Graphical Network and Topology Estimation for Autoregressive Models using Gibbs Sampling

Marija Iloska Yousef El-Laham Mónica F. Bugallo

Department of Electrical and Computer Engineering

Stony Brook University, Stony Brook, NY 11794-2350

{marija.iloska, yousef.ellaham, monica.bugallo}@stonybrook.edu

Abstract

In this paper, we propose novel strategies based on Gibbs sampling for the estimation of the coefficients and topology of a graphical network represented by a first-order vector autoregressive model. As the topology and the coefficients are closely related, obtaining their Markov chains together is a nontrivial task. When incorporating both in a Gibbs-based sampler, the topology samples at each iteration are decisive factors in how information for the corresponding coefficient samples is propagated. We propose new Gibbs-based samplers that differ in the sampling strategies and scanning order used for their operation. We ran a series of experiments on simulated data to analyze and compare the samplers' performances with dimension of data, data size, and choice of prior. The best performing sampler was also applied to real data related to a financial network. Converged Markov chains of coefficient and topology elements of the network attest to the method's validity, and plots illustrating posterior distributions of the predicted data against the observed data indicate promising inference for real data applications.

Keywords: Gibbs sampling, network, topology, vector autoregressive models, financial network.

1 Introduction

Dynamical systems embody processes that vary in time. Examples of such systems are gene expressions [1, 2], animal flocking [3, 4], chemical reactions [5, 6] and many others. For a given system of interest, we want to understand how it functions and predict how it evolves in time. In any system, the underlying processes produce some observable signals, which can be collected and summed into time-indexed data. Clever use of

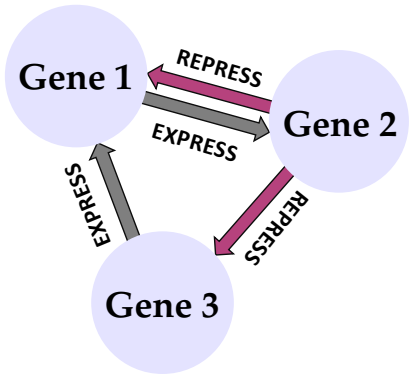


Figure 1: Gene regulatory network.

the observations is crucial to extract the information they contain regarding the inner-workings of the system. One way to model such systems is by using a graphical network. Graphical networks are composed of nodes and edges. The nodes represent the on-going processes in the system, and the edges contain all of the transitionary information governing their evolution. For example, the evolution of gene expressions can be represented as a dynamical system comprised of genes whose interactions are modeled by gene regulatory networks. In the simple gene regulatory network illustrated in Fig. 1, the expression of Gene 1 triggers an expression in Gene 2, however, the expression of Gene 2 triggers a repression in Gene 1 and Gene 3. Essentially, the nodes can be seen as the gene expressions themselves, and the edges as the direction of influence between the gene expressions in time [1]. In a bird flocking system, the graphical network gives the amount of influence that the motion of each individual target (bird) receives on its motion by the trajectories of the rest of the group [3]. In a set of chemical reactions, graphical networks encode information about the kinetics of the reactions involved (i.e. change and rate of change of concentrations of reactants, products, and potential intermediates), as well as the interactions between the chemical species of each reaction [5]. Gene expressions, moving targets, chemical species and generally any component of a given system can be seen as a node of the network, and the connectedness among components as the edges. Since not all of the nodes are expected to interact with each other in time, the network has a structure which maintains the temporal links (or lack thereof) between nodes. This structure is known as the topology of the system [7]. Mathematically, the graphical network is represented by a matrix. Typically, this is a sparse matrix whose nonzero entries signify a connection between the nodes in time, and whose zero entries signify no connection. Its topology however, is represented with a matrix that carries only the connectedness of the nodes, disregarding the strength of the individual connections.

Networks can be directed or undirected [8]. Undirected networks assume a mutual link between two given nodes and are described with symmetric matrices [9]. Directed networks dive deeper into the individual directions of links between nodes, and are described with asymmetric matrices. Inferring directed networks

from time-series data has proven to be a challenging task, as they do not share many of the nice mathematical properties which arise in undirected networks (due to symmetry) [10]. The requirement of inferring a sparse matrix adds quite a bit to the complexity of the task, as there is no direct and/or obvious way of selecting which elements of the matrix to set to zero.

A common and suitable way of modeling dynamical systems for network inference is using autoregression [11]. Vector autoregressive (VAR) models describe the relationship between current and past observations as stochastic processes. Most common approaches to network inference have turned to regularization techniques on VAR models by reformulating the problem as a linear regression, in which the coefficients of the regressors represent the elements of the matrix describing the network [12, 13, 14]. In particular, the least absolute shrinkage and selection operator (LASSO) [15] is capable of simultaneously giving an estimate of the network coefficients and the topology [16, 17]. However, LASSO tends to overselect the coefficients, and can ignore temporal dependence when applied to VAR models [18]. The network and topology estimation problem has also received significant attention from the Bayesian community, commonly as a joint problem along with tracking hidden states. In that context, hidden VAR models have become popular choices for dynamical systems whose latent states can be directly or indirectly measured, but require estimation (usually done through filtering). Many algorithms of iterative nature have been proposed in the literature. For instance, [19] makes use of an automatic relevance determination (ARD) prior structure, which in essence, is a Bayesian hierarchical model that induces sparsity by reducing the posterior distribution of the parameter to almost a point mass at zero. The parameters of the model are estimated by expectation-maximization, and the final estimates are pruned based on an arbitrarily chosen threshold. Implementation of this method was demonstrated in the case where the true states are also unknown (only observed in noise), using particle filtering (PF) [20, 21]. Another method proposed in [22] iteratively uses PF for the states and Kalman filtering (KF) [23] for the elements of the network. The elements are assumed to follow a small Gaussian noise, to allow for the use of KF. Then, the resulting estimates are sparsified via LASSO regularization. Further, [24] uses multiple particle filtering (MPF) to track the evolution of gene expressions, and incorporates gene topology inference by augmenting the topology as an unknown in the problem and finding its posterior. Convergence of MPF, however, is not guaranteed as the particle weights are only approximated [25].

Some other works have turned to Markov chain Monte Carlo (MCMC) [26, 27, 28, 29] sampling. In particular, [30] derives a Gibbs sampler (GS) for topology inference in a hierarchical Bayesian model, which incorporates informative priors based on the degree of the networks. Sparsity is induced by using spike and slab priors for the elements in the network, which are distributions composed of some density (slab) and a point mass (spike) at zero. Another approach in [3] uses KF to track the trajectory of bird flocking data, and estimates their topology with a blocked GS (BGS). A more encompassing view of this method was proposed by [31], in which a particle GS was designed to jointly estimate the hidden states as well as the topology and the network in the model. Generally, proposed models are expressed only in terms of the network and implicitly depend on the topology, making topology inference more challenging. Both previously mentioned works, however, have found a way to explicitly extract the topology in the model, and it is the different base sampling within the GSs that distinguishes the two approaches. Sampling the topology using GSs can be challenging, since the estimate of each of its elements determines the flow of information and estimation of the graphical network elements. This raises important questions related to network topology estimation using Gibbs sampling: How do different methods of sampling affect the estimates? How does the convergence rate change for the different samplers? Do the performances of the methods differ when handling higher dimensional data? Generally, the greatest challenges in Bayesian inference emerge when working with a large number of unknowns, which unfortunately happens in most real-world problems. In the literature, these challenges which arise from the overwhelming number of unknown parameters fall under what is known as the curse of dimensionality [32]. To tackle this issue, it is crucial to understand the behavior of the system under varying conditions. This includes obtaining a clear relationship between number of available observations (data size) and the number of unknown parameters (network size), understanding the influence of the choice of prior distribution, and most importantly, analyzing the empirical performance of the sampling algorithms at hand.[1] This paper offers a new perspective on topology estimation using MCMC schemes on a first-order linear VAR model. More specifically, GSs based on different sampling strategies are derived and evaluated on the estimation of the network and its topology. Theoretically, any GS guarantees convergence [29, 34, 35]. However, since the topology is not explicitly included in the model and it strictly

[1]Note that since our interest focuses on the coefficients and the topology, we make the assumption that the observations are the true states. In a scenario where the evolution of the states requires tracking as well, these methods can easily be coupled with PF based approaches [33] applied to hidden VAR models to capture a full analysis of the considered dynamical system.

dictates the estimation for the network coefficients, its integration in the GS poses additional difficulties such as blocking the proper flow of the sampler or leading to extremely slow convergence. The contributions of this paper are in the design of novel strategies based on Gibbs sampling for network and topology estimation, and a detailed comparative analysis of the proposed methods. In particular, the effects of the scanning order for the sampling, point estimation, marginal versus blocked Gibbs sampling, and matrix versus element wise sampling are considered. Simulations comparing the relationship between different choices of prior, number of observations, and dimension size are shown for synthetic data. The newly proposed Gibbs-based sampler with the highest performance is also applied to a real world problem of a case related to estimation of financial networks. Additionally, LASSO regularization is implemented to both the synthetic and real data to further assess the performance of the considered samplers.

2 Problem Formulation

In a first order VAR model, the current measurement is conditionally independent of all the past measurements given the last measurement, preserving a Markovian property [36]. Bayesian inference is a suitable way to analyze a VAR model [37]. Initially, all the beliefs about the unknown parameters in a model before acquiring any data are summarized by a *prior distribution*. Then, the collected observations formulate a *likelihood function*, which serves to filter out the improbable values from the prior via Bayes' theorem [37]. The resulting updated distribution carries all the relevant information about the unknown parameters, and is known as the *posterior distribution*. In this work, we consider the following first-order linear VAR model:

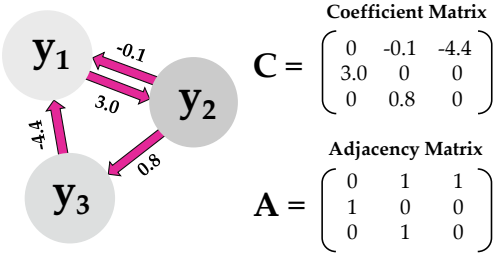


Figure 2: Network of dimension $d_y = 3$.

$$\begin{aligned} \mathbf{y}_1 &\sim p(\mathbf{y}_1) \\ \mathbf{y}_t &= \mathbf{C}\mathbf{y}_{t-1} + \mathbf{u}_t, \quad t = 2, \ldots, T, \end{aligned} \tag{1}$$

with observations $\{\mathbf{y}_t \in \mathbb{R}^{d_y}\}_{t=1}^{T}$ whose transition is governed by the coefficient matrix $\mathbf{C} \in \mathbb{R}^{d_y \times d_y}$, and corrupted by Gaussian noise $\mathbf{u}_t \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbb{I}_{d_y})$. The elements of $\mathbf{C}$ describe the interactions between each of the univariate time-series observations. In particular, the element C_{jk} represents a linear relationship between $y_{j,t}$ and $y_{k,t-1}$. If $y_{k,t-1}$ does not affect $y_{j,t}$, then their corresponding transition element $C_{jk} = 0$. From a graphical point of view, the time-series $\{\mathbf{y}_t \in \mathbb{R}^{d_y}\}_{t=1}^{T}$ are considered nodes in the network, whose edges are laid out in the coefficient matrix $\mathbf{C}$. In this way, $\mathbf{C}$ represents the directed graphical network of the system (see Fig. 2). It is worth mentioning that causal relationships between time-series (commonly modeled with lagged terms) are often examined using Granger causality tests [38], and in that sense, our formulation can be seen as a VAR model with a lag order of 1.

Let $\mathbf{A}$ denote the adjacency matrix of the network $\mathbf{C}$. The adjacency matrix represents the topology network, containing information only about the presence or absence of connections between nodes. More formally, its elements are defined as

$$A_{jk} = \begin{cases} 0, & \text{if } C_{jk} = 0, \\ \\ 1, & \text{if } C_{jk} \neq 0 \end{cases} \tag{2}$$

for $j = 1, \ldots, d_y$, and $k = 1, \ldots, d_y$. Essentially, it holds the causal relationships of the nodes in time, which is crucial to understand the system and its evolution.

In the following sections, we describe and compare the performance of several existing (Algorithms 2, 3 and 7) and newly proposed (Algorithms 4 - 6, and 8 - 11) schemes based on Gibbs sampling for estimation of the topology $\mathbf{A}$ and the network coefficient $\mathbf{C}$. Specifically, we focus on the importance of the scanning order, and emphasize the differences between vanilla Gibbs sampling [39] and blocked Gibbs sampling [40]. Here, vanilla Gibbs sampling refers to sampling each parameter directly from its marginal conditional posterior distribution, as in agreement with the definition of Gibbs sampling (see Section 3 .1), and blocked Gibbs sampling refers to sampling groups of parameters from their joint posterior distribution (see Section 3 .2).

3 Gibbs Sampling

A GS is a MCMC algorithm, which generates samples from the posterior distribution of a set of unknown static parameters in a probabilistic model [39]. It does so by iteratively sampling each unknown from its respective conditional posterior distribution, while keeping the rest of the unknown parameters fixed to the most recently sampled values. Ultimately, a Markov chain is generated whose stationary distribution converges to the true joint posterior, providing a set of samples for each unknown [41, 34].

3 .1 Gibbs Sampler

Here, we illustrate the principle of a GS in a simple case of three unknowns. Let $\boldsymbol{\theta} = [\theta_1, \theta_2, \theta_3]^\top$ contain all the static unknown parameters in a model. Given a set of observations $\mathbf{y}_{1:T}$, we seek to obtain the joint posterior distribution $p(\theta_1, \theta_2, \theta_3|\mathbf{y}_{1:T}) = p(\boldsymbol{\theta}|\mathbf{y}_{1:T})$. A GS iteratively follows the steps

$$\textbf{i)} \quad \theta_1^{(i)} \sim p(\theta_1|\theta_2^{(i-1)}, \theta_3^{(i-1)}, \mathbf{y}_{1:T})$$

$$\textbf{ii)} \quad \theta_2^{(i)} \sim p(\theta_2|\theta_1^{(i)}, \theta_3^{(i-1)}, \mathbf{y}_{1:T})$$

$$\textbf{iii)} \quad \theta_3^{(i)} \sim p(\theta_3|\theta_1^{(i)}, \theta_2^{(i)}, \mathbf{y}_{1:T})$$

for $i = 1, \ldots, I$. The collected samples of each unknown are stored. A burn-in period I_0 is set out to avoid errors introduced by the early samples drawn prior to the convergence of the simulated Markov chain [41]. Thus, the first I_0 samples are discarded, and the Markov chain obtained is $\mathcal{X} = \left\{\theta_1^{(i)}, \theta_2^{(i)}, \theta_3^{(i)}\right\}_{i=I_0+1}^{I}$.

3 .2 Blocked Gibbs Sampler

To implement a GS, it is crucial to be able to not only derive closed form expressions of the conditional posterior distributions, but also sample from them with ease. In most scenarios, this is a nontrivial task. Approaches that sample from posterior approximations have been proposed, however, at a much higher computational cost [42]. One possible alternative is to use a BGS. [40]. A BGS groups two or more parameters and samples from their joint conditional distribution, rather than from their individual marginals.

Continuing with the simple example described above, a BGS which groups, say, θ_1 and θ_2 would sample as

$$\begin{aligned} \textbf{i)} \quad & [\theta_1^{(i)}, \theta_2^{(i)}]^\top \sim p(\theta_1, \theta_2 | \theta_3^{(i-1)}, \mathbf{y}_{1:T}) \\ \textbf{ii)} \quad & \theta_3^{(i)} \qquad\qquad \sim p(\theta_3 | \theta_1^{(i)}, \theta_2^{(i)}, \mathbf{y}_{1:T}) \end{aligned}$$

for $i = 1, \ldots, I$. Similarly as before, a burn-in period I_0 is applied to discard the early samples. In cases where it is tricky to sample directly from the joint distribution, one could sequentially sample from the product terms

$$p(\theta_1, \theta_2 | \theta_3, \mathbf{y}_{1:T}) = p(\theta_2 | \theta_1, \theta_3, \mathbf{y}_{1:T}) p(\theta_1 | \theta_3, \mathbf{y}_{1:T}) \tag{3}$$

in the following order

$$\begin{aligned} \textbf{i)} \quad & \theta_1^{(i)} \sim p(\theta_1 | \theta_3^{(i-1)}, \mathbf{y}_{1:T}) \\ \textbf{ii)} \quad & \theta_2^{(i)} \sim p(\theta_2 | \theta_1^{(i)}, \theta_3^{(i-1)}, \mathbf{y}_{1:T}) \\ \textbf{iii)} \quad & \theta_3^{(i)} \sim p(\theta_3 | \theta_1^{(i)}, \theta_2^{(i)}, \mathbf{y}_{1:T}). \end{aligned}$$

In this way, θ_1 is sampled independently from θ_2, and θ_2 is sampled conditioned on the current θ_1. Depending on the problem at hand, the sampling order of θ_1 and θ_2 can and/or may need to be switched if it is easier to obtain the distributions with the condition reversed i.e. $p(\theta_1 | \theta_2, \theta_3, \mathbf{y}_{1:T})$ and $p(\theta_2 | \theta_3, \mathbf{y}_{1:T})$.

4 Proposed Samplers

In the current form, the VAR model in (1) is parameterized only by $\mathbf{C}$. In order to explicitly include the topology, the model can be alternatively written by decomposing the coefficient matrix into $\mathbf{C} = \mathbf{A} \circ \tilde{\mathbf{C}}$, where $\tilde{\mathbf{C}}$ is the same as the coefficient matrix $\mathbf{C}$ if and only if the network is fully connected (i.e. all $A_{jk} = 1$), and $\circ$ denotes the Hadamard product. The model becomes

$$\mathbf{y}_t = (\mathbf{A} \circ \tilde{\mathbf{C}}) \mathbf{y}_{t-1} + \mathbf{u}_t, \quad t = 2, \ldots, T. \tag{4}$$

Algorithm 1 Gibbs Sampler

1: **Initialization:** Set $A_{jk}^{(0)} \sim Bernoulli(\rho)$, for $j = 1, \ldots, d_y$, for $k = 1, \ldots, d_y$
2: **for** $i = 1, \ldots, I$ **do**
3: Sample $\tilde{\mathbf{C}}^{(i)}$ from the exact conditional

$$\tilde{\mathbf{C}}^{(i)} \sim p(\tilde{\mathbf{C}}|\mathbf{A}^{(i-1)}, \mathbf{y}_{1:T})$$

4: Sample $\mathbf{A}^{(i)}$ from the exact conditional

$$\mathbf{A}^{(i)} \sim p(\mathbf{A}|\tilde{\mathbf{C}}^{(i)}, \mathbf{y}_{1:T})$$

5: **end for**
6: **Output**: Return $\mathcal{X}_{GS} = \{\tilde{\mathbf{C}}^{(i)}, \mathbf{A}^{(i)}\}_{i=I_0+1}^{I}$.

Now, the unknowns for the GS are $\boldsymbol{\theta} = [\tilde{\mathbf{C}}, \mathbf{A}]$. The general structure of the sampler is summarized in Algorithm 1. Once the sampler has converged, it gives us the Markov chain $\mathcal{X}_{GS} = \left\{\tilde{\mathbf{C}}^{(i)}, \mathbf{A}^{(i)}, \right\}_{i=I_0+1}^{I}$.

In general, $\tilde{\mathbf{C}}$ can be sampled either by matrix, vector by vector, or element by element. However, unlike $\tilde{\mathbf{C}}$, sampling $\mathbf{A}$ can only be done element by element as the posterior distribution of the whole matrix $\mathbf{A}$ is intractable. For brevity, we will consider only cases which sample by matrix and element by element. In this section, we discuss 10 different GSs based on various sampling strategies (three existing and seven novel). The newly proposed schemes summarized in Algorithms 4 - 6 differ in the order and manner that sampling is performed, and the schemes in Algorithm 8 - 11 in the overall estimation of the network and its topology by combining point estimates with Gibbs sampling. Detailed derivations for the posterior distributions of each of the samplers can be found in the Appendix. Note that for simplicity in the notation we employ the variable $\mathbf{S}_{\boldsymbol{\theta}}$ to denote a collection of all the most recently obtained (sampled, estimated or initialized) values of some vector of unknowns $\boldsymbol{\theta}$; this includes samples from both iteration i and $i-1$. Further, we use $\boldsymbol{\theta}_{-jk}$ to denote a vector which contains all the elements of $\boldsymbol{\theta}$ excluding the element θ_{jk}.

4 .1 Element Gibbs Sampler

We start by constructing a GS as described in Section 3 .1, which samples $\tilde{\mathbf{C}}$ element by element. We refer to it as element GS (EGS). Sampling the element $\tilde{C}_{jk}$ needs special consideration as it depends on the most recently sampled value of the corresponding topology element A_{jk}. If $A_{jk}^{(i-1)} = 0$, then no information about $\tilde{C}_{jk}$ is carried to iteration i, so we must sample $\tilde{C}_{jk}^{(i)}$ from its prior.

Sampling $\tilde{C}_{jk}$

Let $p(\tilde{C}_{jk}) = \mathcal{N}(0, \sigma_c^2)$ be the prior distribution of $\tilde{C}_{jk}$. Then, its posterior can be found as

$$\begin{aligned} p(\tilde{C}_{jk}|\tilde{\mathbf{C}}_{-jk}, \mathbf{A}, \mathbf{y}_{1:T}) &\propto p(\tilde{C}_{jk}) p(\mathbf{y}_{1:T}|\tilde{C}_{jk}, \tilde{\mathbf{C}}_{-jk}, \mathbf{A}) \\ &\propto p(\tilde{C}_{jk}) \prod_{t=2}^{T} p(\mathbf{y}_t|\mathbf{y}_{t-1}, \tilde{\mathbf{C}}, \mathbf{A}) = \mathcal{N}(\tilde{C}_{jk}|\tilde{\mu}_{jk}, \tilde{\sigma}_{jk}^2), \end{aligned} \tag{5}$$

where we define

$$\tilde{\mu}_{jk} = \frac{\tilde{\sigma}_{jk}^2}{\sigma^2} A_{jk} \sum_{t=2}^{T} (y_{jt} y_{k,t-1} - y_{k,t-1} \sum_{\substack{m=1 \\ m \neq k}}^{d_y} A_{jm} \tilde{C}_{jm} y_{m,t-1}), \qquad \tilde{\sigma}_{jk}^2 = \frac{\sigma_c^2 \sigma^2}{{A_{jk}}^2 \sigma_c^2 \sum_{t=2}^{T} y_{k,t-1}^2 + \sigma^2}. \tag{6}$$

These parameters are conceptually consistent with the sampling dependence of $\tilde{C}_{jk}$ on A_{jk}.

Sampling A_{jk}

The posterior distribution of a single element A_{jk} of the topology for the EGS can be expressed as

$$\begin{aligned} p(A_{jk}|\tilde{\mathbf{C}}, \mathbf{A}_{-jk}, \mathbf{y}_{1:T}) &\propto p(A_{jk}) p(\mathbf{y}_{1:T}|\tilde{\mathbf{C}}, A_{jk}, \mathbf{A}_{-jk}) \\ &\propto p(A_{jk}) \prod_{t=2}^{T} p(\mathbf{y}_t|\mathbf{y}_{t-1}, \tilde{\mathbf{C}}, A_{jk}, \mathbf{A}_{-jk}) \end{aligned} \tag{7}$$

where we assume that $\tilde{\mathbf{C}}$ and A_{jk}, for $j = 1, \ldots, d_y$, for $k = 1, \ldots, d_y$, are independent in their priors, and take the prior $p(A_{jk}) = \text{Bernoulli}(\rho)$. Further, we define

$$\begin{aligned} \alpha_{jk}^{+} &= p(A_{jk} = 1) \prod_{t=2}^{T} p(\mathbf{y}_t|\mathbf{y}_{t-1}, \tilde{\mathbf{C}}, A_{jk} = 1, \mathbf{A}_{-jk}) \\ \alpha_{jk}^{-} &= p(A_{jk} = 0) \prod_{t=2}^{T} p(\mathbf{y}_t|\mathbf{y}_{t-1}, \tilde{\mathbf{C}}, A_{jk} = 0, \mathbf{A}_{-jk}). \end{aligned} \tag{8}$$

to be the unnormalized probabilities that $A_{jk} = 1$ and $A_{jk} = 0$, respectively. From here, the conditional posterior probability that $A_{jk} = 1$ is

$$\alpha_{jk} = \frac{\alpha_{jk}^{+}}{\alpha_{jk}^{+} + \alpha_{jk}^{-}}. \tag{9}$$

The pseudocode for the EGS is shown in Algorithm 2.

Algorithm 2 Element Gibbs Sampler

Initialization:
Set ${A_{jk}}^{(0)} \sim Bernoulli(\rho)$, $\tilde{C}_{jk}^{(0)} \sim \mathcal{N}(0, \sigma_c^2)$, for $j = 1, \ldots, d_y$, for $k = 1, \ldots, d_y$.
for $i = 1, \ldots, I$ **do**
  for $j = 1, \ldots, d_y$ **do**
    for $k = 1, \ldots, d_y$ **do**
      Sample $\tilde{C}_{jk}^{(i)}$ from (6)

$$\tilde{C}_{jk}^{(i)} \sim \mathcal{N}(\tilde{C}_{jk} | \tilde{\mu}_{jk}, \tilde{\sigma}_{jk}^2)$$

      Sample $A_{jk}^{(i-1)}$ from (9)

$$A_{jk}^{(i)} \sim p(A_{jk} | \mathbf{S}_{\tilde{\mathbf{C}}}, \mathbf{S}_{\mathbf{A}_{-jk}}, \mathbf{y}_{1:T})$$

    end for
  end for
end for
Output: $\mathcal{X}_{EGS} = \{\tilde{\mathbf{C}}^{(i)}, \mathbf{A}^{(i)}\}_{i=I_0+1}^{I}$.

Algorithm 3 Element Gibbs Sampler reversed

Initialization:
Set ${A_{jk}}^{(0)} \sim Bernoulli(\rho)$, $\tilde{C}_{jk}^{(0)} \sim \mathcal{N}(0, \sigma_c^2)$, for $j = 1, \ldots, d_y$, for $k = 1, \ldots, d_y$.
for $i = 1, \ldots, I$ **do**
  for $j = 1, \ldots, d_y$ **do**
    for $k = 1, \ldots, d_y$ **do**
      Sample $A_{jk}^{(i)}$ from (9)

$$A_{jk}^{(i)} \sim p(A_{jk} | \mathbf{S}_{\tilde{\mathbf{C}}}, \mathbf{S}_{\mathbf{A}_{-jk}}, \mathbf{y}_{1:T})$$

      Sample $\tilde{C}_{jk}^{(i)}$ from (6)

$$\tilde{C}_{jk}^{(i)} \sim \mathcal{N}(\tilde{C}_{jk} | \tilde{\mu}_{jk}, \tilde{\sigma}_{jk}^2)$$

    end for
  end for
end for
Output: $\mathcal{X}_{EGSr} = \{\tilde{\mathbf{C}}^{(i)}, \mathbf{A}^{(i)}\}_{i=I_0+1}^{I}$.

4 .2 Element Gibbs Sampler reversed

As we want to test the effect of the scanning order in the sampling on the convergence of the methods, we construct a sampler which samples the topology elements first, followed by the coefficient elements. In this case, we simply reverse the order of sampling A_{jk} and $\tilde{C}_{jk}$ and use the same conditional posteriors given in Section 4 .1. We refer to this sampler as element GS reversed (EGSr). Side to side comparison of the algorithms for EGS and EGSr is shown in Algorithms 2 and 3.

4 .3 Matrix Gibbs Sampler

Here, we construct a GS which samples $\tilde{\mathbf{C}}$ by matrix, and the topology $\mathbf{A}$ element by element. This changes the dependence between the elements of $\tilde{\mathbf{C}}$ when sampling, as compared to the EGSs. We refer to this sampler as matrix GS (MGS).

Sampling $\tilde{\mathbf{C}}$

We start by writing

$$p(\tilde{\mathbf{C}} | \mathbf{A}, \mathbf{y}_{1:T}) \propto p(\tilde{\mathbf{C}}) p(\mathbf{y}_{1:T} | \tilde{\mathbf{C}}, \mathbf{A}) \tag{10}$$

where $p(\tilde{\mathbf{C}})$ is the prior of $\tilde{\mathbf{C}}$. First we find the likelihood

$$\begin{aligned} p(\mathbf{y}_{1:T}|\tilde{\mathbf{C}}, \mathbf{A}) &\propto \prod_{t=2}^{T} p(\mathbf{y}_t|\mathbf{y}_{t-1}, \tilde{\mathbf{C}}, \mathbf{A}) \\ &= \prod_{t=2}^{T} \mathcal{N}(\mathbf{y}_t|\mathbf{C}\mathbf{y}_{t-1}, \sigma^2 \mathbb{I}_{d_y}) \\ &\propto \prod_{t=2}^{T} \exp\left(-\frac{1}{2\sigma^2} tr\left[\mathbf{y}_{t-1}^\top \mathbf{C}^\top \mathbf{C}\mathbf{y}_{t-1} - 2\mathbf{y}_{t-1}^\top \mathbf{C}^\top \mathbf{y}_t\right]\right) \\ &\propto \mathcal{MN}(\mathbf{C}|\mathbf{M}, \mathbf{U}, \mathbf{V}) \end{aligned} \tag{11}$$

with $\mathbf{M} = \frac{1}{\sigma^2}\mathbf{U}\sum_{t=2}^{T}\mathbf{y}_t\mathbf{y}_{t-1}^\top$, $\mathbf{U} = \sigma^2(\sum_{t=2}^{T}\mathbf{y}_{t-1}\mathbf{y}_{t-1}^\top)^{-1}$, $\mathbf{V} = \mathbb{I}_{d_y}$.

For convenience, we can use the vectorized version of the matrix normal distribution. Let $\mathbf{c} = vec(\mathbf{C}) = vec(\mathbf{A} \circ \tilde{\mathbf{C}})$, $\tilde{\mathbf{c}} = vec(\tilde{\mathbf{C}})$, $\mathbf{a} = vec(\mathbf{A})$, and the prior $p(\tilde{\mathbf{c}}) = \mathcal{N}(\tilde{\mathbf{c}}|\mathbf{0}, \sigma_c^2 \mathbb{I}_{{d_y}^2})$. Then, it follows that $\mathcal{N}(\mathbf{c}|\boldsymbol{\mu}, \boldsymbol{\Sigma})$ is the multivariate normal equivalent of $\mathcal{MN}(\mathbf{C}|\mathbf{M}, \mathbf{U}, \mathbf{V})$, with $\boldsymbol{\mu} = vec(\mathbf{M})$, and $\boldsymbol{\Sigma} = \mathbf{U} \otimes \mathbf{V}$. In this form, we can write the likelihood as

$$\begin{aligned} p(\mathbf{y}_{1:T}|\tilde{\mathbf{c}}, \mathbf{a}) &\propto \exp\left(-\frac{1}{2}(\mathbf{c}^\top \boldsymbol{\Sigma}^{-1}\mathbf{c} - 2\mathbf{c}^\top \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu})\right) \\ &= \exp\left(-\frac{1}{2}(\tilde{\mathbf{c}}^\top \mathbf{D}_\mathbf{a} \boldsymbol{\Sigma}^{-1} \mathbf{D}_\mathbf{a} \tilde{\mathbf{c}} - 2\tilde{\mathbf{c}}^\top \mathbf{D}_\mathbf{a} \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu})\right), \end{aligned} \tag{12}$$

where $\mathbf{D}_\mathbf{a} \in \mathbb{R}^{{d_y}^2 \times {d_y}^2}$ is a diagonal matrix whose diagonal entries are the elements in $\mathbf{a}$, and we use the fact that $\mathbf{c}^\top = vec(\mathbf{A} \circ \tilde{\mathbf{C}})^\top = \tilde{\mathbf{c}}^\top \mathbf{D}_\mathbf{a}$.

Finally, we combine with the prior to obtain

$$\begin{aligned} p(\tilde{\mathbf{c}})p(\mathbf{y}_{1:T}|\tilde{\mathbf{c}}, \mathbf{a}) &\propto \mathcal{N}(\tilde{\mathbf{c}}|\mathbf{0}, \sigma_c^2 \mathbb{I}_{{d_y}^2}) \mathcal{N}(\tilde{\mathbf{c}}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) \\ &\propto \exp\left(-\frac{1}{2\sigma_c^2}\tilde{\mathbf{c}}^\top \tilde{\mathbf{c}}\right) \exp\left(-\frac{1}{2}(\tilde{\mathbf{c}}^\top \mathbf{D}_\mathbf{a} \boldsymbol{\Sigma}^{-1} \mathbf{D}_\mathbf{a} \tilde{\mathbf{c}} - 2\tilde{\mathbf{c}}^\top \mathbf{D}_\mathbf{a} \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu})\right) \\ &\propto \exp\left(-\frac{1}{2}(\tilde{\mathbf{c}}^\top (\mathbf{D}_\mathbf{a} \boldsymbol{\Sigma}^{-1} \mathbf{D}_\mathbf{a} + \frac{1}{\sigma_c^2})\tilde{\mathbf{c}} - 2\tilde{\mathbf{c}}^\top \mathbf{D}_\mathbf{a} \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu})\right) \\ &= \exp\left(-\frac{1}{2}(\tilde{\mathbf{c}}^\top \tilde{\boldsymbol{\Sigma}}_{vec}^{-1} \tilde{\mathbf{c}} - 2\mathbf{c}^\top \tilde{\boldsymbol{\Sigma}}_{vec}^{-1} \tilde{\boldsymbol{\mu}}_{vec})\right) \\ &\propto \mathcal{N}(\tilde{\mathbf{c}}|\tilde{\boldsymbol{\mu}}_{vec}, \tilde{\boldsymbol{\Sigma}}_{vec}) \end{aligned} \tag{13}$$

with $\tilde{\boldsymbol{\mu}}_{vec} = \tilde{\boldsymbol{\Sigma}}_{vec}\mathbf{D_a}\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}$, $\tilde{\boldsymbol{\Sigma}}_{vec} = (\mathbf{D_a}\boldsymbol{\Sigma}^{-1}\mathbf{D_a} + \frac{1}{\sigma_c^2}\mathbb{I})^{-1}$. Note that, the dependence on the values of all the topology elements A_{jk} is accounted for in the parameters. The covariance components in $\tilde{\boldsymbol{\Sigma}}_{vec}$ are equal to the corresponding σ_{jk}^2 found in (6), however, the expressions of the means slightly differ. The mean $\tilde{\mu}_{jk}$ found in (6) uses samples of the jth row of $\mathbf{C}$, and the mean $\tilde{\boldsymbol{\mu}}_{vec}$ depends solely on the data.

Sampling A_{jk}

Since the topology cannot be sampled by matrix, we sample it element by element using the distribution given by equations (7) - (9).

4 .4 Matrix Gibbs Sampler reversed

Similarly as before, the scanning order for the matrix GS reversed (MGSr) is obtained by simply switching the order of sampling of the elements A_{jk} and the matrix $\tilde{\mathbf{C}}$, using the distributions given in Section 4 .3. Side by side comparison of the MGS and MGSr is presented in Algorithms 4 and 5.

Algorithm 4 Matrix Gibbs Sampler

Initialization: Set $A_{jk}^{(0)} \sim Bernoulli(\rho)$, for $j = 1, \ldots, d_y$, for $k = 1, \ldots, d_y$
for $i = 1, \ldots, I$ **do**
  Sample from (13)

$$\tilde{\mathbf{c}}^{(i)} \sim \mathcal{N}(\tilde{\mathbf{c}}|\tilde{\boldsymbol{\mu}}_{vec}, \tilde{\boldsymbol{\Sigma}}_{vec})$$

  Reshape $\tilde{\mathbf{c}}$ as

$$\tilde{\mathbf{C}}^{(i)} = matrix(\tilde{\mathbf{c}}^{(i)})$$

  for $j = 1, \ldots, dx$ **do**
    for $k = 1, \ldots, dx$ **do**
      Sample $A_{jk}^{(i)}$ from (9)

$$A_{jk}^{(i)} \sim p(A_{jk}|\tilde{\mathbf{C}}^{(i)}, \mathbf{S}_{\mathbf{A}_{-jk}}, \mathbf{y}_{1:T})$$

    end for
  end for
end for
Output: $\mathcal{X}_{MGS} = \{\tilde{\mathbf{C}}^{(i)}, \mathbf{A}^{(i)}\}_{i=I_0+1}^{I}$

Algorithm 5 Matrix Gibbs Sampler reversed

Initialization: Set $\tilde{\mathbf{c}}^{(0)} \sim \mathcal{N}(\mathbf{0}_{d_y{}^2}, \sigma_c^2\mathbb{I}_{d_y{}^2})$
Reshape $\tilde{\mathbf{c}}^{(0)}$ as $\tilde{\mathbf{C}}^{(0)} = matrix(\tilde{\mathbf{c}}^{(0)})$
for $i = 1, \ldots, I$ **do**
  for $j = 1, \ldots, dx$ **do**
    for $k = 1, \ldots, dx$ **do**
      Sample $A_{jk}^{(i)}$ from (9)

$$A_{jk}^{(i)} \sim p(A_{jk}|\tilde{\mathbf{C}}^{(i-1)}, \mathbf{S}_{\mathbf{A}_{-jk}}, \mathbf{y}_{1:T})$$

    end for
  end for
  Sample from (13)

$$\tilde{\mathbf{c}}^{(i)} \sim \mathcal{N}(\tilde{\mathbf{c}}|\tilde{\boldsymbol{\mu}}_{vec}, \tilde{\boldsymbol{\Sigma}}_{vec})$$

  Reshape $\tilde{\mathbf{c}}$ as

$$\tilde{\mathbf{C}}^{(i)} = matrix(\tilde{\mathbf{c}}^{(i)})$$

end for
Output: $\mathcal{X}_{MGSr} = \{\tilde{\mathbf{C}}^{(i)}, \mathbf{A}^{(i)}\}_{i=I_0+1}^{I}$

4 .5 Element Blocked Gibbs Sampler

Here, we construct a BGS as described in Section 3 .2 which jointly samples $\tilde{C}_{jk}$ and A_{jk} in that order. We refer to this sampler as element BGS (EBGS). Jointly sampling the network coefficients and topology elements seems intuitive, and may affect the propagation of information within the sampler. The joint posterior distribution of the elements can be written as

$$p(\tilde{C}_{jk}, A_{jk}|\tilde{\mathbf{C}}_{-jk}, \mathbf{A}_{-jk}, \mathbf{y}_{1:T}) = p(A_{jk}|\tilde{C}_{jk}, \tilde{\mathbf{C}}_{-jk}, \mathbf{A}_{-jk}, \mathbf{y}_{1:T})p(\tilde{C}_{jk}|\tilde{\mathbf{C}}_{-jk}, \mathbf{A}_{-jk}, \mathbf{y}_{1:T}). \tag{14}$$

Sampling $\tilde{C}_{jk}$

Unlike before, here we are not given the value of A_{jk}. In order to find the posterior of $\tilde{C}_{jk}$ we must sum out all the possibilities of A_{jk}, i.e.

$$\begin{aligned} p(\tilde{C}_{jk}|\tilde{\mathbf{C}}_{-jk}, \mathbf{A}_{-jk}, \mathbf{y}_{1:T}) &= \sum_{A_{jk}=0}^{1} p(A_{jk})p(\tilde{C}_{jk}|\tilde{\mathbf{C}}_{-jk}, A_{jk}, \mathbf{A}_{-jk}, \mathbf{y}_{1:T}) \\ &= \sum_{A_{jk}=0}^{1} \rho^{A_{jk}}(1-\rho)^{1-A_{jk}}\mathcal{N}(\tilde{\mu}_{jk}(A_{jk}), \tilde{\sigma}^2_{jk}(A_{jk})) \\ &= \rho\mathcal{N}(\tilde{\mu}_1, \tilde{\sigma}^2_1) + (1-\rho)\mathcal{N}(0, \sigma^2_c) \end{aligned} \tag{15}$$

which results in a mixture of two Gaussian distributions with mixing proportions ρ and $1-\rho$. Note that $\tilde{\mu}_{jk}(A_{jk}=1) = \tilde{\mu}_1$, $\tilde{\sigma}^2_{jk}(A_{jk}=1) = \tilde{\sigma}^2_1$, $\tilde{\mu}_{jk}(A_{jk}=0) = 0$, and $\tilde{\sigma}^2_{jk}(A_{jk}=0) = \sigma^2_c$

Sampling A_{jk}

The distribution required to sample A_{jk} in the EBGS is equal to that required in the EGSs and is given by equations (7) - (9).

4 .6 Element Blocked Gibbs Sampler reversed

Reversing the sampling order in BGSs is not as trivial as in vanilla GSs. Here, we construct a BGS which reverses the order of sampling of the elements $\tilde{C}_{jk}$ and A_{jk}, and we refer to it as EBGS reversed (EBGSr).

In this case, we write the joint posterior as

$$p(\tilde{C}_{jk}, A_{jk}|\tilde{\mathbf{C}}_{-jk}, \mathbf{A}_{-jk}, \mathbf{y}_{1:T}) = p(\tilde{C}_{jk}|\tilde{\mathbf{C}}_{-jk}, A_{jk}, \mathbf{A}_{-jk}, \mathbf{y}_{1:T})p(A_{jk}|\tilde{\mathbf{C}}_{-jk}, \mathbf{A}_{-jk}, \mathbf{y}_{1:T}). \tag{16}$$

and $\tilde{C}_{jk}$ is needed to sample A_{jk}.

Sampling A_{jk}

To find $p(A_{jk}|\tilde{\mathbf{C}}_{-jk}, \mathbf{A}_{-jk}, \mathbf{y}_{1:T})$, we can integrate out all the possible values of $\tilde{C}_{jk}$ by writing

$$\begin{aligned}
p(A_{jk}|\tilde{\mathbf{C}}_{-jk}, \mathbf{A}_{-jk}, \mathbf{y}_{1:T}) &\propto p(A_{jk}|\tilde{\mathbf{C}}_{-jk}, \mathbf{A}_{-jk})p(\mathbf{y}_{1:T}|\tilde{\mathbf{C}}_{-jk}, A_{jk}, \mathbf{A}_{-jk}) \\
&\propto p(A_{jk}) \int p(\tilde{C}_{jk}|\tilde{\mathbf{C}}_{-jk}, A_{jk}, \mathbf{A}_{-jk})p(\mathbf{y}_{1:T}|\tilde{C}_{jk}, \tilde{\mathbf{C}}_{-jk}, A_{jk}, \mathbf{A}_{-jk})d\tilde{C}_{jk} \\
&= p(A_{jk}) \int p(\tilde{C}_{jk})p(\mathbf{y}_1) \prod_{t=2}^{T} p(\mathbf{y}_t|\mathbf{y}_{t-1}, \tilde{\mathbf{C}}, \mathbf{A})d\tilde{C}_{jk} \\
&\propto Bernoulli(\rho) \int \mathcal{N}(\tilde{C}_{jk}|0, \sigma_c^2) \prod_{t=2}^{T} \mathcal{N}(\mathbf{y}_t|\mathbf{C}\mathbf{y}_{t-1}, \sigma^2 \mathbb{I}_{d_y})d\tilde{C}_{jk} \\
&\propto Bernoulli(\rho) \int \exp\left(-\frac{1}{2\tilde{\sigma}_{jk}^2}(-\tilde{\mu}_{jk}^2)\right) \mathcal{N}(\tilde{C}_{jk}|\tilde{\mu}_{jk}, \tilde{\sigma}_{jk}^2)d\tilde{C}_{jk} \\
&= \rho^{A_{jk}}(1-\rho)^{1-A_{jk}} \exp\left(\frac{\tilde{\mu}_{jk}^2}{2\tilde{\sigma}_{jk}^2}\right)(2\pi\tilde{\sigma}_{jk}^2)^{\frac{1}{2}}.
\end{aligned} \tag{17}$$

Similar as in Section (4 .1), we find the unnormalized probabilities that $A_{jk} = 0$ and $A_{jk} = 1$ as

$$\begin{aligned}
\tilde{\alpha}_{jk}^{+} &= \rho(2\pi\tilde{\sigma}_1^2)^{\frac{1}{2}} \exp\left(\frac{\tilde{\mu}_1^2}{2\tilde{\sigma}_1^2}\right) \\
\tilde{\alpha}_{jk}^{-} &= (1-\rho)(2\pi\tilde{\sigma}_c^2)^{\frac{1}{2}}
\end{aligned} \tag{18}$$

where $\tilde{\mu}_1 = \tilde{\mu}_{jk}(A_{jk} = 1)$, and $\tilde{\sigma}_1^2 = \tilde{\sigma}_{jk}^2(A_{jk} = 1)$. Finally, we obtain the probability that $A_{jk} = 1$ as

$$\tilde{\alpha}_{jk} = \frac{\tilde{\alpha}_{jk}^{+}}{\tilde{\alpha}_{jk}^{+} + \tilde{\alpha}_{jk}^{-}}. \tag{19}$$

Algorithm 6 Element Blocked Gibbs Sampler

Initialization:
Set ${A_{jk}}^{(0)} \sim Bernoulli(\rho)$, $\tilde{C}_{jk}^{(0)} \sim \mathcal{N}(0, \sigma_c^2)$, for $j = 1, \ldots, d_y$, for $k = 1, \ldots, d_y$.
for $i = 1, \ldots, I$ **do**
  for $j = 1, \ldots, d_y$ **do**
    for $k = 1, \ldots, d_y$ **do**
      Sample $\tilde{C}_{jk}^{(i)}$ from (15) as

$$\tilde{C}_{jk}^{(i)} \sim p(\tilde{C}_{jk} | \mathbf{S}_{\tilde{\mathbf{C}}_{-jk}}, \mathbf{S}_{\mathbf{A}_{-jk}}, \mathbf{y}_{1:T})$$

      Sample $A_{jk}^{(i-1)}$ from (9)

$$A_{jk}^{(i)} \sim p(A_{jk} | \mathbf{S}_{\tilde{\mathbf{C}}}, \mathbf{S}_{\mathbf{A}_{-jk}}, \mathbf{y}_{1:T})$$

    end for
  end for
end for
Output: $\mathcal{X}_{EBGS} = \{\tilde{\mathbf{C}}^{(i)}, \mathbf{A}^{(i)}\}_{i=I_0+1}^{I}$.

Algorithm 7 Element Blocked Gibbs Sampler reversed

Initialization:
Set ${A_{jk}}^{(0)} \sim Bernoulli(\rho)$, $\tilde{C}_{jk}^{(0)} \sim \mathcal{N}(0, \sigma_c^2)$, for $j = 1, \ldots, d_y$, for $k = 1, \ldots, d_y$.
for $i = 1, \ldots, I$ **do**
  for $j = 1, \ldots, d_y$ **do**
    for $k = 1, \ldots, d_y$ **do**
      Sample $A_{jk}^{(i-1)}$ from (19) as

$$A_{jk}^{(i)} \sim p(A_{jk} | \mathbf{S}_{\tilde{\mathbf{C}}_{-jk}}, \mathbf{S}_{\mathbf{A}_{-jk}}, \mathbf{y}_{1:T})$$

      Sample $\tilde{C}_{jk}^{(i)}$ from (6)

$$\tilde{C}_{jk}^{(i)} \sim \mathcal{N}(\tilde{C}_{jk} | \tilde{\mu}_{jk}, \tilde{\sigma}_{jk}^2)$$

    end for
  end for
end for
Output: $\mathcal{X}_{EBGSr} = \{\tilde{\mathbf{C}}^{(i)}, \mathbf{A}^{(i)}\}_{i=I_0+1}^{I}$.

Sampling $\tilde{C}_{jk}$

The required distribution for sampling $\tilde{C}_{jk}$ in the EBGSr is the same as that of the EGS (and EGSr) and is given by equation (6). The pseudocodes for the EBGS and EBGSr are shown side to side in Algorithms 6 and 7.

Note that the derivations, unfortunately, do not permit constructing matrix BGSs as they require sampling $\mathbf{A}$ as a matrix, which is not possible. In this case, we cannot circumvent this issue with sampling $\mathbf{A}$ element by element, as we would need to sample $\tilde{\mathbf{C}}$ and $\mathbf{A}$ jointly.

4 .7 Maximum A Posteriori Element Gibbs Sampler

We now introduce a group of samplers which combine point estimation with Gibbs sampling to obtain the network and its topology. First, we have the *Maximum A Posteriori* EGS (MAP EGS), which takes the MAP estimate of $\tilde{\mathbf{C}}$ element by element. The MAP estimate is calculated twice: (1) using a fully connected topology $\mathbf{A}$, and (2) using an estimated topology $\hat{\mathbf{A}}$. More specifically, the algorithm is initialized by sampling $\tilde{C}_{jk}$ from its prior for all elements in $\tilde{\mathbf{C}}$, and setting $A_{jk} = 1$ for all elements in $\mathbf{A}$. Then, a MAP estimate $\tilde{C}_{jkMAP}$ of the element $\tilde{C}_{jk}$ is obtained as

Algorithm 8 MAP Element Gibbs Sampler

1: **Initialization:**
Set ${A_{jk}}^{(0)} = 1$, $\tilde{C}_{jk}^{(0)} \sim \mathcal{N}(0, \sigma_c^2)$,
for $j = 1, \ldots, d_y$, for $k = 1, \ldots, d_y$.
2: Obtain MAP estimate of $\tilde{C}_{jk}$ from (20) as

$$\tilde{C}_{jkMAP} = \arg\max_{\tilde{C}_{jk}} p(\tilde{C}_{jk}|\mathbf{S}_{\tilde{\mathbf{C}}_{-jk}}, \mathbf{A}^{(0)}, \mathbf{y}_{1:T})$$

for $j = 1, \ldots, d_y$, for $k = 1, \ldots, d_y$.
3: **for** $i = 1, \ldots, I$ **do**
4: **for** $j = 1, \ldots, d_y$ **do**
5: **for** $k = 1, \ldots, d_y$ **do**
6: Sample $A_{jk}^{(i)}$ from (9)

$$A_{jk}^{(i)} \sim p(A_{jk}|\tilde{\mathbf{c}}_{MAP}, \mathbf{S}_{\mathbf{A}_{-jk}}, \mathbf{y}_{1:T})$$

7: **end for**
8: **end for**
9: **end for**
10: Obtain MAP estimate of $\tilde{C}_{jk}$ from (20) as

$$\tilde{C}_{jkMAP} = \arg\max_{\tilde{C}_{jk}} p(\tilde{C}_{jk}|\mathbf{S}_{\tilde{\mathbf{C}}_{-jk}}, \hat{\mathbf{A}}, \mathbf{y}_{1:T})$$

for $j = 1, \ldots, d_y$, for $k = 1, \ldots, d_y$.
11: **Output**: $\tilde{\mathbf{c}}_{MAP}$, $\mathcal{X}_{MAPEGS} = \{\mathbf{A}^{(i)}\}_{i=I_0+1}^{I}$.

Algorithm 9 ML Element Gibbs Sampler

1: **Initialization:**
Set ${A_{jk}}^{(0)} = 1$, $\tilde{C}_{jk}^{(0)} \sim \mathcal{N}(0, \sigma_c^2)$,
for $j = 1, \ldots, d_y$, for $k = 1, \ldots, d_y$.
2: Obtain ML estimate of $\tilde{C}_{jk}$ from (21) as

$$\tilde{C}_{jkML} = \arg\max_{\tilde{C}_{jk}} p(\mathbf{y}_{1:T}|\tilde{\mathbf{C}}, \mathbf{A}^{(0)})$$

for $j = 1, \ldots, d_y$, for $k = 1, \ldots, d_y$.
3: **for** $i = 1, \ldots, I$ **do**
4: **for** $j = 1, \ldots, d_y$ **do**
5: **for** $k = 1, \ldots, d_y$ **do**
6: Sample $A_{jk}^{(i)}$ from (9)

$$A_{jk}^{(i)} \sim p(A_{jk}|\tilde{\mathbf{c}}_{ML}, \mathbf{S}_{\mathbf{A}_{-jk}}, \mathbf{y}_{1:T})$$

7: **end for**
8: **end for**
9: **end for**
10: Obtain ML estimate of $\tilde{C}_{jk}$ from (21) as

$$\tilde{C}_{jkML} = \arg\max_{\tilde{C}_{jk}} p(\mathbf{y}_{1:T}|\tilde{\mathbf{C}}, \hat{\mathbf{A}})$$

for $j = 1, \ldots, d_y$, for $k = 1, \ldots, d_y$.
11: **Output**: $\tilde{\mathbf{c}}_{ML}$, $\mathcal{X}_{MLEGS} = \{\mathbf{A}^{(i)}\}_{i=I_0+1}^{I}$

$$\tilde{C}_{jkMAP} = \arg\max_{\tilde{C}_{jk}} p(\tilde{C}_{jk}|\mathbf{S}_{\tilde{\mathbf{C}}_{-jk}}, \mathbf{A}, \mathbf{y}_{1:T}), \tag{20}$$

where $p(\tilde{C}_{jk}|\mathbf{S}_{\tilde{\mathbf{C}}_{-jk}}, \mathbf{A}, \mathbf{y}_{1:T})$ is the posterior distribution of $\tilde{C}_{jk}$. Let $\tilde{\mathbf{c}}_{MAP}$ denote a vector containing the estimated elements $\tilde{C}_{jkMAP}$ for all j and k. Then, Gibbs sampling is used to acquire $\hat{\mathbf{A}}$ according to (7)-(9), using the $\tilde{\mathbf{c}}_{MAP}$ obtained in the previous step. Once the 0s in the topology are fixed ($\hat{\mathbf{A}}$ is obtained), the corresponding elements in $\tilde{\mathbf{C}}$ are fixed to 0, and the rest are re-estimated as in (20). The pseudocode of the MAP EGS is summarized in Algorithm 8.

4 .8 Maximum Likelihood Element Gibbs Sampler

The *Maximum Likelihood* EGS (ML EGS) is equivalent in structure to the MAP EGS, where instead of MAP, maximum likelihood (ML) is used for point estimation of all the elements $\tilde{C}_{jk}$. In particular, the ML

estimate $\tilde{C}_{jkML}$ of every element $\tilde{C}_{jk}$ is obtained as

$$\tilde{C}_{jkML} = \arg\max_{\tilde{C}_{jk}} p(\mathbf{y}_{1:T}|\tilde{\mathbf{C}}, \mathbf{A}), \tag{21}$$

where $p(\mathbf{y}_{1:T}|\tilde{\mathbf{C}}, \mathbf{A})$ is the likelihood distribution. Similarly as in Section 4 .7, let $\tilde{\mathbf{c}}_{ML}$ denote a vector containing the estimated elements $\tilde{C}_{jkML}$ for all j and k, which is a quantity used in the sampling of $\mathbf{A}$. The comparison of these two samplers (MAP EGS and ML EGS) highlights the differences in performance between MAP and ML estimation, especially in the results from the GS. The pseudocode for the ML EGS is given in Algorithm 9, together with that of the MAP EGS.

4 .9 Maximum A Posteriori Matrix Gibbs Sampler

Following Sections 4 .7 and 4 .8, we propose the Maximum A Posteriori MGS (MAP MGS), which samples the topology $\mathbf{A}$ according to (7) - (9), using the matrix MAP estimate of $\tilde{\mathbf{C}}$. In this case, the initial MAP estimate $\tilde{\mathbf{C}}_{MAP}$ is obtained by matrix as

$$\tilde{\mathbf{C}}_{MAP} = \arg\max_{\tilde{\mathbf{C}}} p(\tilde{\mathbf{C}}|\mathbf{y}_{1:T}, \mathbf{A}), \tag{22}$$

assuming a fully connected topology. This ensures that the estimate is obtained solely based on the data and the prior $p(\tilde{\mathbf{C}})$. Similarly as before, once $\hat{\mathbf{A}}$ is obtained from the GS, the 0s in $\tilde{\mathbf{C}}$ are fixed, and the rest of the elements are re-estimated together using MAP. The pseudocode of the MAP MGS is provided in Algorithm 10.

4 .10 Maximum Likelihood Matrix Gibbs Sampler

Finally, we propose the Maximum Likelihood MGS (ML MGS), equivalent in structure to the MAP MGS, where instead of MAP, the matrix ML estimate of $\tilde{\mathbf{C}}$ is used. The matrix ML estimate $\tilde{\mathbf{C}}_{ML}$ is obtained as

$$\tilde{\mathbf{C}}_{ML} = \arg\max_{\tilde{\mathbf{C}}} p(\mathbf{y}_{1:T}|\tilde{\mathbf{C}}, \mathbf{A}), \tag{23}$$

Algorithm 10 MAP Matrix Gibbs Sampler

Initialization: Set $A_{jk}^{(0)} = 1$, for $j = 1, \ldots, d_y$, for $k = 1, \ldots, d_y$

Obtain MAP estimate of $\tilde{\mathbf{C}}$ from (22) as

$$\tilde{\mathbf{C}}_{MAP} = \arg\max_{\tilde{\mathbf{C}}} p(\tilde{\mathbf{C}}|\mathbf{y}_{1:T}, \mathbf{A}^{(0)}) \tag{24}$$

for $i = 1, \ldots, I$ **do**
  for $j = 1, \ldots, dx$ **do**
    for $k = 1, \ldots, dx$ **do**
      Sample $A_{jk}^{(i)}$ from (9) as

$$A_{jk}^{(i)} \sim p(A_{jk}|\tilde{\mathbf{C}}_{MAP}, \mathbf{S}_{\mathbf{A}_{-jk}}, \mathbf{y}_{1:T})$$

    end for
  end for
end for
Return: $\mathcal{X}_{MAPMGS} = \{\mathbf{A}^{(i)}\}_{i=I_0+1}^{I}$.
Re-estimate from (22) as

$$\tilde{\mathbf{C}}_{MAP} = \arg\max_{\tilde{\mathbf{C}}} p(\tilde{\mathbf{C}}|\mathbf{y}_{1:T}, \hat{\mathbf{A}}) \tag{25}$$

Output: $\tilde{\mathbf{C}}_{MAP}$, $\hat{\mathbf{A}}$.

Algorithm 11 ML Matrix Gibbs Sampler

Initialization: Set $A_{jk}^{(0)} = 1$, for $j = 1, \ldots, d_y$, for $k = 1, \ldots, d_y$

Obtain ML estimate of $\tilde{\mathbf{C}}$ from (23) as

$$\tilde{\mathbf{C}}_{ML} = \arg\max_{\tilde{\mathbf{C}}} p(\mathbf{y}_{1:T}|\tilde{\mathbf{C}}, \mathbf{A}^{(0)}) \tag{26}$$

for $i = 1, \ldots, I$ **do**
  for $j = 1, \ldots, dx$ **do**
    for $k = 1, \ldots, dx$ **do**
      Sample $A_{jk}^{(i)}$ from (9) as

$$A_{jk}^{(i)} \sim p(A_{jk}|\tilde{\mathbf{C}}_{ML}, \mathbf{S}_{\mathbf{A}_{-jk}}, \mathbf{y}_{1:T})$$

    end for
  end for
end for
Return: $\mathcal{X}_{MLMGS} = \{\mathbf{A}^{(i)}\}_{i=I_0+1}^{I}$.
Re-estimate from (23) as

$$\tilde{\mathbf{C}}_{ML} = \arg\max_{\tilde{\mathbf{C}}} p(\mathbf{y}_{1:T}|\tilde{\mathbf{C}}, \hat{\mathbf{A}}) \tag{27}$$

Output: $\tilde{\mathbf{C}}_{ML}$, $\hat{\mathbf{A}}$.

assuming a fully connected topology. Once the estimate $\hat{\mathbf{A}}$ is obtained from the GS, the corresponding connected elements in $\tilde{\mathbf{C}}$ are re-estimated together using ML, and the rest are fixed to 0. The pseudocode of the ML MGS is summarized in Algorithm 11, in parallel with that of the MAP MGS.

The construction of point estimate GSs (Sections 4 .7 - 4 .10) was motivated by their simplicity. As they are based on a slightly different model, they are not expected to converge towards the same topology as the first group of the samplers (Sections 4 .1 - 4 .6). However, their performance will provide a clear sense of the trade-off between simplicity and accuracy.

4 .11 Least Absolute Shrinkage and Selection Operator

The LASSO [15] is a popular optimization-based method which simultaneously performs parameter estimation and elimination. We include it here as we will compare its performance to the considered GSs. The cost function is the same as that of well-known least squares plus an L_1 penalty term that helps remove redundant parameters in the model. This attribute makes LASSO a perfect candidate approach to estimating the sparse matrix $\mathbf{C}$.

Let $\mathbf{y}_{j,2:T} = [y_{j2}, y_{j3}, \ldots, y_{jT}]^\top \in \mathbb{R}^{(T-1)}$ and $\mathbf{Y}_{T-1} = [\mathbf{y}_{1,1:T-1}, \mathbf{y}_{2,1:T-1}, \ldots, \mathbf{y}_{d_y,1:T-1}] \in \mathbb{R}^{(T-1)\times d_y}$. The model in (1) can be written as

$$\mathbf{y}_{j,2:T} = \mathbf{Y}_{T-1}\mathbf{c}_j + \mathbf{u}_j, \qquad for \quad j = 1, \ldots, d_y, \tag{28}$$

where $\mathbf{c}_j \in \mathbb{R}^{d_y}$ denotes the jth row of $\mathbf{C}$, and $\mathbf{u}_j \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbb{I}_{T-1})$. The objective is to construct the cost function and optimize it over the parameters $\mathbf{c}_j$ for each row in $\mathbf{C}$. Ultimately, we obtain the estimate $\hat{\mathbf{c}}_j$ as

$$\hat{\mathbf{c}}_j = _{\mathbf{c}_j \in \mathbb{R}^{d_y}} \left\{ ||\mathbf{y}_{j,2:T} - \mathbf{Y}_{T-1}\mathbf{c}_j||_2^2 - \lambda_j ||\mathbf{c}_j||_1 \right\}, \qquad for \quad j = 1, \ldots, d_y, \tag{29}$$

with shrinkage parameter λ_j. Interestingly, LASSO can be interpreted from a Bayesian perspective by assigning a double exponential prior to the regressor coefficients [43]. In this case, we assign a Laplace prior to the elements in $\mathbf{c}_j$ parameterized as $L(C_{jk}|0, \frac{\sigma_c}{\sqrt{2}})$, which gives a shrinkage parameter $\lambda_j = 2\sqrt{2}\frac{\sigma^2}{\sigma_c}$.

5 Results & Discussion

The proposed samplers were applied and evaluated on synthetic data to obtain and analyze performance based on different factors including dimension size, data size, prior variance, and also to compare convergence rates. Then, the best performing sampler was selected and applied to a financial system. As a popular regression and feature selection method, LASSO was implemented on both the synthetic and real data and compared to all of the samplers. Details and discussions are provided in the following subsections.

5 .1 Synthetic Data

We carried out experiments varying the dimension of the state vector d_y, the number of observations T, and the prior variance σ_c^2. The noise variance σ^2 is inherent to the data, and it was assumed to be known. The number of iterations was $I = 3500$ with a burn-in period of $I_0 = 1500$. The estimate $\hat{\mathbf{A}}$ was obtained by taking the element-wise mode of the topology samples, and the estimate $\hat{\bar{\mathbf{C}}}$ by taking the element-wise mean of the network coefficients samples from each iteration. The final estimate of the coefficient matrix $\mathbf{C}$

was calculated using the Hadamard product $\hat{\mathbf{C}} = \hat{\tilde{\mathbf{C}}} \circ \hat{\mathbf{A}}$. The topology estimation was evaluated using the classical metrics: precision (PR), recall (RE), and F-score (FS) defined as

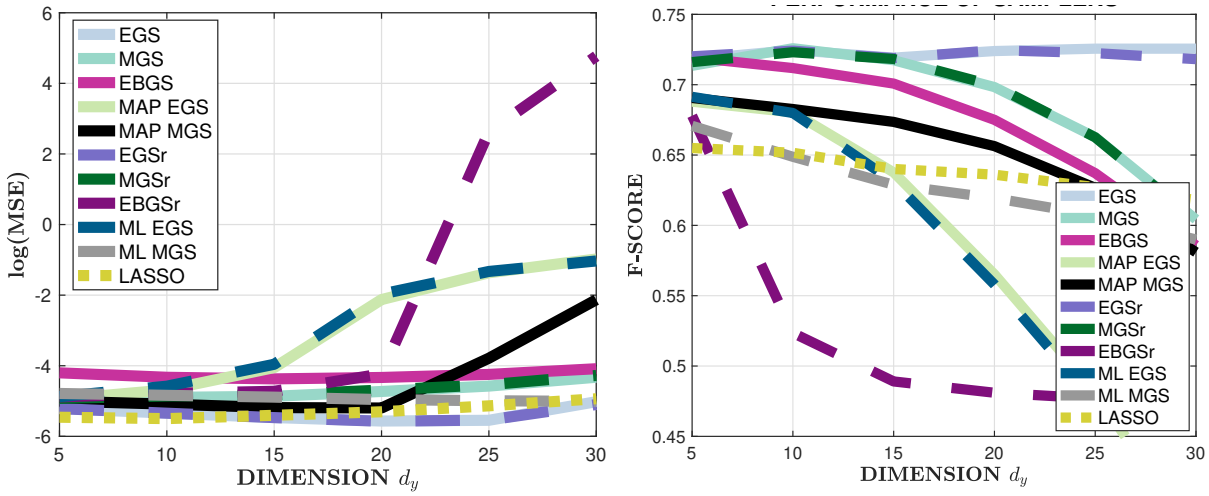
$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \quad \text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \quad \text{F-score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

where TP denotes true positives, TN true negatives, FP false positives, and FN false negatives. Further, the network estimation was evaluated using the mean squared error (MSE), given by the Frobenius norm:

$$\text{MSE}(\hat{\mathbf{C}}) = \frac{1}{d_y^2} \sum_{j=1}^{d_y} \sum_{k=1}^{d_y} (C_{jk} - \hat{C}_{jk})^2.$$

The computed FSs and MSEs were averaged over $R = 100$ independent runs. At each run, a new time series was generated to account for data variability. Note that the MSE values depend on the topology estimation of the sampler, since the estimate $\hat{\mathbf{C}}$ holds the structure of $\hat{\mathbf{A}}$. Finally, LASSO was implemented as described in Section 4 .11, with a 10 fold cross-validation for R = 100 independent runs as well.

The performance of the samplers as a function of dimension d_y with a fixed number of observations $T = 100$ and prior variance $\sigma_c^2 = 0.1$ are plotted in Fig. 3. We tested $d_y \in \{5, 10, 15, 20, 25, 30\}$, before some of the samplers collapsed from numerical issues. At first glance, it is clear that the EGSs (EGS and EGSr), as well as the MGSs (MGS and MGSr) yield indistinguishable performance, respectively, implying that reversing the order of sampling in vanilla GSs does not make a difference. This is expected, since reversing the order in this case requires use of the same exact sampling distributions. The dimension size does not seem to affect the performance of the EGSs, in contrast to the performance of the MGSs, which gradually drops past dimension $d_y = 15$. This drift in estimation may stem from the difference in dependence of the posterior means on the network coefficients. More specifically, the posterior mean $\tilde{\mu}_{jk}$ from (6) uses the samples from the rest of the elements in the jth row of $\tilde{\mathbf{C}}$, whereas the posterior $\tilde{\boldsymbol{\mu}}_{vec}$ from (13) only relies on information from the time-series data, the topology, and the prior of $\tilde{\mathbf{C}}$, but not from any of the samples of $\tilde{\mathbf{C}}$. This discrepancy becomes significant for higher dimensions. Moreover, the EGSs are less likely to cause any numerical instability when implemented, as they are purely based on scalar computations. The MGSs require matrix operations such as inversion (using Cholesky decomposition [44]) and become unstable



(a) MSE of coefficient matrix $\hat{\mathbf{C}}$.

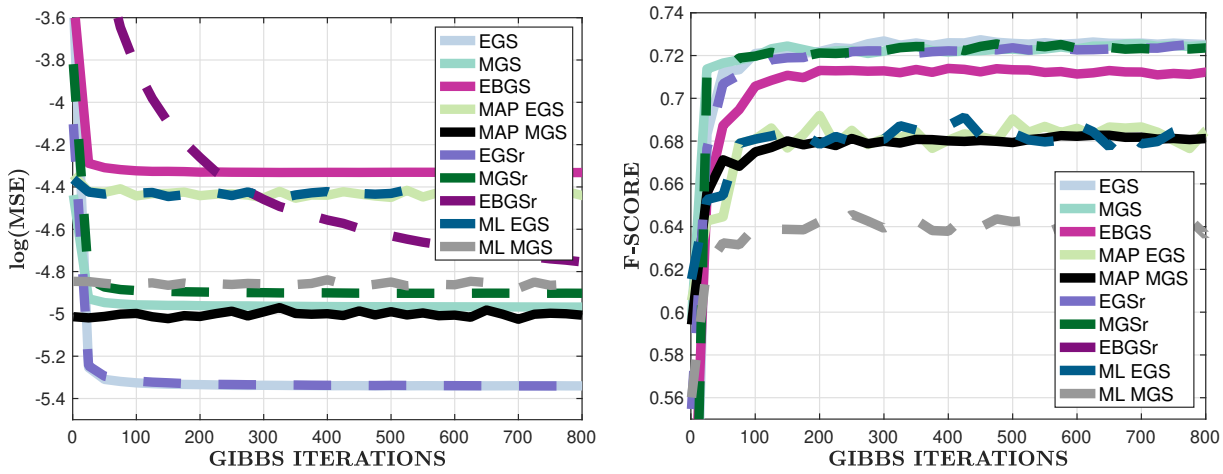
(b) F-score of topology $\hat{\mathbf{A}}$.

Figure 3: Performance of samplers with increasing dimensions. Conditions: $T = 100$, $\sigma_c^2 = 0.1$.

for higher dimensions. The implementation resorts to approximations when numbers are small, resulting in seemingly singular matrices i.e. numerical underflow [45].

Theoretically, the conditional sampling in both of the element blocked samplers (EBGS and EBGSr) should ultimately yield samples drawn from the same joint posterior distribution. However, the two samplers exhibit drastically different behaviors. The performance of the EBGSr is considerably worse than the rest of the samplers. It is worth pointing out that its FS does not seem to be affected by its corresponding MSE, although both estimates are poor. In the EBGSr, the topology element is sampled first. A closer look at the probability of $A_{jk} = 0$, (i.e., $\tilde{\alpha}_{jk}^{-}$ from (18)), reveals that its expression is formulated mainly with prior knowledge of $\tilde{C}_{jk}$ and A_{jk}, and any data dependence is canceled out when evaluating it with $A_{jk} = 0$. In contrast, the expression of α_{jk}^{-} from (8), comprises both data and samples from the jth row of $\tilde{\mathbf{C}}$ and $\mathbf{A}$. This distinction facilitates the isolated behavior for the FS as well as an ineffective use of information within the EBGSr. Overall, while jointly sampling the topology and network elements is intuitively sound, implementing it by conditional sampling adversely affects its performance and may not be the best choice for the network topology estimation task.

Both of the point estimate element samplers (MAP EGS and ML EGS) significantly drop in performance with increasing dimensions. Like the EGSs, the point estimate EGSs use the rest of the elements in the jth row of $\tilde{\mathbf{C}}$ for estimation. However, the key difference is that the EGSs perform sampling of each element,



(a) MSE of coefficient matrix $\hat{\mathbf{C}}$.

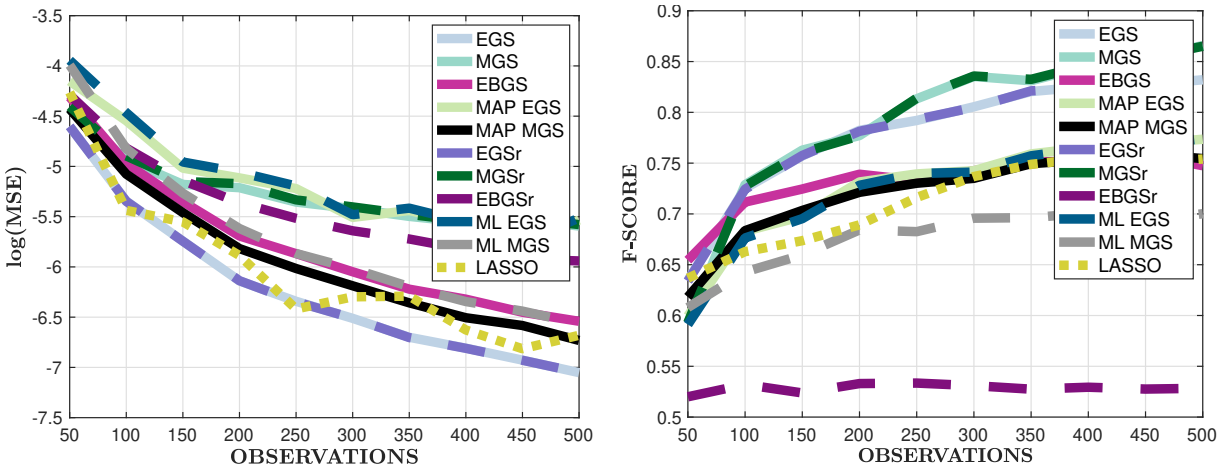
(b) F-score of topology $\hat{\mathbf{A}}$.

Figure 4: Rate of convergence of samplers. Conditions: $d_y = 10$, $\sigma_c^2 = 0.1$, $T = 100$.

allowing for continuous improvement over GS iterations, whereas the point estimate EGSs only estimate the elements in $\tilde{\mathbf{C}}$ once before and after the GS for $\mathbf{A}$. Moreover, while the point estimate matrix samplers (MAP MGS and ML MGS) are based on the same idea as MAP EGS and ML EGS, the superior performance of the point estimate MGSs is that the estimation used relies solely on the data, which in this case bring more information than elements in $\tilde{\mathbf{C}}$.

LASSO's performance with increasing dimensions of the state is more stable than the rest of the samplers with the exception of the EGSs. It competes with the point estimate matrix samplers and may even be preferred over EBGS and MGSs at higher dimensions, especially over MGSs when it comes to numerical stability. However, its curve is significantly lower than that of EGSs at the conditions specified in Fig. 3.

Figure 4 lays out a close up of the MSE and FS convergence of each sampler for a system of dimension $d_y = 10$, with prior variance $\sigma_c^2 = 0.1$, and data size $T = 100$. As before, the convergence rate for the two EGSs as well as the two MGSs is comparable, respectively. The MGS set converges slightly faster than the EGS set. So it is better to use an MGS over an EGS in the case where $d_y < 15$. It is worth pointing out that the MGSs have worse MSE than the EGSs even for $d_y < 15$. Potentially, this could be a consequence of numerical instability in the implementation. Overall, the EBGSr stands out with the slowest MSE convergence among the rest of the samplers. It is apparent that even though its MSE convergence slowly improves with iterations, its FS convergence fixates quickly, suggesting once again that the topology



(a) MSE of coefficient matrix $\hat{\mathbf{C}}$.

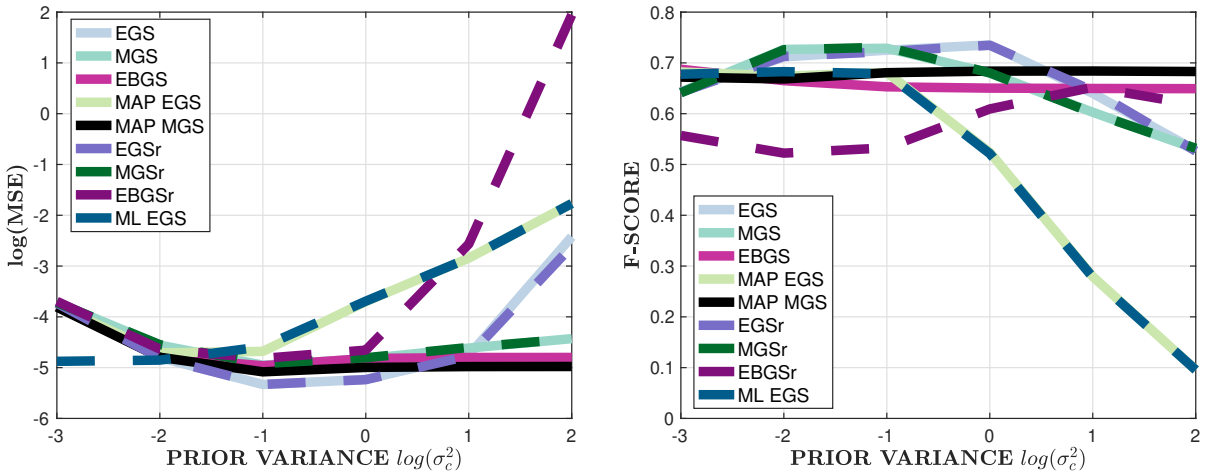
(b) F-score of topology $\hat{\mathbf{A}}$.

Figure 5: Performance of samplers with number of observations. Conditions: $d_y = 10$, $\sigma_c^2 = 0.1$.

samples were more or less unaffected by the network samples. Additionally, note that randomization of the order of sampling at each iteration was tested for the EGSs and EBGSs for increasing dimensions of state, as well as convergence rate. The results obtained were essentially the same as those from ordered sampling presented in Fig. 3 and Fig. 4, and for that reason were not included in the paper.

In Fig 5, we look at the performance as a function of the data size. In particular, we ran the methods for $T \in \{50, 100, 150, 200, 250, 300, 350, 400, 450, 500\}$ observations, fixing $d_y = 10$ and $\sigma_c^2 = 0.1$. The relative expected accuracy of each of the GSs can be beneficial in a real world problem where the number of observations are limited. Generally, the FS estimation of the EGS and MGS sets is comparable even with data size. However, a bigger discrepancy is apparent in the MSE estimation, with superiority of the EGS set. The FS of the EBGSr seems unaffected by data size. This is, once again, a consequence of the data cancellation when computing $\tilde{\alpha}_{jk}^{-}$. LASSO's performance also improves with data size and its FS is comparable to that of the MAP MGS and EBGS but considerably outperformed by the EGSs and MGSs. Interestingly though, its MSE performance is one of the best and fluctuates close with that of the EGSs. This can be explained by LASSO's property to shrink too many parameters to 0. Many of the elements in $\mathbf{C}$ may be close to 0 (but not 0) and be estimated as 0's. In the topology $\mathbf{A}$, however, this estimation would result in a missed edge.

In Fig. 6 we compare the samplers' performances with informative and non-informative priors, i.e.,



(a) MSE of coefficient matrix $\hat{\mathbf{C}}$.

(b) F-score of topology $\hat{\mathbf{A}}$.

Figure 6: Performance of samplers with varying prior variance. Conditions: $T = 100$, $d_y = 10$.

$\sigma_c^2 \in \{0.001, 0.01, 0.1, 1, 10, 100\}$, fixing to $T = 100$ number of observations, and testing only for dimension $d_y = 10$. The performances of the EGSs and MGSs considerably degrade with less informative priors. The EGS seems to be slightly more robust than the MGS, but eventually the two curves meet for higher variances. The MAP MGS as well as the EBGS seem to be less affected by the choice of prior, indicating their inference is more heavily based on the data. The point estimate EGSs significantly degrade with less informative priors, especially since both algorithms are initialized using the prior. Note that the ML EGS is included in this plot because its initialization uses the prior (see Algorithm 9), but the estimation itself is independent of the prior. The ML MGS however, is completely independent of the prior and its performance is a fixed value (FS: 0.6487 and log(MSE): -4.8409), for the conditions specified above. Theoretically, LASSO's Bayesian interpretation allows its performance to be tested for different priors as described in Section 4 .11. However, the choices of prior in this case are too restricted and shrink all the parameters to 0 except for a prior with variance $\sigma_c^2 = 100$ (FS: 0.6436 and log(MSE): -4.2267) or higher. For these reasons, the results from the ML MGS and LASSO are not included in Fig. 6. As before, the FS of the EBGSr leaves an inconsistent pattern. The relationships obtained between performance and dimension size, data size, prior variance, and convergence rate in the Figs. 3 - 6 set forth the EGSs and MGSs as the most promising samplers to use. If the posterior distributions of the unknowns are intractable and do not permit the use of these GSs, then the EBGS, MAP MGS and/or even the ML MGS can serve as alternative approaches. However,

the implementation of the EBGSr is not recommended for topology estimation due to its inconsistency and generally weak performance. The use of the point estimate EGSs is also not recommended as they require good initialization, and do not perform well for higher dimensions.

5 .2 Computational Complexity

The computational complexity for each of the samplers is summarized in Table 1. All of the samplers except the EBGSr sample $\mathbf{A}$ in the same manner, which happens to be the dominating operation in terms of computational cost. While the EBGSr holds the lowest complexity, it substantially lacks in performance. LASSO's complexity is lower than the rest of the samplers, and its use would be preferred over the ML MGS, and the EBGSr. However, LASSO's performance is comparable to that of the MAP MGS, and one is faced with the typical trade-off between cost and performance when choosing between the two, especially for higher dimensions. Ideally, the best choice would be the EGS or EGSr as they perform consistently better than the rest, while sharing the same computational cost. When edge detection is not the main concern in a given problem, then LASSO would suffice for data fitting and prediction.

Table 1 Computational complexity of each algorithm.

EGS	$\mathcal{O}(d_y^4T + d_y^3T^2)$	MGS	$\mathcal{O}(d_y^4T + d_y^3T^2)$	EBGS	$\mathcal{O}(d_y^4T + d_y^3T^2)$	MAP EGS	$\mathcal{O}(d_y^4T + d_y^3T^2)$
EGSr	$\mathcal{O}(d_y^4T + d_y^3T^2)$	MGSr	$\mathcal{O}(d_y^4T + d_y^3T^2)$	EBGSr	$\mathcal{O}(d_y^3T)$	ML EGS	$\mathcal{O}(d_y^4T + d_y^3T^2)$
MAPMGS	$\mathcal{O}(d_y^4T + d_y^3T^2)$	ML MGS	$\mathcal{O}(d_y^4T + d_y^3T^2)$	LASSO	$\mathcal{O}(d_y^3 + d_y^2T)$		

5 .3 Application: Financial Network

A market index is a statistical measure that quantifies the stock market and provides a means of monitoring market performance [46]. The dynamics of major stock indices are rich in information related to the global economy and the overall relationships of the market economies across countries. Extracting such information benefits many financial practices such as portfolio optimization, risk management, trading, and understanding market crisis [47, 48]. In this paper, we considered the daily prices of 27 major market indices in 30 days over the last month of 2020, i.e., a time-series data of $d_y = 27$ and $T = 30$. The data was obtained from [49] and the chosen major market indices were: SMI, TWII, KS11, NZX50, AORD, SSE, HSI, NYSE, JSEC, BIST100, OMXC20, ATX, IBEX, BEL20, GDAXI, AEX, CAC40, MERV, TSX, BVSP, IXIC, GSPC, DJI, ATHEX, MXX, FTSEMIB, N225. The financial network to be inferred represents the price correlation

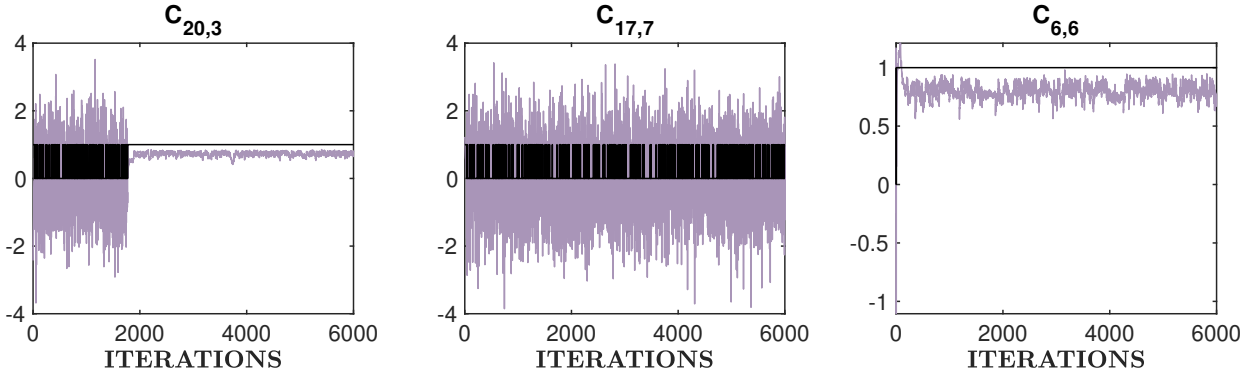


Figure 7: Markov chains of selected coefficient (purple) and topology (black) elements.

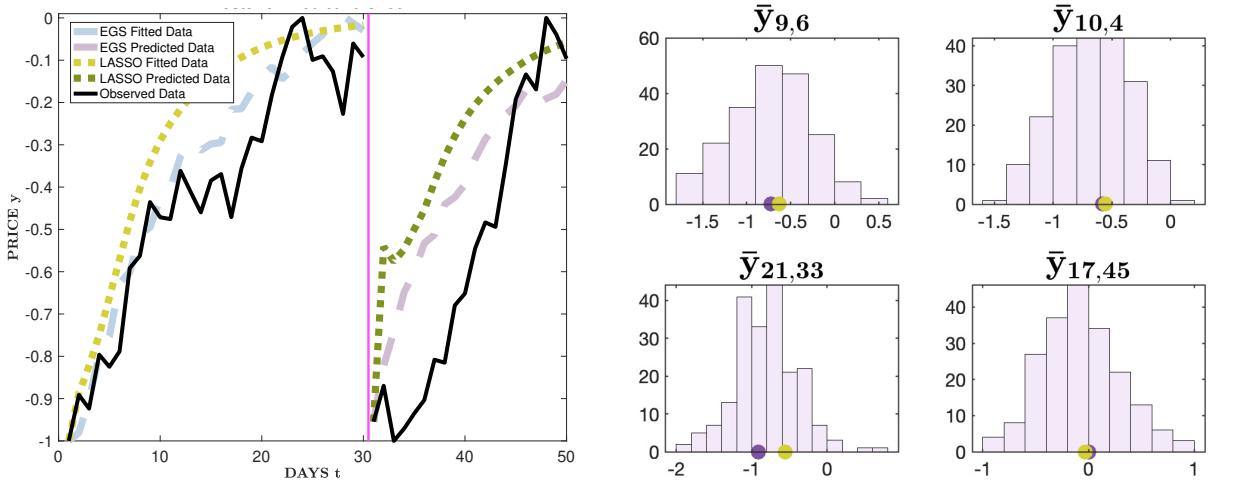
matrix, and its topology indicates the influences these global market indices send out and receive from the other global market indices in the network. In other words, the market indices serve as nodes in the network, and the edges capture their interactions. While more data is available, the GSs constructed do not account for a varying network topology, which is expected over longer periods of time. With such a small collection of data, the topology changes (if any) are expected to be minor.

The EGS, the most promising sampler, was used for inference. Since pricing data was used, standardizing the data was necessary to reduce multicollinearity. To check the convergence of the sampler, Markov chains of randomly selected coefficients were plotted in Fig. 7. The chains such as that of $C_{17,7}$ in Fig. 7 were created from continuous sampling from the priors of A_{jk} and $\tilde{C}_{jk}$, indicating that $C_{17,7}$ was most likely 0. And in fact, the final estimate $\hat{C}_{17,7}$ from the sampler was found to be 0. Further, we ran posterior predictive checks to examine the validity of the model and its predictions. This was done by obtaining the predictive posterior distribution as

$$p(\bar{\mathbf{y}}_{1:T+\tau}|\mathbf{y}_{1:T}) = \int p(\bar{\mathbf{y}}_{1:T+\tau}|\tilde{\mathbf{C}}, \mathbf{A})p(\tilde{\mathbf{C}}, \mathbf{A}|\mathbf{y}_{1:T})d\tilde{\mathbf{C}}d\mathbf{A} \tag{30}$$

where $\bar{\mathbf{y}}_{1:T+\tau}$ was new simulated data, and $\tau \geq 0$. To simulate new data $\bar{\mathbf{y}}_{1:T+\tau}$ from $p(\bar{\mathbf{y}}_{1:T+\tau}|\mathbf{y}_{1:T})$, we used an MC approximation to the integral in (30), i.e.

$$\hat{p}(\bar{\mathbf{y}}_{1:T+\tau}|\mathbf{y}_{1:T}) = \frac{1}{M}\sum_{m=1}^{M} p(\bar{\mathbf{y}}_{1:T+\tau}|\tilde{\mathbf{C}}^{(m)}, \mathbf{A}^{(m)}) \tag{31}$$



(a) Predicted mean trajectory of a randomly selected stock index. The vertical magenta line separates the model fitting and model predictions.

(b) Posterior predictive distributions of randomly selected indices y_{jt}. The yellow dot indicates the value found by LASSO. The dark purple dot indicates the true data point value.

Figure 8: Posterior predictive checks.

with $[\tilde{\mathbf{C}}^{(m)}, \mathbf{A}^{(m)}]^\top \sim p(\tilde{\mathbf{C}}, \mathbf{A}|\mathbf{y}_{1:T})$, for $m = 1, \ldots, M$. Essentially, we first drew M network and topology samples, which practically are obtained from the GS used once convergence is reached. Then, using these samples, M data points were simulated via the likelihood as $\bar{\mathbf{y}}^{(m)}_{1:T+\tau} \sim p(\bar{\mathbf{y}}_{1:T+\tau}|\tilde{\mathbf{C}}^{(m)}, \mathbf{A}^{(m)})$, for $m = 1, \ldots, M$. The data simulated for $\tau = 0$ was used to determine the goodness of fit of the model by plotting it against the observed data and discerning the discrepancies. Data simulated for $\tau \geq 1$ was used to evaluate the model's predictive ability, by plotting the new simulated data against data unseen by the model (i.e., not used for inference). In both cases, the starting point was taken to be the observed data point (i.e., $\bar{\mathbf{y}}_1 = \mathbf{y}_1$ and $\bar{\mathbf{y}}_{T+1} = \mathbf{y}_{T+1}$). The model fitting ($\tau = 0$) and prediction ($\tau = 20$) for $M = 200$ are shown using the trajectory of a randomly selected market index plotted in Fig. 8 a). For such data, it is generally not recommended to use a value of τ that large as the topology may have changed, however, the predictions in this case proved satisfactory. Histograms describing the predictive posterior distributions of a few randomly selected stock indices at given time-points are also presented in Fig. 8 b). The uncertainty in the predictive posteriors can be reduced by approximating with a larger number of samples M.

LASSO with 10 - fold cross-validation was also implemented for fitting and prediction of the financial system, and plotted against the real data, as well as against the EGS fitting and prediction. Figure 8 shows that both LASSO and EGS give reasonable fitting and predictions, with LASSO overshooting by a small

amount. This agrees with the results in Fig. 3 a) and Fig. 5 a). However, a higher discrepancy in performance can be seen when comparing estimated links with LASSO finding 67% of the connections to be absent (i.e. 0's in $\mathbf{C}$), and the EGS only 33%. This example demonstrates the tendency of LASSO to overselect coefficients (shrink a large number of parameters to 0). The difference in estimated edges becomes even more prominent in cases where the presence and absence of edges requires accurate and physical interpretation.

6 Conclusion

In this paper, we introduce novel strategies inspired by Gibbs sampling for estimation of network coefficients and topology in a first-order vector autoregressive model. The samplers differed in their sampling strategies (marginal vs blocked (and conditional)) and their scanning order (by element vs by matrix and their reversed orders). The contribution of this effort is in the construction and implementation of the new samplers, as well as in elucidating advantages and concerns related to their performances. The differences in their performances were highlighted using synthetic data through plots illustrating the effects of dimension size, data size, choice of prior variance, and convergence rate. The results manifest clear relationships between the above mentioned parameters and performances using different sampling strategies and scanning order for network coefficients and topology estimation. The best performing sampler was applied to data of stock market indices to infer a financial network (price correlation matrix) and its topology, capturing the direction and amount of influence between stock markets. Posterior predictive checks demonstrated the validity of the model fitting and predictions. The popular LASSO was implemented on both the synthetic and real data as a reference base for assessing the samplers' performances. The discussions provided in this paper can aid experts and practitioners in decision making for research and applications in network topology estimations using Gibbs samplers.

Appendix

This appendix gives the detailed derivations of the conditional posterior distributions for each of the unknowns used in the samplers i.e., $\tilde{\mathbf{C}}, \tilde{C}_{jk}, \mathbf{A}, A_{jk}$ under different conditions. Additionally, the posterior distributions of $\mathbf{C}$ and C_{jk} are also included.

A POSTERIORS OF C

A .1 Matrix Normal Posterior of C

General Using Bayes theorem, the posterior distribution of $\mathbf{C}$ can be expressed as

$$\begin{aligned} p(\mathbf{C}|\mathbf{y}_{1:T}) &\propto p(\mathbf{y}_{1:T}|\mathbf{C})p(\mathbf{C}) \\ &= p(\mathbf{y}_t|\mathbf{C}, \mathbf{y}_{T-1})p(\mathbf{y}_{1:T-1}|\mathbf{C})p(\mathbf{C}) \\ &\propto p(\mathbf{C}) \prod_{t=2}^{T} p(\mathbf{y}_T|\mathbf{C}, \mathbf{y}_{T-1}) \end{aligned} \tag{A1.1}$$

Prior $p(\mathbf{C})$ We assume that

$$C_{jk} \sim \mathcal{N}(0, \sigma_c^2), \quad j = 1, \ldots, d_y, \quad k = 1, \ldots, d_y. \tag{A1.2}$$

If we vectorize $\mathbf{C}$ as $\mathbf{c} = vec(\mathbf{C})$, i.e., stack each element C_{jk} in a row vector $\mathbf{c}$, we can rewrite the prior as $p(\mathbf{c}) = \mathcal{N}(\mathbf{0}_{d_y^2}, \sigma_c^2 \mathbb{I}_{d_y^2})$. Further we can take the matrix normal form of the prior as

$\mathbf{C} \sim \mathcal{MN}(\mathbf{C}|\mathbf{M}_0, \mathbf{U}_0, \mathbf{V}_0)$ with $\mathbf{M}_0 = \mathbf{0}_{d_y \times d_y}$ $\mathbf{U}_0 = \sigma_c^2 \mathbb{I}_{d_y}$ $\mathbf{V}_0 = \mathbb{I}_{d_y}$.

Product of Likelihoods The product term in (A1.1) can be expressed as

$$\begin{aligned} \prod_{t=2}^{T} p(\mathbf{y}_t|\mathbf{C}, \mathbf{y}_{t-1}) &= \prod_{t=2}^{T} \mathcal{N}(\mathbf{C}\mathbf{y}_{t-1}, \sigma^2 \mathbb{I}_{d_y}) \\ &\propto \prod_{t=2}^{T} exp\left(-\frac{1}{2\sigma^2}(\mathbf{y}_t - \mathbf{C}\mathbf{y}_{t-1})^\top (\mathbf{y}_t - \mathbf{C}\mathbf{y}_{t-1})\right) \\ &= \prod_{t=2}^{T} exp\left(-\frac{1}{2\sigma^2}(\mathbf{y}_t^\top \mathbf{y}_t - 2\mathbf{y}_t^\top \mathbf{C}\mathbf{y}_{t-1} + \mathbf{y}_{t-1}^\top \mathbf{C}^\top \mathbf{C}\mathbf{y}_{t-1})\right) \\ &\propto \prod_{t=2}^{T} exp\left(-\frac{1}{2\sigma^2} tr\left[\mathbf{y}_{t-1}^\top \mathbf{C}^\top \mathbf{C}\mathbf{y}_{t-1} - 2\mathbf{y}_t^\top \mathbf{C}\mathbf{y}_{t-1}\right]\right) \end{aligned}$$

Using the cyclic shifting property of traces, we write

$$\begin{aligned}
&\prod_{t=2}^{T} exp\left(-\frac{1}{2\sigma^2} tr\left[\mathbf{y}_{t-1}^\top \mathbf{C}^\top \mathbf{C}\mathbf{y}_{t-1} - 2\mathbf{y}_t^\top \mathbf{C}\mathbf{y}_{t-1}\right]\right) \\
&= exp\left(-\frac{1}{2\sigma^2}\sum_{t=2}^{T} tr\left[\mathbf{C}^\top \mathbf{y}_{t-1}\mathbf{y}_{t-1}^\top \mathbf{C} - 2\mathbf{C}^\top \mathbf{y}_t \mathbf{y}_{t-1}^\top\right]\right) \\
&= exp\left(-\frac{1}{2\sigma^2} tr\left[\mathbf{C}^\top \sum_{t=2}^{T}\mathbf{y}_{t-1}\mathbf{y}_{t-1}^\top \mathbf{C} - 2\mathbf{C}^\top \sum_{t=2}^{T}\mathbf{y}_t \mathbf{y}_{t-1}^\top\right]\right) \\
&= exp\left(-\frac{1}{2} tr\left[\mathbf{V}^{-1}(\mathbf{C}-\mathbf{M})^\top \mathbf{U}^{-1}(\mathbf{C}-\mathbf{M})\right]\right) \\
&\propto \mathcal{MN}(\mathbf{C}|\mathbf{M},\mathbf{U},\mathbf{V})
\end{aligned} \tag{A1.3}$$

with $\mathbf{M} = \frac{1}{\sigma^2}\mathbf{U}\sum_{t=2}^{T}\mathbf{y}_t\mathbf{y}_{t-1}^\top \qquad \mathbf{U} = \sigma^2(\sum_{t=2}^{T}\mathbf{y}_{t-1}\mathbf{y}_{t-1}^\top)^{-1} \qquad \mathbf{V} = \mathbb{I}_{d_y}$.

Matrix Normal Posterior Combining the prior with the product of likelihoods, we get

$$\begin{aligned}
p(\mathbf{C}|\mathbf{y}_{1:T}) &\propto p(\mathbf{C})p(\mathbf{y}_{1:T}|\mathbf{C}) \\
&\propto \mathcal{MN}(\mathbf{C}|\mathbf{M}_0,\mathbf{U}_0,\mathbf{V}_0)\cdot\mathcal{MN}(\mathbf{C}|\mathbf{M},\mathbf{U},\mathbf{V}) \\
&\propto exp\left(-\frac{1}{2} tr\left[\mathbf{V}^{-1}(\mathbf{C}-\mathbf{M})^\top\mathbf{U}^{-1}(\mathbf{C}-\mathbf{M}) + \mathbf{V}_0^{-1}(\mathbf{C}-\mathbf{M}_0)^\top\mathbf{U}_0^{-1}(\mathbf{C}-\mathbf{M}_0)\right]\right) \\
&\propto exp\left(-\frac{1}{2} tr\left[\mathbf{C}^\top\mathbf{U}^{-1}\mathbf{C} - 2\mathbf{C}^\top\mathbf{U}^{-1}\mathbf{M} + \mathbf{C}^\top\mathbf{U}_0^{-1}\mathbf{C}\right]\right) \\
&= exp\left(-\frac{1}{2} tr\left[\mathbf{C}^\top(\mathbf{U}^{-1}+\mathbf{U}_0^{-1})\mathbf{C} - 2\mathbf{C}^\top\mathbf{U}^{-1}\mathbf{M}\right]\right) \\
&= exp\left(-\frac{1}{2} tr\left[\mathbf{C}^\top(\mathbf{U}^{-1}+\mathbf{U}_0^{-1})\mathbf{C} - 2\mathbf{C}(\mathbf{U}^{-1}+\mathbf{U}_0^{-1})(\mathbf{U}^{-1}+\mathbf{U}_0^{-1})^{-1}\mathbf{U}^{-1}\mathbf{M}\right]\right) \\
&= exp\left(-\frac{1}{2} tr\left[\mathbf{C}^\top\mathbf{U}_{post}^{-1}\mathbf{C} - 2\mathbf{C}\mathbf{U}_{post}^{-1}\mathbf{M}_{post}\right]\right) \\
&\propto \mathcal{MN}(\mathbf{C}|\mathbf{M}_{post},\mathbf{U}_{post},\mathbf{V}_{post})
\end{aligned} \tag{A1.4}$$

with $\qquad \mathbf{M}_{post} = \mathbf{U}_{post}\mathbf{U}^{-1}\mathbf{M} \qquad \mathbf{U}_{post} = (\mathbf{U}^{-1}+\mathbf{U}_0^{-1})^{-1} \qquad \mathbf{V}_{post} = \mathbb{I}_{d_y}$.

Vector form of Posterior Let $\mathbf{c} = vec(\mathbf{C})$. We can now write

$$p(\mathbf{c}|\mathbf{y}_{1:T}) = p(vec(\mathbf{C})|\mathbf{y}_{1:T}) \propto \mathcal{N}(\boldsymbol{\mu}_{post}, \boldsymbol{\Sigma}_{post}) \tag{A1.5}$$

with $\qquad \boldsymbol{\mu}_{vec} = vec(\mathbf{M}_{post}) \qquad \boldsymbol{\Sigma}_{vec} = \mathbf{U}_{post} \otimes \mathbf{V}_{post}$.

A .2 Element Posterior of C_{jk}

$$p(C_{jk}|\mathbf{y}_{1:T}, \mathbf{C}_{-jk}) \propto p(C_{jk}) \cdot \prod_{t=2}^{T} p(\mathbf{y}_t|\mathbf{y}_{t-1}, C_{jk}, \mathbf{C}_{-jk}) \tag{A2.1}$$

Prior $p(C_{jk})$ As in Section A .1 we take the prior as

$$C_{jk} \sim \mathcal{N}(0, \sigma_c^2), \quad j = 1, \ldots, d_y, \quad k = 1, \ldots, d_y.$$

Product of Likelihoods

$$\begin{aligned}
\prod_{t=2}^{T} p(\mathbf{y}_t|\mathbf{y}_{t-1}, C_{jk}, C_{-jk}) &\propto \prod_{t=2}^{T} exp\left(\frac{1}{2\sigma^2}(\mathbf{y}_t - \mathbf{C}\mathbf{y}_{t-1})^\top(\mathbf{y}_t - \mathbf{C}\mathbf{y}_{t-1})\right) \\
&\propto \prod_{t=2}^{T} exp\left(-\frac{1}{2\sigma^2}[\mathbf{y}_{t-1}^\top\mathbf{C}^\top\mathbf{C}\mathbf{y}_{t-1} - 2\mathbf{y}_{t-1}^\top\mathbf{C}^\top\mathbf{y}_t]\right) \\
&= \prod_{t=2}^{T} exp\left(-\frac{1}{2\sigma^2}\left[\sum_{n=1}^{d_y}(\sum_{m=1}^{d_y} C_{nm}y_{m,t-1})^2 - 2\sum_{n=1}^{d_y} y_{nt}\sum_{m=1}^{d_y} C_{nm}y_{m,t-1}\right]\right) \\
&\propto exp\left(-\frac{1}{2\sigma^2}\sum_{t=2}^{T}\left[(C_{jk}y_{k,t-1})^2 + 2C_{jk}y_{k,t-1}\sum_{\substack{m=1\\m\neq k}}^{d_y} C_{jm}y_{j,t-1} - 2C_{jk}y_{jt}y_{k,t-1}\right]\right) \\
&\propto exp\left(-\frac{1}{2\sigma^2}\sum_{t=2}^{T}\left[(C_{jk}y_{k,t-1})^2 - 2C_{jk}(y_{jt}y_{k,t-1} - y_{k,t-1}\sum_{\substack{m=1\\m\neq k}}^{d_y} C_{jm}y_{j,t-1})\right]\right) \\
&= exp\left(-\frac{1}{2\sigma^2}\left[{C_{jk}}^2\sum_{t=2}^{T} y_{k,t-1}^2 - 2C_{jk}\sum_{t=2}^{T}(y_{jt}y_{k,t-1} - y_{k,t-1}\sum_{\substack{m=1\\m\neq k}}^{d_y} C_{jm}y_{j,t-1})\right]\right) \\
&= exp\left(-\frac{1}{2\sigma^2}(\sum_{t=2}^{T} y_{k,t-1}^2)\left[{C_{jk}}^2 - 2C_{jk}\frac{\sum_{t=2}^{T}(y_{jt}y_{k,t-1} - y_{k,t-1}\sum_{\substack{m=1\\m\neq k}}^{d_y} C_{jm}y_{j,t-1})}{\sum_{t=2}^{T} y_{k,t-1}^2}\right]\right) \\
&= exp\left(-\frac{1}{2\sigma_{jk}^2}\left[{C_{jk}}^2 - 2C_{jk}\mu_{jk}\right]\right) \\
&\propto \mathcal{N}(C_{jk}|\mu_{jk}, \sigma_{jk}^2)
\end{aligned} \tag{A2.3}$$

with

$$\mu_{jk} = \frac{\sigma_{jk}^2}{\sigma^2}\left(\sum_{t=2}^{T} y_{j,t}y_{k,t-1} - \sum_{t=2}^{T} y_{k,t-1}(\sum_{\substack{m=1\\m\neq k}}^{d_y} C_{jm}y_{m,t-1})\right), \qquad \sigma_{jk}^2 = \frac{\sigma^2}{\sum_{t=2}^{T} y_{t-1,k}^2}. \tag{A2.4}$$

Normal Posterior Combining the prior and the likelihood, we get

$$\begin{aligned} p(C_{jk}|\mathbf{y}_{1:T}, \mathbf{C}_{-jk}) &\propto \mathcal{N}(0, \sigma_c^2)\mathcal{N}(\mu_{jk}, \sigma_{jk}^2) \\ &\propto \mathcal{N}(C_{jk}|\mu_{post}, \sigma_{post}^2) \end{aligned} \tag{A2.5}$$

with

$$\mu_{post} = \frac{\sigma_{post}^2}{\sigma_{jk}^2}\mu_{jk}, \qquad \sigma_{post}^2 = \frac{\sigma_{jk}^2\sigma_c^2}{\sigma_{jk}^2 + \sigma_c^2}. \tag{A2.6}$$

B POSTERIORS OF $\tilde{\mathbf{C}}$

B .1 Matrix Normal Posterior of $\tilde{\mathbf{C}}$

Let $\mathbf{c} = vec(\mathbf{C}) = vec(\mathbf{A} \circ \tilde{\mathbf{C}})$, $\tilde{\mathbf{c}} = vec(\tilde{\mathbf{C}})$, $\mathbf{a} = vec(\mathbf{A})$, and $\mathbf{D_a}$ be a diagonal matrix whose diagonal entries are the elements in $\mathbf{a}$. Additioanlly, let $\boldsymbol{\mu} = vec(\mathbf{M})$, and $\boldsymbol{\Sigma} = \mathbf{U} \otimes \mathbf{V}$ given in (A1.3). We then write

$$\begin{aligned} p(\mathbf{y}_{1:T}|\mathbf{c}) &\propto exp\left(-\frac{1}{2}(\mathbf{c}^\top\boldsymbol{\Sigma}^{-1}\mathbf{c} - 2\mathbf{c}^\top\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu})\right) \\ &= exp\left(-\frac{1}{2}(\tilde{\mathbf{c}}^\top\mathbf{D_a}\boldsymbol{\Sigma}^{-1}\mathbf{D_a}\tilde{\mathbf{c}} - 2\tilde{\mathbf{c}}^\top\mathbf{D_a}\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu})\right) \end{aligned} \tag{B1.1}$$

where we use the property

$$\begin{aligned} \mathbf{c}^\top &= [c_{11}, c_{12}, \ldots, c_{1d_y}, c_{21}, c_{22}, \ldots, c_{d_y d_y}] \\ &= [a_{11}\tilde{c}_{11}, a_{12}\tilde{c}_{12}, \ldots, a_{1d_y}\tilde{c}_{1d_y}, a_{21}\tilde{c}_{21}, a_{22}\tilde{c}_{22}, \ldots, a_{d_y d_y}\tilde{c}_{d_y d_y}] \\ &= \tilde{\mathbf{c}}^\top\mathbf{D_a} \end{aligned} \tag{B1.2}$$

We take the prior of $\tilde{C}_{jk}$ to be the same as that of C_{jk}. Combining, we get

$$\begin{aligned} p(\tilde{\mathbf{c}})p(\mathbf{y}_{1:T}|\mathbf{c}) &\propto \mathcal{N}(\tilde{\mathbf{c}}|\mathbf{0}, \sigma_c^2 \mathbb{I}_{{d_y}^2})\mathcal{N}(\tilde{\mathbf{c}}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) \\ &\propto exp\left(-\frac{1}{2\sigma_c^2}\mathbf{c}^\top\mathbf{c}\right) exp\left(-\frac{1}{2}(\mathbf{c}^\top\boldsymbol{\Sigma}^{-1}\mathbf{c} - 2\mathbf{c}^\top\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu})\right) \\ &\propto exp\left(-\frac{1}{2}(\tilde{\mathbf{c}}^\top(\mathbf{D_a}\boldsymbol{\Sigma}^{-1}\mathbf{D_a} + \frac{1}{\sigma_c^2})\tilde{\mathbf{c}} - 2\tilde{\mathbf{c}}^\top\mathbf{D_a}\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu})\right) \\ &= exp\left(-\frac{1}{2}(\tilde{\mathbf{c}}^\top\boldsymbol{\Sigma}_{vec}^{-1}\tilde{\mathbf{c}} - 2\tilde{\mathbf{c}}^\top\tilde{\boldsymbol{\Sigma}}_{vec}^{-1}\tilde{\boldsymbol{\mu}}_{vec})\right) \\ &\propto \mathcal{N}(\tilde{\mathbf{c}}|\tilde{\boldsymbol{\mu}}_{vec}, \tilde{\boldsymbol{\Sigma}}_{vec}) \end{aligned} \tag{B1.3}$$

with

$$\begin{aligned} &\tilde{\boldsymbol{\mu}}_{vec} = \tilde{\boldsymbol{\Sigma}}_{vec}\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu} && \boldsymbol{\mu} = vec(\mathbf{M}) && \mathbf{M} = (\sum_{t=2}^{T}\mathbf{y}_t\mathbf{y}_{t-1}^\top)\mathbf{U} \\ &\tilde{\boldsymbol{\Sigma}}_{vec} = (\mathbf{D_a}\boldsymbol{\Sigma}^{-1}\mathbf{D_a} + \frac{1}{\sigma_c^2}\mathbb{I})^{-1} && \boldsymbol{\Sigma} = \mathbf{U}\otimes\mathbb{I}_{d_y} && \mathbf{U} = \sigma^2(\sum_{t=2}^{T}\mathbf{y}_{t-1}\mathbf{y}_{t-1}^\top)^{-1}. \end{aligned} \tag{B1.4}$$

B .2 Posterior of $\tilde{C}_{jk}$ given A_{jk}

The posterior of $\tilde{C}_{jk}$ assuming we know A_{jk} is

$$\begin{aligned} p(\tilde{C}_{jk}|\tilde{\mathbf{C}}_{-jk}, \mathbf{A}, \mathbf{y}_{1:T}) &\propto p(\tilde{C}_{jk})p(\mathbf{y}_1)\prod_{t=2}^{T}p(\mathbf{y}_t|\mathbf{y}_{t-1}, \mathbf{A}, \tilde{\mathbf{C}}) \\ &\propto \mathcal{N}(\tilde{C}_{jk}|0, \sigma_c^2)\prod_{t=2}^{T}\mathcal{N}(\mathbf{y}_t|\mathbf{C}\mathbf{y}_{t-1}, \sigma^2\mathbb{I}_{d_y}) \end{aligned} \tag{B2.1}$$

Product of likelihoods

We rewrite the product of likelihoods as

$$\begin{aligned}
&\prod_{t=2}^{T}\mathcal{N}(\mathbf{y}_t|\mathbf{C}\mathbf{y}_{t-1},\sigma^2\mathbb{I}_{d_y}) \\
&\qquad= \prod_{t=2}^{T} exp\left(-\frac{1}{2\sigma^2}(\mathbf{y}_t-\mathbf{C}\mathbf{y}_{t-1})^\top(\mathbf{y}_t-\mathbf{C}\mathbf{y}_{t-1})\right) \\
&\qquad= exp\left(-\frac{1}{2\sigma^2}\sum_{t=2}^{T}(\mathbf{y}_t-\mathbf{C}\mathbf{y}_{t-1})^\top(\mathbf{y}_t-\mathbf{C}\mathbf{y}_{t-1})\right) \\
&\qquad= exp\left(-\frac{1}{2\sigma^2}\sum_{t=2}^{T}(-2\mathbf{y}_{t-1}^\top\mathbf{C}^\top\mathbf{y}_t+\mathbf{y}_{t-1}^\top\mathbf{C}^\top\mathbf{C}\mathbf{y}_{t-1})\right) \\
&\qquad= exp\left(-\frac{1}{2\sigma^2}\sum_{t=2}^{T}\left[\sum_{n=1}^{d_y}(\sum_{m=1}^{d_y}C_{nm}y_{m,t-1})^2-2\sum_{n=1}^{d_y}(y_{nt}\sum_{m=1}^{d_y}C_{nm}y_{m,t-1})\right]\right).
\end{aligned} \tag{B2.2}$$

We now separate the terms that include C_{jk} from those that don't. The first term can be expanded to:

$$\sum_{n=1}^{d_y}(\sum_{m=1}^{d_y}C_{nm}y_{m,t-1})^2=(C_{jk}y_{k,t-1})^2+2C_{jk}y_{k,t-1}\sum_{\substack{m=1\\m\neq k}}^{d_y}C_{nm}y_{m,t-1}+L_{-jk}$$

and the second term to:

$$2\sum_{n=1}^{d_y}(y_{nt}\sum_{m=1}^{d_y}C_{nm}y_{m,t-1})=2C_{jk}y_{jt}y_{k,t-1}+L'_{-jk}$$

where L_{-jk} and L'_{-jk} are the terms that do not involve $\tilde{C}_{jk}$ and A_{jk} in the first and second term, respectively.

From here, we continue to write

$$\begin{aligned}
&\prod_{t=2}^{T}\mathcal{N}(\mathbf{y}_t|\mathbf{C}\mathbf{y}_{t-1},\sigma^2\mathbb{I}_{d_y}) \\
&\qquad= exp\left(-\frac{1}{2\sigma^2}\sum_{t=2}^{T}\left[(C_{jk}y_{k,t-1})^2+2C_{jk}y_{k,t-1}(\sum_{\substack{m=1\\m\neq k}}^{d_y}C_{jm}y_{m,t-1})-2y_{jt}C_{jk}y_{k,t-1})\right]\right) \\
&\qquad= exp\left(-\frac{1}{2\sigma^2}\sum_{t=2}^{T}\left[{C_{jk}}^2y_{k,t-1}^2-2C_{jk}(y_{jt}y_{k,t-1}-y_{k,t-1}\sum_{\substack{m=1\\m\neq k}}^{d_y}C_{jm}y_{m,t-1})\right]\right) \\
&\qquad= exp\left(-\frac{1}{2\sigma^2}\left[{C_{jk}}^2\sum_{t=2}^{T}y_{k,t-1}^2-2C_{jk}\sum_{t=2}^{T}(y_{jt}y_{k,t-1}-y_{k,t-1}\sum_{\substack{m=1\\m\neq k}}^{d_y}C_{jm}y_{m,t-1})\right]\right).
\end{aligned} \tag{B2.3}$$

Combine with prior

$$
\begin{aligned}
&p(\tilde{C}_{jk}) \prod_{t=2}^{T} p(\mathbf{y}_t|\mathbf{C}\mathbf{y}_{t-1}, \sigma^2) \\
&\quad \propto exp\left(-\frac{1}{2\sigma_c^2}\tilde{C}_{jk}^2\right) exp\left(-\frac{1}{2\sigma^2}\left[\tilde{C}_{jk}^2 {A_{jk}}^2 \sum_{t=2}^{T} y_{k,t-1}^2 - 2\tilde{C}_{jk}A_{jk}\sum_{t=2}^{T}(y_{jt}y_{k,t-1} - y_{k,t-1}\sum_{\substack{m=1\\m\neq k}}^{d_y} C_{jm}y_{m,t-1})\right]\right) \\
&\quad = exp\left(-\frac{1}{2}\left[\tilde{C}_{jk}^2(\frac{{A_{jk}}^2\sum_{t=2}^{T} y_{k,t-1}^2}{\sigma^2} + \frac{1}{\sigma_c^2}) - 2\tilde{C}_{jk}\frac{A_{jk}\sum_{t=2}^{T}(y_{jt}y_{k,t-1} - y_{k,t-1}\sum_{\substack{m=1\\m\neq k}}^{d_y} C_{jm}y_{m,t-1})}{\sigma^2}\right]\right) \\
&\quad = exp\left(-\frac{1}{2}(\frac{{A_{jk}}^2\sum_{t=2}^{T} y_{k,t-1}^2}{\sigma^2} + \frac{1}{\sigma_c^2})\left[\tilde{C}_{jk}^2 - 2\tilde{C}_{jk}\frac{A_{jk}\sum_{t=2}^{T}(y_{jt}y_{k,t-1} - y_{k,t-1}\sum_{\substack{m=1\\m\neq k}}^{d_y} C_{jm}y_{m,t-1})}{\sigma^2(\frac{{A_{jk}}^2\sum_{t=2}^{T} y_{k,t-1}^2}{\sigma^2} + \frac{1}{\sigma_c^2})}\right]\right) \\
&\quad = exp\left(-\frac{1}{2\sigma_{jk}^2}\left[\tilde{C}_{jk}^2 - 2\tilde{C}_{jk}\mu_{jk}\right]\right) \\
&\quad \propto \mathcal{N}(\tilde{C}_{jk}|\tilde{\mu}_{jk}, \tilde{\sigma}_{jk}^2),
\end{aligned}
\tag{B2.4}
$$

where we define

$$
\tilde{\mu}_{jk} = \frac{\tilde{\sigma}_{jk}^2}{\sigma^2}A_{jk}\sum_{t=2}^{T}(y_{jt}y_{k,t-1} - y_{k,t-1}\sum_{\substack{m=1\\m\neq k}}^{d_y} A_{jm}\tilde{C}_{jm}y_{m,t-1}), \qquad \tilde{\sigma}_{jk}^2 = \frac{\sigma_c^2\sigma^2}{{A_{jk}}^2\sigma_c^2\sum_{t=2}^{T} y_{k,t-1}^2 + \sigma^2}. \tag{B2.5}
$$

B .3 Posterior of $\tilde{C}_{jk}$ NOT given A_{jk}

To obtain the posterior of $\tilde{C}_{jk}$ without the knowledge of A_{jk}, we must marginalize by summing out all the possibilities for A_{jk} as

$$
\begin{aligned}
p(\tilde{C}_{jk}|\tilde{\mathbf{C}}_{-jk}, \mathbf{A}_{-jk}, \mathbf{y}_{1:T}) &= \sum_{A_{jk}=0}^{1} p(A_{jk})p(\tilde{C}_{jk}|A_{jk}, \tilde{\mathbf{C}}_{-jk}\mathbf{A}_{-jk}, \mathbf{y}_{1:T}) \\
&= p(A_{jk}=0)p(\tilde{C}_{jk}|A_{jk}=0, \tilde{\mathbf{C}}_{-jk}\mathbf{A}_{-jk}, \mathbf{y}_{1:T}) + p(A_{jk}=1)p(\tilde{C}_{jk}|A_{jk}=1, \tilde{\mathbf{C}}_{-jk}\mathbf{A}_{-jk}, \mathbf{y}_{1:T}) \\
&= (1-\rho)\mathcal{N}(\tilde{C}_{jk}|\mu_0, \sigma_0^2) + \rho\mathcal{N}(\tilde{C}_{jk}|\mu_1, \sigma_1^2).
\end{aligned}
\tag{B3.1}
$$

The result is a mixture Gaussian with

$$\begin{aligned} \mu_0 &= \tilde{\mu}_{jk}(A_{jk}=0) = 0, \qquad & \sigma_0^2 &= \tilde{\sigma}_{jk}^2(A_{jk}=0) = \sigma_c^2, \\ \mu_1 &= \tilde{\mu}_{jk}(A_{jk}=1), & \sigma_1^2 &= \tilde{\sigma}_{jk}^2(A_{jk}=1) \end{aligned} \tag{B3.2}$$

and $\tilde{\mu}_{jk}(A_{jk})$ and $\tilde{\sigma}_{jk}^2(A_{jk})$ are given in (B2.5) .

C POSTERIORS OF A_{jk}

C .1 Posterior of A_{jk} given $\tilde{C}_{jk}$

The posterior distribution of A_{jk} can be written as

$$\begin{aligned} p(A_{jk}|\tilde{\mathbf{C}}, \mathbf{A}_{-jk}, \mathbf{y}_{1:T}) &\propto p(A_{jk})p(\mathbf{A}_{-jk})p(\tilde{\mathbf{C}})p(\mathbf{y}_{1:T}|A_{jk}, \mathbf{A}_{-jk}, \tilde{\mathbf{C}}) \\ &\propto p(A_{jk}) \prod_{t=2}^{T} p(\mathbf{y}_t|\mathbf{y}_{t-1}, A_{jk}, \mathbf{A}_{-jk}, \tilde{\mathbf{C}}). \end{aligned} \tag{C1.1}$$

We assume that $\tilde{\mathbf{C}}$ and A_{jk}, for $j = 1, \ldots, d_y$, for $k = 1, \ldots, d_y$, are independent in their priors. We take the prior of A_{jk} to be Bernoulli(ρ). Now we define

$$\begin{aligned} \alpha_{jk}^{+} &= p(A_{jk}=1) \prod_{t=2}^{T} p(\mathbf{y}_t|\mathbf{y}_{t-1}, A_{jk}=1, \mathbf{A}_{-jk}, \tilde{\mathbf{C}}) \\ \alpha_{jk}^{-} &= p(A_{jk}=0) \prod_{t=2}^{T} p(\mathbf{y}_t|\mathbf{y}_{t-1}, A_{jk}=0, \mathbf{A}_{-jk}, \tilde{\mathbf{C}}). \end{aligned} \tag{C1.2}$$

to be the unnormalized probabilities that $A_{jk} = 1$ and $A_{jk} = 0$, respectively. From here, the conditional posterior probability that $A_{jk} = 1$ is

$$\alpha_{jk} = \frac{\alpha_{jk}^{+}}{\alpha_{jk}^{+} + \alpha_{jk}^{-}}. \tag{C1.3}$$

C .2 Posterior of A_{jk} NOT given $\tilde{C}_{jk}$

To obtain the posterior distribution of A_{jk} without the knowledge of $\tilde{C}_{jk}$, we must integrate out the space of $\tilde{C}_{jk}$. We write the posterior as

$$\begin{aligned} p(A_{jk}|\mathbf{A}_{-jk}, \tilde{\mathbf{C}}_{-jk}, \mathbf{y}_{1:T}) &\propto p(\mathbf{y}_{1:T}|A_{jk}, \mathbf{A}_{-jk}, \tilde{\mathbf{C}}_{jk}) p(A_{jk}|\mathbf{A}_{-jk}, \tilde{\mathbf{C}}_{-jk}) \\ &\propto p(A_{jk}) \int p(\tilde{C}_{jk}|A_{jk}, \mathbf{A}_{-jk}, \tilde{\mathbf{C}}_{-jk}) p(\mathbf{y}_{1:T}|A_{jk}, \mathbf{A}_{-jk}, \tilde{\mathbf{C}}_{-jk}, \tilde{C}_{jk}) d\tilde{C}_{jk} \\ &\propto p(A_{jk}) \int p(\tilde{C}_{jk}) p(\mathbf{y}_1) \prod_{t=2}^{T} p(\mathbf{y}_t|\mathbf{y}_{t-1}, \mathbf{A}, \tilde{\mathbf{C}}) d\tilde{C}_{jk} \\ &\propto Bernoulli(\rho) \int \mathcal{N}(\tilde{C}_{jk}|0, \sigma_c^2) \prod_{t=2}^{T} \mathcal{N}(\mathbf{y}_t|\mathbf{C}\mathbf{y}_{t-1}, \sigma^2 \mathbb{I}_{d_y}) d\tilde{C}_{jk}. \end{aligned} \tag{C2.1}$$

C .2.1 Solving Integral

We know that from (B2.4)

$$\begin{aligned} &\int \mathcal{N}(\tilde{C}_{jk}|0, \sigma_c^2) \prod_{t=2}^{T} \mathcal{N}(\mathbf{y}_t|\mathbf{C}\mathbf{y}_{t-1}, \sigma^2 \mathbb{I}_{d_y}) d\tilde{C}_{jk} \\ &\quad \propto \int exp\left(-\frac{1}{2\sigma_{jk}^2}(\tilde{C}_{jk}^2 - 2\tilde{C}_{jk}\tilde{\mu}_{jk})\right) d\tilde{C}_{jk} \\ &\quad = \int exp\left(-\frac{1}{2\tilde{\sigma}_{jk}^2}(\tilde{C}_{jk}^2 - 2\tilde{C}_{jk}\tilde{\mu}_{jk} + \tilde{\mu}_{jk}^2 - \tilde{\mu}_{jk}^2)\right) d\tilde{C}_{jk} \\ &\quad = exp\left(-\frac{1}{2\tilde{\sigma}_{jk}^2}(-\tilde{\mu}_{jk}^2)\right) \int exp\left(-\frac{1}{2\tilde{\sigma}_{jk}^2}(\tilde{C}_{jk}^2 - 2\tilde{C}_{jk}\tilde{\mu}_{jk} + \tilde{\mu}_{jk}^2)\right) d\tilde{C}_{jk} \\ &\quad = exp\left(-\frac{1}{2\tilde{\sigma}_{jk}^2}(-\tilde{\mu}_{jk}^2)\right) (2\pi\tilde{\sigma}_{jk}^2)^{\frac{1}{2}} \end{aligned} \tag{C2.2}$$

Now, going back to (C2.1), we write

$$\begin{aligned} p(A_{jk}|\mathbf{A}_{-jk}, \tilde{\mathbf{C}}_{-jk}, \mathbf{y}_{1:T}) &\propto Bernoulli(\rho) \int \mathcal{N}(\tilde{C}_{jk}|0, \sigma_c^2) \prod_{t=2}^{T} \mathcal{N}(\mathbf{y}_t|\mathbf{C}\mathbf{y}_{t-1}, \sigma^2 \mathbb{I}_{d_y}) d\tilde{C}_{jk} \\ &= \rho^{A_{jk}} (1-\rho)^{1-A_{jk}} (2\pi\tilde{\sigma}_{jk}^2)^{\frac{1}{2}} exp\left(\frac{\tilde{\mu}_{jk}^2}{2\tilde{\sigma}_{jk}^2}\right). \end{aligned} \tag{C2.3}$$

The unnormalized probabilities that $A_{jk} = 0$ and $A_{jk} = 1$, respectively, are

$$\begin{aligned}\tilde{\alpha}_{jk}^{+} &= p(A_{jk} = 1|\mathbf{A}_{-jk}, \tilde{\mathbf{C}}_{-jk}, \mathbf{y}_{1:T}) = \rho(2\pi\tilde{\sigma}_1^2)^{\frac{1}{2}} \exp\left(\frac{\tilde{\mu}_1^2}{2\tilde{\sigma}_1^2}\right) \\ \tilde{\alpha}_{jk}^{-} &= p(A_{jk} = 0|\mathbf{A}_{-jk}, \tilde{\mathbf{C}}_{-jk}, \mathbf{y}_{1:T}) = (1-\rho)(2\pi\tilde{\sigma}_c^2)^{\frac{1}{2}}\end{aligned} \tag{C2.4}$$

Finally, the conditional probability that $A_{jk} = 1$ is

$$\tilde{\alpha}_{jk} = \frac{\tilde{\alpha}_{jk}^{+}}{\tilde{\alpha}_{jk}^{+} + \tilde{\alpha}_{jk}^{-}}. \tag{C2.5}$$

Acknowledgements

The authors would like to express special thanks to the reviewers whose constructive feedback helped greatly improve the clarity and quality of the paper. This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

References

[1] R. Rabadán and A. J. Blumberg, *Topological Data Analysis for Genomics and Evolution: Topology in Biology*. Cambridge University Press, 2019.

[2] I. Shmulevich and E. R. Dougherty, *Probabilistic Boolean networks: the modeling and control of gene regulatory networks*. SIAM, 2010.

[3] J. Murphy, E. Ozkan, P. Bunch, and S. J. Godsill, "Sparse structure inference for group and network tracking," in *2016 19th International Conference on Information Fusion (FUSION)*. IEEE, 2016, pp. 1208–1214.

[4] M. Ballerini, N. Cabibbo, R. Candelier, A. Cavagna, E. Cisbani, I. Giardina, V. Lecomte, A. Orlandi, G. Parisi, A. Procaccini *et al.*, "Interaction ruling animal collective behavior depends on topological rather than metric distance: Evidence from a field study," *Proceedings of the national academy of sciences*, vol. 105, no. 4, pp. 1232–1237, 2008.

[5] P. J. Hansen and P. C. Jurs, "Chemical applications of graph theory. part i. fundamentals and topological indices," *Journal of Chemical Education*, vol. 65, no. 7, p. 574, 1988.

[6] J. Andrés, L. Gracia, P. González-Navarrete, and V. S. Safont, "Chemical structure and reactivity by means of quantum chemical topology analysis," *Computational and Theoretical Chemistry*, vol. 1053, pp. 17–30, 2015.

[7] C. W. Patty, *Foundations of topology*. Jones & Bartlett Learning, 2009.

[8] A. Ortega, P. Frossard, J. Kovačević, J. M. Moura, and P. Vandergheynst, "Graph signal processing: Overview, challenges, and applications," *Proceedings of the IEEE*, vol. 106, no. 5, pp. 808–828, 2018.

[9] C. Zhang, D. Florêncio, and P. A. Chou, "Graph signal processing-a probabilistic framework," *Microsoft Res., Redmond, WA, USA, Tech. Rep. MSR-TR-2015-31*, 2015.

[10] A. Gavili and X.-P. Zhang, "On the shift operator, graph frequency, and optimal filtering in graph signal processing," *IEEE Transactions on Signal Processing*, vol. 65, no. 23, pp. 6303–6318, 2017.

[11] C. W. Granger, "Some recent development in a concept of causality," *Journal of econometrics*, vol. 39, no. 1-2, pp. 199–211, 1988.

[12] D. Hallac, Y. Park, S. Boyd, and J. Leskovec, "Network inference via the time-varying graphical lasso." New York, NY, USA: Association for Computing Machinery, 2017. [Online]. Available: https://doi.org/10.1145/3097983.3098037

[13] C. Charbonnier, J. Chiquet, and C. Ambroise, "Weighted-lasso for structured network inference from time course data," *arXiv preprint arXiv:0910.1723*, 2009.

[14] S. W. Han, G. Chen, M.-S. Cheon, and H. Zhong, "Estimation of directed acyclic graphs through two-stage adaptive lasso for gene network inference," *Journal of the American Statistical Association*, vol. 111, no. 515, pp. 1004–1019, 2016.

[15] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 58, no. 1, pp. 267–288, 1996.

[16] P. A. Valdés-Sosa, J. M. Sánchez-Bornot, A. Lage-Castellanos, M. Vega-Hernández, J. Bosch-Bayard, L. Melie-García, and E. Canales-Rodríguez, "Estimating brain functional connectivity with sparse multivariate autoregression," *Philosophical Transactions of the Royal Society B: Biological Sciences*, vol. 360, no. 1457, pp. 969–981, 2005.

[17] A. Arnold, Y. Liu, and N. Abe, "Temporal causal modeling with graphical granger methods," in *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2007, pp. 66–75.

[18] S. Song and P. J. Bickel, "Large vector auto regressions," *arXiv preprint arXiv:1106.3915*, 2011.

[19] J. Murphy and S. J. Godsill, "Sequential sparse system estimation for linear systems with nonlinear observations," in *2016 19th International Conference on Information Fusion (FUSION)*. IEEE, 2016, pp. 606–611.

[20] N. J. Gordon, D. J. Salmond, and A. F. Smith, "Novel approach to nonlinear/non-gaussian bayesian state estimation," in *IEE proceedings F (radar and signal processing)*, vol. 140, no. 2. IET, 1993, pp. 107–113.

[21] P. M. Djuric, J. H. Kotecha, J. Zhang, Y. Huang, T. Ghirmai, M. F. Bugallo, and J. Miguez, "Particle filtering," *IEEE signal processing magazine*, vol. 20, no. 5, pp. 19–38, 2003.

[22] A. Noor, E. Serpedin, M. Nounou, and H. Nounou, "Inferring gene regulatory networks via nonlinear state-space models and exploiting sparsity," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 9, no. 4, pp. 1203–1211, 2012.

[23] R. E. Kalman, "A new approach to linear filtering and prediction problems," 1960.

[24] Ç. Taşdemir, M. F. Bugallo, and P. M. Djurić, "A particle-based´ approach for topology estimation of gene networks," in *2017 IEEE 7th International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*. IEEE, 2017, pp. 1–5.

[25] P. M. Djuric, T. Lu, and M. F. Bugallo, "Multiple particle filtering," in *2007 IEEE International Conference on Acoustics, Speech and Signal Processing-ICASSP'07*, vol. 3. IEEE, 2007, pp. III–1181.

[26] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, "Equation of state calculations by fast computing machines," *The journal of chemical physics*, vol. 21, no. 6, pp. 1087–1092, 1953.

[27] N. Metropolis and S. Ulam, "The monte carlo method," *Journal of the American statistical association*, vol. 44, no. 247, pp. 335–341, 1949.

[28] W. K. Hastings, "Monte carlo sampling methods using markov chains and their applications," 1970.

[29] S. Geman and D. Geman, "Stochastic relaxation, gibbs distributions, and the bayesian restoration of images," *IEEE Transactions on pattern analysis and machine intelligence*, no. 6, pp. 721–741, 1984.

[30] Y. Fan, X. Wang, and Q. Peng, "Inference of gene regulatory networks using bayesian nonparametric regression and topology information," *Computational and mathematical methods in medicine*, vol. 2017, 2017.

[31] M. Iloska, Y. El-Laham, and M. F. Bugallo, "A particle gibbs sampling approach to topology inference in gene regulatory networks," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 5855–5859.

[32] T. Bengtsson, P. Bickel, B. Li *et al.*, "Curse-of-dimensionality revisited: Collapse of the particle filter in very large scale systems," in *Probability and statistics: Essays in honor of David A. Freedman*. Institute of Mathematical Statistics, 2008, pp. 316–334.

[33] C. Andrieu, A. Doucet, and R. Holenstein, "Particle markov chain monte carlo methods," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 72, no. 3, pp. 269–342, 2010.

[34] S. Brooks, A. Gelman, G. Jones, and X.-L. Meng, *Handbook of markov chain monte carlo*. CRC press, 2011.

[35] A. Gelman, J. B. Carlin, H. S. Stern, D. B. Dunson, A. Vehtari, and D. B. Rubin, *Bayesian data analysis*. CRC press, 2013.

[36] L. Rabiner and B. Juang, "An introduction to hidden markov models," *ieee assp magazine*, vol. 3, no. 1, pp. 4–16, 1986.

[37] S. Särkkä, *Bayesian filtering and smoothing*. Cambridge University Press, 2013, vol. 3.

[38] C. W. Granger, "Investigating causal relations by econometric models and cross-spectral methods," *Econometrica: journal of the Econometric Society*, pp. 424–438, 1969.

[39] R. M. Neal, *Probabilistic inference using Markov chain Monte Carlo methods*. Department of Computer Science, University of Toronto Toronto, Ontario, Canada, 1993.

[40] J. P. Hobert and C. J. Geyer, "Geometric ergodicity of gibbs and block gibbs samplers for a hierarchical random effects model," *Journal of Multivariate Analysis*, vol. 67, no. 2, pp. 414–430, 1998.

[41] C. Geyer, "Introduction to markov chain monte carlo," *Handbook of markov chain monte carlo*, vol. 20116022, p. 45, 2011.

[42] G. Rodrigues, D. J. Nott, and S. A. Sisson, "Likelihood-free approximate gibbs sampling," *Statistics and Computing*, pp. 1–17, 2020.

[43] T. Park and G. Casella, "The bayesian lasso," *Journal of the American Statistical Association*, vol. 103, no. 482, pp. 681–686, 2008.

[44] A. Krishnamoorthy and D. Menon, "Matrix inversion using cholesky decomposition," in *2013 signal processing: Algorithms, architectures, arrangements, and applications (SPA)*. IEEE, 2013, pp. 70–72.

[45] M. T. Heath, *Scientific Computing: An Introductory Survey, Revised Second Edition*. SIAM, 2018.

[46] M. Madaleno and C. Pinho, "International stock market indices comovements: a new look," *International Journal of Finance & Economics*, vol. 17, no. 1, pp. 89–102, 2012.

[47] Y. Li, X.-F. Jiang, Y. Tian, S.-P. Li, and B. Zheng, "Portfolio optimization based on network topology," *Physica A: Statistical Mechanics and its Applications*, vol. 515, pp. 671–681, 2019.

[48] Y. Tang, J. J. Xiong, Y. Luo, and Y.-C. Zhang, "How do the global stock markets influence one another? evidence from finance big data and granger causality directed network," *International Journal of Electronic Commerce*, vol. 23, no. 1, pp. 85–109, 2019.

[49] "Stock market quotes & financial news," *Investing.com*. Accessed 2021.