Graphical Models in Heavy-tailed Markets

a talk by

José Vinícius de M. Cardoso, Jiaxi Ying, and Daniel P. Palomar
The Hong Kong University of Science and Technology

Thirty-fourth Conference on Neural Information Processing Systems (NeurIPS 2021), Virtual

December, 2021

MLE for the Laplacian Matrix

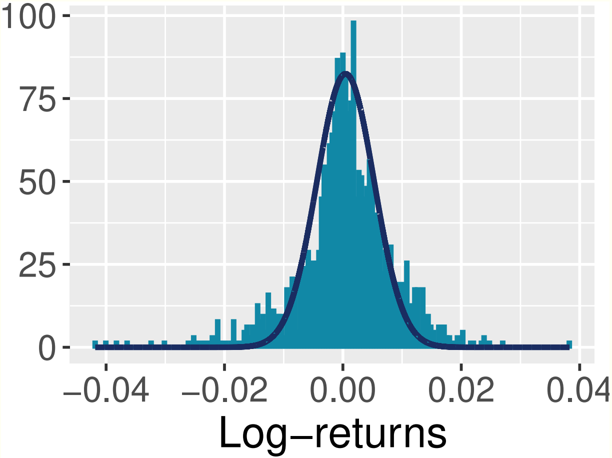
- data generating process: Laplacian constrained Gaussian Markov random field (LGMRF) with rank $p-1$
- its $p \times p$ precision matrix $\boldsymbol{L}$ is modeled as a combinatorial graph Laplacian
- state-of-the-art[1]:

$$\begin{array}{ll} \underset{\boldsymbol{L} \succeq \boldsymbol{0}}{\text{minimize}} & \text{tr}\left(\boldsymbol{L}\boldsymbol{S}\right) - \log\det\left(\boldsymbol{L} + \frac{1}{p}\boldsymbol{1}\boldsymbol{1}^\top\right), \\ \text{subject to} & \boldsymbol{L}\boldsymbol{1} = \boldsymbol{0},\ L_{ij} = L_{ji} \leq 0, \end{array}$$

- where $\boldsymbol{S} = \frac{1}{n}\boldsymbol{X}^\top\boldsymbol{X}$ is the sample covariance matrix
- con: sensitive to outliers or may not be adequate in case $\boldsymbol{X}$ is heavy-tailed distributed

[1] J. Ying, J. V. de M. Cardoso, and D. P. Palomar. Nonconvex sparse graph learning under Laplacian-structured graphical model. In Advances in Neural Information Processing Systems (NeurIPS), 2020

Heavy-tails in Financial Markets



S. I. Resnick. Heavy-Tail Phenomena: Probabilistic and Statistical Modeling. Springer-Verlag New York, 2007.

Proposed Formulations

Student-t Graph Learning Formulation

- assuming $\boldsymbol{x}$ follows a Student-t distribution with positive semidefinite inverse scatter matrix $\boldsymbol{\Theta}$ modeled as a combinatorial graph Laplacian
- the pdf of $\boldsymbol{x}$ is then

$$p(\boldsymbol{x}) \propto \sqrt{\det^*(\boldsymbol{\Theta})}\left(1 + \frac{\boldsymbol{x}^\top \boldsymbol{\Theta} \boldsymbol{x}}{\nu}\right)^{-\dfrac{\nu + p}{2}}, \ \nu > 2$$

- given n realizations of $\boldsymbol{x}$, the robustified version of the MLE for connected graph learning is:

$$\begin{array}{ll} \underset{\boldsymbol{w} \geq \boldsymbol{0}, \boldsymbol{\Theta} \succeq \boldsymbol{0}}{\text{minimize}} & \dfrac{p+\nu}{n} \displaystyle\sum_{i=1}^{n} \log\left(1 + \frac{\boldsymbol{x}_i^\top \mathcal{L}\boldsymbol{w} \boldsymbol{x}_i}{\nu}\right) - \log\det\left(\boldsymbol{\Theta} + \frac{1}{p}\mathbf{1}\mathbf{1}^\top\right), \\ \text{subject to} & \boldsymbol{\Theta} = \mathcal{L}\boldsymbol{w}, \ \mathfrak{d}\boldsymbol{w} = \boldsymbol{d}, \end{array}$$

- where $\mathcal{L}$ is a linear operator that maps a vector of edge weights $\boldsymbol{w}$ into a valid Laplacian matrix and $\mathfrak{d}\boldsymbol{w} \triangleq \text{diag}\left(\mathcal{L}\boldsymbol{w}\right)$

k-component Graphs

- $\text{rank}(\mathcal{L}\boldsymbol{w}) = p - k$
- Fan's[2] theorem:

$$\sum_{i=1}^{k} \lambda_i\left(\mathcal{L}\boldsymbol{w}\right) = \underset{\boldsymbol{V} \in \mathbb{R}^{p\times k}, \boldsymbol{V}^\top \boldsymbol{V} = \boldsymbol{I}}{\text{minimize}} \ \text{tr}\left(\boldsymbol{V}^\top \mathcal{L}\boldsymbol{w}\boldsymbol{V}\right)$$

- k-component heavy-tailed graph learning:

$$\begin{array}{ll} \underset{\boldsymbol{w}\geq\boldsymbol{0}, \boldsymbol{\Theta}\succeq 0, \boldsymbol{V}}{\text{minimize}} & \dfrac{p+\nu}{n}\displaystyle\sum_{i=1}^{n}\log\left(1 + \dfrac{\boldsymbol{x}_i^\top \mathcal{L}\boldsymbol{w}\boldsymbol{x}_i}{\nu}\right) - \log\det^*\left(\boldsymbol{\Theta}\right) + \eta\text{tr}(\mathcal{L}\boldsymbol{w}\boldsymbol{V}\boldsymbol{V}^\top), \\ \text{subject to} & \boldsymbol{\Theta} = \mathcal{L}\boldsymbol{w},\ \text{rank}(\boldsymbol{\Theta}) = p - k, \mathfrak{d}\boldsymbol{w} = \boldsymbol{d},\ \boldsymbol{V}^\top\boldsymbol{V} = \boldsymbol{I},\ \boldsymbol{V} \in \mathbb{R}^{p\times k}. \end{array}$$

- we employ the alternating direction method of multipliers (ADMM) and majorization-minimization (MM) to find stationary points of the proposed optimization problems
- see our supplementary material for convergence proofs 😄

[2] K. Fan. On a theorem of Weyl concerning eigenvalues of linear transformations I. Proceedings of the National Academy of Sciences, 35(11):652–655, 1949.

Experiments

Datasets and Benchmark Algorithms

Datasets (Log-returns)

- US Stock Market ($p = 82$ S&P500 stocks, $n = 1006$ daily observations)
- Foregin Exchange ($p = 34$ currencies, $n = 522$ daily observations)
- Cryptocurrencies ($p = 41$ most traded cryptos, $n = 1218$ daily observations)

Benchmark Models

- sparse models for connected graphs: GLE[3], NGL[4]
- k-component graphs: CLR[5], SGL[6]

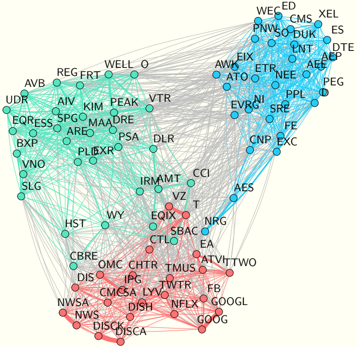
[3] L. Zhao *et al*. Optimization algorithms for graph Laplacian estimation via ADMM and MM. IEEE TSP 2019.

[4] J. Ying *et al*. Nonconvex sparse graph learning under Laplacian-structured graphical model. NeurIPS, 2020.

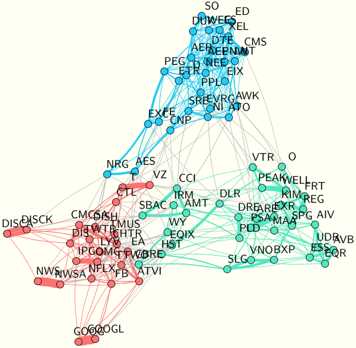
[5] F. Nie *et al*. The constrained Laplacian rank algorithm for graph-based clustering. AAAI, 2016.

[6] S. Kumar *et al*. Structured graph learning via Laplacian spectral constraints. NeurIPS, 2019.

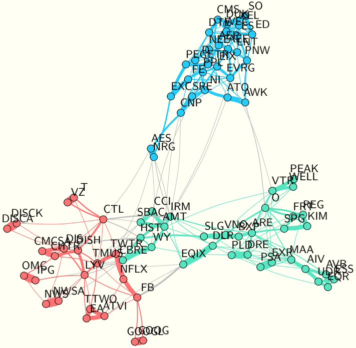
US Stock Market



(a) GLE, modularity $= 0.31$

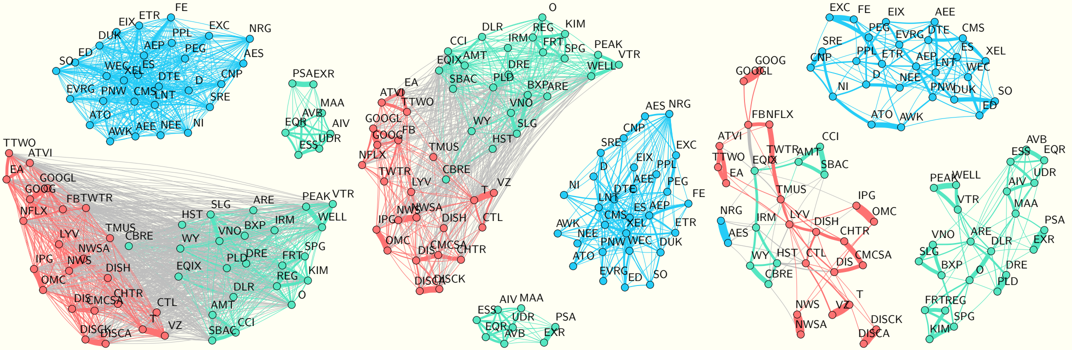


(b) NGL, modularity $= 0.49$



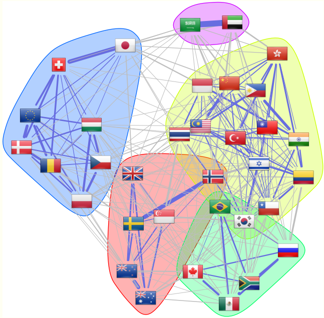
(c) proposed, modularity $= 0.54$

US Stock Market

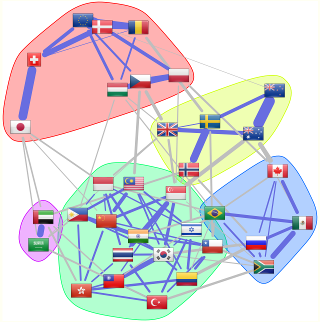


(a) SGL, modularity $= 0.29$ (b) CLR, modularity $= 0.33$ (c) proposed, modularity $= 0.56$

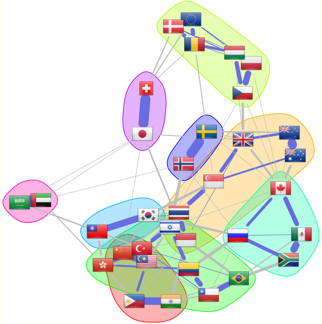
Foreign Exchange



(a) GLE, modularity $= 0.34$

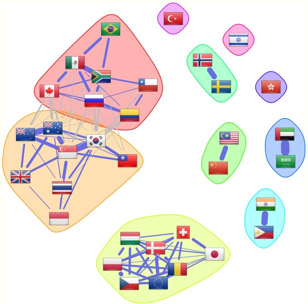


(b) NGL, modularity $= 0.46$

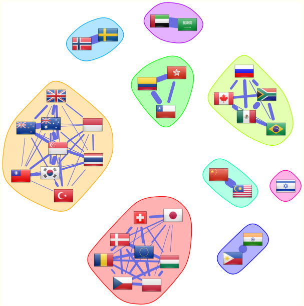


(c) proposed, modularity $= 0.58$

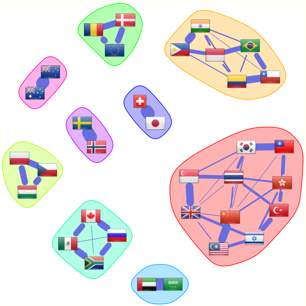
Foreign Exchange



(a) SGL, modularity $= 0.62$

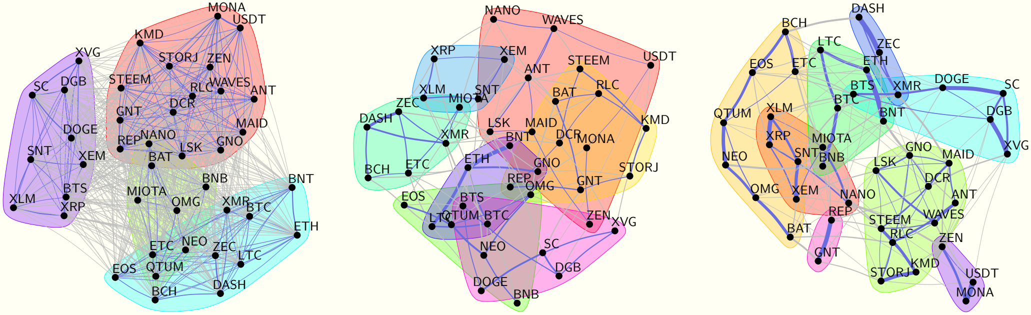


(b) CLR, modularity $= 0.79$



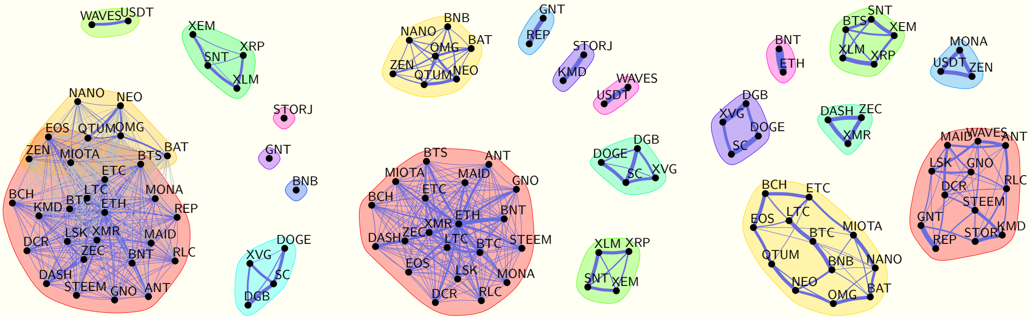
(c) proposed, modularity $= 0.84$

Cryptocurrencies



(a) GLE, modularity $= 0.19$ (b) NGL, modularity $= 0.40$ (c) proposed, modularity $= 0.52$

Cryptocurrencies



(a) SGL, modularity $= 0.36$ (b) CLR, modularity $= 0.66$ (c) proposed, modularity $= 0.79$

Thank You!!