Replication files for "Spectral estimation of large stochastic blockmodels with discrete nodal covariates"

Angelo Mele

Package installation

To run this code you need to install the `grdpg` package from Github.

```
library(devtools)
install_github("/meleangelo/grdpg")
```

Examples

The examples in the text are generated with the file

- simulations.R

These results are shown in Tables 1, 2, 3 and 4 in the paper.

Monte carlo

Disclaimer: if you run the Monte Carlo code for replication, please note that each network simulation generates large matrices and may overwhelm your RAM. In the simulations we use a PC with 64GB RAM, and for the simulations with $n = 10000$ we can only use 10 processors. If you use more processors you may run out of memory.

In the Monte Carlo experiments we estimate a model with two binary observed covariates,

$$\mathbf{Z}_i \sim Bernoulli(b_z) \quad and \quad \mathbf{W}_i \sim Bernoulli(b_w) \tag{1}$$

and vary the probabilities b_z and b_w, as well as the correlation among the two variables. We estimate the following model in each Monte Carlo design

$$\log\left(\frac{P_{ij}}{1-P_{ij}}\right) = \mathbf{X}_i^T\mathbf{X}_j + \beta_1 \mathbf{1}_{\{\mathbf{z}_i=\mathbf{z}_j\}} + \beta_2 \mathbf{1}_{\{\mathbf{w}_i=\mathbf{w}_j\}}. \tag{2}$$

The Monte Carlo design considers networks of sizes $n = 2000, 5000, 10000$ and we set the number of blocks to $K = 2$. For all the simulations the parameter value that generates the data is $\beta = (0.5, 0.75)$ and the centers of the blocks are $\nu = (-1.5, 1.0)$.

The Monte Carlo experiments follow 5 different designs, as shown in the following table

Design	π_1	b_z	b_w	correlation
1	0.5	0.5	0.5	independent
2	0.5	0.5	0.5	0.3

Design	π_1	b_z	b_w	correlation
3	0.3	0.5	0.5	independent
4	0.3	0.4	0.6	independent
5	0.3	0.4	0.6	0.3

- Design 1: mc_multiple_covariates.R
- Design 2: mc_multiple_covariates_correlated.R
- Design 3: mc_multiple_covariates_unbalanced.R
- Design 4: mc_multiple_covariates_unbalanced_unbalcov.R
- Design 5: mc_multiple_covariates_unbalanced_unbalcov_correlated.R

These results are shown in Tables 5, 6 and 7 in the paper.

Variance plug-in estimator

The plug-in estimator for the variance is tested using the code

- TestingVarianceFormula.R

The results are in Table 8.

Empirical Application to Facebook data

The code for descriptive statistics and estimation is

- FacebookHarvardLarge.R

and the data are cotained in the Matlab file

- Harvard1.mat

The results are in Table 9 and 10 in the paper.