrix} B_{a,i} \\ B_{a,-i} \end{bmatrix}$$

$$S_{oa} = \begin{bmatrix} S_{oa,i} \\ S_{oa,-i} \end{bmatrix}$$

$$M_o = \Omega_o^{-1} = \begin{bmatrix} M_{ii} & M_{i,-i} \\ M_{-i,i} & M_{-i,-i} \end{bmatrix}$$



Rewriting (***) w.r.t. $B_{a,i}$, it is equivalent to minimize

$$F_a^i(B_{a,i}) = \operatorname{tr}(S_a B_{a,i}^T M_{ii} B_{a,i}) + 2\operatorname{tr}(S_a B_{a,i}^T M_{-i,i} B_{a,-i}) + 2\operatorname{tr}(B_{a,i}^T S_{oa,i}) + 2\lambda I(B_{a,i} \neq 0). \quad (****)$$

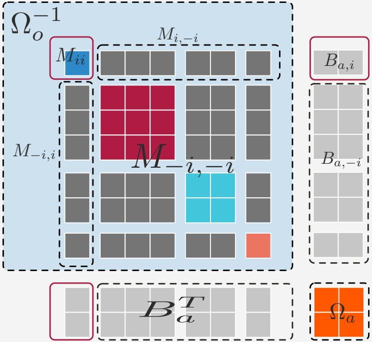
Acceleration (I)

Recap (****)

$$F_a^i(B_{a,i}) = \mathrm{tr}(S_a B_{a,i}^T \boxed{M_{ii}} B_{a,i}) + 2\,\mathrm{tr}(S_a B_{a,i}^T \boxed{M_{-i,i}} B_{a,-i})$$
$$+2\,\mathrm{tr}(B_{a,i}^T S_{oa,i}) + 2\lambda I(B_{a,i} \neq 0),$$

which depends on

$$M_o = \Omega_o^{-1} = \begin{bmatrix} M_{ii} & M_{i,-i} \\ M_{-i,i} & M_{-i,-i} \end{bmatrix}$$



Acceleration (I) (cont.)

Recalling the inverse formula

$$\Omega^{-1} = \begin{bmatrix} \Omega_o^{-1} + \Omega_o^{-1} B_a \Delta_o^{-1} B_a^T \Omega_o^{-1} & -\Omega_o^{-1} B_a \Delta_o^{-1} \\ -\Delta_o^{-1} B_a^T \Omega_o^{-1} & \Delta_o^{-1} \end{bmatrix},$$

we have

$$M_o = \boxed{\Omega_o^{-1}} = \Sigma_o - G_a \boxed{\Sigma_a^{-1}} G_a^T.$$

And we can also update Σ using M_o and submatrices of Ω.

Acceleration (II)

Minimizing (****) is to solve

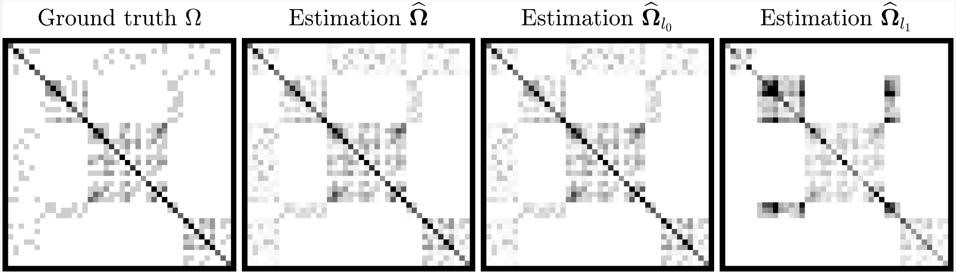
$$M_{ii}B^*_{a,i} = -(M_{i,-i}B_{a,-i}S_a + S_{oa,i})S_a^{-1}.$$

This can be solved fast by matrix *Conjugate Gradient* method.

Outline

Simulations

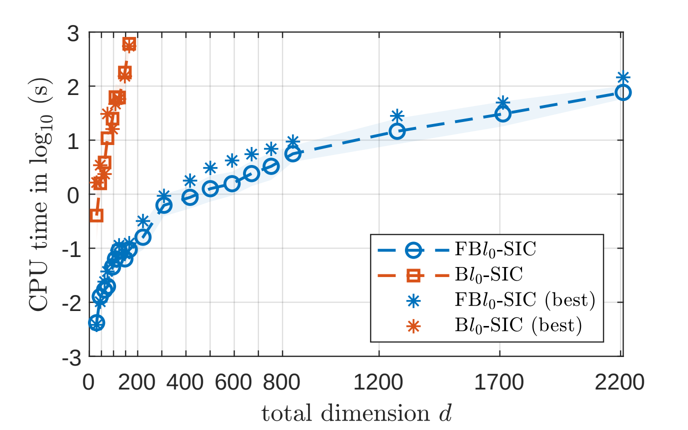
Example in Marjanovic and Solo (2018), no loss of accuracy:



Simulations (cont.)

Speed comparison:

- Marjanovic and Solo (2018): 7 hours to solve 40, 165-dim problems
- Ours: 1 hour to solve 40, 2200-dim problems



Conclusions

Summary of contributions:

- an improved implementation of coordinate descent;
- speed acceleration by smart matrix inversion and matrix CG;
- an algorithm to solve large-scale problems.

Thank you!

Bibliography

Fornito, A., Zalesky, A., & Bullmore, E. (2016). *Fundamentals of Brain Network Analysis.* Academic Press.

Kolar, M., Liu, H., & Xing, E. (2013). Markov Network Estimation From Multi-attribute Data. *Proceedings of the 30th International Conference on Machine Learning*, *28*(3), 73–81. Retrieved from `http://proceedings.mlr.press/v28/kolar13a.html`

Marjanovic, G. & Solo, V. (2018). Vector l_0 Sparse Conditional Independence Graphs. In *2018 ieee international conference on acoustics, speech and signal processing (icassp)* (pp. 2731–2735). IEEE.