Master Program in
Data Science Methodology
2023-2024

Bayesian Graphical Modelling with External Network Data

Authors:
Harry Morley, Rafael Gallegos Cortés, Ross Fleming

Directors:
Jack Jewson & Nicolas Bianco

June 9, 2024

Abstract

We propose a Bayesian framework designed to facilitate the incorporation of external network information to the adaptive Bayesian GLASSO model to improve estimates of conditional independence. Using stock market data as a motivating test case for our method, we build a network based on commodity flows between sectors and use this to determine the level of regularisation applied to the elements of our precision matrix. We show through simulated examples that we are able to learn all parameters using Bayesian methods, and on real data, our model worked well when the network effect was significant.

Resumen

Planteamos una metodología completamente Bayesiana al incorporar información de una red externa al modelo adaptativo Bayesiano GLASSO. Usamos datos del mercado de valores como un caso de prueba para nuestro método, construimos una red basada en los flujos de mercancías entre sectores, lo que utilizamos para determinar el nivel de regularización aplicado a los elementos de nuestra matriz de precisión. Se muestra, por medio de ejemplos simulados, que somos capaces de calcular los parámetros utilizando métodos Bayesianos, pero nuestro modelo no alcanza el rendimiento de los métodos de predicción comunes en datos reales, debido a problemas de convergencia de nuestros algoritmos.

Keywords: Graphical Lasso, Bayesian models, Optimal Portfolio.
Palabras clave: Graphical Lasso, Modelos Bayesianos, Portafolio óptimo.

Bayesian Graphical Modelling with External Network Data

Harry Morley, Rafael Gallegos Cortés, Ross Fleming

June 9, 2024

Abstract

We propose a Bayesian framework designed to facilitate the incorporation of external network information to the adaptive Bayesian GLASSO model to improve estimates of conditional independence. Using stock market data as a motivating test case for our method, we build a network based on commodity flows between sectors and use this to determine the level of regularisation applied to the elements of our precision matrix. We show through simulated examples that we are able to learn all parameters using Bayesian methods, and on real data, our model worked well when the network effect was significant.

Keywords: Graphical Lasso, Bayesian models, Optimal Portfolio.

1 Introduction

Estimating the precision matrix is a fundamental problem in high-dimensional multivariate analysis, which is employed in contemporary problems across many disciplines, including in deciphering functional connectivity between brain regions (Wang et al., 2006); in various applications across economics and finance (Fan et al., 2016); and in genomics Kuismin and Sillanpää (2017). Estimation of the precision matrix is particularly valuable in the context of Gaussian graphical models. Consider a d-dimensional random vector $Y = (Y_1, \ldots, Y_d)$ where Y follows a multivariate normal distribution, $\mathcal{N}_d(0, \Sigma)$. The elements of Y correspond to the set of vertices in an undirected graph $G = (V, E)$, and the edge set, E, describes the conditional independence of each element to any other. E is informed by the elements of the precision matrix, $\Theta = \Sigma^{-1}$. If Y_i is conditionally independent of Y_j, the corresponding element, Θ_{ij}, is zero. Therefore, an accurate estimation of the precision matrix is equivalent to performing variable selection (Cox and Wermuth, 2014).

However, when working with high dimensional data, Θ becomes difficult to estimate accurately. In particular, if $d > 2$, a sample estimate of Θ from the data is biased (Stein, 1960; Efron and Morris, 1973), and if $d > N$ where N is the sample size, Σ is singular, and a sample estimate is unobtainable. Consequentially, much of the recent literature in multivariate analysis has explored novel techniques to produce accurate estimates of the

precision matrix. These methods are often reliant the assumption that the precision matrix is truly sparse. If this condition is satisfied, various penalisation methods can be employed to improve estimates. A complete review of many of these methods is available in Heinävaara et al. (2016), but in the interest of brevity we restrict the following discussion to GLASSO-based procedure.

The application of ℓ_1 penalties to graphical models was introduced in Meinshausen and Bühlmann (2006), who showed a sparse graphical model could be estimated by fitting a LASSO to each variable, using the other variables as predictors, and taking the variable Θ_{ij} to be zero only if both the coefficient of variable i's effect on j and the reverse effect was zero. They showed, asymptotically, that this method consistently selects those elements of Θ that are nonzero. Subsequently, several authors demonstrated that this simple algorithm can be viewed as an approximation of the maximisation of the ℓ_1 penalised log-likelihood and introduced their own algorithms to solve it (e.g., Yuan and Lin, 2007).

Friedman et al. (2007) introduced the Graphical LASSO (GLASSO) algorithm. The paper reformulates the estimation as an optimisation problem using an ℓ_1 penalty, given by:

$$\min_{\Theta \succeq 0} \{\text{tr}(S\Theta) - \log\det(\Theta) + \lambda\|\Theta\|_1\} \tag{1}$$

Where $\text{tr}(S\Theta)$ is the trace of the product of the empirical covariance matrix S and Θ, fitting the model to the observed data; $\lambda\|\Theta\|_1$ is the regularization term that encourages sparsity in Θ through the ℓ_1 norm, and λ is a non-negative regularization parameter that controls the level of sparsity, and is determined using some information criterion, like the BIC (Schwarz, 1978). Because the number of non-zero elements in Θ is effectively constrained by the penalty, the computation is tractable even in cases in which the number of covariates is greater than the number of observations of the data.

While this property is desirable, GLASSO is known to be unstable in high dimensions if the dataset contains even a single outlier; Loh and Tan (2018) derived the following error scaling relationship:

$$O\left(\sqrt{\frac{\log p}{N}}\right) + O(\epsilon)$$

where p is the dimension of the data, N the number of samples, and ϵ the error inherent in the model. This demonstrates two properties of the GLASSO's performance: in high dimensional settings, a large sample size greatly improves performance, and any so-called contamination of the data, caused by outliers, has an adverse effect. In this work, we aim to improve the accuracy of the GLASSO estimator using two modifications of the standard theory and then apply it to a prototypical application of the GLASSO.

1. *We use a hierarchical Bayesian framework to facilitate sampling from the posterior distribution*

2. *We incorporate external network data into the model to supplement the information in the core dataset*

Bayesian paradigms have enjoyed significant uptake in statistical modelling problems across many disciplines since the development of robust computational algorithms like Markov

Chain Monte Carlo (MCMC); see van de Schoot et al. (2021) for a review. Its popularity owes in large part to its convenient modelling properties; for instance, one can incorporate a prior into analysis of empirical data, meaning researchers can use previous studies to supplement smaller datasets and still produce robust results. However, our prior is uninformative (see 2.2), only directing the estimation of the precision matrix towards two criteria necessary for subsequent analysis (positive definiteness and sparsity). Nonetheless, the reader should keep in mind that informative priors could be set.

We use a Bayesian framework to capitalise on two other convenient properties. First, by sampling from the posterior, we obtain full distributions over the parameters of the model. This is preferable to the point estimates obtained using information criteria via classical GLASSO because we can infer confidence intervals. Secondly, a Bayesian framework facilitates a hierarchical structure, enabling the incorporation of supplementary data into the model. While traditional GLASSO algorithms can produce estimates of the precision matrix even when $N > p$, the accuracy in their estimation is inherently limited by the poor data-to-parameter ratio. To improve accuracy, we introduce an informative $p \times p$ network which, in tandem with the data itself, is used to estimate the parameters of the generative distribution of the precision matrix. The incorporation of external network data was described in Jewson et al. (2022), in which the researchers attempted to model stock returns using the company returns time series, but included in the network data on the publicly announced political and economic risks of each. As in their paper, our model allows for multiple networks to be added. They used the EBIC to fit the network coefficients in the model, and reported some numerical instability in their point estimates. We fit the parameters using MCMC, which outputs a full posterior distribution. Any instability in point estimates should be evident in the shape of this posterior.

As a practical case study, we try to produce an optimal portfolio using our model's estimation of the covariance matrix with an external network, computed with economic sector information from the Bureau of Labor Statistics. We build this network using sector commodity flow matrices. This should help us to estimate the extent of the connection between stocks. Once we have our precision matrix, we use optimal portfolio theory to produce stock portfolios. Devised by Markowitz (1952), this is a technique used to yield the maximum returns attainable from a set of stocks given a risk tolerance (or vice versa). The theory argues that, if both the correlation between a set of stocks and their returns are known, a fund can be distributed amongst the stocks to create a portfolio with a fixed level of risk. Given a set of p stocks, let r_i denote the rate of return of stock i. Then we may define a random vector, Y where:

$$Y = \begin{pmatrix} r_1 \\ r_2 \\ \vdots \\ r_p \end{pmatrix} \tag{2}$$

Set $\mu_i = E(r_i)$, $m = (\mu_1, \mu_2, \ldots, \mu_p)^T$, and $\text{cov}(Y) = \Sigma$. If $w = (w_1, w_2, \ldots, w_p)^T$ is a set of weights associated with a portfolio, then the rate of return of this portfolio is given by $r = \sum_{i=1}^{p} r_i w_i$, and r is also a random variable with mean $m^T w$ and variance $w^T \Sigma w$. We can define the *mean variance efficient portfolio* by:

$$x_{EV} = \frac{1}{1^T \Sigma^{-1} \mu} \Sigma^{-1} \mu, \tag{3}$$

where 1 is a vector of all ones. We can also define the *minimum variance portfolio* by

$$x_{MV}(\Sigma) = \frac{1}{1^T \Sigma^{-1} 1} \Sigma^{-1} 1. \tag{4}$$

These derivations have this closed form if we assume short selling stocks is permitted. In both, Σ is the inverse precision matrix Θ estimated by our model.

It is worth mentioning that in the absence of a reliable estimate of the precision matrix, optimal portfolio theory performs poorly, sometimes worse than a simple mean-weighted portfolio Michaud (1989). Last, there may be small inaccuracies in estimate of the covariance between two stocks that may not be statistically significant, but have a deep effect on the resulting portfolio, which is exacerbated as the number of stocks is large. Best and Grauer (1991) found that increasing the number of stocks from 10 to 100 caused the sensitivity of optimal portfolio weights to perturbations to be on the order of 4000 times greater. Additionally, Kan and Zhou (2007) showed that when the relation between the number of stocks and observations is not small, estimation errors in forecast returns and covariances produce large interaction effects, making the optimal weights unstable and unreliable. When computing precision matrices in practise, is common to encounter scenarios where the number of variables p exceeds the number of observations N. In such instances, the precision matrix cannot be directly computed due to the imbalance between p and N.

2 Model

2.1 Bayesian Adaptive GLASSO With External Network Data

Our full model, the adaptive Bayesian GLASSO with external network data functionality, is given below:

$$y_i \sim \mathcal{N}_p\left(0, \Theta^{-1}\right), \ i = 1, \ldots, N \tag{5}$$

$$\theta_{jj} \sim \text{Exp}\left(\frac{\lambda_0}{2}\right), \ j = 1, \ldots, p \tag{6}$$

$$\theta_{jk} \sim \text{DExp}\left(1/\lambda_{jk}\right) \tag{7}$$

$$\lambda_{jk} = \exp\left(\beta_0 + \sum_{q=1}^{Q} \beta_q A_{jk}^{(q)}\right), \ k < j \tag{8}$$

$$\beta_0 \sim \mathcal{N}\left(\mu_{\beta_0}, \sigma_{\beta_0}^2\right) \tag{9}$$

$$\beta_q \sim \mathcal{N}\left(0, \sigma_{\beta_q}^2\right), \ q = 1, \ldots, Q \tag{10}$$

Here y_i are the observations, assumed to be normally distributed. θ_{jj} are the diagonal elements of the precision matrix, drawn from an exponential distribution with λ_0 as the rate parameter. The θ_{jk}'s are the off-diagonal elements of the precision matrix, drawn from a double exponential distribution with rate parameter $1/\lambda_{jk}$. In the context of the double exponential distribution, a larger rate parameter corresponds to a lower rate of decay in the distribution's tails (less values centred around 0). In our context, this means that higher λ_{jk} values correspond to greater shrinkage (more values of Θ being set to zero). The parameters λ_{jk} are determined by taking the exponential of a linear combination of variables β_q and the external network data, $A^{(q)}$, where $A^{(q)}_{jk}$ denotes a single entry of the network matrix $A^{(q)}$.

Our model incorporates sparsity in the precision matrix by assigning a double exponential prior to the off-diagonal elements. This encourages many of these elements to be zero, effectively determining which variables (nodes in the graph) are conditionally independent of one another. In a linear regression setting, LASSO induces sparsity in the coefficient estimates by using an ℓ_1 penalty, and this is equivalent to assuming a double exponential prior on the coefficients. Tibshirani (1996) shows that LASSO estimates are equivalent to the mode of the posterior distribution if the regression coefficients are assumed to have independent and identically distributed double exponential priors.

There are a few important differences between our model and the GLASSO model from Wang (2012). Our model more closely follows the Bayesian adaptive GLASSO detailed in the same paper. This method places different parameters λ_{jk} on different off-diagonal elements θ_{jk}. In our framework, the amount of shrinkage depends on external network data.

The standard Bayesian GLASSO has only one λ, which is the same in the priors for θ_{jj} and θ_{jk}. Our modified version uses unique λ_{jk}'s, capable of shrinking each element of Θ individually. The problem with having only one λ value is that while we obtain small θ_{jk} for large values of λ, as desired, we also obtain small values for θ_{jj}. Having only one λ also introduces scale invariance problems. A more complete discussion into scale invariance is provided in Carter et al. (2023). Our model improves standard GLASSO by considering different penalisation on entries of Θ, and now large λ_{jk} values will induce shrinkage on the λ_{jk}'s.

The fundamental difference between our model and the Bayesian adaptive graphical LASSO is in the method by which we estimate the shrinkage parameters. Our methodology facilitates the inclusion of external network data to improve our estimate of Θ. In our model, λ_{jk} is determined by taking the exponential of a linear combination of learnt parameters β_q with external networks $A^{(q)}$. If a network $A^{(q)}$ is informative, it will have a negative β_q value associated with it. So, if there is a large value $A^{(q)}_{jk}$ in an informative network this, λ_{jk} is small, decreasing the penalization for the value θ_{jk}. If j and k are linked in an informative network, less shrinkage is applied to θ_{jk}.

Informative networks should be similar to the precision matrix in their sparsity and their structure. High values in one should correspond to high values on the other. These networks can be binary, or weighted.

Importantly, each network $A^{(q)}$ has a learned β_q associated with it, indicating how useful this network is in the prediction of Θ. In practise, this means that multiple networks can be introduced to the model, and only informative networks will affect the shrinkage.

Determining the β's using Bayesian methods provides full posterior distributions of the

parameters, not just point estimates. This gives a complete picture of the uncertainty and variability of the parameters, allowing for probabilistic interpretations. It also allows us to assign informative priors, if we have prior beliefs about where β_q may lie in the state space.

2.2 Priors

We set the priors in 5 to 10 based on two major assumptions:

1. Sparsity: we anticipate that the estimate of Θ will be sparse. Therefore, we want a prior that assigns a high probability density to sparse configurations of Θ

2. Positive definiteness: we aim for a Θ drawn i.i.d. from the GLASSO prior to be positive definite with high probability

We set the initial θ_{jj} on the assumption that the data will be standardised. If this is the case, then θ_{jj} will necessarily be greater than 1 for all j. We set $\lambda_0 = 0.5$ accordingly, to give prior probability of 0.75 that $\theta_{jj} > 1$. This reinforces the expectation of relatively large values for the diagonal elements of the precision matrix, which corresponds to strong conditional dependence after standardization.

To initialise θ_{jk}, we perform prior elicitation on β_0 and β_q. To simplify the prior elicitation, we make two assumptions: that the prior mean for β_q is 0, and that the prior variance is the same for all $\beta_0, \beta_1, ...\beta_Q$. We set the mean of β_q to zero to express no prior beliefs about whether the network helps positively or negatively. This simplifies prior elicitation, and allows untested networks to be added to the model.

To ensure sparsity in Θ, we define a sparsity measure. We suppose the average node is connected to r other nodes. For our data we take $r = 5$. This gives us $5 \times p$ edges in the graph and a sparsity of $\frac{5p}{p(p-1)} = 5/(p-1)$. We then set θ_{jk} to zero if its partial correlation ρ_{jk} is such that $|\rho_{jk}| < 0.01$. This sets a very stringent criterion for defining significant relationships, thereby promoting a sparse structure by considering only strong and significant connections as non-zero. We use this to set the parameters μ_{β_0} and $\sigma^2_{\beta_0}$. To do this we propose many different values of μ_{β_0} and $\sigma^2_{\beta_0}$, and after computing sparsity and positive definiteness for all combinations of values, we identify the combination that best matches the desired sparsity and probability of positive definiteness. For our data, we choose $\mu_{\beta_0} = 9.75$ and $\sigma_{\beta_0} = 4.75$.

2.3 Implementation

The practical implementation of our model is programmed in R (R Core Team, 2024). We first use a block Gibbs sampler to estimate the covariance matrix. We then extend this algorithm to determine the β_q's and λ_{jk}'s. To do this, we use the Metropolis-Hastings algorithm. For the block Gibbs sampler, we closely follow Wang (2012). We outline the basic approach here, but for more detail see his paper. To estimate Θ, we formulate it as:

$$p(\Theta|\lambda_{jk,j<k}) = C^{-1}_{\lambda_{jk,j<k}} \left\{ \prod_{j<k} \text{DExp}(\theta_{jk}|\lambda_{jk}) \right\} \left\{ \prod_{i=1}^{p} \text{Exp}\left(\theta_{jj}|\frac{\lambda_0}{2}\right) \right\} \mathbf{1}_{\Theta \in M^+}, \tag{11}$$

where C is a normalising constant independent of λ_{jk}. We need to represent 11 as a hierarchical model to enable us to use block Gibbs sampling. To do this, we use a useful

property of the double exponential distribution by modeling it as a scale mixture of normal distributions (West, 1987).

$$p(\theta \mid \tau, \lambda_{jk}) = C_\tau^{-1} \left\{ \prod_{i<j} \left(\frac{1}{\sqrt{2\pi\tau_{jk}}} \exp\left(-\frac{\theta_{jk}^2}{2\tau_{jk}} \right) \right) \prod_{i=1}^{p} \left(\frac{\lambda_0}{2} \exp\left(-\frac{\lambda_0 \theta_{jj}^2}{2} \right) \right) \right\} \Theta_{M+} \tag{12}$$

$$\tau_{jk} \sim C_\tau \prod_{i<j} \left(\frac{\lambda_{jk}^2}{2} \exp\left(-\frac{\lambda_{jk}^2 \tau_{jk}}{2} \right) \right) \tag{13}$$

Here $\tau_{jk,j<k}$ are latent scale parameters. Equations 12 and 13 together form a hierarchical model that provides a more tractable approach to sampling from the distribution specified in Equation 11. This is easier to sample from, because the distributions involved are conjugate as seen in the Gaussian and exponential cases in Equations 12 and 13, respectively, compared to directly sampling from a double exponential distribution as in Equation 11.

We iteratively sample Θ one row at a time whilst ensuring the matrix remains positive definite. To update τ_{jk}, we use the fact that the parameters $1/\tau_{jk}$ have a inverse Gaussian conditional posterior distribution, $\mathrm{IG}(\mu', \lambda')$ with

$$\mu'_{jk} = \sqrt{\lambda_{jk}^2 / \theta_{jk}^2}, \qquad \lambda'_{jk} = \lambda_{jk}^2 \tag{14}$$

Finally, we get a full prediction for Θ. We outline the full procedure in algorithm 1.

2.4 Metropolis-Hastings Sampling

To obtain the regularization parameters λ_{jk}, we need to estimate the values of β_q. We use the Metropolis-Hastings algorithm, a Markov Chain Monte Carlo method used for sampling from probability distributions when direct sampling is difficult. Briefly, the Metropolis-Hastings algorithm extends the Metropolis algorithm, which itself is an adaptation of a random walk combined with an acceptance/rejection criterion. This modification enables convergence to the desired target distribution, as detailed in Gelman et al. (2013). We cannot use Gibbs sampling because there is no closed form for β_q. We start by assigning the prior on β_q:

$$\beta_q \sim \mathcal{N}\left(\mu_{\beta_q}, \sigma_{\beta_q}^2 \right) \tag{15}$$

where $\mu_{\beta_0}, \sigma_{\beta_0}^2$ are fixed.
Following standard Metropolis-Hastings procedure, we then propose a new value for β,

$$\beta_q^* \sim \mathcal{N}\left(\beta_q^{old}, \epsilon_q \right) \tag{16}$$

Then we calculate the acceptance probability,

$$\alpha = min \left\{ 1, \frac{\prod_{i=k} p\left(\theta_{ik} | \beta_q^*\right) p\left(\beta_q^*\right)}{\prod_{i=k} p\left(\theta_{ik} | \beta_q^{old}\right) p\left(\beta_q^{old}\right)} \right\} \tag{17}$$

Here $p\left(\theta_{ik} | \beta_q\right)$ is a double exponential, and $p\left(\beta_q\right)$ is the β_q prior.

Next, we draw a random number from a uniform distribution, $U \sim \mathcal{U}(0,1)$. If $U < \alpha$ we accept β_q^* as the new β_q^{old}.

Repeat this process for all $\beta_{q,q=0...Q}$, where Q is the number of external networks. The λ_{jk} values can now be determined through equation 8. This algorithm is run for M draws from the posterior. We then discard the first half of the iterations (burn-in), and average Θ over the remaining iterations, to obtain our final precision matrix.

In the Metropolis-Hastings algorithm, the acceptance rate should be between 20% and 50%, for all β_q's (Gelman et al., 2013). To achieve this, the ϵ values in equation 16 need to be tuned. We found the optimal epsilons via trial and error.

Algorithm 1: Adaptive Bayesian GLASSO with External Network Data (Metropolis-Hastings)

Initialize: Initialize $\theta_{jk}^{(0)}$, $\theta jj^{(0)}$, $\lambda_0^{(0)}$, $\lambda_{jk}^{(0)}$, $\tau_{jk}^{(0)}$, $\beta_0^{(0)}$, $\beta_1^{(0)}$, ..., $\beta_q^{(0)}$. Fix ϵ_0, ϵ_1, ..., ϵ_q.
for *h=1...M* **do**
  Set $M = h + 1$
  for *j=1...p* **do**
    Partition Θ as in A.3
    Sample $\upsilon \sim \text{Ga}(N/2 + 1, (s_{22} + \lambda_0/2)$ and $\phi \sim \mathcal{N}(-C_{s_{21}}, C)$,
     $C = \left\{(s_{22} + \lambda)\Theta_{11}^{-1} + diag(\tau_{12}^{-1})\right\}^{-1}$
    Update $\theta_{21} = \phi$, $\theta_{12} = \phi'$, $\theta_{22} = \upsilon + \phi'\Theta_{11}^{-1}\phi$
    for $j < k$ **do**
      Sample $\mu_{jk} \sim \text{IG}(\mu', \lambda')$ and update $\tau_{jk} = 1/\mu_{jk}$
    end
    for *Q=1...q* **do**
      Propose new $\beta_q^* \sim \mathcal{N}(\beta_{old}, \epsilon)$
      Compute acceptance probability α (see 17)
      if $\mathcal{U}_{[0,1]} < \alpha$ **then**
        Set $\beta^{(h+1)} = \beta^*$ **else**
          Set $\beta^{(h+1)} = \beta^{(h)}$.
        end
      end
    end
  end
end
return Θ

3 Network

Our network aims to capture information about the interaction between firms in their production processes. The edges of the Network represent the intermediate use of the inputs in one firm to produce some commodity.

The network was constructed using data on inter-industry relationships (Input-Output

Matrix) provided by the Bureau of Labor Statistics (BLS), Office of Occupational Statistics and Employment Projections (U.S. Bureau of Labor Statistics, 2024).

Input-output data illustrate the commodity flows from production, through their intermediate use by industries, to their final acquisition by consumers. Annually, the BLS develops these insights into a series of matrices or tables, utilizing input-output data originally compiled by the Bureau of Economic Analysis. These data adhere to the North American Industrial Classification System (NAICS). NAICS is a classification standard utilized by U.S. statistical agencies to organize business establishments, based on a supply-based or production-oriented framework which groups establishments into industries by similarity in production processes.

We focus on the "USE" Matrix, which encompasses intermediate inter-industry inputs, including intermediate inter-industry sales and a column with final demand for the specified year. Each column corresponds to total industry output, while each row equates to total commodity output.

For our initial analysis, we used the "USE" matrix to model interactions within the Network. We presumed a dynamic network structure and used the corresponding information to the stocks in the dataset for each year. The structure of the USE input-output matrix, categorized by NAICS codes, is presented in Table 1.

Table 1: USE Input-Output Matrix

	Industry 111	Industry 112	$\cdots$	Industry 8139	$\cdots$	Value Added	Total Commodity Output
Commodity 111	$m_{1,1}$	$m_{1,2}$	$\ldots$	$m_{1,177}$	$\ldots$	$m_{1,193}$	c_1
Commodity 112	$m_{2,1}$	$m_{2,2}$	$\ldots$	$m_{2,177}$	$\ldots$	$m_{2,193}$	c_2
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
Commodity 8139	$m_{177,1}$	$m_{177,2}$	$\ldots$	$m_{177,177}$	$\ldots$	$m_{177,193}$	c_{177}
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
Value Added	$m_{193,1}$	$m_{193,2}$	$\ldots$	$m_{193,177}$	$\ldots$	$m_{193,193}$	c_{193}
Total Industry Output	i_1	i_2	$\ldots$	i_{177}	$\ldots$	i_{193}	t

The element m_{ij} indicates the production of commodity j by industry i (e.g., $m_{111,112}$ is the production of the commodities that belong to the 112 sector, by industries classified in the 111 sector), and t_i denotes the total output of each industry.

It is important to note the discrepancy between industry codes used in the stock market (CIK) and NAICS codes. To bridge this gap, we obtain the corresponding three-digit NAICS code for each stock market entity from the webpage of Zoominfo (Zoominfo, 2024).

To construct the adjacency matrix for the network, we assign the percentage for the commodity sector that belongs to stock indicated by the row in the correspond industry, that is, the value $e_{i,j} \in E$ from the stocks $i \in V$ to stock j is $\frac{m_{i,j}}{c_i - m_{i,j}}$. These elements

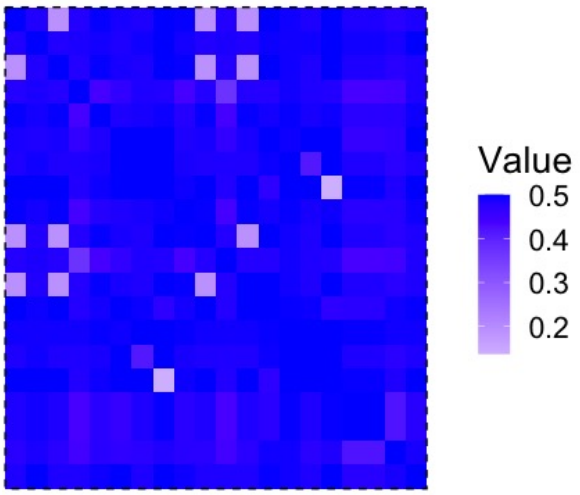


Figure 1: Transformed network values for 2021, $p = 20$ stocks

Table 2: Graph Edge weights

Statistic	2013	2015	2022	Avg. (2013-2022)
Average weights	1.99	2.12	2.08	2.03
Max weights	45.15	40.34	41.29	43.35
Standard deviation	4.96	5.22	5.09	4.98
25%	0.10	0.09	0.09	0.10
50%	0.49	0.40	0.33	0.40
75%	1.29	1.35	1.27	1.37

represent the percentage that the intermediate use of one industry contributes to the total production of another industry.

Subsequently, to ensure symmetry in the matrix and to reflect accurately the interdependencies between industries in both directions, we calculate the average between the element and its corresponding transpose for each pair i, j. This approach ensures that the undirected graph, G, equitably represents the bidirectional interdependency relationships between industries in commodity production.

We do this for every year, from 2008-2022. The resulting networks are fully connected networks with an average edge value of 2.05% and a standard deviation of 4.99%. The maximum weight is 41.86%, and 50% of the weights are lower than 0.40%, while 75% are lower than 1.34%. Table 2 displays some statistics for several years, and the weight value distribution. Figure 2 shows the network for 2022, its edge weights, and clusters. Figure 1 shows the change in the network from 2021 to 2022 for a subsample of 20 stocks. This plot demonstrates some small year-on-year changes in the network, although its structure is generally constant.

To verify whether we expect the network to be informative, we plot the lower triangular values of our network against their corresponding elements in a precision matrix obtained through GLASSO for one year. To achieve the shape evident in the graph, we use the transformation $|0.5 - A|$. This makes the relationship clearer. We also use the transformed

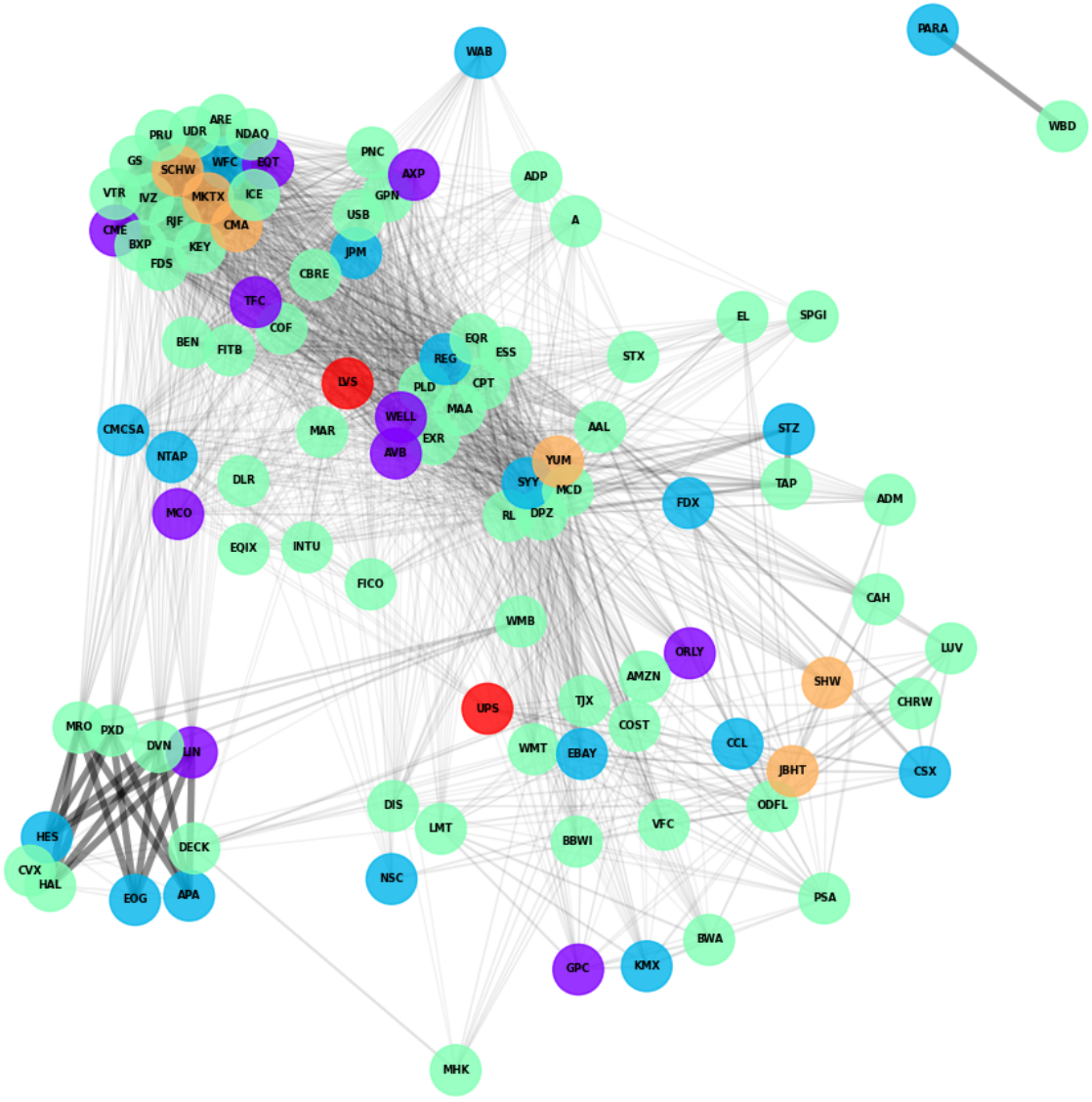


Figure 2: Graph Assets Network 2022. The colors represent the communities (clusters) detected by the Louvain algorithm, indicating groups of nodes that are densely connected to each other. The edges shown have weights greater than 1%, with higher values represented by greater width and darker color.

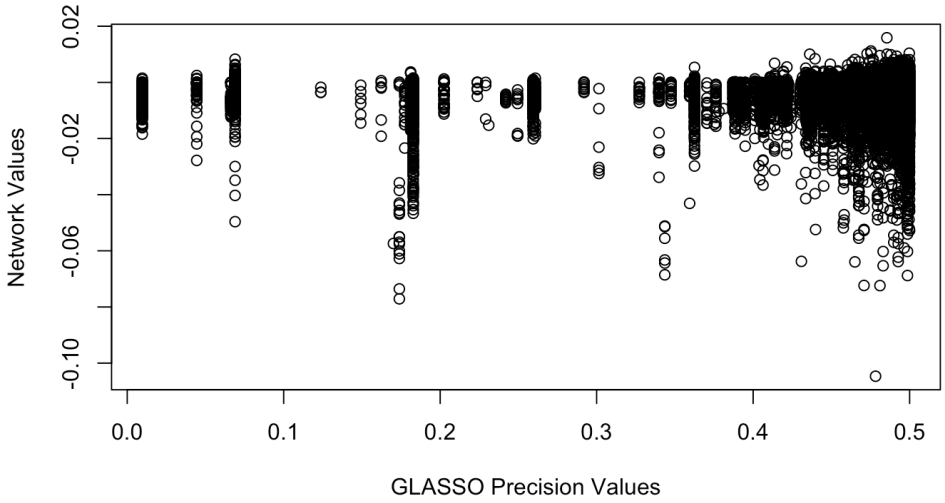


Figure 3: Network vs GLASSO Precision Matrix Estimate

network in our model. Here we estimate the precision matrix using the GLASSO package (Friedman et al., 2019), which is similar in structure to the one obtained using our model. There is is a correlation between the precision matrix and the network evident in figure 3.

4 Empirical analysis

4.1 Data

To test our model, we use the daily log returns of the closing stock price adjusted for stock splits from 2008 to 2023, from the S&P 500 index. That is on average $N = 252$ observations per year to train and to forecast.

4.2 Simulation Study

To verify our method and implementation, we ran a simulation study, using a simulated covariance matrix and simulated networks for $p = 20$ randomly selected stocks. To synthesis our data matrix S, we generate normally distributed data for our p stocks, with mean 0 and variance equal to the covariance matrix for our 20 stocks. We then set all values < 0.1 to 0. For our networks, we construct two sample matrices: $A^{(1)}$; a very informative network, where a 1 is given to an element A_{jk} if j and k are strongly correlated (have a covariance ≥ 0.2), and a 0 otherwise. We can see in figure 4 that the network is informative, as expected. Our second network $A^{(2)}$ is simply a randomised network, where 20% of off-diagonal entries have been set to 1 randomly, and the rest are 0. With these matrices, we expect to see a negative β_1 value, and a β_2 value of 0.

We run our algorithm with this simulated data. We run $20,000$ iterations, and use a burn-in of $10,000$. We get the diagnostics plots in figure 5.

In the first column we can see the chain plots. From these plots, we can see that the β chains appear to converge well. The density plots are smooth and narrow, again showing

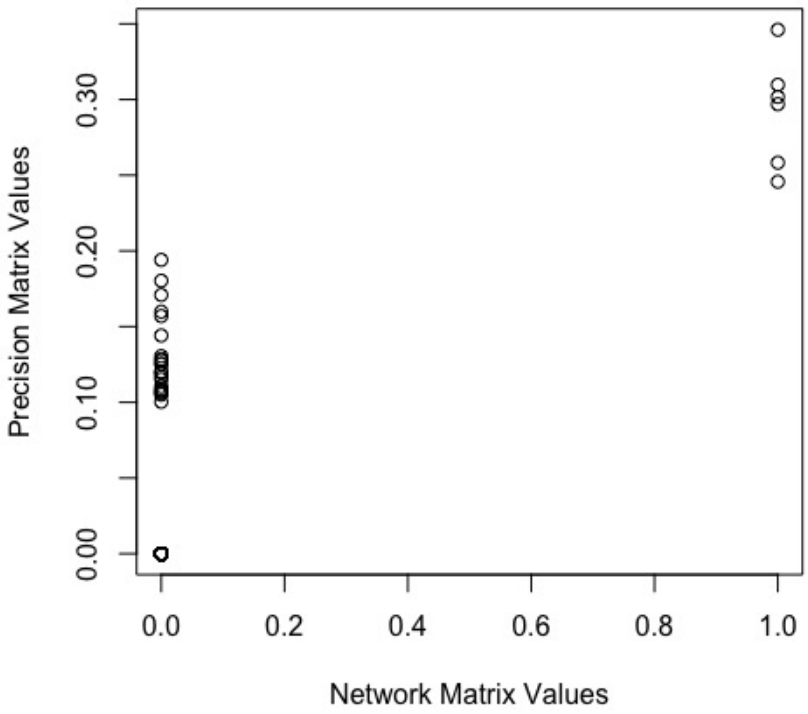
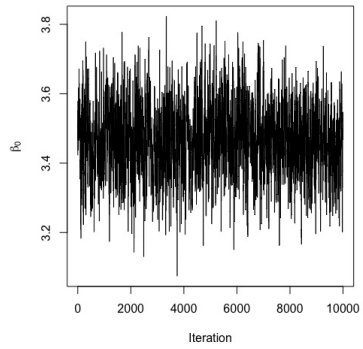
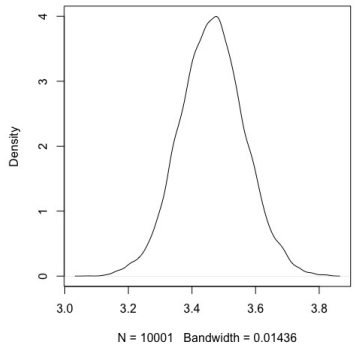


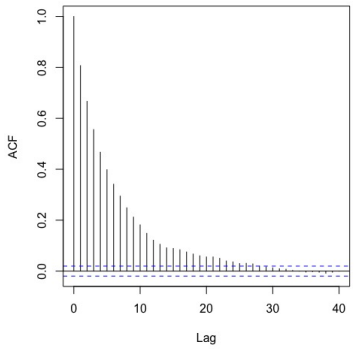
Figure 4: $A^{(1)}$ informative network vs GLASSO estimates



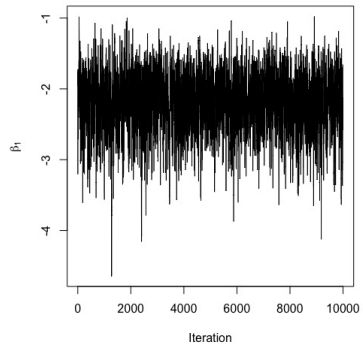
(a) β_0 Chain



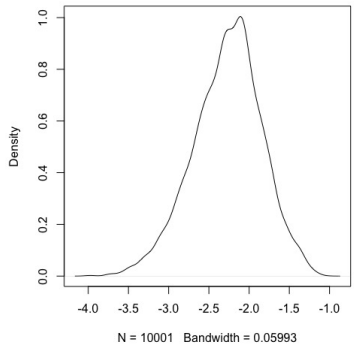
(b) β_0 Density



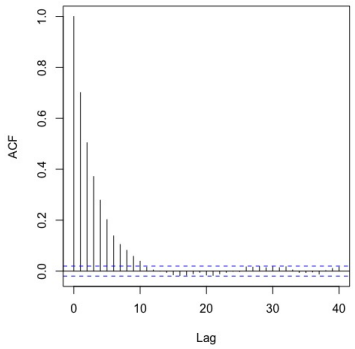
(c) β_0 ACF



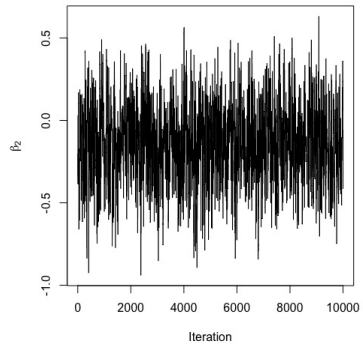
(d) β_1 Chain



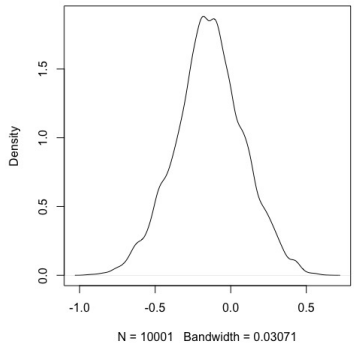
(e) β_1 Density



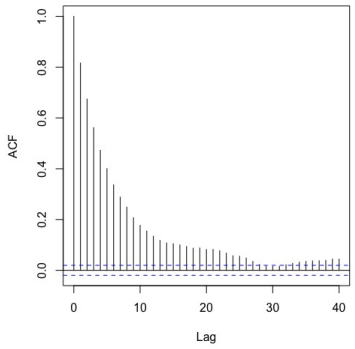
(f) β_1 ACF



(g) β_2 Chain

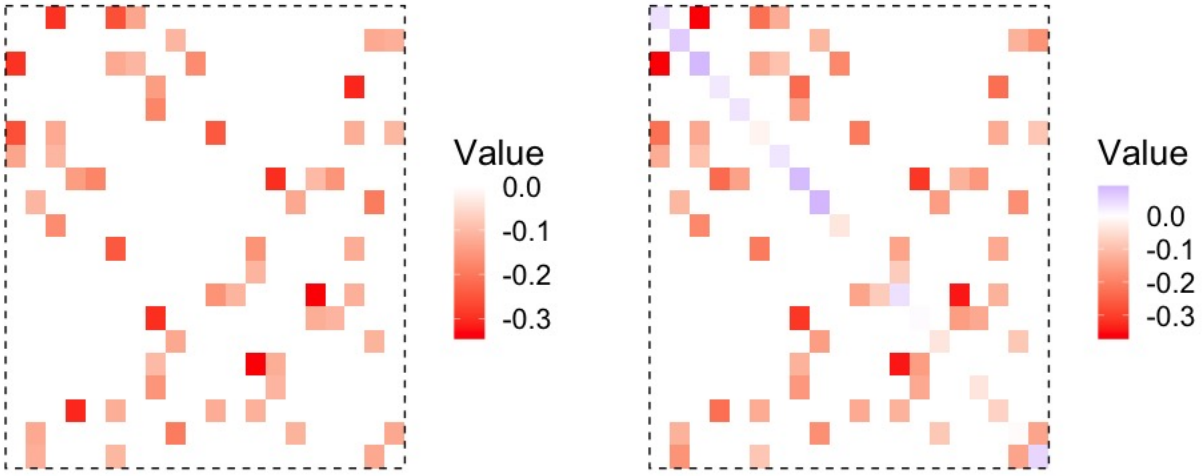


(h) β_2 Density



(i) β_2 ACF

Figure 5: Diagnostic plots for β in Simulation study, $(20,000$ iterations with a burn in of $10,000)$



(a) True Θ (b) Θ obtained using both networks

Figure 6: Θ Heatmaps From the Simulation Study

effective convergence. The ACF plots show the autocorrelation function for β_q, used to analyze the correlation between the values of β_q at different time lags. The exponentially decreasing bars indicate that the correlation diminishes as the lag increases. All together, these plots show that are our algorithm works well for this controlled test case.

Averaging over the 10,000 iterations, we the following values for β: $\beta_0 = 3.4$, $\beta_1 = -2.1$ and $\beta_2 = -0.09$. β_1 was negative, correctly indicating that the network $A^{(1)}$ was informative and correlated with the precision matrix. We also see that β_2 is close to zero ,indicating $A^{(2)}$ is uninformative. It isn't exactly 0 due the random nature of the matrix meaning that some values in $A^{(2)}$ will be informative.

Another important check for proof of concept is that the Bayesian network GLASSO actually alters the precision matrix. Figure 6a represents the precision matrix obtained by inverting the variance-covariance matrix of the selected stock data. It is then standardized so that its diagonal elements are all 1. Finally, small elements (absolute value less than 0.1) in this standardized precision matrix are set to zero to sparsity the matrix. Figure 6b shows the precision matrix estimate obtained through using our algorithm, with the informative network $A^{(1)}$. We can see there are significant differences between (a) and (b), as we would expect.

This simulation study provides a proof of concept for our model. We have shown that, if the network is, indeed, informative, we should expect the model to use the external information to update values of the precision matrix. As such, we move to an analysis of a real-world problem in which our model is applicable.

4.3 Stock Market Data

To test the effectiveness of our model in practical applications, we used stock market data to estimate optimal portfolios. We trained the model using one year's worth of data at a time. To do this, we chose $p = 100$ stocks at random from the S&P dataset of 500 stocks, and for

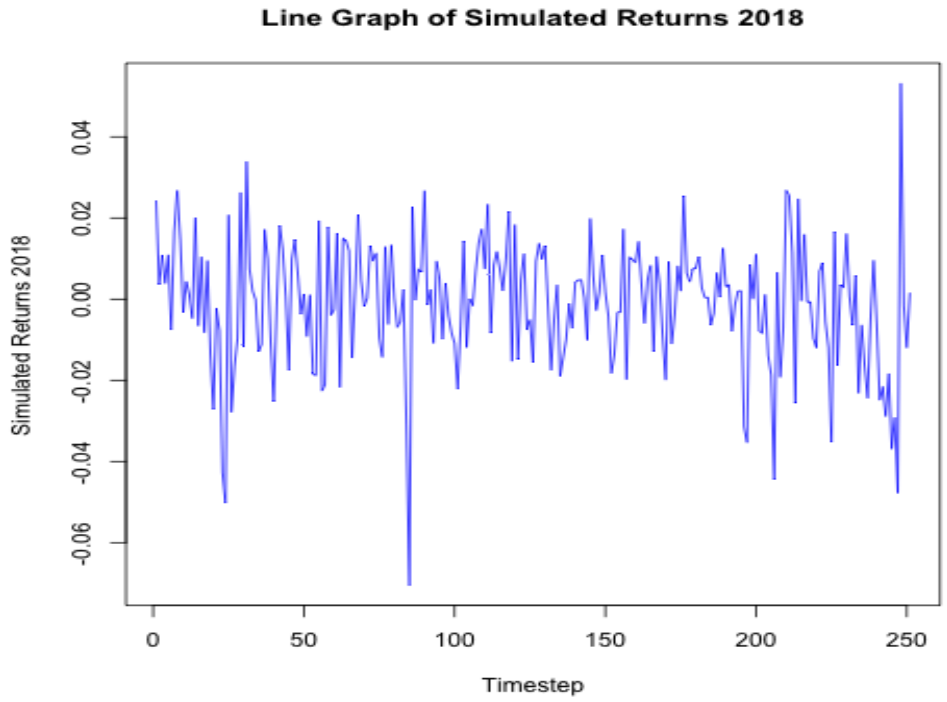


Figure 7: Simulated daily percentage returns for 2018 using calculated portfolio

each year used our economic commodity flow networks A^{year} as external network data. After initialising the model with the priors specified in 2.2, S and A^{year}, we performed $M = 20,000$ iterations (with a burn-in of 10000) for each test year. We then used equation 3 to calculate the weights for the optimal portfolio, and simulated our portfolio on the following years data (see figure 7 for an example of simulated returns over one year). We iterated this procedure for 10 years, from 2013-2022, to analyse how our model performed over a longer time period.

In order to quantify our model's performance, we compare it with two benchmarks:

1. An equally weighted portfolio - assigning uniform weights of $1/p$, where $p = 100$, to every stock.

2. Our model with no external network data - allows us to investigate the effect A^{year} had on our model

Despite its simplicity, the equally weighted portfolio is a reputable benchmark that reflects the state of the market as a whole. It is well diversified, meaning it should exhibit far less volatility than a single stock, and its members constitute some of the most profitable companies in the world, meaning its returns are guaranteed to align with the state of the market in a given year. The no network model provides a relevant baseline for our procedure; in effect, it performs a GLASSO procedure. Any returns greater than this benchmark are attributable directly to the network data.

Overall, our model performed poorly slightly worse than the benchmarks over the full period (figure 10), but we attribute this result largely to the failure of the Markov chains to converge. Figures 12 display sample plots for a several years, and it is evident there

is no convergence after 20,000 iterations. Classical diagnostics confirmed that the Markov chain did not behave as we would have liked; the effective sample size was under ten and autocorrelation high over a lag of fifty (>0.85) for most years.

Nonetheless, the model did perform well in some years (see Figure 8), and in years in which it did not match up to the benchmarks (Figure 9), its performance was not substantially worse. To investigate whether the years in which the model performed better than the benchmark were due to the information provided by the network, we examined the relationship between β_1 and and the performance of the model relative to the no-network benchmark in figure 11. In this figure, we plotted the average β_1 value from the Markov chain after the burn-in against the difference between the cumulative portfolio performances of our model and the no-network model at the end of each year. While the sample is not large enough to draw too strong a conclusion about their relationship, there does seem to be a downwards trend. As a reminder, a lower value for β_1 implies the network is informative. This plot provides tentative evidence that, despite the poor convergence of the beta chains, the network may have had a tangible, positive effect on the portfolio's performance.

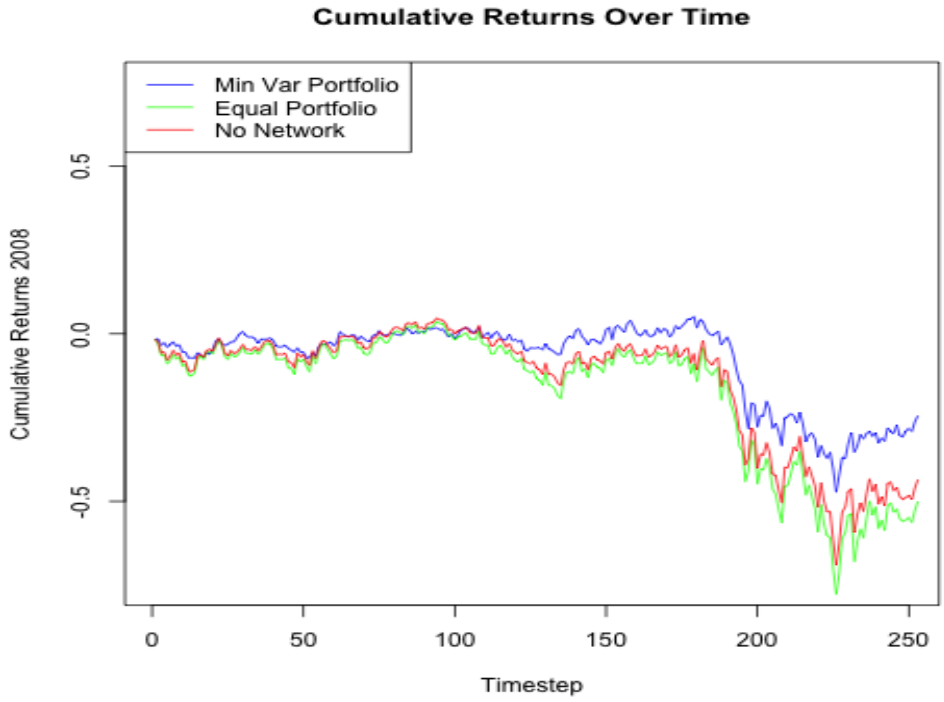


Figure 8: Cumulative Returns in 2008

Given the aforementioned convergence issues, it would be unwise to discount the effectiveness of our external network. While the chains did not converge to a narrow range, the β_1 distribution means were often negative and the entire distribution non-zero, indicating that the network is informative. We believe this is the case (figure 3), and so a more effective sampling scheme likely would have produced better results.

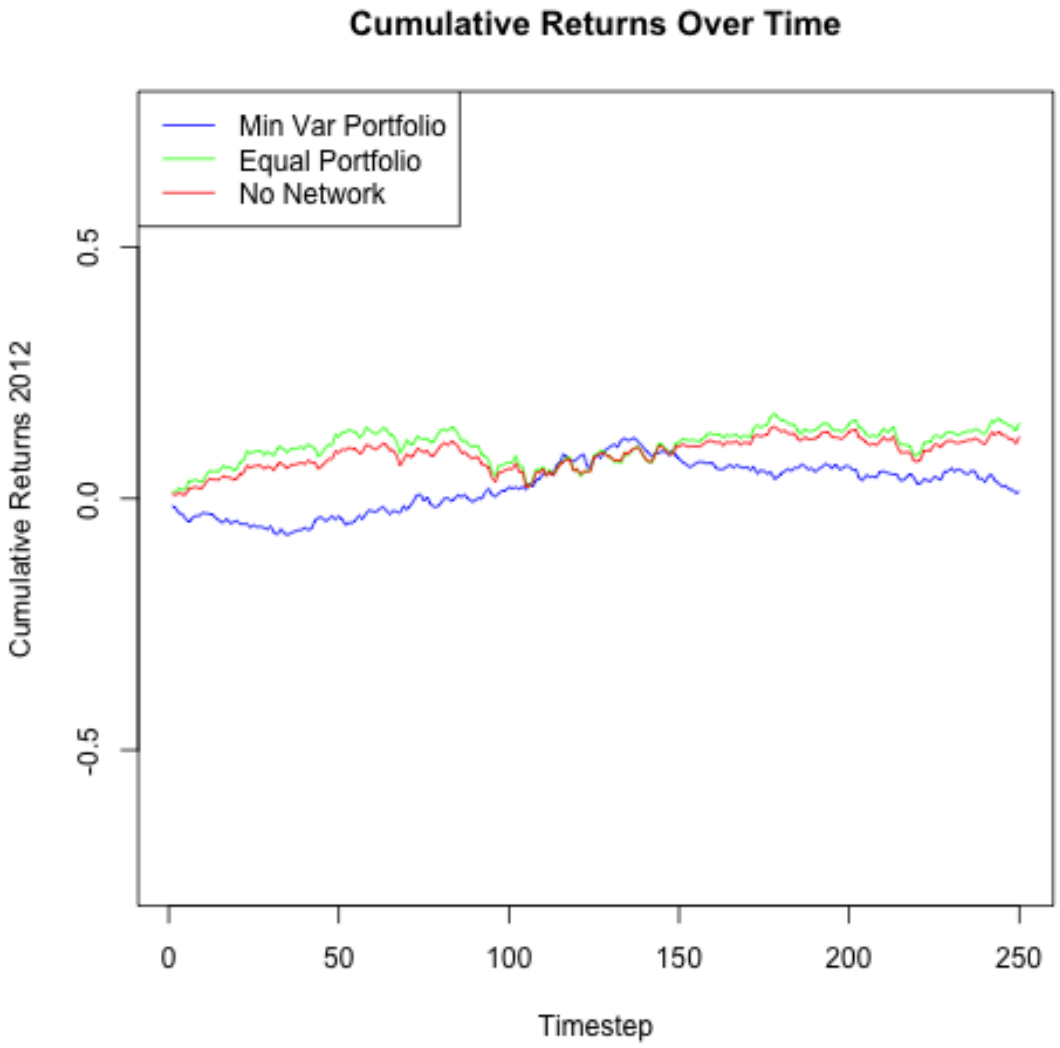


Figure 9: Cumulative Returns in 2011

5 Discussion

5.1 Findings

To verify our method and implementation, we conducted a simulation study using a simulated covariance matrix and networks for a small sample ($p = 20$) of stocks. We demonstrated that our method works well in this low dimensional case, under controlled conditions.

Our results on real market data was more mixed. We tested our model using historical stock market data, with $p = 100$ stocks to estimate optimal portfolios over a 10-year period from 2008-2023. Despite some years of good performance, overall, our model with network data underperformed compared to both an equally weighted portfolio and a model without network data. This was likely due to convergence issues in the algorithm, with effective sample sizes significantly below expectations. We did see evidence of our commodity flow

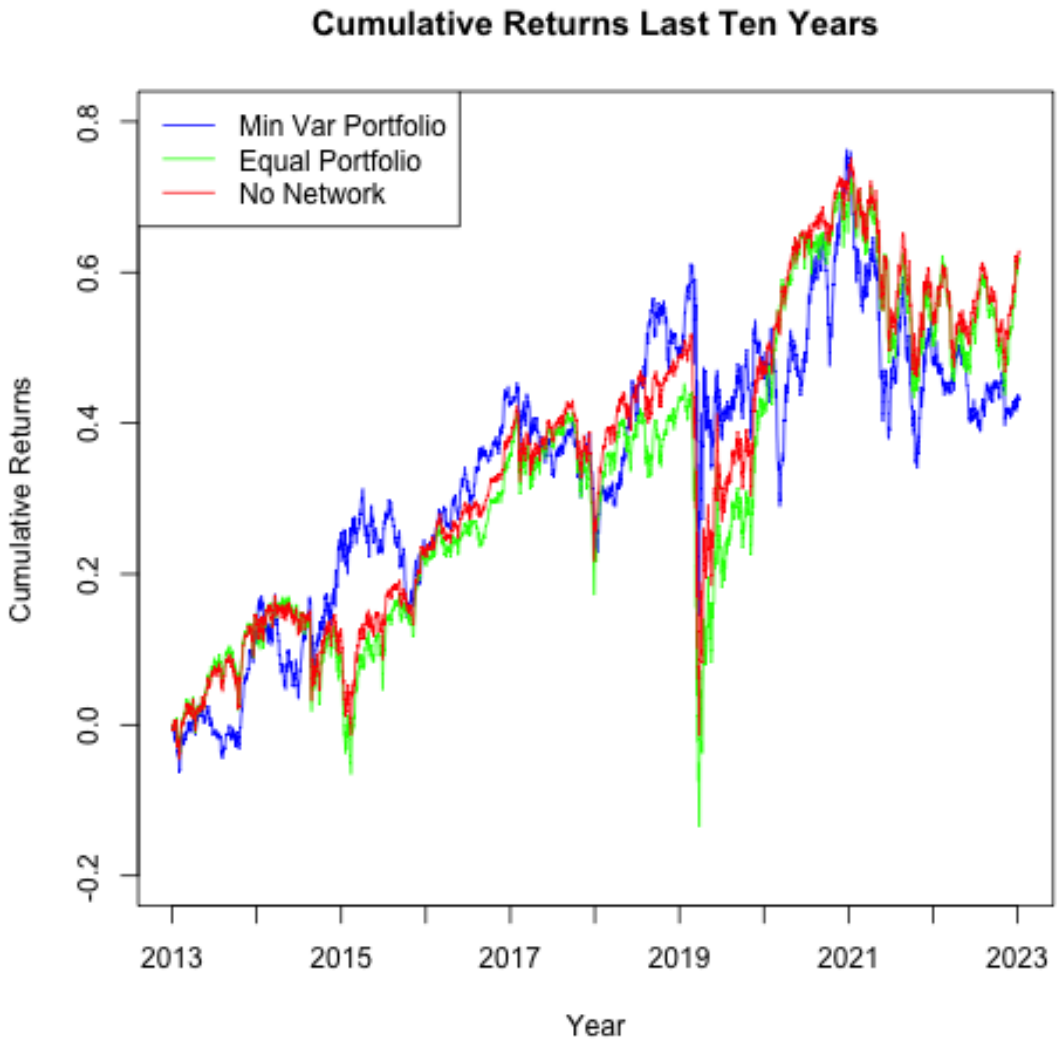


Figure 10: Cumulative Returns (2013-2023)

networks working in some years, (figure 11) and we suspect that fixing the convergence issues may significantly improve our results.

5.2 Markov Chain

The extent to which we can draw conclusions on the efficacy of our model was hampered by the failure of the markov chains to converge. We employed a random walk - Metropolis-Hastings - sampling schema. We now believe this was a poor choice of algorithm for our task for a couple of reasons: first, we had initially planned to sample in a high dimensional space - one in which p was almost double N. However, although random walk MCMC is guaranteed to converge, the time it takes to do so can vary greatly. When dealing with high dimensional spaces, it often requires many more samples (Au and Beck, 2001). An intuition for why full exploration of the posterior is given by considering the complexity of the log-likelihood

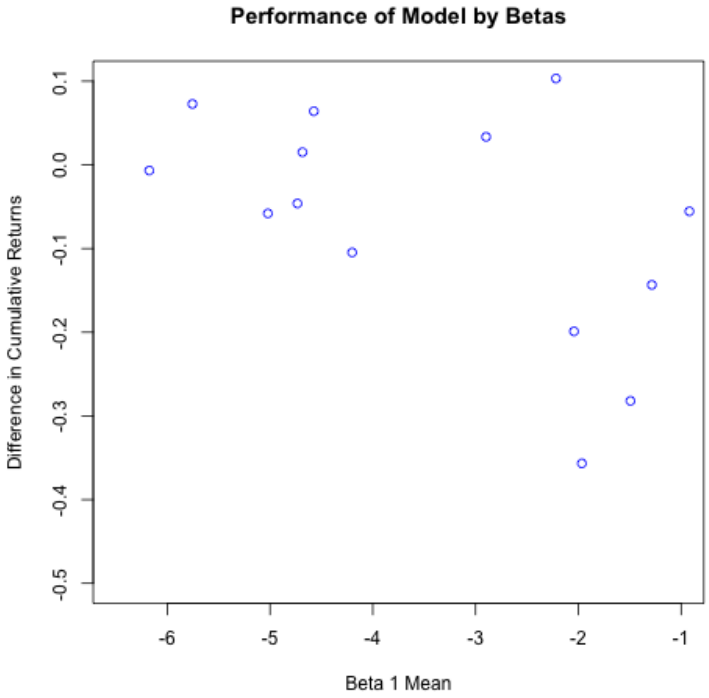
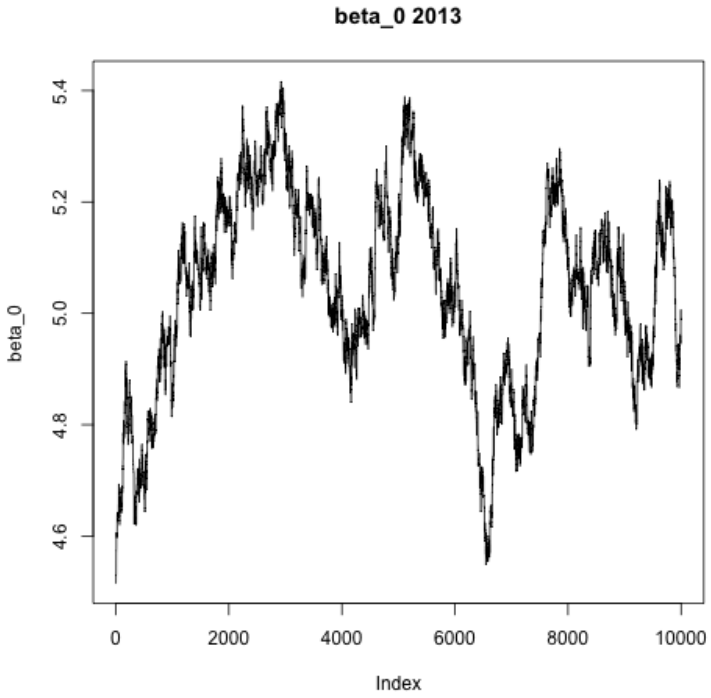


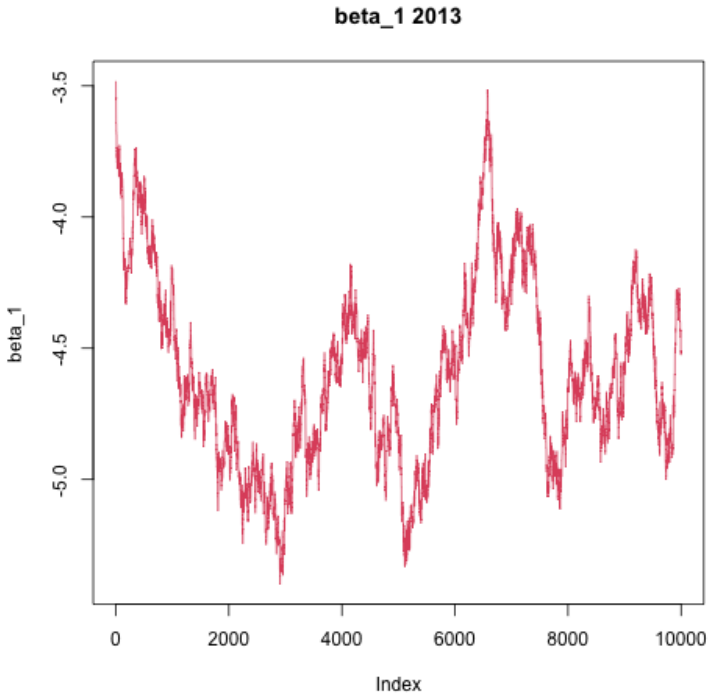
Figure 11: β_1 vs. Portfolio Performance

space used to evaluate the proposal. In our case, we sum the log-likelihood of every element of the precision matrix under a Laplace distribution with rate parameter determined by a non-linear transformation of the betas and the external network. Any perturbation to the betas in the proposal is likely to affect the log-likelihood greatly because of the large number ($\frac{p \times p}{2} - p$) of elements over which it is summed. For a chain to explore the state space effectively, it is imperative that some rare events are sampled. This is the purpose of the draw from the uniform distribution in the algorithm - in some cases, it pays to take a step towards a less likely neighbourhood in case of greater likelihood on the other side. In our case, though, the change in log-likelihood may have been so great that those events were too rare too occur at all, and the state space remained unexplored in reasonable time. This was borne out in our results. During testing, we ran several sets of 50,000 iterations, but the parameters still failed to converge to a stable value. We note that an informative prior would probably be of particular value in decreasing the time to converge, since the parameters are initialised in the optimal neighbourhood.

Secondly, given the instability in our Markov Chains, we should consider that the posterior distribution may be multimodal. That is to say, there are several combinations of β_1 and β_2 that fit the data well. In this case, the wide range of possible values of each beta obtained by the simulation might be a consequence of its oscillation between these modes. posterior modes are located sufficiently close to one another, the chain might take especially long to converge to the greatest mode because of the allure of other mode.



(a) β_0 Chain



(b) β_1 Chain

Figure 12: β Chains from 2013 (20,000 iterations with burn-in = 10,000)

5.3 Portfolio Computation

We introduced the optimal portfolio theory briefly in the introduction, but perhaps failed to convey the extent to which freedom of choice is afforded to the investor in its computation. Throughout our stock market study, we used the minimum variance portfolio described in 4. We chose not to display the maximum returns portfolio (3) to avoid overwhelming the reader with twice the number of plots. Additionally, since most investors prioritize security, the maximum returns portfolio, being highly volatile, might not align with their preferences. However, it should also be noted that the closed-form solutions to the optimal portfolio problem necessarily permit the short selling of securities (in effect designating a negative weight to a stock, or betting against it). If a portfolio contains stocks that are short sold, in order for the weights to sum to one the investor must put in more money than originally intended. Thus, in these cases, the investor is more exposed to the market than in a traditional 'long-only' portfolio like the S&P 500 or the equally weighted portfolio. In the interest of theoretical optimality, we allowed stocks to be short sold, but any investor looking to use our model for their own investments should note that a more secure portfolio would be achieved by constructing instead a 'long-only' portfolio, which can be viewed as a convex optimisation problem and solved using a quadratic program (Polson and Tew, 2000). Unlike the one-size-fits-all closed-form solutions, the optimisation method allows the investor to choose their desired returns-volatility trade-off by setting constraints on the minimum and maximum weight (as well as the nonnegativity constraint). We hope that the reader appreciates the diversity in optimal portfolio construction, and understands why we chose to limit our study to just one methodology.

5.4 Future Directions

There are several modifications to our methodology to improve its performance. First, we discuss assumptions we made to facilitate our model and different model designs that would not have been subject to these conditions.

Recall that our model assumes sparsity in the precision matrix. There are cases in which sparsity cannot be assumed. For example, during large market crashes, Jr and Franca (2011) showed that price dynamics across markets become homogenous, and more generally, the volatility of stock markets is correlated with the correlation between them. In cases such as these, we should expect the precision matrix to be comprised primarily of non-zero entries, and the assumption for penalisation to be void. To achieve instead conditional sparsity, one can control for common factors between the stocks, and then estimate via some penalisation method a precision matrix whose entries represent only the correlation of idiosyncratic components of the stocks. Knowledge of the common factors driving the correlation is not necessary – one can use principal component analysis to estimate their effect Stock and Watson (2002). For simplicity, our model quixotically assumes sparsity across all time periods, but this likely does not hold, for example, during the COVID-19 pandemic or American subprime mortgage crisis periods. Secondly, we model the time series as a multivariate normal distribution. This, again, is for simplicity, and provides a reasonable approximation. However, other distributions, like the multivariate-t, would better model our data. Indeed, Saha et al. (2009) illustrated the inadequacy of a Gaussian

assumption in modelling stock prices. They analysed Dow Jones returns from 1900 – 2006 and found the percentage of trading days during which price movements were three standard deviations or more from their mean was 1.58%, substantially more than the 0.3% expected under a Gaussian paradigm. Such fat-tailed behaviour lends itself to elliptical modelling, which relaxes the assumption on the underlying distribution of the data. Han and Liu (2017) describe a method to estimate the precision matrix using a rank-based (i.e., non-parametric) approach which performs well in high-dimensions. The inclusion of these techniques in our method would likely marginally improve results.

While our commodity flow networks, did appear to be informative, they were built in a slightly naive way, due to the market data not matching exactly with the sector data. With further research, it would be possible to build a more accurate network. We also note that we don't exploit one advantage of our model, it's ability to include more than one network. This could improve inference, particularly in years where our commodity flow network was less informative.

The most notable point of improvement in our method is in the use of the Metropolis Hastings sampling schema. As described in 5.2, the chain failed to explore the posterior distribution adequately in reasonable time. We now believe that Hamiltonian Monte Carlo would have been a more suitable method to ensure our beta estimates were accurate.

6 Conclusion

We proposed, tested and implemented a fully Bayesian GLASSO algorithm, with external network data, building on the work by Jewson et al. (2022). We aimed to improve on their method by determining the β_q parameters in a fully Bayesian way, with the use of the Metropolis-Hastings algorithm. This improvement allowed us to incorporate prior beliefs about our networks into our models and provide a distribution for the β_q's. In order to improve our performance on historical finance data, we built a network for stocks from the S&P 500 index, based on commodity flows between companies.

Our model worked as expected in a controlled test case ($p = 20$), and performed well in isolated years for higher dimensions ($p = 100$) with real data. In the latter case, the algorithm didn't converge in a feasible number of iterations, and our portfolio performed worse than the market average (equally weighted portfolio) over 10 years. Despite this, we saw that our commodity flow network may have improved the performance of our model in some years.

To improve this convergence, we suggest using a more sophisticated sampling scheme, like Hamiltonian Monte Carlo, to explore the posterior distribution more efficiently. Additionally, better results could be achieved by modelling the data using a fat-tailed distribution, or an estimation technique that does not assume sparsity in the precision matrix. Finally, one might also wish to explore different portfolio construction techniques based on their own risk profile.

Overall, the simulation study showed that our method was built on sound theoretical foundations and on stock market data, it outperformed market benchmarks for some years.

References

Au, S.-K. and Beck, J. L. (2001), 'Estimation of small failure probabilities in high dimensions by subset simulation', *Probabilistic engineering mechanics* **16**(4), 263–277.

Best, M. and Grauer, R. (1991), 'On the sensitivity of mean-variance-efficient portfolios to changes in asset means: Some analytical and computational results', *Review of Financial Studies* **4**, 315–42.

Carter, J. S., Rossell, D. and Smith, J. Q. (2023), 'Partial correlation graphical lasso', *Scandinavian Journal of Statistics* **51**(1), 32–63.

Cox, D. R. and Wermuth, N. (2014), *Multivariate dependencies: Models, analysis and interpretation*, Chapman and Hall/CRC.

Efron, B. and Morris, C. (1973), 'Stein's estimation rule and its competitors–an empirical bayes approach', *Journal of the American Statistical Association* **68**(341), 117–130.

Fan, J., Liao, Y. and Liu, H. (2016), 'An overview of the estimation of large covariance and precision matrices', *The Econometrics Journal* **19**(1), C1–C32.

Friedman, J., Hastie, T. and Tibshirani, R. (2007), 'Sparse inverse covariance estimation with the graphical lasso', *Biostatistics* **9**(3), 432–441.

Friedman, J., Hastie, T. and Tibshirani, R. (2019), *Graphical Lasso: Estimation of Gaussian Graphical Models*.
URL: *http://www-stat.stanford.edu/ tibs/glasso*

Gelman, A., Carlin, J., Stern, H., Dunson, D., Vehtari, A. and Rubin, D. (2013), *Bayesian Data Analysis, Third Edition*, Chapman & Hall/CRC Texts in Statistical Science, Taylor & Francis.
URL: *https://books.google.es/books?id=ZXL6AQAAQBAJ*

Han, F. and Liu, H. (2017), 'Statistical analysis of latent generalized correlation matrix estimation in transelliptical distribution', *Bernoulli* **23**, 23–57.

Heinävaara, O., Leppä-Aho, J., Corander, J. and Honkela, A. (2016), 'On the inconsistency of 1-penalised sparse precision matrix estimation', *BMC bioinformatics* **17**, 99–107.

Jewson, J., Li, L., Battaglia, L., Hansen, S., Rossell, D. and Zwiernik, P. (2022), 'Graphical model inference with external network data', *arXiv preprint arXiv:2210.11107* .

Jr, L. and Franca, I. (2011), 'Correlation of financial markets in times of crisis', *Physica A: Statistical Mechanics and its Applications* **391**.

Kan, R. and Zhou, G. (2007), 'Optimal portfolio choice with parameter uncertainty', *Journal of Financial and Quantitative Analysis* **42**, 621–656.

Kuismin, M. O. and Sillanpää, M. J. (2017), 'Estimation of covariance and precision matrix, network structure, and a view toward systems biology', *Wiley Interdisciplinary Reviews: Computational Statistics* **9**(6), e1415.

Loh, P.-L. and Tan, X. L. (2018), 'High-dimensional robust precision matrix estimation: Cellwise corruption under -contamination'.

Markowitz, H. (1952), 'Portfolio selection', *The Journal of Finance* **7**(1), 77–91.

Meinshausen, N. and Bühlmann, P. (2006), 'Consistent neighbourhood selection for high-dimensional graphs with the lasso', *Statist Surv* **2**.

Michaud, R. (1989), 'The markowitz optimization enigma: Is 'optimized' optimal?', *Financial Analysts Journal - FINANC ANAL J* **45**, 31–42.

Nitzsche, D. (2005), *Quantitative Financial Economics: Stocks, Bonds and Foreign Exchange*, Wiley.

Polson, N. G. and Tew, B. V. (2000), 'Bayesian portfolio selection: An empirical analysis of the s&p 500 index 1970–1996', *Journal of Business & Economic Statistics* **18**(2), 164–173.

R Core Team (2024), *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria.
URL: *https://www.R-project.org/*

Saha, A., Malkiel, B. G. and Grecu, A. (2009), 'The clustering of extreme movements: Stock prices and the weather', *Journal of Investment Management* **First Quarter 2009**.

Schwarz, G. (1978), 'Estimating the dimension of a model', *The annals of statistics* pp. 461–464.

Stein, C. (1960), 'Estimation with quadratic loss'.

Stock, J. H. and Watson, M. W. (2002), 'Forecasting using principal components from a large number of predictors', *Journal of the American Statistical Association* **97**(460), 1167–1179.

Tibshirani, R. (1996), 'Regression shrinkage and selection via the lasso', *Journal of the Royal Statistical Society Series B: Statistical Methodology* **58**(1), 267–288.

U.S. Bureau of Labor Statistics (2024), 'Inter-industry relationships (input-output matrix)'. Available at: `https://www.bls.gov/emp/data/input-output-matrix.htm` [Accessed: 2024-04-29].

van de Schoot, R., Depaoli, S., King, R., Kramer, B., Märtens, K., Tadesse, M. G., Vannucci, M., Gelman, A., Veen, D., Willemsen, J. et al. (2021), 'Bayesian statistics and modelling', *Nature Reviews Methods Primers* **1**(1), 1.

Wang, H. (2012), 'Bayesian Graphical Lasso Models and Efficient Posterior Computation', *Bayesian Analysis* **7**(4), 867 – 886.

Wang, J., Kang, Y. and Kemmer, P. (2006), 'An efficient and reliable statistical method for estimating functional connectivity in large scale brain networks using partial correlation', *Front. Neurosci* **10**, 123.

West, M. (1987), 'On scale mixtures of normal distributions', *Biometrika* **74**(3), 646–648.

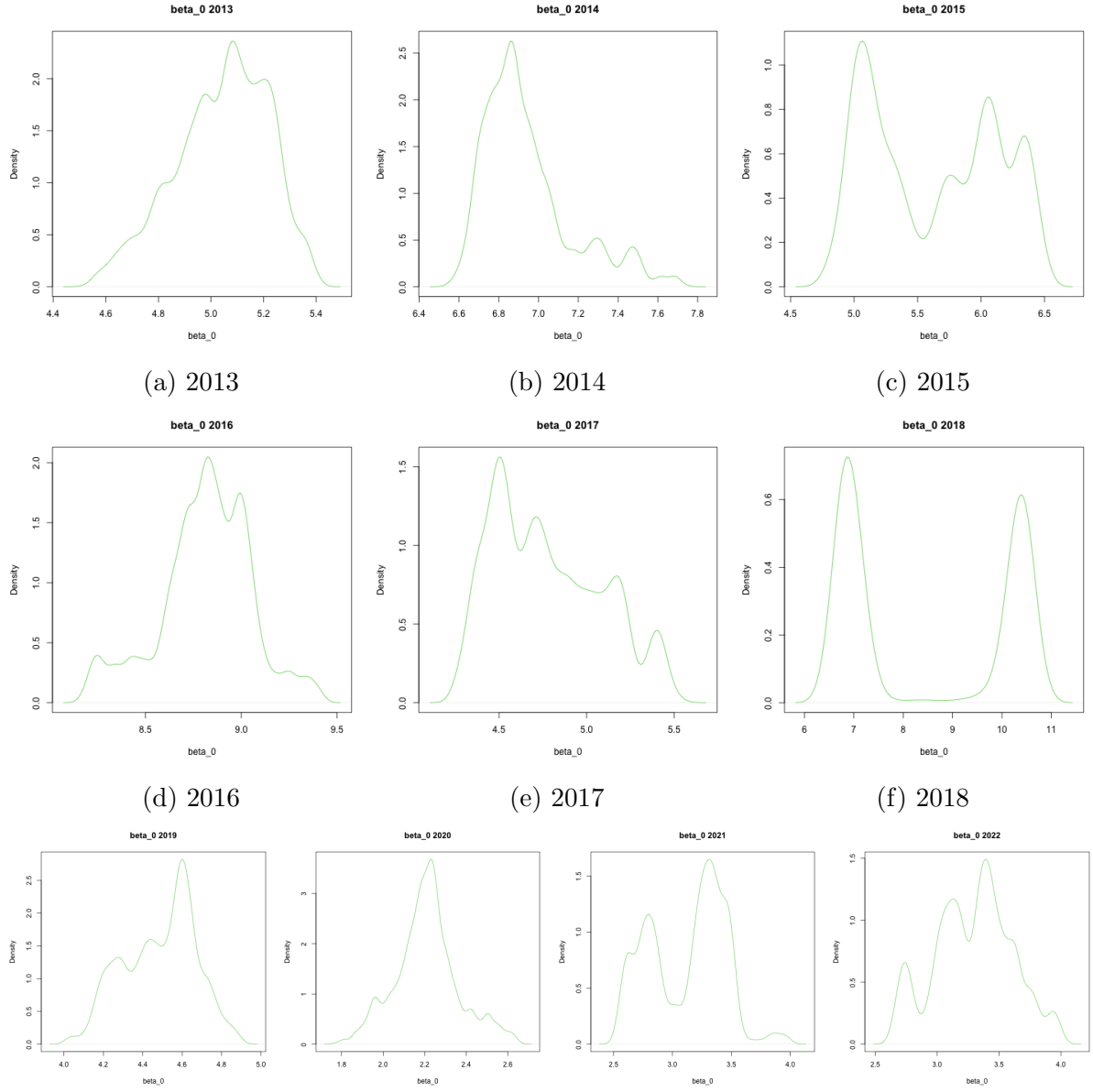
Yuan, M. and Lin, Y. (2007), 'Model selection and estimation in the gaussian graphical model', *Biometrika* **94**(1), 19–35.

Zoominfo (2024). Available at: `https://www.zoominfo.com` [Accessed: 2024-04-04].

A Appendix

A.1 β_0 and β_1 Distributions

This appendix shows the β_0 y β_1 y distributions for the informative networks for all years. The Figure 13 display β_0 distributions plots. Most of the highest modes are in the range of 3 and 5. As we mentioned, most of the distributions present multiple closely spaced modes, which is clear in figure 14f.



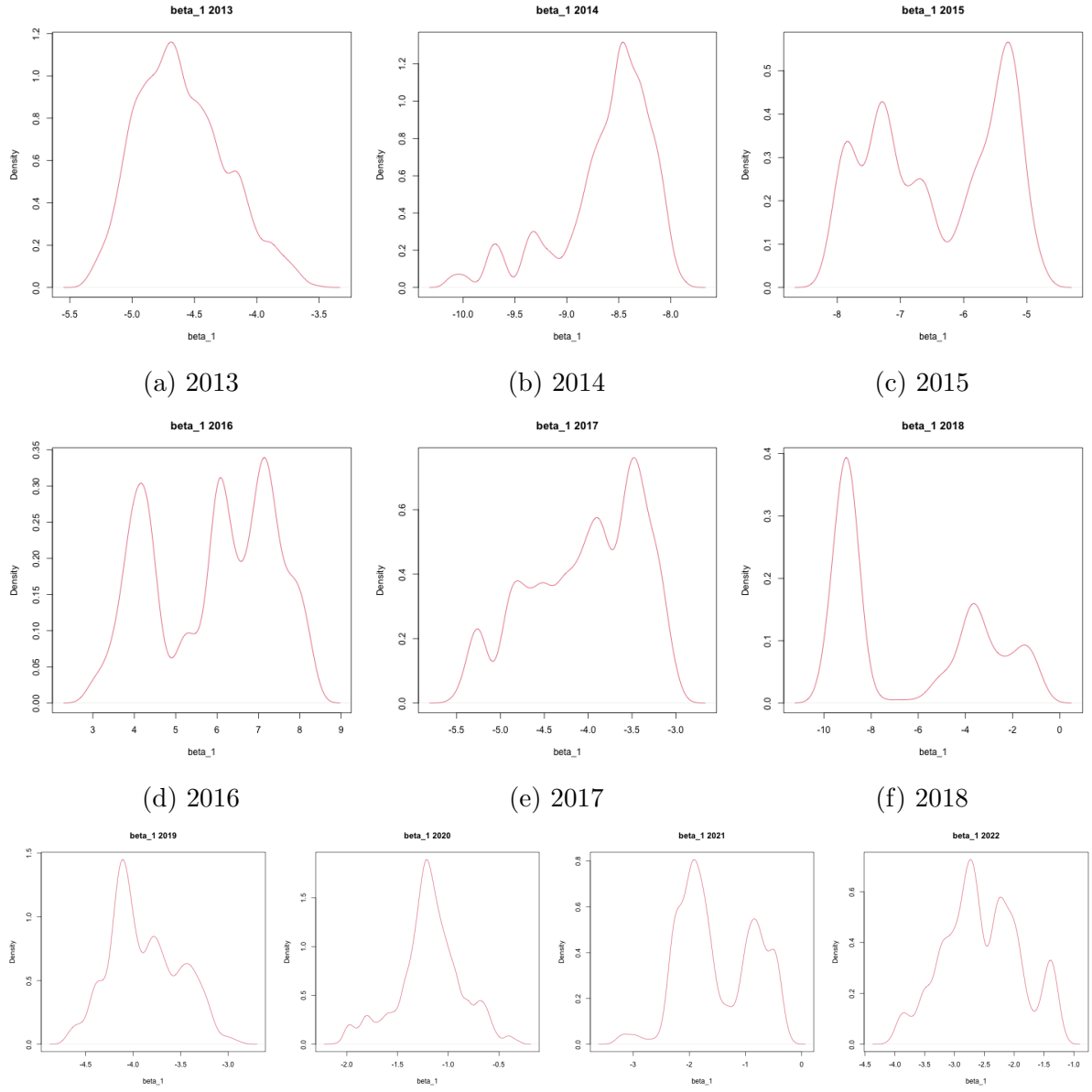
(a) 2013 (b) 2014 (c) 2015

(d) 2016 (e) 2017 (f) 2018

(g) 2019 (h) 2020 (i) 2021 (j) 2022

Figure 13: β_0 Distribution Plots

Now, we show the β_1 distributions for the informative networks for all years (Figure 14). We can note that the highest modes are negative except for 2016. As we mentioned, there are some challenging regions for the Metropolis-Hastings algorithm. This is evident in 2016 and 2017, the former has many closely spaced modes, while the latter has flat zones and a mode before the highest mode.



(a) 2013 (b) 2014 (c) 2015

(d) 2016 (e) 2017 (f) 2018

(g) 2019 (h) 2020 (i) 2021 (j) 2022

Figure 14: β_1 Distribution Plots

A.2 Markowitz Portfolio Theory

As we mentioned, the final portfolio comes from the Markowitz Portfolio. So, in this section we mention some of the key assumptions in our case (Nitzsche, 2005; Efron and Morris, 1973).

- Risk aversion: Investors are rational and prefer to minimize risk for a given level of expected return or maximize expected return for a given level of risk.
- Expected Returns: Investors base their decisions on the expected returns and variances (or standard deviations) of returns for each asset and the covariances between asset returns.
- Single Period: The investment decision is considered for a single time period, and investors plan their portfolios for this one period.
- Normal Distribution of Returns: Asset returns are normally distributed, allowing the use of mean and variance as the key measures for return and risk.
- No Transaction Costs: There are no taxes, transaction costs, or restrictions on short selling.
- Divisibility: All assets are infinitely divisible, meaning investors can hold fractional shares of any asset.
- Homogeneous Expectations: All investors have access to the same information and therefore have the same expectations regarding future returns, variances, and covariances of assets.
- Risk-Free Rate: There is a risk-free rate at which investors can lend or borrow money.

A.3 Gibbs Sampler Θ Partition

In the Gibbs sampler, we partition Θ, τ and υ in the following way:

$$\Theta = \begin{pmatrix} \Theta_{11} & \theta'_{12} \\ \theta_{12} & \theta_{22} \end{pmatrix}, \quad S = \begin{pmatrix} S_{11} & s_{12} \\ s'_{12} & s_{22} \end{pmatrix}, \quad \upsilon = \begin{pmatrix} \upsilon_{11} & \tau_{12} \\ \tau'_{12} & 0 \end{pmatrix} \tag{18}$$