# High Dimensional Models

## Time-Varying Graphical Lasso
## David Hallac, Youngsuk Park, Stephen Boyd, Jure Leskovec (2017)

Andrew Boomer & Jacob Pichelmann

Toulouse School of Economics
M2 EEE

March 18, 2021

# Overview

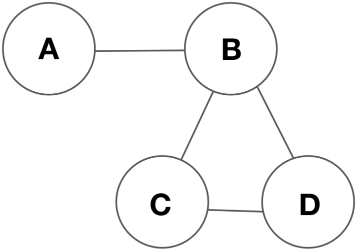
1. Introduction to Graphical Models
    - Important Properties
2. Gaussian Graphical Model
3. Time Varying Graphical Lasso (TGLV)
    - Altered Optimisation Problem
    - ADMM
4. Practical Application of TGVL
    - Comparison to Static Graphical Lasso
    - Changing the penalty function

# Graphical Models

- Graphical models offer a way to encode conditional dependencies between $p$ random variables $X_1, \cdots, X_p$ by a graph $g$
- A graph consists of a vertex set $V = \{1, 2, \cdots, p\}$ and an edge set $E \subset V \times V$
- We focus on undirected graphical models, i.e. no distinction between an edge $(s, t) \in E$ and the edge $(t, s)$.

Consider the following example:

Figure: Undirected Graphical Model



# Factorization Property

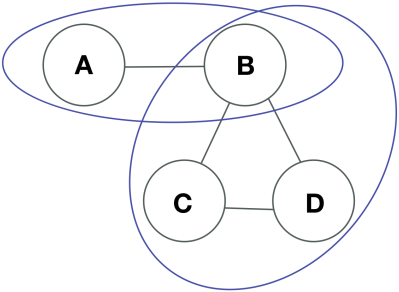
A graph clique $C \subseteq V$ is a fully-connected subset of the vertex set, i.e. $(s, t) \in E \forall s, t \in C$. (Hastie, Tibshirani, & Wainwright, 2015)

$$\mathbb{P}(A, B, C, D) \propto \phi(A, B)\phi(B, C, D)$$

$$\mathbb{P}(X) = \frac{1}{Z} \prod_{c \in C} \phi c(x_c)$$

where $Z = \sum_{x \in X^p} \prod_{c \in C} \phi_c\left(x_c\right)$.

Figure: Maximal Cliques

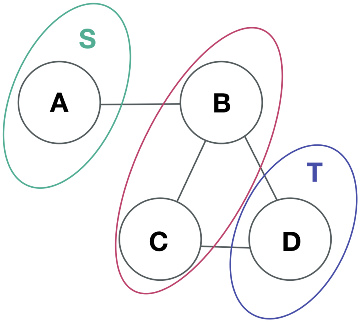


# Markov Property

Any two subsets $S$ and $T$ are conditionally independent given a separating subset $Y$. A random vector $X$ is Markov with respect to $g$ if

$$X_S \perp\!\!\!\perp X_T | X_Y \text{ for all cut sets } S \subset V.$$

Figure: Separating Set: $\{B, C\}$



# Equivalence of Properties

- Hammersley-Clifford theorem:

  *For any strictly positive distribution the distribution of $X$ factorizes according to the graph $g$ if and only if the random vector $X$ is Markov with respect to the graph.* (Hastie et al., 2015)[1]

[1] https://sites.stat.washington.edu/mmp/courses/stat535/fall10/Handouts/l3-mrf.pdf

# Gaussian Graphical Model

$X$ follows a Gaussian distribution:

$$X \sim \mathcal{N}(\mu, \Sigma)$$

If $\Sigma$ is positive definite, distribution has density on $\mathbb{R}^p$

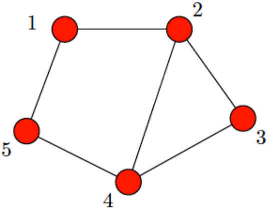
$$f(x \mid \mu, \Sigma) = (2\pi)^{-p/2} (\det \Theta)^{1/2} e^{-(x-\mu)^T \Theta (x-\mu)/2}$$

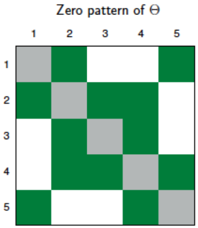
where $\Theta = \Sigma^{-1}$ is the **Precision matrix** of the distribution.

Empirical covariance $S = \frac{1}{n-1} \sum_{i=1}^{n} (x_i - \mu)(x_i - \mu)'$

# Gaussian Graphical Model

We can represent a multivariate Gaussian distribution as a graphical model. Whenever $X$ factorizes according to the graph $g$ we must have $\Theta_{st} = 0$ for any pair $(s, t) \notin E$. This gives a correspondence between the zero pattern of $\Theta$ and the edge structure of $g$.



# Estimating the graph structure $\Leftrightarrow \Theta$

- Suppose $X$ denotes samples from a multivariate Gaussian distribution with $\mu = 0$ and precision matrix $\Theta \in \mathbb{R}^{p \times p}$
- We can write the log-likelihood of the multivariate Gaussian as

$$\mathcal{L}(\Theta; X) = \frac{1}{N} \sum_{i=1}^{N} \log \mathbb{P}_{\Theta}(x_i) = \log det\Theta - trace(S\Theta)$$

- So why not just estimate by MLE to obtain $\widehat{\Theta}_{ML}$?
  1. A sparse graph increases interpretability, prevents overfitting.
  2. In real world applications often times $p > N$, then MLE solution does not exist.

# $\ell_1$ Norm Regularisation

Sparsity can be achieved by adding a penalty term to the optimisation problem. Using the $\ell_1$ norm yields the familiar lasso estimator.

$$\hat{\Theta} = \mathrm{argmin}_{\Theta \geq 0} \left( \mathrm{tr}(S\Theta) - \log\det(\Theta) + \lambda \|\Theta\|_{\mathrm{od},1} \right)$$

where $\|\Theta\|_{\mathrm{od},1}$ is the $\ell_1$-norm of the off-diagonal entries of $\Theta$.

# Challenge: The Network Structure Can Change Over Time

In many real world settings (e.g. financial markets) the structure of the complex system changes over time.

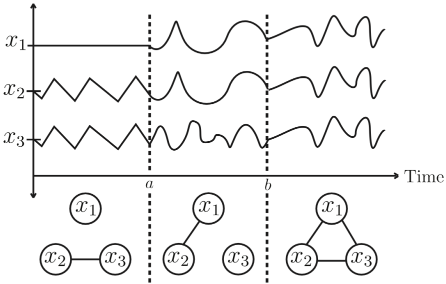



Figure: Example of Changing Network Structure (Hallac et al., 2017)

# Solution: Optimization on a Chain Graph (TVGL)

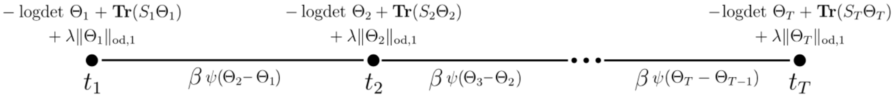



Figure: (Hallac et al., 2017)

$$\underset{\Theta \in S^p_{++}}{\text{minimize}} \quad \sum_{i=1}^{T} Tr(S_i\Theta_i) - logdet(\Theta_i) + \lambda\,\|\Theta_i\|_{\text{od},1} + \beta \sum_{i=2}^{T} \psi\left(\Theta_i - \Theta_{i-1}\right)$$

- $\psi$ is the function applied the change in the graph structure
- $\beta$ is the penalty parameter applied to sum of $\psi$ functions

# Choice of $\psi$

1. **A few edges changing at a time** - $\psi(X) = \sum_{i,j} |X_{i,j}|$
   - Encourages neighboring graphs to be identical
   - Best used when only a few nodes are expected to change
2. **Smoothly varying over time** - $\psi(X) = \sum_{i,j} X_{i,j}^2$
   - Causes smooth transition of graphical models
   - Severe deviations penalized (sum of squares)
3. **Perturbed node** - $\psi(X) = \min_{V:V+V^T=X} \sum_j \left\|[V]_j\right\|_2$
   - Allows single node to change all edge relationships at once with minimal penalty
   - Used when looking for single node restructuring

# Optimization Algorithm: ADMM

- ADMM (Alternating Direction Method of Multipliers) is an general technique that can be used on any convex optimization problem.
- ADMM has a couple main advantages compared to standard gradient descent based methods: (1) Can be applied to nonsmooth functions, (2) Can be distributed across multiple independent machines
- To put ADMM into context, we show how it can be used to solve a generic optimization problem

# Optimization Algorithm: ADMM

## General Example

We can take the generic minimization problem

$$\underset{x}{argmin}\, f(x) \quad s.t.\ x \in \mathcal{C}$$

And separate it into two functions, $f$ and $g$, where $g$ is the indicator of $\mathcal{C}$

$$\underset{x}{argmin}\, f(x) + g(z) \quad s.t.\ x - z = 0$$

The variable $z$ is known as a consensus variable, and the constraint ensures final convergence between $x$ and $z$

# Optimization Algorithm: ADMM

## Proximal Operators/Proximal Gradient Descent

ADMM optimization used by authors relies on proximal gradient descent. Proximal gradient descent uses proximal operators, defined as:

$$\mathrm{prox}_{\rho f}(v) = \operatorname*{argmin}_x \left( f(x) + \frac{1}{\rho} \|x - v\|_2^2 \right)$$

The ADMM iteration based update method is:

$$\begin{aligned} x^{k+1} &:= \operatorname*{argmin}_x \left( f(x) + (\rho/2) \left\| x - z^k + y^k \right\|_2^2 \right) \\ z^{k+1} &:= \Pi_C \left( x^{k+1} + y^k \right) \\ y^{k+1} &:= y^k + \rho(x^{k+1} - z^{k+1}) \end{aligned}$$

Iterations stop when $y^k \to y^{k+1}$ ($x - z = 0$ constraint satisfied)

# Optimization Algorithm: ADMM

## TVGL ADMM Application Overview

For the TVGL, the authors introduce 3 consensus variables: $(Z_0, Z_1, Z_2)$

1. $Z_0$ is the consensus variable for the $\Theta_i$ within $|\Theta_i|_{od,1}$
2. $(Z_1, Z_2)$ correspond to $(\Theta_i, \Theta_{i-1})$ within $\Psi(\Theta_i - \Theta_{i-1})$

The augmented lagrangian for the TVGL then is:

$$
\begin{aligned}
\mathcal{L}_\rho(\Theta, Z, U) = \sum_{i=1}^{T} -l\left(\Theta_i\right) + \lambda \left\|Z_{i,0}\right\|_{\text{od, }1} + \beta \sum_{i=2}^{T} \psi\left(Z_{i,2} - Z_{i-1,1}\right) \\
+ (\rho/2) \sum_{i=1}^{T} \left( \left\|\Theta_i - Z_{i,0} + U_{i,0}\right\|_F^2 - \left\|U_{i,0}\right\|_F^2 \right) \\
+ (\rho/2) \sum_{i=2}^{T} \left( \left\|\Theta_{i-1} - Z_{i-1,1} + U_{i-1,1}\right\|_F^2 - \left\|U_{i-1,1}\right\|_F^2 \right. \\
\left. + \left\|\Theta_i - Z_{i,2} + U_{i,2}\right\|_F^2 - \left\|U_{i,2}\right\|_F^2 \right)
\end{aligned}
$$

# Optimization Algorithm: ADMM

## TVGL ADMM Application Overview

Finally, the update procedure for the $k^{th}$ iteration in the TVGL is

(a) $\Theta^{k+1} := \underset{\Theta \in S^p_{++}}{\operatorname{argmin}} \mathcal{L}_\rho \left( \Theta, Z^k, U^k \right)$ (b)

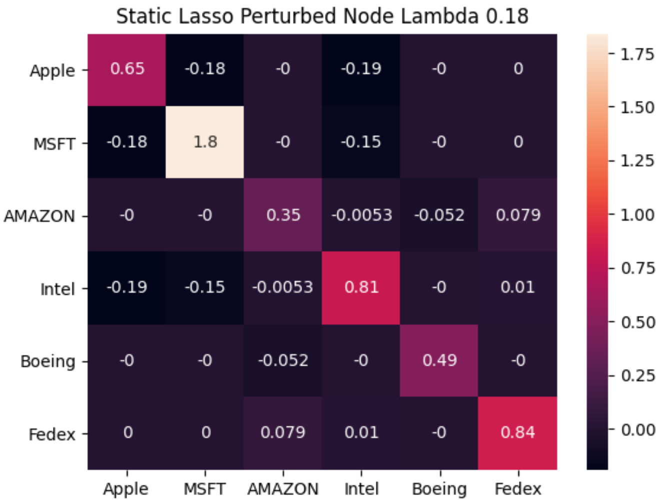
$$Z^{k+1} = \begin{bmatrix} Z_0^{k+1} \\ Z_1^{k+1} \\ Z_2^{k+1} \end{bmatrix} := \underset{Z_0, Z_1, Z_2}{\operatorname{argmin}} \mathcal{L}_\rho \left( \Theta^{k+1}, Z, U^k \right) \text{ (c)}$$

$$U^{k+1} = \begin{bmatrix} U_0^{k+1} \\ U_1^{k+1} \\ U_2^{k+1} \end{bmatrix} := \begin{bmatrix} U_0^k \\ U_1^k \\ U_2^k \end{bmatrix} + \begin{bmatrix} \Theta^{k+1} - Z_0^{k+1} \\ \left( \Theta_{1+1}^{k+1}, \ldots, \Theta_{T-1}^{k+1} \right) - Z_1^{k+1} \\ \left( \Theta_2^{k+1}, \ldots, \Theta_T^{k+1} \right) - Z_2^{k+1} \end{bmatrix}$$

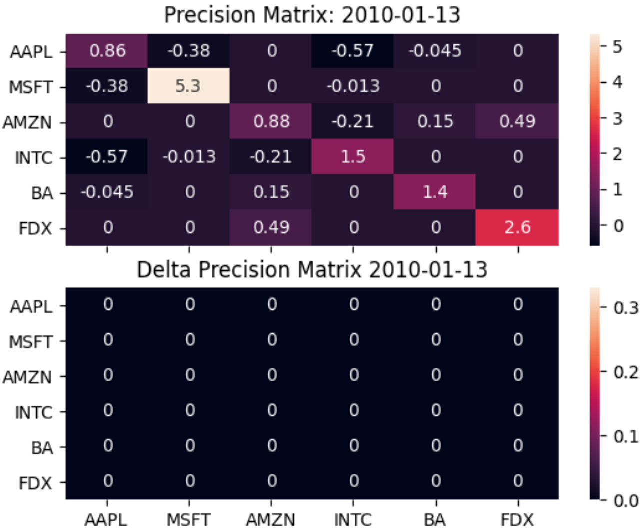
# TVGL Application: Data

- Replication of authors' TVGL application to stock price data
- Panel of six stocks (labeled in graphs), comprising trading days from 13/01/2010 to 19/03/2010
- Authors application focuses on changes in the network structure of Apple in the graph over time

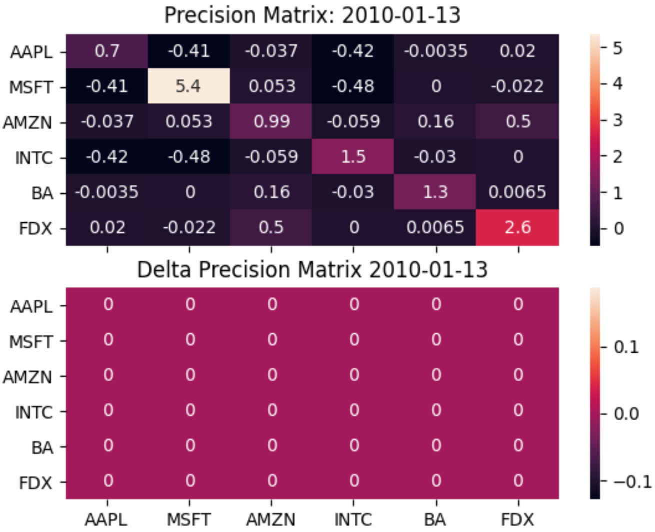
# Static LASSO



# TVGL (Perturbed Node)



# TVGL (Smoothly Varying)



# Changing $\psi$

## Temporal Deviation of Precision Matrix

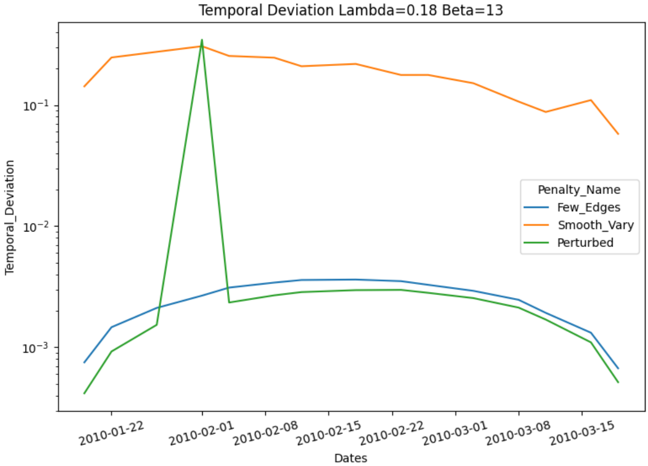



Figure: Temporal Deviation Psi Comparison

# Changing $\psi$

## Temporal Deviation of Precision Matrix

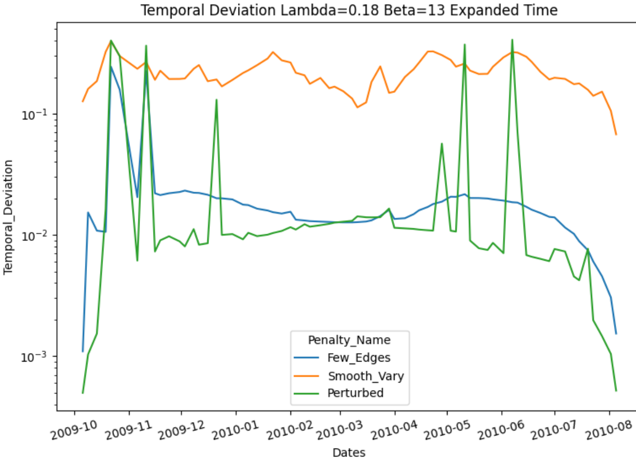



Figure: Temporal Deviation Psi Comparison Expanded Timespan

# Changing $\psi$

## Temporal Deviation of Precision Matrix

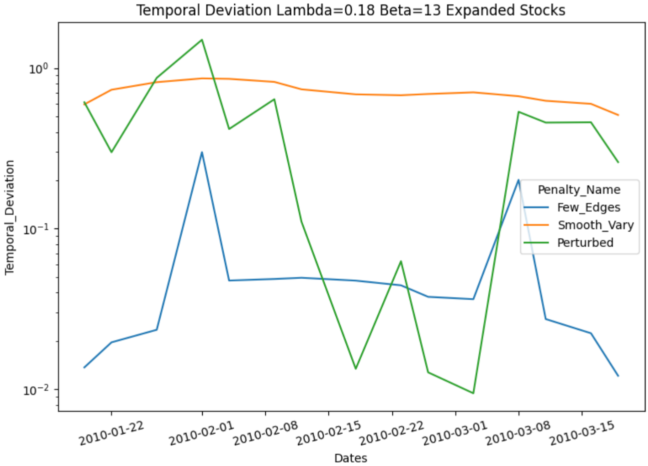



Figure: Temporal Deviation Psi Comparison Expanded Stock Set

# References


Boyd, S., Parikh, N., & Chu, E. (2011). *Distributed optimization and statistical learning via the alternating direction method of multipliers*. Now Publishers Inc.

Friedman, J., Hastie, T., & Tibshirani, R. (2008). Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, *9*(3), 432–441.

Giraud, C. (2014). *Introduction to high-dimensional statistics* (Vol. 138). CRC Press.

Hallac, D., Park, Y., Boyd, S., & Leskovec, J. (2017). Network inference via the time-varying graphical lasso. , 205–213.

Hastie, T., Tibshirani, R., & Wainwright, M. (2015). *Statistical learning with sparsity: the lasso and generalizations*. CRC press.