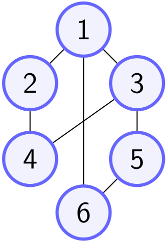
A partial orthogonalization method for simulating covariance and concentration graph matrices

Irene Córdoba[1], Gherardo Varando[1,2], Concha Bielza[1], Pedro Larrañaga[1]

[1]Universidad Politécnica de Madrid
[2]University of Copenhagen

2018

Symmetric positive definite matrices and undirected graphs



$$\begin{pmatrix} \cdot & \cdot & \cdot & 0 & 0 & \cdot \\ \cdot & \cdot & 0 & \cdot & 0 & 0 \\ \cdot & 0 & \cdot & \cdot & \cdot & 0 \\ 0 & \cdot & \cdot & \cdot & 0 & 0 \\ 0 & 0 & \cdot & 0 & \cdot & \cdot \\ \cdot & 0 & 0 & 0 & \cdot & \cdot \end{pmatrix}$$

Given an undirected graph $G = (V, E)$:

- $\mathcal{M}_G = \{\mathbf{M} \in \mathbb{R}^{p \times p}$ s.t $m_{j,i} = m_{i,j} = 0$ if $(i, j) \notin E\}$
- $\mathbb{S}(G) = \mathbb{S} \cap \mathcal{M}_G$ the set of symmetric matrices compatible with the graph G
- $\mathbb{S}^{>0}(G) = \mathbb{S}^{>0} \cap \mathcal{M}_G$ the set of symmetric positive definite matrices compatible with the graph G

Matrices in $\mathbb{S}^{>0}(G)$ parametrize Gaussian graphical models as covariance and concentration models.

To test structure recovering algorithm with synthetic dataset:

1. Generate a random graph G
2. Generate $\mathbf{\Sigma} \in \mathbb{S}^{>0}(G)$
3. Sample data from $\mathcal{N}(\mu, \mathbf{\Sigma})$ (or $\mathcal{N}(\mu, \mathbf{\Sigma}^{-1})$)
4. Use some structure learning algorithm over the generated dataset
5. Compare the known structure G with the recovered one

Diagonally dominant matrices

$$\begin{pmatrix} \cdot & 0.1980 & 0.3760 & 0 & 0 & 0.1797 \\ 0.1980 & \cdot & 0 & 0.4059 & 0 & 0 \\ 0.3760 & 0 & \cdot & 0.3938 & 0.9971 & 0 \\ 0 & 0.4059 & 0.3938 & \cdot & 0 & 0 \\ 0 & 0 & 0.9971 & 0 & \cdot & 0.4481 \\ 0.1797 & 0 & 0 & 0 & 0.4481 & \cdot \end{pmatrix}$$

Diagonal dominance method

- Generate a random matrix in $\mathbb{S}(G)$ (e.g. i.i.d. $U(0,1)$)

Diagonally dominant matrices

$a_{i,i} = \sum_{j \neq i} |a_{i,j}| + \epsilon_i$

$$\begin{pmatrix} 2.5356 & 0.1980 & 0.3760 & 0 & 0 & 0.1797 \\ 0.1980 & 2.3027 & 0 & 0.4059 & 0 & 0 \\ 0.3760 & 0 & 3.4829 & 0.3938 & 0.9971 & 0 \\ 0 & 0.4059 & 0.3938 & 2.0663 & 0 & 0 \\ 0 & 0 & 0.9971 & 0 & 1.7646 & 0.4481 \\ 0.1797 & 0 & 0 & 0 & 0.4481 & 1.0863 \end{pmatrix}$$

Diagonal dominance method

- Generate a random matrix in $\mathbb{S}(G)$ (e.g. i.i.d. $U(0,1)$)
- Choose the diagonal to make the matrix diagonally dominant (plus a random positive perturbation e.g. $U(0,1)$)

- The diagonal dominance method is very fast and easy to implement, thus has been used extensively
- Difficulties arise when using this method in the validation of structure learning algorithm (Kramer *et al.* 2009, Cai *et al.* 2011)
- If $\mathbf{M} \in \mathbb{S}^{>0}(G)$ is generated with the diagonally dominance method then:

$$r_{i,j} = \frac{|m_{i,j}|}{m_{i,i}} \quad \text{are "small"}$$

- In particular if we consider graphs with constant edge density and $p \to \infty$:

$$r_{i,j} < \frac{|m_{i,j}|}{\sum_{j \neq i} |m_{i,j}|} \to 0 \text{ a.s.}$$

Partial orthogonalization

- For every full rank $\mathbf{Q} \in \mathbb{R}^{p \times p}$, $\mathbf{Q}\mathbf{Q}^t \in \mathbb{S}^{>0}(G)$

$$\mathbf{\Sigma} = \mathbf{Q}\mathbf{Q}^t = \begin{pmatrix} q_{1,1} & q_{1,2} & \cdots & q_{1,p} \\ \vdots & \vdots & \vdots & \vdots \\ q_{i,1} & q_{i,2} & \cdots & q_{i,p} \\ \vdots & \vdots & \vdots & \vdots \\ q_{p,1} & q_{p,2} & \cdots & q_{p,p} \end{pmatrix} \begin{pmatrix} q_{1,1} & \cdots & q_{j,1} & \cdots & q_{1,p} \\ \vdots & \vdots & q_{j,2} & \cdots & \vdots \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ q_{1,p} & \cdots & q_{j,p} & \cdots & q_{p,p} \end{pmatrix}$$

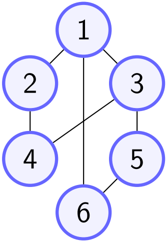
$\mathbf{q}_i$ $\mathbf{q}_j^t$

$$\sigma_{i,j} = \mathbf{q}_i \cdot \mathbf{q}_j$$

$$\sigma_{i,j} = 0 \Leftrightarrow \mathbf{q}_i \perp \mathbf{q}_j$$

- Generate random $\mathbf{Q} \in \mathbb{R}^{p \times p}$ and then orthogonalize the appropriate rows

Partial orthogonalization



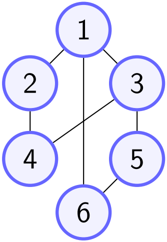
$$\begin{pmatrix} 0.680 & 0.477 & 0.084 & 0.958 & 0.415 & 0.433 \\ 0.555 & 0.991 & 0.535 & 0.300 & 0.654 & 0.041 \\ 0.315 & 0.443 & 0.709 & 0.288 & 0.384 & 0.955 \\ 0.457 & 0.272 & 0.352 & 0.712 & 0.474 & 0.326 \\ 0.189 & 0.437 & 0.274 & 0.796 & 0.548 & 0.571 \\ 0.723 & 0.954 & 0.688 & 0.023 & 0.274 & 0.395 \end{pmatrix}$$

Partial orthogonalization method

- Generate a random matrix $\mathbf{Q} \in \mathbb{R}^{p \times p}$ (e.g. i.i.d. $U(0,1)$)

Partial orthogonalization

$$\mathbf{q}_3 = \mathbf{q}_3 - \frac{\mathbf{q}_3 \cdot \mathbf{q}_2}{||\mathbf{q}_2||^2}\mathbf{q}_2$$

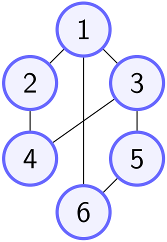


$$\begin{pmatrix} 0.680 & 0.477 & 0.084 & 0.958 & 0.415 & 0.433 \\ 0.555 & 0.991 & 0.535 & 0.300 & 0.654 & 0.041 \\ -0.047 & -0.204 & 0.359 & 0.091 & -0.043 & 0.928 \\ 0.457 & 0.272 & 0.352 & 0.712 & 0.474 & 0.326 \\ 0.189 & 0.437 & 0.274 & 0.796 & 0.548 & 0.571 \\ 0.723 & 0.954 & 0.688 & 0.023 & 0.274 & 0.395 \end{pmatrix}$$

Partial orthogonalization method

- Generate a random matrix $\mathbf{Q} \in \mathbb{R}^{p \times p}$ (e.g. i.i.d. $U(0,1)$)
- For each row $\mathbf{q}_i$, $i = 1, \dots, p$ orthogonalize $\mathbf{q}_i$ with respect to $\langle \mathbf{q}_j$ s.t. $(i,j) \notin E$ and $j < i \rangle$

Partial orthogonalization

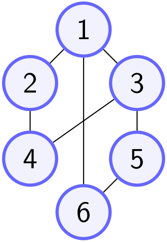


0.680	0.477	0.084	0.958	0.415	0.433
0.555	0.991	0.535	0.300	0.654	0.041
−0.047	−0.204	0.359	0.091	−0.043	0.928
−0.056	−0.088	0.288	−0.011	0.160	−0.001
−0.308	0.140	−0.054	0.026	0.068	0.218
0.188	0.054	0.179	−0.305	−0.246	−0.029

Partial orthogonalization method

- Generate a random matrix $\mathbf{Q} \in \mathbb{R}^{p \times p}$ (e.g. i.i.d. $U(0,1)$)
- For each row $\mathbf{q}_i$, $i = 1, \ldots, p$ orthogonalize $\mathbf{q}_i$ with respect to $\langle \mathbf{q}_j$ s.t. $(i,j) \notin E$ and $j < i \rangle$

Partial orthogonalization

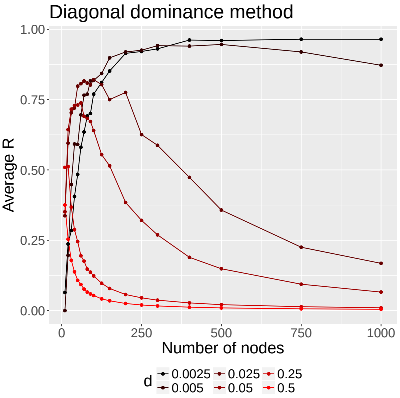


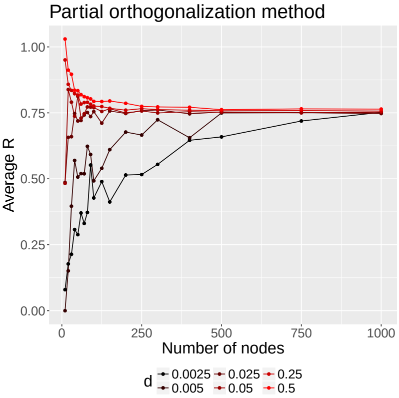
$$\boldsymbol{\Sigma} = \begin{pmatrix} 1.974 & 1.471 & 0.371 & 0 & 0 & -0.238 \\ 1.471 & 2.095 & 0 & 0.137 & 0 & 0 \\ 0.371 & 0 & 1.045 & 0.115 & 0.168 & 0 \\ 0 & 0.137 & 0.115 & 0.120 & 0 & 0 \\ 0 & 0 & 0.168 & 0 & 0.170 & -0.099 \\ -0.238 & 0 & 0 & 0 & -0.099 & 0.228 \end{pmatrix}$$

Partial orthogonalization method

- Generate a random matrix $\mathbf{Q} \in \mathbb{R}^{p \times p}$ (e.g. i.i.d. $U(0,1)$)
- For each row $\mathbf{q}_i$, $i = 1, \ldots, p$ orthogonalize $\mathbf{q}_i$ with respect to $\langle \mathbf{q}_j$ s.t. $(i,j) \notin E$ and $j < i \rangle$
- Compute $\boldsymbol{\Sigma} = \mathbf{Q}\mathbf{Q}^t$

Numerical experiments: $R = \max_{i \neq j} r_{i,j} = \max_{i \neq j} \frac{|\sigma_{i,j}|}{\sigma_{i,i}}$





Numerical experiments: execution time

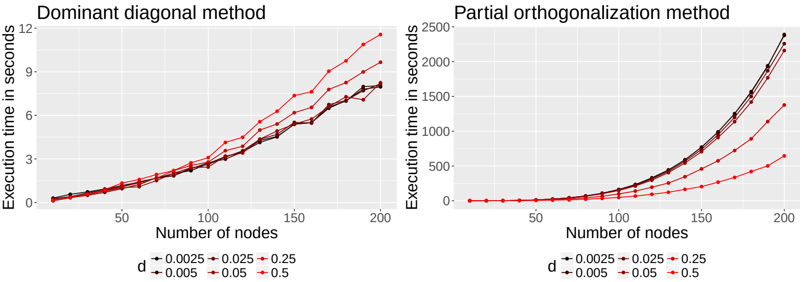
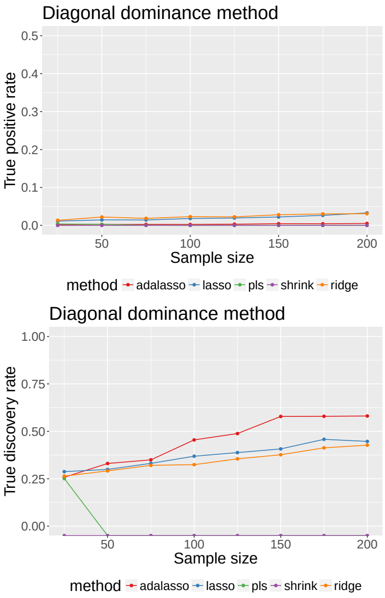


Figure: Execution time to simulate 5000 matrices.

Replication of the comparisons in Kramer *et al.* 2009



Implementation and R package

- The algorithm for partial orthogonalization is **implemented in C**
- Both methods can be found in the R package gmat available on **CRAN**
- The scripts to replicate the experiments are available at `https://github.com/irenecrsn/spdug`

CRAN 0.1.0

downloads 304

```r
# Generate a random undirected graph structure
ug <- igraph::sample_gnp(n = 10, p = 0.25)

# Generate 10 matrices via partial orthogon.
gmat::port(N = 10, ug = ug)

# Generate 10 matrices via diagonal dominance
gmat::diagdom(N = 10, ug = ug)
```

Future work: Implementation and execution time

$\mathbf{Q} \leftarrow$ random $p \times p$ matrix
for $i = 1, \ldots, p$ **do**
    for $j = 1, \ldots, i-1$ **and** $j \not\sim_G i$ **do**
        $\tilde{\mathbf{q}}_j \leftarrow \mathbf{q}_j$
        for $k = 1, \ldots, j-1$ **and** $k \not\sim_G i$ **do**
            $\tilde{\mathbf{q}}_j \leftarrow \tilde{\mathbf{q}}_j - proj_{\tilde{\mathbf{q}}_k}(\tilde{\mathbf{q}}_j)$
        end for
    end for
    for $j = 1, \ldots, i-1$ **and** $j \not\sim_G i$ **do**
        $\mathbf{q}_i \leftarrow \mathbf{q}_i - proj_{\tilde{\mathbf{q}}_j}(\mathbf{q}_i)$
    end for
end for
return $\mathbf{Q}\mathbf{Q}^t$

For every row we need to compute the orthogonal basis of the space

$\langle \mathbf{q}_j : i \not\sim_G j, j < i \rangle$

We can use a part of the basis computed in the previous step

Future work: Induced distribution

- Given an initial random (i.i.d.) matrix how is the induced distribution on $\mathbb{S}^{>0}(G)$?
- How different initial random matrices (e.g. Gaussian, uniform entries) affect the generated matrices?

A partial orthogonalization method for simulating covariance and concentration graph matrices

Irene Córdoba[1], Gherardo Varando[1,2], Concha Bielza[1], Pedro Larrañaga[1]

[1]Universidad Politécnica de Madrid
[2]University of Copenhagen

2018

Thank you for the attention!