Time-Varying Graphical Lasso

Group Members: *Davide Messina, Matteo Lenzi, Fabio Marcaurelio*
Graphical Models - FISE 2024-2025
Instructor: *Phuong Thuy Vo*

Date: January 26, 2025

Abstract

This document explores the Time-Varying Graphical Lasso (TVGL) method for dynamic network inference, based on the work of Hallac et al. Our group chose this topic for its relevance in analyzing evolving systems, particularly in contexts like finance. We present the mathematical formulation, implementation details, and experimental results derived from applying the algorithm to real-world datasets.

Contents

1 Introduction **2**

2 Literature Review **2**
2.1 TVGL 3

3 Methodology **4**
3.1 Mathematical Formulation 4
3.2 Assumptions and possible extensions 7

4 Implementation **7**
4.1 Data 7
4.2 Precision Matrices Before, During, and After COVID 8
4.3 Graphs Before, During, and After COVID 9
4.4 Bayesian Optimization for Hyperparameter Tuning 10

5 Conclusion **11**

1 Introduction

2 Literature Review

In financial markets and other fields, series of multivariate observations are often time-stamped, meaning each variation of a variable is recorded alongside its precise time of occurrence. For example, in the stock market, we might analyze the time-dependent changes in the prices of different shares, exploring how they evolve and whether dependencies exist between them.

A possible way to model such dependencies is through Markov Random Fields (MRFs). These are mathematical frameworks where a set of random variables is structured according to a graph that is undirected and follows specific Markov properties. Unlike Bayesian networks, which are directed and acyclic, MRFs allow for the representation of dependencies that may form cycles, making them particularly versatile for capturing relationships that Bayesian networks cannot. MRFs can represent a joint probability distribution of variables, offering flexibility in modeling dependencies. Fonte: https://www.cs.umd.edu/ gihan/resources/em509/project/markov-random-fields.pdf

To better understand this concept, let's consider an example based on Figure 1. Suppose $x_1, x_2, andx_3$ represent the stock prices of three assets changing over time. Correspondingly, for each time period, we can construct an MRF where the nodes represent the three companies, and the edges indicate dependencies between them. If, during a specific time interval, we observe similarities in the movement of two stock prices, we connect the corresponding nodes in the graph. While this can be done intuitively in simple cases, dealing with a large volume of data requires inferring a **time-varying network** that uncovers all possible dependencies.

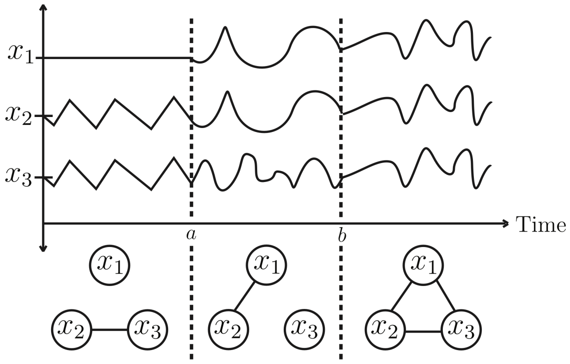


Figure 1: Enter Caption

To build a time-varying network, relationships between entities are modeled using a time-varying **inverse covariance matrix** ($\Sigma^{-1}(t)$), which captures the conditional dependencies between variables. If an element of the matrix $\Sigma^{-1}_{ij}(t)$ equals zero, nodes i and j are *conditionally independent* at time t, while a nonzero value indicates a *structural dependency* between them. Thus, an edge exists between nodes i and j at time t if $\Sigma^{-1}_{ij}(t) \neq 0$, representing a dynamic dependency.

- **Static Networks**: The matrix Σ^{-1} is constant over time. In this case, the network inference is known as the *graphical lasso problem*, and efficient algorithms exist to solve it.

- **Dynamic Networks**: The matrix $\Sigma^{-1}(t)$ changes over time, reflecting evolving dependencies. Standard graphical lasso methods are not suitable here and solving for time-varying networks is computationally more expensive, requiring specialized approaches like *Time-Varying Graphical Lasso (TVGL).*

In a static network, relationships between entities are fixed. For example, in financial markets, you could calculate the correlation between the stock prices of different companies over a specific period, assuming it remains constant. This would form a static network, where edges represent fixed dependencies between companies.

In contrast, for dynamic networks, where dependencies evolve, the inverse covariance matrix changes over time. For example, the correlation between stock prices might fluctuate due to market conditions. To model these dynamic dependencies, methods like TVGL are used.

2.1 TVGL

The Time-Varying Graphical Lasso (TVGL) is a method designed to estimate the inverse covariance matrix, $\Sigma^{-1}(t)$, of a multivariate data sequence. This estimation simultaneously pursues three primary objectives:

1. **Matching Empirical Observations:** This ensures that the estimated networks are consistent with the observed data. By aligning with the empirical covariance, TVGL provides a robust foundation for understanding the structure of the data.

2. **Sparsity:** Imposing sparsity on the inverse covariance matrix prevents overfitting, which is crucial when dealing with high-dimensional data. Sparse representations also enhance interpretability, as they naturally lead to graphical models where connections between nodes are minimized, highlighting only the most significant relationships.

3. **Temporal Consistency:** Recognizing that adjacent time points in many real-world applications often exhibit similar underlying structures, TVGL enforces temporal smoothness. This ensures that the networks do not vary arbitrarily across time but rather evolve in a controlled manner.

To achieve temporal consistency, TVGL incorporates penalties that regulate the degree of change in the network structure over time. These penalties are designed to model different dynamic patterns in the temporal evolution of the networks. Five distinct types of penalties will be presented, each imposing a unique constraint on temporal evolution, enabling TVGL to accommodate a diverse array of real-world applications.

To solve this optimization problem efficiently, since no scalable methods exist for solving it, TVGL employs the Alternating Direction Method of Multipliers (ADMM). This scalable message-passing algorithm breaks the problem into smaller subproblems, each of which can be solved analytically. Closed-form solutions are derived for the subproblems associated with each penalty type, significantly accelerating computation.

3 Methodology

3.1 Mathematical Formulation

Optimization Approach

The estimation problem in TVGL is formulated as minimizing an objective function that balances three key components: data fidelity, sparsity, and temporal regularization. Mathematically, this can be expressed as:

$$\min_{\Theta_t \in \mathbb{S}_+^p} \sum_{t=1}^{T} [-\log\det(\Theta_t) + \operatorname{Tr}(S_t\Theta_t)] + \lambda \sum_{t=1}^{T} \|\Theta_t\|_{od,1} + \beta \sum_{t=2}^{T} \psi(\Theta_t - \Theta_{t-1}),$$

where S_t denotes the empirical covariance matrix, λ governs the level of sparsity, and β controls the temporal penalty ψ. In the following sections, we will derive this function step by step.

Inference Framework

The inference process for time-varying networks involves the following key steps:

- **Input:** Multivariate time series data $\{x_t\}$, typically organized as a $T \times N$ matrix, where T is the number of time steps and N is the number of variables.
- **Output:** A sequence of graph structures Θ_t that represent the evolving relationships among variables over time.
- **Method:** The procedure consists of two main steps:
  1. Estimation of the precision matrix $\Sigma^{-1}(t)$, that is the inverse of the covariance matrix $\Sigma(t)$, for each time step t.
  2. Application of penalties to enforce *sparsity* and *temporal coherence* in the inferred graphs.

The focus is on balancing the fidelity of the estimated graphs with their interpretability and temporal consistency.

Step 1: Precision Matrix Estimation

The precision matrix $\Theta = \Sigma^{-1}$ provides information about conditional independence between variables. Estimating it involves using the empirical covariance matrix S and incorporating constraints like sparsity through optimization methods.

Given a multivariate time series data $\{x_t\}_{t=1}^{T}$, the sample covariance matrix is computed as:

$$S_t = \frac{1}{n} \sum_{i=1}^{n} x_{i,t} x_{i,t}^{\mathrm{T}},$$

where n is the number of samples at each time step.

The optimization problem for each t is expressed as:

$$\min_{\Theta \in S_{++}^{p}} -\log\det(\Theta_t) + \mathrm{Tr}(S_t \Theta_t) + \lambda \|\Theta\|_{od,1},$$

where:

- S_{++}^{p} denotes the space of positive-definite matrices.
- λ is a regularization parameter that controls the sparsity of Θ.
- The term $-\log\det(\Theta_t)$ enforces Θ_t to remain positive definite.
- $\mathrm{Tr}(S_t\Theta_t)$ ensures a connection between Θ_t and the empirical covariance matrix S_t.
- $\|\Theta\|_{od,1}$ is the sparsity-inducing penalty, which encourages a sparse structure in Θ.

The combined term $-\log\det(\Theta_t) + \mathrm{Tr}(S_t\Theta_t)$ ensures that Θ_t aligns well with the empirical data. The parameter λ plays a crucial role in determining the sparsity of the solution:

- A high value of λ promotes greater sparsity, resulting in fewer connections in the estimated network.
- A low value of λ reduces sparsity, allowing more connections in the network structure.

Step 2: Penalties for Sparsity and Temporal Coherence

Sparsity Penalty

To ensure sparsity, an ℓ_1 penalty is applied:

$$\|\Theta(t)\|_{od,1} = \sum_{i \neq j} |\Theta_{ij}(t)|,$$

which acts on the off-diagonal elements of the precision matrix. This penalty encourages many elements of Θ to become exactly zero, reducing the complexity of the graph.

Temporal Coherence Penalty

Temporal coherence is enforced using a penalty term $\psi(X)$, where $X = \Theta(t), \Theta(t+1)$, which encourages smooth transitions between consecutive graphs. Various forms of this penalty can be used:

1. ℓ_1 penalty: $\psi(X) = \sum_{i,j} \|X_{i,j}\|$
   Encourages sparsity by allowing only a few edges to change at each timestamp, making it suitable for scenarios with minimal structural variations.

2. ℓ_2 penalty (Global restructuring): $\psi(X) = \sum_j \|[X]_j\|_2$
   Facilitates global restructuring, enabling the detection of significant regime shifts by encouraging entire graphs to remain constant except at a few timestamps.

3. Laplacian penalty: $\psi(X) = \sum_{i,j} X_{i,j}^2$
   Promotes smooth transitions by penalizing large deviations between consecutive graphs, ideal for smoothly evolving network structures.

4. ℓ_∞ penalty: $\psi(X) = \sum_j (\max \|X_{i,j}\|)$
   Allows block-wise restructuring by permitting clusters of nodes to change their internal connections while keeping the rest of the network constant.

5. Perturbed node: $\psi(X) = \min \sum \|[V]_j\|_2$
   Encourages minimal penalty when a node rewires all its edges at a given time, while the rest of the network remains unchanged.

The choice of penalty depends on the specific application and desired properties of the inferred graphs.

Similar to the parameter λ, the choice of the parameter β is equally crucial, as it governs the temporal coherence of the solution:

- A high value of β ensures that the network evolves slowly over time, resulting in greater temporal consistency across adjacent time steps.
- A low value of β allows the network to change more rapidly, leading to reduced temporal coherence but accommodating faster dynamics in the network structure.

Complete TVGL Problem

In conclusion, the complete optimization problem for TVGL is:

$$\min_{\Theta_t \in \mathbb{S}_+^p} \sum_{t=1}^{T} [-\log\det(\Theta_t) + \mathrm{Tr}(S_t\Theta_t)] + \lambda \sum_{t=1}^{T} \|\Theta_t\|_{od,1} + \beta \sum_{t=2}^{T} \psi(\Theta_t - \Theta_{t-1}),$$

This formulation effectively balances the trade-off between sparsity and temporal coherence, allowing for the inference of interpretable and consistent dynamic networks.

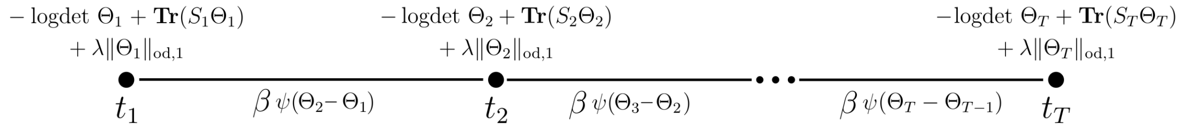

$-\,\text{logdet}\ \Theta_1 + \mathbf{Tr}(S_1\Theta_1)$
$+\,\lambda\|\Theta_1\|_{\text{od},1}$
$-\,\text{logdet}\ \Theta_2 + \mathbf{Tr}(S_2\Theta_2)$
$+\,\lambda\|\Theta_2\|_{\text{od},1}$
$-\text{logdet}\ \Theta_T + \mathbf{Tr}(S_T\Theta_T)$
$+\,\lambda\|\Theta_T\|_{\text{od},1}$
t_1
$\beta\,\psi(\Theta_2-\Theta_1)$
t_2
$\beta\,\psi(\Theta_3-\Theta_2)$
$\beta\,\psi(\Theta_T-\Theta_{T-1})$
t_T

Figure 2: Enter Caption

3.2 Assumptions and possible extensions

In this study, we assume that the readings are synchronous, meaning that $t_i - t_{i-1}$ is constant for all i. While the original paper presents advanced extensions of the time-varying graphical lasso to address challenges such as asynchronous observations, intermediate network inference, and real-time streaming updates, these methods are not the focus of our work. This choice was guided by the nature of our dataset, as described in Section 4?, which consists of uniformly sampled observations that do not require high-frequency updates or intermediate inferences. Additionally, our primary aim was to capture the broader structural evolution of the network rather than focusing on fine-grained or real-time adjustments, rendering these advanced methods unnecessary for our specific study.

4 Implementation

In this work, the TVGL model is applied to a set of global stock indices to analyze their dynamics before, during, and after the COVID-19 pandemic. The original implementation of the model was in Python 2, but for this project, the TVGL file was converted to Python 3 to ensure compatibility with modern libraries and environments. The goal is to study how the precision matrices of these indices have evolved over time, particularly focusing on the structural changes that occurred during the pandemic. This analysis is crucial to understanding the shifts in market relationships and volatility as a result of global economic disruptions caused by the pandemic, for this reason in our analysis of stock market indices, we employ the ℓ_2 penalty due to its ability to detect abrupt structural changes and regime shifts in the network. This approach is well-suited for understanding significant transitions in financial markets, where global restructuring often reflects key economic events or market dynamics. The implementation of the model includes various steps, from the preparation of historical data to the optimization of model parameters using a Bayesian algorithm.

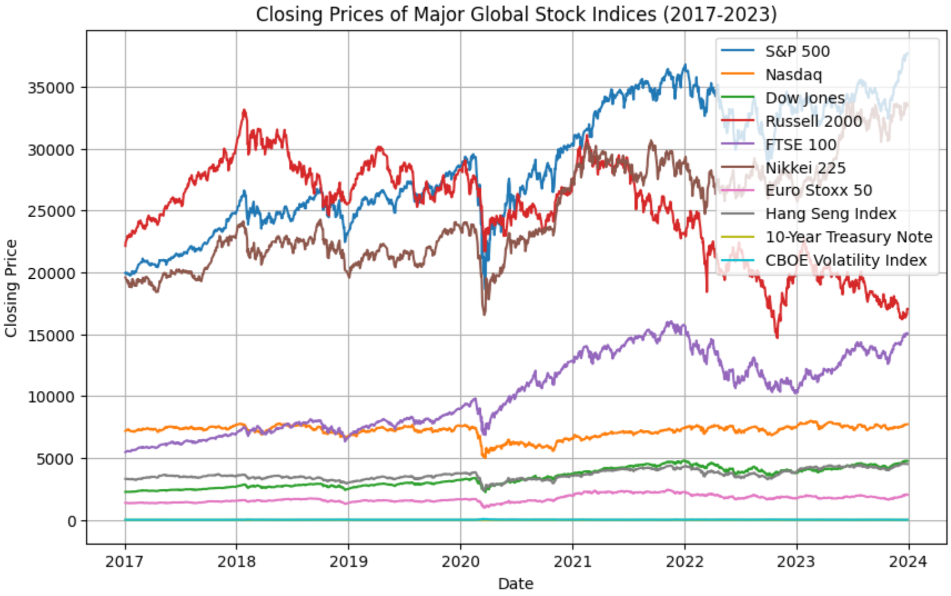
4.1 Data

The first step in applying the TVGL model is the preparation of historical stock index data. In this study, data for major global stock indices was collected using Yahoo Finance, with a focus on three distinct periods: pre-COVID, during COVID, and post-COVID. The selected indices are: S&P 500 (USA), Nasdaq (USA), Dow Jones Industrial Average (USA), Russell 2000 (USA), FTSE 100 (UK), Nikkei 225 (JAPAN), Euro Stoxx 50 (EU), Hang Seng Index (HONG KONG), CBOE Volatility Index (GLOBAL), 10-Year Treasury Note (USA). The data was collected for three time periods:

- **Pre-COVID**: January 1, 2017 to December 31, 2019
- **During COVID**: January 1, 2020 to December 31, 2021

- **Post-COVID**: January 1, 2022 to December 31, 2023

Once the data was collected, it was concatenated into a single dataset, combining the three periods mentioned above. This step ensures that the data is organized in a continuous time series, with no missing values between the different periods.
It is important to note that the data used in this analysis was not asynchronous, meaning that all indices share the same time stamps and there were no discrepancies in the timing of observations across the different periods. This uniformity allows for a more accurate analysis, as the data for all indices is aligned in time.
Finally, the raw closing prices were plotted to provide a visual overview of the data. The following plot illustrates the closing prices of the selected indices from 2017 to 2023:



As shown in the plot, the major stock indices experienced a significant shock during the COVID-19 period. This sharp decline reflects the market's reaction to the economic uncertainty caused by the pandemic.

4.2 Precision Matrices Before, During, and After COVID

In this analysis, we computed the precision matrices for the three periods: before, during, and after the COVID-19 pandemic. These matrices provide insight into the relationships and dependencies between the various stock indices in the dataset. To examine how the market dynamics have evolved, we used the TVGL model with the following settings:

- Lambda and Beta were both set to 10 for testing purposes. These values were chosen as an initial experiment to explore the behavior of the model without extensive tuning. In future iterations, the hyperparameters could be optimized for more precise results.

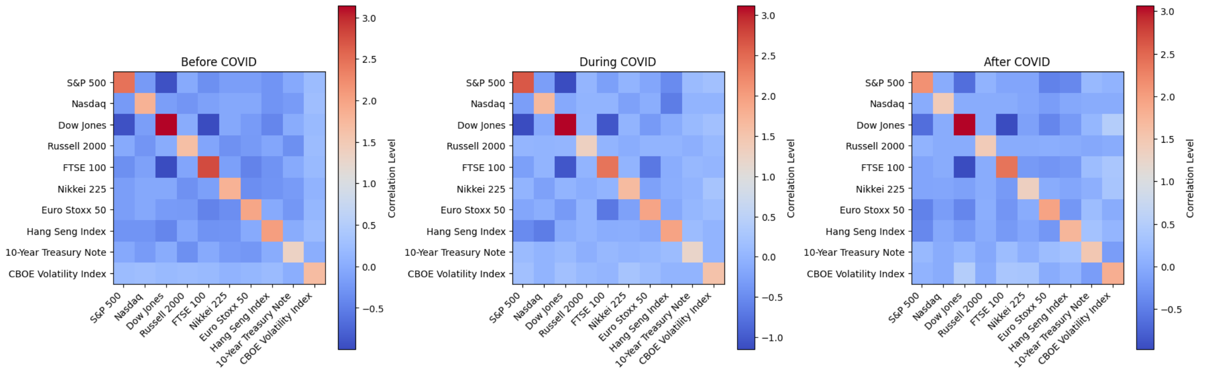
- The L2 penalty was chosen to handle the market volatility during the COVID-19 period. The sharp fluctuations in stock indices made it essential to prevent overfitting, as extreme changes could lead to unstable models. L2 regularization helps by penalizing large coefficients, ensuring the model remains more robust and generalizable, even during the heightened volatility caused by the pandemic. This choice is particularly relevant for studying the period of the COVID-19 pandemic, which introduced unprecedented disruptions in global financial markets. By leveraging the ℓ_2 penalty, we identify significant transitions in the network of stock market indices, shedding light on the systemic impacts of the pandemic on financial structures.

- Slice size was set to 50. This refers to the number of time points considered in each slice of the model. A slice size of 50 was chosen as a balance between computational efficiency and sufficient data representation in each slice.

With these settings, we calculated the precision matrices for each of the three periods. We then used the Frobenius norm to measure the differences between the precision matrices of the different periods. The Frobenius norm provides an overall measure of how different two matrices are, this metric is particularly useful for understanding the magnitude of differences between matrices in an intuitive way. The results are as follows:

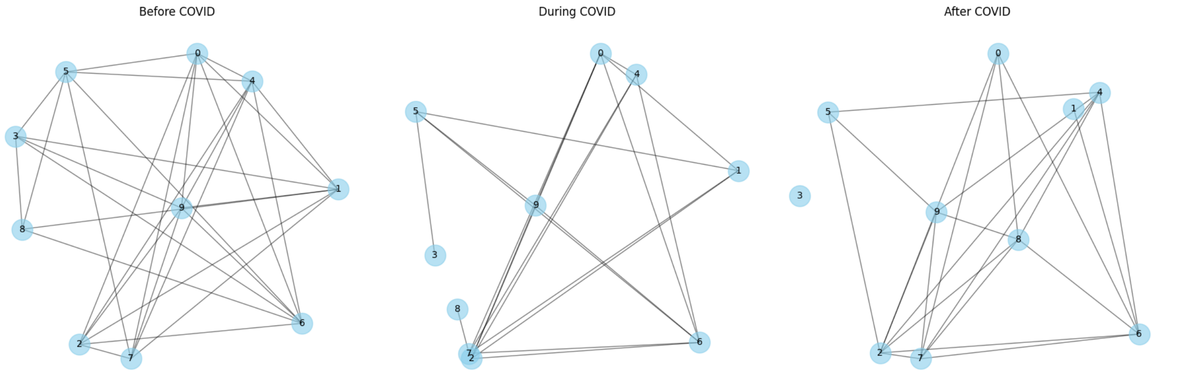
- Diff before and during COVID: **1.6904**
- Diff during and after COVID: **2.0223**

These values highlight the significant structural changes in the market during and after the COVID-19 period. The first difference, between the pre-COVID and during-COVID periods, indicates a major shift in the relationships between the indices, reflecting the shock caused by the pandemic. The second difference, between the during-COVID and post-COVID periods, shows a further adjustment in the market dynamics as the global economy began to recover, albeit with slightly less pronounced changes.



4.3 Graphs Before, During, and After COVID

To visualize how the relationships between the stock indices evolved over time, we plotted the graphs of the precision matrices for each period: before, during, and after COVID-19. These matrices represent the correlation structure between the indices, and their changes over time provide insights into how market relationships were affected by the pandemic.



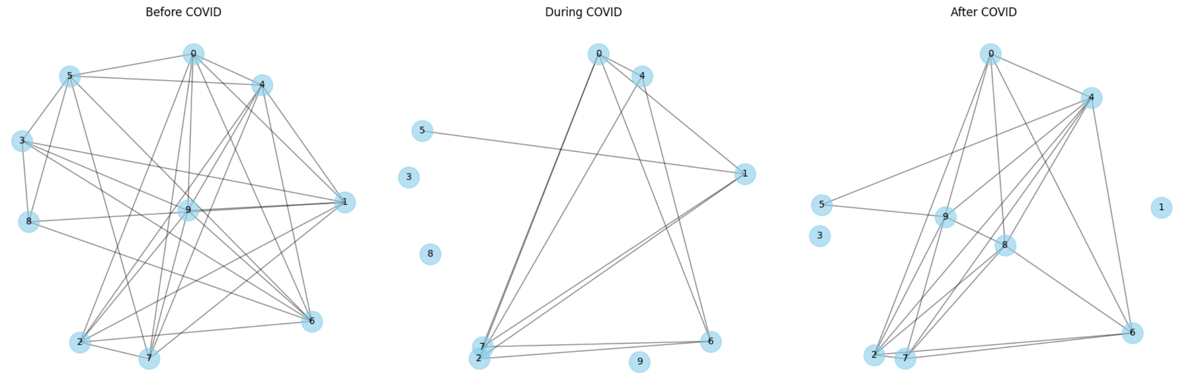
- **Before COVID**: The graph during this period shows the stable correlations among the stock indices, reflecting a relatively consistent and predictable market behavior.
- **During COVID**: The graph for this period reveals a significant shift in the relationships. The pandemic caused market shocks, leading to higher volatility and more erratic behavior across the indices. This is reflected in a less consistent structure in the precision matrix, indicating that the correlations between the indices were less stable during this time.
- **After COVID**: The post-COVID graph shows a return to more stable, yet slightly altered, correlations. Although the market recovered, the relationships between indices did not return to their pre-COVID patterns, possibly due to structural changes in the global economy.

4.4 Bayesian Optimization for Hyperparameter Tuning

In our implementation, we employed **Bayesian Optimization** to fine-tune the hyperparameters β and λ of the Time-Varying Graphical Lasso (TVGL) model. Given the complexity of the model and the computational cost of evaluating its performance for each combination of β and λ, Bayesian Optimization was chosen as an efficient and effective alternative to exhaustive search methods.

Methodology:
We defined an objective function to minimize the Frobenius norm of the difference between the last precision matrix of the model and the average precision matrix across all time slices. This metric ensures that the selected parameters produce consistent precision matrices while respecting the temporal structure of the data.

Results and Relevance:
The optimized parameters improved the model's ability to capture temporal and structural changes in the financial indices across the pre-, during, and post-COVID periods. This approach was particularly critical during the COVID-19 period, which introduced extreme market volatility and abrupt structural shifts. The application of Bayesian Optimization significantly enhanced the performance of our TVGL model by effectively tuning the hyperparameters β and λ. This optimization resulted in a notable reduction in the Frobenius norm, which measures the distance between precision matrices over different time periods.



As shown in the three graphs representing the precision matrices before, during, and after COVID-19, increasing the value of λ has a significant impact on the graph structure. Specifically, a higher λ enforces greater sparsity in the precision matrices, which translates to fewer connections (edges) between nodes. This effect is particularly useful in identifying robust relationships while eliminating weaker or noise-induced connections. Consequently, with higher λ, only the strongest dependencies between indices are preserved, simplifying the graph and making it easier to interpret.

On the other hand, increasing β affects the temporal regularization of the precision matrices. A higher β encourages smoother transitions between consecutive matrices over time, which reduces the impact of sudden changes and ensures that the graph evolves more gradually. This property is particularly relevant when analyzing periods with stable market conditions or when aiming to capture long-term structural dependencies. However, excessively high values of β may mask abrupt changes in the graph, potentially overlooking critical events such as market shocks.

5 Conclusion

In this project, we implemented the Temporal-Variation Graphical Lasso (TVGL) model to analyze the correlations between various global stock indices before, during, and after the COVID-19 pandemic. The model successfully captured the dynamic nature of these correlations over time, providing valuable insights into how the relationships between financial assets evolved during periods of market turbulence.

Advantages of the TVGL Model:
The primary strength of the TVGL model lies in its ability to model dynamic networks, such as the correlations between stock indices, and track their evolution over time. This is particularly useful in the context of financial markets, where the relationships between assets are rarely static. The use of the graphical lasso regularization allows us to identify sparse networks, which simplifies the interpretation of complex financial data and reduces the risk of overfitting.
Moreover, the ability of TVGL to handle temporal data makes it highly suitable for analyzing events like the COVID-19 pandemic, which caused significant shocks to the global economy and affected the correlations between different financial assets. By adjusting the model's parameters, such as λ and β, we were able to fine-tune the results and obtain

a clearer picture of how market relationships changed during and after the pandemic. The application of the L2 penalty also helped ensure that the model remains stable and generalizable, particularly in the context of limited or noisy data. This is important when dealing with financial time series, which can often exhibit high levels of volatility and irregular patterns.

Limitations of the TVGL Model:

Despite its strengths, the TVGL model also has some limitations. One notable challenge is the computational cost associated with fitting the model, especially as the size of the dataset increases. The need to optimize the hyperparameters and run the model for multiple time slices can lead to long processing times, which can be a limitation when working with large datasets or requiring real-time analysis.

Furthermore, the model assumes that the structure of the graph is sparse and that the relationships between the nodes can be captured using a graphical lasso approach. While this assumption is generally valid for many financial networks, it may not always hold true in cases where assets exhibit complex, non-linear relationships or when external factors, such as geopolitical events, significantly impact the market.

Additionally, the TVGL model relies on the availability of clean, high-quality data. Any missing or erroneous data points can impact the accuracy of the estimated correlation matrices and the subsequent analysis. In our case, we addressed this issue by removing any rows with missing values, but this may not always be feasible in practice, particularly with real-time market data.