fnets: An R Package for Network Estimation and Forecasting via Factor-Adjusted VAR Modelling

by Dom Owens, Haeran Cho, and Matteo Barigozzi

Abstract Vector autoregressive (VAR) models are useful for modelling high-dimensional time series data. This paper introduces the package fnets, which implements the suite of methodologies proposed by (Barigozzi, Cho, and Owens 2023) for the network estimation and forecasting of high-dimensional time series under a factor-adjusted vector autoregressive model, which permits strong spatial and temporal correlations in the data. Additionally, we provide tools for visualising the networks underlying the time series data after adjusting for the presence of factors. The package also offers data-driven methods for selecting tuning parameters including the number of factors, the order of autoregression, and thresholds for estimating the edge sets of the networks of interest in time series analysis. We demonstrate various features of fnets on simulated datasets as well as real data on electricity prices.

1 Introduction

Vector autoregressive (VAR) models have been popularly adopted for modelling time series data across many disciplines including economics (Koop, 2013), finance (Barigozzi and Brownlees, 2019), neuroscience (Kirch et al., 2015), and systems biology (Shojaie and Michailidis, 2010). By fitting VAR models to data, we can infer dynamic interdependence between the variables and forecast future values. In particular, by inferring the non-zero elements of the VAR parameter matrices, we can find a network representation of the data which embeds Granger causal linkages. Besides, by estimating the precision matrix (inverse of the covariance matrix) of the VAR innovations, we can define a network representing their contemporaneous dependencies by means of partial correlations. Finally, the inverse of the long-run covariance matrix of the data simultaneously captures lead-lag and contemporaneous co-movements of the variables. For further discussions on the network interpretation of VAR modelling, we refer to Dahlhaus (2000), Eichler (2007), Billio et al. (2012) and Barigozzi and Brownlees (2019).

Fitting VAR models to the data can quickly become a high-dimensional problems since the number of parameters grows quadratically with the dimensionality of the data. There exists a mature literature on ℓ_1-regularisation methods for estimating VAR models in high dimensions under suitable sparsity assumptions on the parameters (Basu and Michailidis, 2015; Han et al., 2015; Kock and Callot, 2015; Medeiros and Mendes, 2016; Nicholson et al., 2020; Liu and Zhang, 2021). Consistency of such methods is derived under the assumption that the spectral density matrix of the data has bounded eigenvalues. However, in many applications, the datasets exhibit strong serial and cross-sectional correlations which leads to the violation of this assumption. As a motivating example, we introduce a dataset of node-specific prices in the PJM (Pennsylvania, New Jersey and Maryland) power pool area in the United States, see Energy price data for further details. Figure 1 demonstrates that the leading eigenvalue of the long-run covariance matrix (i.e. spectral density matrix at frequency 0) increases linearly as the dimension of the data increases, which implies the presence of latent common factors in the panel data (Forni et al., 2000). Additionally, the left panel of Figure 2 shows the inadequacy of fitting a VAR model to such data under the sparsity assumption via ℓ_1-regularisation methods, unless the presence of strong correlations is accounted for by a *factor-adjustment* step as in the right panel.

Barigozzi et al. (2023) propose the FNETS method for factor-adjusted VAR modelling of high-dimensional, second-order stationary time series. Under their proposed model, the data is decomposed into two latent components such that the *factor-driven* component accounts for pervasive leading, lagging or contemporaneous co-movements of the variables, while the remaining *idiosyncratic* dynamic dependence between the variables is modelled by a sparse VAR process. Then, FNETS provides tools for inferring the networks underlying the latent VAR process and forecasting.

In this paper, we present an R package named **fnets** which implements the FNETS method. It provides a range of user-friendly tools for estimating and visualising the networks representing the interconnectedness of time series variables, and for producing forecasts. In addition, **fnets** includes a range of methods for selecting tuning parameters ranging from the number of factors and the VAR order, to regularisation and thresholding parameters adopted for producing sparse and interpretable networks. The main routine of **fnets** outputs an object of S3 class `fnets` which is supported by a `plot` method for network visualisation and a `predict` method for time series forecasting.

There exist several packages for fitting VAR models and their extensions to high-dimensional time series, see **LSVAR** (Bai, 2021), **sparsevar** (Vazzoler, 2021), **nets** (Brownlees, 2020), **mgm** (Haslbeck

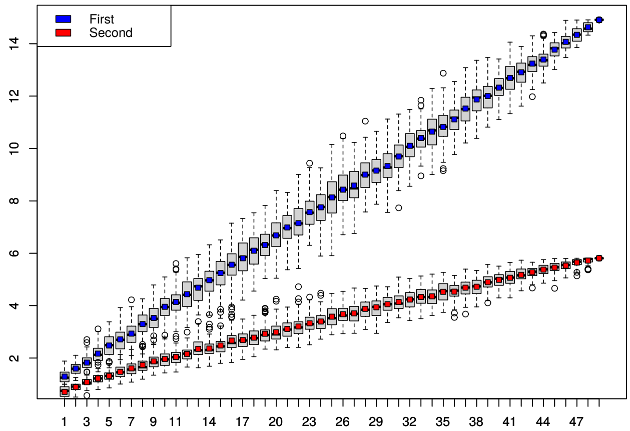

Figure 1: Box plots of the two largest eigenvalues (y-axis) of the long-run covariance matrix estimated from the energy price data collected between 01/01/2021 and 19/07/2021 ($n = 200$), see Real data example for further details. Cross-sections of the data are randomly sampled 100 times for each given dimension $p \in \{2, \ldots, 50\}$ (x-axis) to produce the box plots.

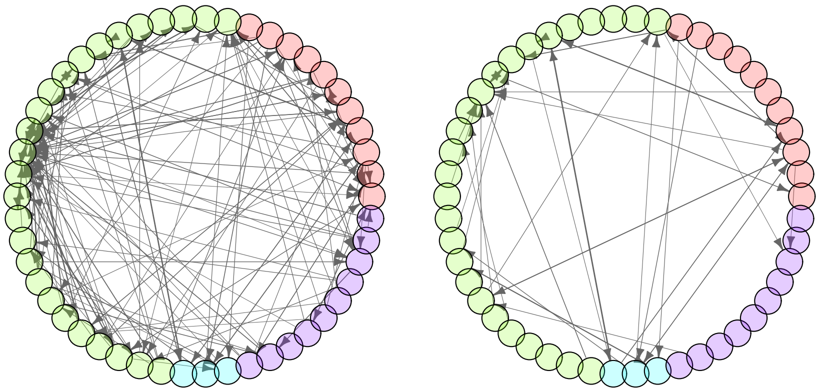

Figure 2: Granger causal networks defined in (5) obtained from fitting a VAR(1) model to the energy price data analysed in Figure 1, without (left) and with (right) the factor adjustment step outlined in FNETS: Network estimation. Edge weights (proportional to the size of coefficient estimates) are visualised by the width of each edge, and the nodes are coloured according to their groupings, see Real data example for further details.

and Waldorp, 2020), **graphicalVAR** (Epskamp et al., 2018), **BigVAR** (Nicholson et al., 2017), and **bigtime** (Wilms et al., 2021). There also exist R packages for time series factor modelling such as **dfms** (Krantz and Bagdziunas, 2023) and **sparseDFM** (Mosley et al., 2023), and **FAVAR** (Bernanke et al., 2005) for Bayesian inference of factor-augmented VAR models. The advantage of **fnets** over the above-mentioned packages is its ability to handle strong cross-sectional and serial correlations in the data via factor-adjustment step performed in the frequency domain. In addition, the FNETS method operates under the most general approach to high-dimensional time series factor modelling termed the Generalised Dynamic Factor (GDFM), first proposed in Forni et al. (2000) and further investigated in Forni et al. (2015). Accordingly, **fnets** is the first R package to provide tools for high-dimensional panel data analysis under the GDFM, such as fast computation of spectral density and autocovariance matrices via the Fast Fourier Transform, but it is flexible enough to allow for more restrictive static factor models. While there exist some packages for network-based time series modelling (e.g. **GNAR**, Knight et al., 2020), we highlight that the goal of **fnets** is to learn the networks underlying a time series and does not require a network as an input.

2 FNETS methodology

In this section, we introduce the factor-adjusted VAR model and describe the FNETS methodology proposed in Barigozzi et al. (2023) for network estimation and forecasting of high-dimensional time series. We limit ourselves to describing the key steps of FNETS and refer to the above paper for its comprehensive treatment.

2.1 Factor-adjusted VAR model

A zero-mean, p-variate process $\boldsymbol{\xi}_t$ follows a VAR(d) model if it satisfies

$$\boldsymbol{\xi}_t = \sum_{\ell=1}^{d} \mathbf{A}_\ell \boldsymbol{\xi}_{t-\ell} + \boldsymbol{\Gamma}^{1/2} \boldsymbol{\varepsilon}_t, \tag{1}$$

where $\mathbf{A}_\ell \in R^{p\times p}$, $1 \le \ell \le d$, determine how future values of the series depend on the past values. For the p-variate random vector $\boldsymbol{\varepsilon}_t = (\varepsilon_{1t}, \ldots, \varepsilon_{pt})^\top$, we assume that ε_{it} are independently and identically distributed (i.i.d.) for all i and t with $\mathbb{E}(\varepsilon_{it}) = 0$ and $\mathsf{Var}(\varepsilon_{it}) = 1$. Then, the positive definite matrix $\boldsymbol{\Gamma} \in R^{p\times p}$ is the covariance matrix of the innovations $\boldsymbol{\Gamma}^{1/2}\boldsymbol{\varepsilon}_t$.

In the literature on factor modelling of high-dimensional time series, the factor-driven component exhibits strong cross-sectional and/or serial correlations by 'loading' finite-dimensional vectors of factors linearly. Among many time series factor models, the GDFM (Forni et al., 2000) provides the most general approach where the p-variate factor-driven component $\boldsymbol{\chi}_t$ admits the following representation

$$\boldsymbol{\chi}_t = \mathcal{B}(L)\mathbf{u}_t = \sum_{\ell=0}^{\infty} \mathbf{B}_\ell \mathbf{u}_{t-\ell} \ \text{ with } \ \mathbf{u}_t = (u_{1t}, \ldots, u_{qt})^\top \text{ and } \mathbf{B}_\ell \in R^{p\times q}, \tag{2}$$

for some fixed q, where L stands for the lag operator. The q-variate random vector $\mathbf{u}_t$ contains the common factors which are loaded across the variables and time by the filter $\mathcal{B}(L) = \sum_{\ell=0}^{\infty} \mathbf{B}_\ell L^\ell$, and it is assumed that u_{jt} are i.i.d. with $\mathbb{E}(u_{jt}) = 0$ and $\mathsf{Var}(u_{jt}) = 1$. The model (2) reduces to a static factor model (Bai, 2003; Stock and Watson, 2002; Fan et al., 2013), when $\mathcal{B}(L) = \sum_{\ell=0}^{s} \mathbf{B}_\ell L^\ell$ for some finite integer $s \ge 0$. Then, we can write

$$\boldsymbol{\chi}_t = \boldsymbol{\Lambda}\mathbf{F}_t \ \text{ where } \ \mathbf{F}_t = (\mathbf{u}_t^\top, \ldots, \mathbf{u}_{t-s}^\top)^\top \ \text{ and } \ \boldsymbol{\Lambda} = [\mathbf{B}_0, \ldots, \mathbf{B}_s] \tag{3}$$

with $r = q(s+1)$ as the dimension of static factors $\mathbf{F}_t$. Throughout, we refer to the models (2) and (3) as *unrestricted* and *restricted* to highlight that the latter imposes more restrictions on the model.

Barigozzi et al. (2023) propose a factor-adjusted VAR model under which we observe a zero-mean, second-order stationary process $\mathbf{X}_t = (X_{1t}, \ldots, X_{pt})^\top$ for $t = 1, \ldots, n$, that permits a decomposition into the sum of the unobserved components $\boldsymbol{\xi}_t$ and $\boldsymbol{\chi}_t$, i.e.

$$\mathbf{X}_t = \boldsymbol{\xi}_t + \boldsymbol{\chi}_t. \tag{4}$$

We assume that $\mathbb{E}(\varepsilon_{it} u_{jt'}) = 0$ for all i, j, t and t' as is commonly assumed in the literature, such that $\mathbb{E}(\xi_{it}\chi_{i't'}) = 0$ for all $1 \le i, i' \le p$ and $t, t' \in \mathbb{Z}$.

2.2 Networks

Under (4), it is of interest to infer three types of networks representing the interconnectedness of $\mathbf{X}_t$ after factor adjustment. Let $\mathcal{V} = \{1, \ldots, p\}$ denote the set of vertices representing the p cross-sections. Then, the VAR parameter matrices, $\mathbf{A}_\ell = [A_{\ell,ii'}, 1 \le i, i' \le p]$, encode the directed network $\mathcal{N}^{\mathrm{G}} = (\mathcal{V}, \mathcal{E}^{\mathrm{G}})$ representing Granger causal linkages, where the set of edges are given by

$$\mathcal{E}^{\mathrm{G}} = \left\{(i, i') \in \mathcal{V} \times \mathcal{V} : \ A_{\ell,ii'} \ne 0 \text{ for some } 1 \le \ell \le d\right\}. \tag{5}$$

Here, the presence of an edge $(i, i') \in \mathcal{E}^{\mathrm{G}}$ indicates that $\xi_{i',t-\ell}$ Granger causes ξ_{it} at some lag $1 \le \ell \le d$ (Dahlhaus, 2000).

The second network contains undirected edges representing contemporaneous cross-sectional dependence in VAR innovations $\boldsymbol{\Gamma}^{1/2}\boldsymbol{\varepsilon}_t$, denoted by $\mathcal{N}^{\mathrm{C}} = (\mathcal{V}, \mathcal{E}^{\mathrm{C}})$. We have $(i, i') \in \mathcal{E}^{\mathrm{C}}$ if and only if the partial correlation between the i-th and i'-th elements of $\boldsymbol{\Gamma}^{1/2}\boldsymbol{\varepsilon}_t$ is non-zero, which in turn is given by $-\delta_{ii'}/\sqrt{\delta_{ii} \cdot \delta_{i'i'}}$ where $\boldsymbol{\Gamma}^{-1} = \boldsymbol{\Delta} = [\delta_{ii'}, 1 \le i, i' \le p]$ (Peng et al., 2009). Hence, the set of edges for $\mathcal{N}^{\mathrm{C}}$ is given by

$$\mathcal{E}^{\mathrm{C}} = \left\{(i, i') \in \mathcal{V} \times \mathcal{V} : \, i \ne i' \text{ and } -\frac{\delta_{ii'}}{\sqrt{\delta_{ii} \cdot \delta_{i'i'}}} \ne 0\right\}, \tag{6}$$

Finally, we can summarise the aforementioned lead-lag and contemporaneous relations between the variables in a single, undirected network $\mathcal{N}^{\mathrm{L}} = (\mathcal{V}, \mathcal{E}^{\mathrm{L}})$ by means of the long-run partial correlations of $\boldsymbol{\xi}_t$. Let $\boldsymbol{\Omega} = [\omega_{ii'}, 1 \le i, i' \le p]$ denote the inverse of the zero-frequency spectral density (a.k.a. long-run covariance) of $\boldsymbol{\xi}_t$, which is given by $\boldsymbol{\Omega} = 2\pi\mathcal{A}^\top(1)\boldsymbol{\Delta}\mathcal{A}(1)$ with $\mathcal{A}(z) = \mathbf{I} - \sum_{\ell=1}^d \mathbf{A}_\ell z^\ell$. Then, the long-run partial correlation between the i-th and i'-th elements of $\boldsymbol{\xi}_t$, is obtained as $-\omega_{ii'}/\sqrt{\omega_{ii} \cdot \omega_{i'i'}}$ (Dahlhaus, 2000), so the edge set of $\mathcal{N}^{\mathrm{L}}$ is given by

$$\mathcal{E}^{\mathrm{L}} = \left\{ (i, i') \in \mathcal{V} \times \mathcal{V} : i \neq i' \text{ and } -\frac{\omega_{ii'}}{\sqrt{\omega_{ii} \cdot \omega_{i'i'}}} \neq 0 \right\}. \tag{7}$$

2.3 FNETS: Network estimation

We describe the three-step methodology for estimating the networks $\mathcal{N}^{\mathrm{G}}$, $\mathcal{N}^{\mathrm{C}}$ and $\mathcal{N}^{\mathrm{L}}$. Throughout, we assume that the number of factors, either q under the more general model in (2) or r under the restricted model in (3), and the VAR order d, are known, and discuss its selection in Tuning parameter selection.

Step 1: Factor adjustment

The autocovariance (ACV) matrices of $\boldsymbol{\xi}_t$, denoted by $\boldsymbol{\Gamma}_\xi(\ell) = \mathbb{E}(\boldsymbol{\xi}_{t-\ell}\boldsymbol{\xi}_t^\top)$ for $\ell \ge 0$ and $\boldsymbol{\Gamma}_\xi(\ell) = (\boldsymbol{\Gamma}_\xi(-\ell))^\top$ for $\ell < 0$, play a key role in network estimation. Since $\boldsymbol{\xi}_t$ is not directly observed, we propose to adjust for the presence of the factor-driven $\boldsymbol{\chi}_t$ and estimate $\boldsymbol{\Gamma}_\xi(\ell)$. For this, we adopt a frequency domain-based approach and perform the dynamic principal component analysis (PCA). Spectral density matrix $\boldsymbol{\Sigma}_x(\omega)$ of a time series $\{\mathbf{X}_t\}_{t\in\mathbb{Z}}$ aggregates information of its ACV $\boldsymbol{\Gamma}_x(\ell)$, $\ell \in \mathbb{Z}$, at a specific frequency $\omega \in [-\pi, \pi]$, and is obtained by the Fourier transform $\boldsymbol{\Sigma}_x(\omega) = (2\pi)^{-1}\sum_{\ell=-\infty}^{\infty} \boldsymbol{\Gamma}_x(\ell)\exp(-\iota\ell\omega)$ where $\iota = \sqrt{-1}$. Denoting the sample ACV matrix of $\mathbf{X}_t$ at lag ℓ by

$$\widehat{\boldsymbol{\Gamma}}_x(\ell) = \frac{1}{n}\sum_{t=\ell+1}^{n} \mathbf{X}_{t-\ell}\mathbf{X}_t^\top \text{ when } \ell \ge 0 \quad \text{and} \quad \widehat{\boldsymbol{\Gamma}}_x(\ell) = (\widehat{\boldsymbol{\Gamma}}_x(-\ell))^\top \text{ when } \ell < 0,$$

we estimate the spectral density of $\mathbf{X}_t$ by

$$\widehat{\boldsymbol{\Sigma}}_x(\omega_k) = \frac{1}{2\pi}\sum_{\ell=-m}^{m} K\left(\frac{\ell}{m}\right)\widehat{\boldsymbol{\Gamma}}_x(\ell)\exp(-\iota\ell\omega_k), \tag{8}$$

where $K(\cdot)$ denotes a kernel, m the kernel bandwidth (for its choice, see Tuning parameter selection) and $\omega_k = 2\pi k/(2m+1)$ the Fourier frequencies. We adopt the Bartlett kernel as $K(\cdot)$, which ensures positive semi-definiteness of $\widehat{\boldsymbol{\Sigma}}_x(\omega)$ and also $\widehat{\boldsymbol{\Gamma}}_\xi(\ell)$ estimating $\boldsymbol{\Gamma}_\xi(\ell)$ obtained as described below.

Performing PCA on $\widehat{\boldsymbol{\Sigma}}_x(\omega_k)$ at each ω_k, we obtain the estimator of the spectral density matrix of $\boldsymbol{\chi}_t$ as $\widehat{\boldsymbol{\Sigma}}_\chi(\omega_k) = \sum_{j=1}^q \widehat{\mu}_{x,j}(\omega_k)\widehat{\mathbf{e}}_{x,j}(\omega_k)(\widehat{\mathbf{e}}_{x,j}(\omega_k))^*$, where $\widehat{\mu}_{x,j}(\omega_k)$ denotes the j-th largest eigenvalue of $\widehat{\boldsymbol{\Sigma}}_x(\omega_k)$, $\widehat{\mathbf{e}}_{x,j}(\omega_k)$ its associated eigenvector, and for any vector $\mathbf{a} \in \mathbb{C}^n$, we denote its transposed complex conjugate by $\mathbf{a}^*$. Then taking the inverse Fourier transform of $\widehat{\boldsymbol{\Sigma}}_\chi(\omega_k)$, $-m \le k \le m$, leads to an estimator of $\boldsymbol{\Gamma}_\chi(\ell)$, the ACV matrix of $\boldsymbol{\chi}_t$, as

$$\widehat{\boldsymbol{\Gamma}}_\chi(\ell) = \frac{2\pi}{2m+1}\sum_{k=-m}^{m}\widehat{\boldsymbol{\Sigma}}_\chi(\omega_k)\exp(\iota\ell\omega_k) \quad \text{for} \quad -m \le \ell \le m.$$

Finally, we estimate the ACV of $\boldsymbol{\xi}_t$ by

$$\widehat{\boldsymbol{\Gamma}}_\xi(\ell) = \widehat{\boldsymbol{\Gamma}}_x(\ell) - \widehat{\boldsymbol{\Gamma}}_\chi(\ell). \tag{9}$$

When we assume the restricted factor model in (3), the factor-adjustment step is simplified as it suffices to perform PCA in the time domain, i.e. eigenanalysis of the sample covariance matrix $\widehat{\boldsymbol{\Gamma}}_x(0)$. Denoting the eigenvector of $\widehat{\boldsymbol{\Gamma}}_x(0)$ associated with its j-th largest eigenvalue by $\widehat{\mathbf{e}}_{x,j}$, we obtain $\widehat{\boldsymbol{\Gamma}}_\xi(\ell) = \widehat{\boldsymbol{\Gamma}}_x(\ell) - \widehat{\mathbf{E}}_x\widehat{\mathbf{E}}_x^\top\widehat{\boldsymbol{\Gamma}}_x(\ell)\widehat{\mathbf{E}}_x\widehat{\mathbf{E}}_x^\top$ where $\widehat{\mathbf{E}}_x = [\widehat{\mathbf{e}}_{x,j}, 1 \le j \le r]$.

Step 2: Estimation of $\mathcal{N}^{\mathrm{G}}$

Recall from (5) that $\mathcal{N}^{\mathrm{G}}$, representing Granger causal linkages, has its edge set determined by the VAR transition matrices $\mathbf{A}_\ell$, $1 \le \ell \le d$. By the Yule-Walker equation, we have $\boldsymbol{\beta} = [\mathbf{A}_1, \ldots, \mathbf{A}_d]^\top =$

$\mathbf{G}(d)^{-1}\mathbf{g}(d)$, where

$$\mathbf{G}(d) = \begin{bmatrix} \mathbf{\Gamma}_\xi(0) & \mathbf{\Gamma}_\xi(-1) & \dots & \mathbf{\Gamma}_\xi(-d+1) \\ \mathbf{\Gamma}_\xi(1) & \mathbf{\Gamma}_\xi(0) & \dots & \mathbf{\Gamma}_\xi(-d+2) \\ & & \ddots & \\ \mathbf{\Gamma}_\xi(d-1) & \mathbf{\Gamma}_\xi(d-2) & \dots & \mathbf{\Gamma}_\xi(0) \end{bmatrix} \quad \text{and} \quad \mathbf{g}(d) = \begin{bmatrix} \mathbf{\Gamma}_\xi(1) \\ \mathbf{\Gamma}_\xi(2) \\ \vdots \\ \mathbf{\Gamma}_\xi(d) \end{bmatrix}. \tag{10}$$

We propose to estimate $\boldsymbol{\beta}$ as a regularised Yule-Walker estimator based on $\widehat{\mathbf{G}}(d)$ and $\widehat{\mathbf{g}}(d)$, each of which is obtained by replacing $\mathbf{\Gamma}_\xi(\ell)$ with $\widehat{\mathbf{\Gamma}}_\xi(\ell)$, see (9), in the definition of $\mathbf{G}(d)$ and $\mathbf{g}(d)$.

For any matrix $\mathbf{M} = [m_{ij}] \in R^{n_1 \times n_2}$, let $|\mathbf{M}|_1 = \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} |m_{ij}|$, $|\mathbf{M}|_\infty = \max_{1 \le i \le n_1} \max_{1 \le j \le n_2} |m_{ij}|$ and $\mathrm{tr}(\mathbf{M}) = \sum_{i=1}^{n_1} m_{ii}$ when $n_1 = n_2$. We consider two estimators of $\boldsymbol{\beta}$. Firstly, we adopt a Lasso-type estimator which solves an ℓ_1-regularised M-estimation problem

$$\widehat{\boldsymbol{\beta}}^{\text{las}} = \arg\min_{\mathbf{M} \in \mathbb{R}^{pd \times p}} \mathrm{tr}\left(\mathbf{M}^\top \widehat{\mathbf{G}}(d)\mathbf{M} - 2\mathbf{M}^\top \widehat{\mathbf{g}}(d)\right) + \lambda |\mathbf{M}|_1 \tag{11}$$

with a tuning parameter $\lambda > 0$. In the implementation, we solve (11) via the fast iterative shrinkage-thresholding algorithm (FISTA, Beck and Teboulle, 2009). Alternatively, we adopt a constrained ℓ_1-minimisation approach closely related to the Dantzig selector (DS, Candes and Tao, 2007):

$$\widehat{\boldsymbol{\beta}}^{\text{DS}} = \arg\min_{\mathbf{M} \in \mathbb{R}^{pd \times p}} |\mathbf{M}|_1 \quad \text{subject to} \quad \left|\widehat{\mathbf{G}}(d)\mathbf{M} - \widehat{\mathbf{g}}(d)\right|_\infty \le \lambda \tag{12}$$

for some tuning parameter $\lambda > 0$. We divide (12) into p sub-problems and obtain each column of $\widehat{\boldsymbol{\beta}}^{\text{DS}}$ via the simplex algorithm (using the function `lp` in **lpSolve** (Berkelaar et al., 2020)), which is performed in parallel with **doParallel** and **foreach** (Microsoft and Weston, 2022a,b).

Barigozzi et al. (2023) establish the consistency of both $\widehat{\boldsymbol{\beta}}^{\text{las}}$ and $\widehat{\boldsymbol{\beta}}^{\text{DS}}$ but, as is typically the case for ℓ_1-regularisation methods, they do not achieve exact recovery of the support of $\boldsymbol{\beta}$. Hence we propose to estimate the edge set of $\mathcal{N}^{\text{G}}$ by thresholding the elements of $\widehat{\boldsymbol{\beta}}$ with some threshold $\mathfrak{t} > 0$, where either $\widehat{\boldsymbol{\beta}} = \widehat{\boldsymbol{\beta}}^{\text{las}}$ or $\widehat{\boldsymbol{\beta}} = \widehat{\boldsymbol{\beta}}^{\text{DS}}$, i.e.

$$\widetilde{\boldsymbol{\beta}}(\mathfrak{t}) = \left[\widehat{\beta}_{ij} \cdot \mathbb{I}_{\{|\widehat{\beta}_{ij}| > \mathfrak{t}\}}, 1 \le i \le pd, 1 \le j \le p\right]. \tag{13}$$

We discuss cross validation and information criterion methods for selecting λ, and a data-driven choice of $\mathfrak{t}$, in Tuning parameter selection.

Step 3: Estimation of $\mathcal{N}^{\mathbf{C}}$ and $\mathcal{N}^{\mathbf{L}}$

From the definitions of $\mathcal{N}^{\text{C}}$ and $\mathcal{N}^{\text{L}}$ given in (6) and (7), their edge sets are obtained by estimating $\mathbf{\Delta} = \mathbf{\Gamma}^{-1}$ and $\mathbf{\Omega} = 2\pi\mathcal{A}^\top(1)\mathbf{\Delta}\mathcal{A}(1)$. Suppose that we are given $\widehat{\boldsymbol{\beta}} = [\widehat{\mathbf{A}}_1, \dots, \widehat{\mathbf{A}}_d]^\top$, some estimator of the VAR parameter matrices obtained as in either (11) or (12). Then, a natural estimator of $\mathbf{\Gamma}$ arises from the Yule-Walker equation $\mathbf{\Gamma} = \mathbf{\Gamma}_\xi(0) - \sum_{\ell=1}^d \mathbf{A}_\ell \mathbf{\Gamma}_\xi(\ell) = \mathbf{\Gamma}_\xi(0) - \boldsymbol{\beta}^\top \mathbf{g}$, as $\widehat{\mathbf{\Gamma}} = \widehat{\mathbf{\Gamma}}_\xi(0) - \widehat{\boldsymbol{\beta}}^\top \widehat{\mathbf{g}}$. In high dimensions, it is not feasible or recommended to directly invert $\widehat{\mathbf{\Gamma}}$ to estimate $\mathbf{\Delta}$. Therefore, we adopt a constrained ℓ_1-minimisation method motivated by the CLIME methodology of Cai et al. (2011).

Specifically, the CLIME estimator of $\mathbf{\Delta}$ is obtained by first solving

$$\check{\mathbf{\Delta}} = \arg\min_{\mathbf{M} \in \mathbb{R}^{p \times p}} |\mathbf{M}|_1 \quad \text{subject to} \quad \left|\widehat{\mathbf{\Gamma}}\mathbf{M} - \mathbf{I}\right|_\infty \le \eta, \tag{14}$$

and applying a symmetrisation step to $\check{\mathbf{\Delta}} = [\check{\delta}_{ii'}, 1 \le i, j \le p]$ as

$$\widehat{\mathbf{\Delta}} = [\widehat{\delta}_{ii'}, 1 \le i, i' \le p] \text{ with } \widehat{\delta}_{ii'} = \check{\delta}_{ii'} \cdot \mathbb{I}_{\{|\check{\delta}_{ii'}| \le |\check{\delta}_{i'i}|\}} + \check{\delta}_{i'i} \cdot \mathbb{I}_{\{|\check{\delta}_{i'i}| < |\check{\delta}_{ii'}|\}}. \tag{15}$$

for some tuning parameter $\eta > 0$. Cai et al. (2016) propose ACLIME, which improves the CLIME estimator by selecting the parameter η in (15) adaptively. It first produces the estimators of the diagonal entries $\delta_{ii}, 1 \le i \le p$, as in (15) with $\eta_1 = 2\sqrt{\log(p)/n}$ as the tuning parameter. Then these estimates are used for adaptive tuning parameter selection in the second step. We provide the full description of the ACLIME estimator along with the details of its implementation in ACLIME estimator of the Appendix.

Given the estimators $\widehat{\mathcal{A}}(1) = \mathbf{I} - \sum_{\ell=1}^d \widehat{\mathbf{A}}_\ell$ and $\widehat{\mathbf{\Delta}}$, we estimate $\mathbf{\Omega}$ by $\widehat{\mathbf{\Omega}} = 2\pi\widehat{\mathcal{A}}^\top(1)\widehat{\mathbf{\Delta}}\widehat{\mathcal{A}}(1)$. In Barigozzi et al. (2023), $\widehat{\mathbf{\Delta}}$ and $\widehat{\mathbf{\Omega}}$ are shown to be consistent in ℓ_∞- and ℓ_1-norms under suitable sparsity assumptions. However, an additional thresholding step as in (13) is required to guarantee consistency in estimating the support of $\mathbf{\Delta}$ and $\mathbf{\Omega}$ and consequently the edge sets of $\mathcal{N}^{\mathrm{C}}$ and $\mathcal{N}^{\mathrm{L}}$. We discuss data-driven selection of these thresholds and η in Tuning parameter selection.

2.4 FNETS: Forecasting

Following the estimation procedure, FNETS performs forecasting by estimating the best linear predictor of $\mathbf{X}_{n+a}$ given $\mathbf{X}_t$, $t \le n$, for a fixed integer $a \ge 1$. This is achieved by separately producing the best linear predictors of χ_{n+a} and ξ_{n+a} as described below, and then combining them.

Forecasting the factor-driven component

For given $a \ge 0$, the best linear predictor of χ_{n+a} given $\mathbf{X}_t$, $t \le n$, under (2) is

$$\chi_{n+a|n} = \sum_{\ell=0}^{\infty} \mathbf{B}_{\ell+a}\mathbf{u}_{n-\ell}.$$

Forni et al. (2015) show that the model (2) admits a low-rank VAR representation with $\mathbf{u}_t$ as the innovations under mild conditions, and Forni et al. (2017) propose the estimators of $\mathbf{B}_\ell$ and $\mathbf{u}_t$ based on this representation which make use of the estimators of the ACV of χ_t obtained as described in Step 1. Then, a natural estimator of $\chi_{n+a|n}$ is

$$\widehat{\chi}^{\mathrm{unr}}_{n+a|n} = \sum_{\ell=0}^{K} \widehat{\mathbf{B}}_{\ell+a}\widehat{\mathbf{u}}_{n-\ell} \tag{16}$$

for some truncation lag K. We refer to $\widehat{\chi}^{\mathrm{unr}}_{n+a|n}$ as the *unrestricted* estimator of $\chi_{n+a|n}$ as it is obtained without imposing any restrictions on the factor model (2).

When χ_t admits the static representation in (3), we can show that $\chi_{n+a|n} = \mathbf{\Gamma}_\chi(-a)\mathbf{E}_\chi\boldsymbol{\mathcal{M}}_\chi^{-1}\mathbf{E}_\chi^\top\chi_n$, where $\boldsymbol{\mathcal{M}}_\chi \in R^{r\times r}$ is a diagonal matrix with the r eigenvalues of $\mathbf{\Gamma}_\chi(0)$ on its diagonal and $\mathbf{E}_\chi \in R^{p\times r}$ the matrix of the corresponding eigenvectors; see Section 4.1 of Barigozzi et al. (2023) and also Forni et al. (2005). This suggests an estimator

$$\widehat{\chi}^{\mathrm{res}}_{n+a|n} = \widehat{\mathbf{\Gamma}}_\chi(-a)\widehat{\mathbf{E}}_\chi\widehat{\boldsymbol{\mathcal{M}}}_\chi^{-1}\widehat{\mathbf{E}}_\chi^\top\mathbf{X}_n, \tag{17}$$

where $\widehat{\boldsymbol{\mathcal{M}}}_\chi$ and $\widehat{\mathbf{E}}_\chi$ are obtained from the eigendecomposition of $\widehat{\mathbf{\Gamma}}_\chi(0)$. We refer to $\widehat{\chi}^{\mathrm{res}}_{n+a|n}$ as the *restricted* estimator of $\chi_{n+a|n}$. As a by-product, we obtain the in-sample estimators of χ_t, $t \le n$, as $\widehat{\chi}_{t|n} = \widehat{\chi}_t$, with either of the two estimators in (16) and (17).

Forecasting the latent VAR process

Once the VAR parameters are estimated either as in (11) or (12), we produce an estimator of $\xi_{n+a|n} = \sum_{\ell=1}^{d}\mathbf{A}_\ell\xi_{n+a-\ell}$, the best linear predictor of ξ_{n+a} given $\mathbf{X}_t$, $t \le n$, as

$$\widehat{\xi}_{n+a|n} = \sum_{\ell=1}^{\max(1,a)-1}\widehat{\mathbf{A}}_\ell\widehat{\xi}_{n+a-\ell|n} + \sum_{\ell=\max(1,a)}^{d}\widehat{\mathbf{A}}_\ell\widehat{\xi}_{n+a-\ell}. \tag{18}$$

Here, $\widehat{\xi}_{n+1-\ell} = \mathbf{X}_{n+1-\ell} - \widehat{\chi}_{n+1-\ell}$ denotes the in-sample estimator of $\xi_{n+1-\ell}$, which may be obtained with either of the two (in-sample) estimators of the factor-driven component in (16) and (17).

3 Tuning parameter selection

3.1 Factor numbers q and r

The estimation and forecasting tools of the FNETS methodology require the selection of the number of factors, i.e. q under the unrestricted factor model in (2), and r under the restricted, static factor model in (3). Under (2), there exists a large gap between the q leading eigenvalues of the spectral density matrix of $\mathbf{X}_t$ and the remainder which diverges with p (see also Figure 1). We provide two methods for selecting the factor number q, which make use of the postulated eigengap using $\widehat{\mu}_{x,j}(\omega_k)$, $1 \le j \le p$, the eigenvalues of the spectral density estimator of $\mathbf{X}_t$ in (8) at a given Fourier frequency ω_k, $-m \le k \le m$.

Hallin and Liška (2007) propose an information criterion for selecting the number of factors under the model (2) and further, a methodology for tuning the multiplicative constant in the penalty. Define

$$\mathrm{IC}(b, c) = \log\left(\frac{1}{p}\sum_{j=b+1}^{p}\frac{1}{2m+1}\sum_{k=-m}^{m}\widehat{\mu}_{x,j}(\omega_k)\right) + b\cdot c\cdot \mathrm{pen}(n, p), \tag{19}$$

where $\mathrm{pen}(n, p) = \min(p, m^2, \sqrt{n/m})^{-1/2}$ by default (for other choices of the information criterion, see Appendix A), and $c > 0$ a constant. Provided that $\mathrm{pen}(n, p) \to 0$ sufficiently slowly, for an arbitrary value of c, the factor number q is consistently estimated by the minimiser of $\mathrm{IC}(b, c)$ over $b \in \{0, \ldots, \bar{q}\}$, with some fixed $\bar{q}$ as the maximum allowable number of factors. However, this is not the case in finite sample, and Hallin and Liška (2007) propose to simultaneously select q and c. First, we identify $\widehat{q}(n_l, p_l, c) = \arg\min_{0\le b\le\bar{q}} \mathrm{IC}(n_l, p_l, b, c)$ where $\mathrm{IC}(n_l, p_l, b, c)$ is constructed analogously to $\mathrm{IC}(b, c)$, except that it only involves the sub-sample $\{X_{it},\, 1 \le i \le p_l,\, 1 \le t \le n_l\}$, for sequences $0 < n_1 < \ldots < n_L = n$ and $0 < p_1 < \ldots < p_L = p$. Then, denoting the sample variance of $\widehat{q}(n_l, p_l, c)$, $1 \le l \le L$, by $S(c)$, we select $\widehat{q} = \widehat{q}(n, p, \widehat{c})$ with $\widehat{c}$ corresponding to the second interval of stability with $S(c) = 0$ for the mapping $c \mapsto S(c)$ as c increases from 0 to some $c_{\max}$ (the first stable interval is where $\bar{q}$ is selected with a very small value of c). Figure 3 plots $\widehat{q}(n, p, c)$ and $S(c)$ for varying values of c obtained from a dataset simulated in Data simulation. In the implementation of this methodology, we set $n_l = n - (L - l)\lfloor n/20\rfloor$ and $p_l = \lfloor 3p/4 + lp/40\rfloor$ with $L = 10$, and $\bar{q} = \min(50, \lfloor\sqrt{\min(n-1, p)}\rfloor)$.

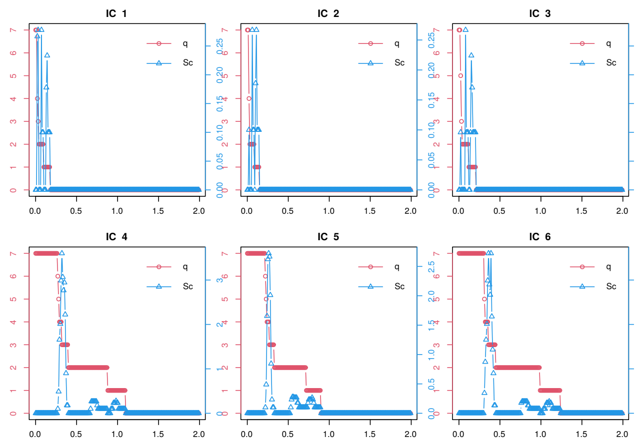

Figure 3: Plots of c against $\widehat{q}(n, p, c)$ (in circles, y-axis on the left) and $S(c)$ (in triangles, y-axis on the right) with the six IC (see Appendix A) implemented in the function `factor.number` of **fnets**, on a dataset simulated as described in Data simulation (with $n = 500$, $p = 50$ and $q = 2$). With the default choice of IC in (19) (IC_5), we obtain $\widehat{q} = \widehat{q}(n, p, \widehat{c}) = 2$ correctly estimating $q = 2$.

Alternatively, we can adopt the ratio-based estimator $\widehat{q} = \arg\min_{1\le b\le\bar{q}} \mathrm{ER}(b)$ proposed in Avarucci et al. (2022), where

$$\mathrm{ER}(b) = \left(\sum_{k=-m}^{m}\widehat{\mu}_{x,b+1}(\omega_k)\right)^{-1}\left(\sum_{k=-m}^{m}\widehat{\mu}_{x,b}(\omega_k)\right). \tag{20}$$

These methods are readily modified to select the number of factors r under the restricted factor model in (3), by replacing $(2m+1)^{-1}\sum_{k=-m}^{m}\widehat{\mu}_{x,j}(\omega_k)$ with $\widehat{\mu}_{x,j}$, the j-th largest eigenvalues of the sample covariance matrix $\widehat{\mathbf{\Gamma}}_x(0)$. We refer to Bai and Ng (2002) and Alessi et al. (2010) for the discussion of the information criterion-based method in this setting, and Ahn and Horenstein (2013) for that of the eigenvalue ratio-based method.

3.2 Threshold $\mathfrak{t}$

Motivated by Liu et al. (2021), we propose a method for data-driven selection of the threshold $\mathfrak{t}$, which is applied to the estimators of $\mathbf{A}_\ell$, $1 \le \ell \le d$, $\mathbf{\Delta}$ or $\mathbf{\Omega}$ for estimating the edge sets of $\mathcal{N}^{\mathrm{G}}$, $\mathcal{N}^{\mathrm{C}}$ or $\mathcal{N}^{\mathrm{L}}$, respectively, see also (13).

Let $\mathbf{B} = [b_{ij}] \in R^{m\times n}$ denote a matrix for which a threshold is to be selected, i.e. $\mathbf{B}$ may be either $\widehat{\boldsymbol{\beta}} = [\widehat{\mathbf{A}}_1, \ldots, \widehat{\mathbf{A}}_d]^\top$, $\widehat{\mathbf{\Delta}}_0$ ($\widehat{\mathbf{\Delta}}$ with diagonals set to zero), or $\widehat{\mathbf{\Omega}}_0$ ($\widehat{\mathbf{\Omega}}$ with diagonals set to zero), obtained from Steps 2 and 3 of FNETS. We work with $\widehat{\mathbf{\Delta}}_0$ and $\widehat{\mathbf{\Omega}}_0$ since we do not threshold the diagonal entries of $\widehat{\mathbf{\Delta}}$ and $\widehat{\mathbf{\Omega}}$. As such estimators have been shown to achieve consistency in ℓ_∞-norm, we expect there exists a large gap between the entries of $\mathbf{B}$ corresponding to true positives and false positives. Further, it is expected that the number of edges reduces at a faster rate when increasing the threshold from 0 towards this (unknown) gap, compared to when increasing the threshold from the gap to $|\mathbf{B}|_\infty$. Therefore, we propose to identify this gap by casting the problem as that of locating a single change point in the trend of the ratio of edges to non-edges,

$$\text{Ratio}_k = \frac{|\mathbf{B}(\mathfrak{t}_k)|_0}{\max(N - |\mathbf{B}(\mathfrak{t}_k)|_0, 1)}, \quad k = 1, \ldots, M.$$

Here, $\mathbf{B}(\mathfrak{t}) = [b_{ij} \cdot \mathbb{I}_{\{|b_{ij}|>\mathfrak{t}\}}]$, $|\mathbf{B}(\mathfrak{t})|_0 = \sum_{i=1}^{m_1} \sum_{j=1}^{m_2} \mathbb{I}_{\{|b_{ij}|>\mathfrak{t}\}}$ and $\{\mathfrak{t}_k,\, 1 \le k \le M :\, 0 = \mathfrak{t}_1 < \mathfrak{t}_2 < \cdots < \mathfrak{t}_M = |\mathbf{B}|_\infty\}$ denotes a sequence of candidate threshold values. We recommend using an exponentially growing sequence for $\{\mathfrak{t}_k\}_{k=1}^M$ since the size of the false positive entries tends to be very small. The quantity N in the denominator of Ratio_k is set as $N = p^2 d$ when $\mathbf{B} = \widehat{\boldsymbol{\beta}}$, and $N = p(p-1)$ when $\mathbf{B} = \widehat{\mathbf{\Delta}}_0$ or $\mathbf{B} = \widehat{\mathbf{\Omega}}_0$. Then, from the difference quotient

$$\text{Diff}_k = \frac{\text{Ratio}_k - \text{Ratio}_{k-1}}{\mathfrak{t}_k - \mathfrak{t}_{k-1}}, \quad k = 2, \ldots, M,$$

we compute the cumulative sum (CUSUM) statistic

$$\text{CUSUM}_k = \sqrt{\frac{k(M-k)}{M}} \left| \frac{1}{k} \sum_{l=2}^{k} \text{Diff}_l - \frac{1}{M-k} \sum_{l=k+1}^{M} \text{Diff}_l \right|, \quad k = 2, \ldots, M-1,$$

and select $\mathfrak{t}_{\text{ada}} = \mathfrak{t}_{k^*}$ with $k^* = \arg\max_{2 \le k \le M-1} \text{CUSUM}_k$. For illustration, Figure 4 plots Ratio_k and CUSUM_k against candidate thresholds for the dataset simulated in Data simulation.

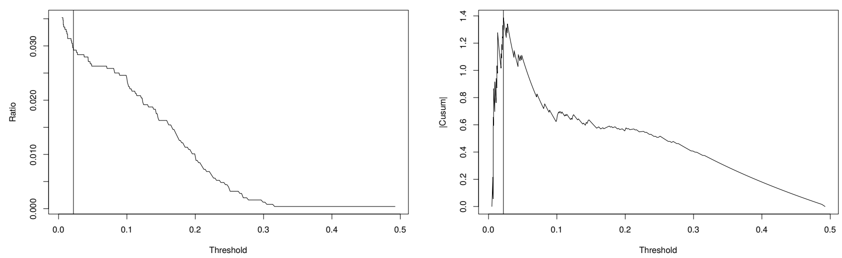

Figure 4: Ratio_k (left) and CUSUM_k (right) plotted against $\mathfrak{t}_k$ when $\mathbf{B} = \widehat{\boldsymbol{\beta}}^{\text{las}}$ obtained from the data simulated in Data simulation with $n = 500$ and $p = 50$, as a Lasso estimator of the VAR parameter matrix, with the selected $\mathfrak{t}_{\text{ada}}$ denoted by the vertical lines.

3.3 VAR order d, λ and η

Step 2 and Step 3 of the network estimation methodology of FNETS involve the selection of the tuning parameters λ and η (see (11), (12) and (14)) and the VAR order d. While there exist a variety of methods available for VAR order selection in fixed dimensions (Lütkepohl, 2005, Chapter 4), the data-driven selection of d in high dimensions remains largely unaddressed with a few exceptions (Nicholson et al., 2020; Krampe and Margaritella, 2021; Zheng, 2022). We suggest two methods for jointly selecting λ and d for Step 2. The first method is also applicable for selecting η in Step 3.

Cross validation

Cross validation (CV) methods have been popularly adopted for tuning parameter and model selection. Bergmeir et al. (2018) study the usage of a conventional CV procedure that randomly partitions the data, in the time series settings when the model is correctly specified. However, such arguments do not apply to our problem since the VAR process is latent. Instead, we propose to adopt a modified CV procedure that bears resemblance to out-of-sample evaluation or rolling forecasting validation (Wang and Tsay, 2021), for simultaneously selecting d and λ in Step 2. For this, the data is partitioned into L folds, $\mathcal{I}_l = \{n_l^\circ + 1, \ldots, n_{l+1}^\circ\}$ with $n_l^\circ = \min(l\lceil n/L \rceil, n)$, $1 \le l \le L$, and each fold is split into a training set $\mathcal{I}_l^{\text{train}} = \{n_l^\circ + 1, \ldots, \lceil (n_l^\circ + n_{l+1}^\circ)/2 \rceil\}$ and a test set $\mathcal{I}_l^{\text{test}} = \mathcal{I}_l \setminus \mathcal{I}_l^{\text{train}}$. On each fold, $\boldsymbol{\beta}$ is estimated from $\{\mathbf{X}_t,\ t \in \mathcal{I}_l^{\text{train}}\}$ as either the Lasso (11) or the Dantzig selector (12) estimators with λ as the tuning parameter and some b as the VAR order, say $\widehat{\boldsymbol{\beta}}_l^{\text{train}}(\lambda, b)$, using which we compute the CV measure

$$\text{CV}(\lambda, b) = \sum_{l=1}^{L} \text{tr}\left(\widehat{\boldsymbol{\Gamma}}_{\xi,l}^{\text{test}}(0) - (\widehat{\boldsymbol{\beta}}_l^{\text{train}}(\lambda, b))^\top \widehat{\mathbf{g}}_l^{\text{test}}(b) - (\widehat{\mathbf{g}}_l^{\text{test}}(b))^\top \widehat{\boldsymbol{\beta}}_l^{\text{train}}(\lambda, b) + (\widehat{\boldsymbol{\beta}}_l^{\text{train}}(\lambda, b))^\top \widehat{\mathbf{G}}_l^{\text{test}}(b) \widehat{\boldsymbol{\beta}}_l^{\text{train}}(\lambda, b)\right),$$

where $\widehat{\boldsymbol{\Gamma}}_{\xi,l}^{\text{test}}(\ell)$, $\widehat{\mathbf{G}}_l^{\text{test}}(b)$ and $\widehat{\mathbf{g}}_l^{\text{test}}(b)$ are generated analogously as $\widehat{\boldsymbol{\Gamma}}_\xi(\ell)$, $\widehat{\mathbf{G}}(b)$ and $\widehat{\mathbf{g}}(b)$, respectively, from the test set $\{\mathbf{X}_t,\ t \in \mathcal{I}_l^{\text{test}}\}$. Although we do not directly observe $\boldsymbol{\xi}_t$, the measure $\text{CV}(\lambda, b)$ gives an approximation of the prediction error. Then, we select $(\widehat{\lambda}, \widehat{d}) = \arg\min_{\lambda \in \Lambda, 1 \le b \le \bar{d}} \text{CV}(\lambda, b)$, where Λ is a grid of values for λ, and $\bar{d} \ge 1$ is a pre-determined upper bound on the VAR order. A similar approach is taken for the selection of η with a Burg matrix divergence-based CV measure:

$$\text{CV}(\eta) = \sum_{l=1}^{L} \text{tr}\left(\widehat{\boldsymbol{\Delta}}_l^{\text{train}}(\eta) \widehat{\boldsymbol{\Gamma}}_l^{\text{test}}\right) - \log\left|\widehat{\boldsymbol{\Delta}}_l^{\text{train}}(\eta) \widehat{\boldsymbol{\Gamma}}_l^{\text{test}}\right| - p.$$

Here, $\widehat{\boldsymbol{\Delta}}_l^{\text{train}}(\eta)$ denotes the estimator of $\boldsymbol{\Delta}$ with η as the tuning parameter from $\{\mathbf{X}_t,\ t \in \mathcal{I}_l^{\text{train}}\}$, and $\widehat{\boldsymbol{\Gamma}}_l^{\text{test}}$ the estimator of $\boldsymbol{\Gamma}$ from $\{\mathbf{X}_t,\ t \in \mathcal{I}_l^{\text{test}}\}$, see Step 3 for the descriptions of the estimators. In the numerical results reported in Simulations, the sample size is relatively small (ranging between $n = 200$ and $n = 500$ while $p \in \{50, 100, 200\}$ and the number of parameters increasing with p^2), and we set $L = 1$ which returns reasonably good performance. When more observations are available, relative to the dimensionality, we may use the number of folds greater than one.

Extended Bayesian information criterion

Alternatively, to select the pair (λ, d) in Step 2, we propose to use the extended Bayesian information criterion (eBIC) of Chen and Chen (2008), originally proposed for variable selection in high-dimensional linear regression. Let $\widetilde{\boldsymbol{\beta}}(\lambda, b, \mathfrak{t}_{\text{ada}})$ denote the thresholded version of $\widehat{\boldsymbol{\beta}}(\lambda, b)$ as in (13) with the threshold $\mathfrak{t}_{\text{ada}}$ chosen as described in Threshold t. Then, letting $s(\lambda, b) = |\widetilde{\boldsymbol{\beta}}(\lambda, b, \mathfrak{t}_{\text{ada}})|_0$, we define

$$\text{eBIC}_\alpha(\lambda, b) = \frac{n}{2} \log\left(\mathcal{L}(\lambda, b)\right) + s(\lambda, b) \log(n) + 2\alpha \log\binom{bp^2}{s(\lambda, b)}, \quad \text{where} \tag{21}$$
$$\mathcal{L}(\lambda, b) = \text{tr}\left(\widehat{\mathbf{G}}(b) - (\widetilde{\boldsymbol{\beta}}(\lambda, b))^\top \widehat{\mathbf{g}}(b) - (\widehat{\mathbf{g}}(b))^\top \widetilde{\boldsymbol{\beta}}(\lambda, b) + (\widetilde{\boldsymbol{\beta}}(\lambda, b))^\top \widehat{\mathbf{G}}(b) \widetilde{\boldsymbol{\beta}}(\lambda, b)\right).$$

Then, we select $(\widehat{\lambda}, \widehat{d}) = \arg\min_{\lambda \in \Lambda, 1 \le b \le \bar{d}} \text{eBIC}_\alpha(\lambda, b)$. The constant $\alpha \in (0, 1)$ determines the degree of penalisation which may be chosen from the relationship between n and p. Preliminary simulations suggest that $\alpha = 0$ is a suitable choice for the dimensions (n, p) considered in our numerical studies.

3.4 Other tuning parameters

Motivated by theoretical results reported in Barigozzi et al. (2023), we select the kernel bandwidth for Step 1 of FNETS as $m = \lfloor 4(n/\log(n))^{1/3} \rfloor$. In forecasting the factor-driven component as in (16), we set the truncation lag at $K = 20$, as it is expected that the elements of $\mathbf{B}_\ell$ decay rapidly as ℓ increases for short-memory processes.

4 Package overview

fnets is available from the Comprehensive R Archive Network (CRAN). The main function, `fnets`, implements the FNETS method for the input data and returns an object of S3 class `fnets`. `fnets.var` implements Step 2 of the FNETS methodology estimating the VAR parameters only, and is applicable directly for VAR modelling of high-dimensional time series; its outputs are of class `fnets`. `fnets.factor.model` performs factor modelling under either of the two models (2) and (3), and returns an object of class `fm`. We provide `predict` methods for the objects of classes and `fm`, and a `plot` method for the `fnets` class objects. Prior to using these functions to fit VAR models, we recommend to perform a unit root test and, if necessary, transform the time series such that it is stationary. In this section, we demonstrate how to use the functions included with the package.

4.1 Data simulation

For illustration, we generate an example dataset of $n = 500$ and $p = 50$, following the model described in (4). **fnets** provides functions for this purpose. For given n and p, the function `sim.var` generates the VAR(1) process following (1) with $d = 1$, $\mathbf{\Gamma}$ as supplied to the function ($\mathbf{\Gamma} = \mathbf{I}$ by default), and $\mathbf{A}_1$ generated as described in Simulations. The function `sim.unrestricted` generates the factor-driven component under the unrestricted factor model in (2) with q dynamic factors ($q = 2$ by default) and the filter $\mathcal{B}(L)$ generated as in model (C1) of Simulations.

```
set.seed(111)
n <- 500
p <- 50
x <- sim.var(n, p)$data + sim.unrestricted(n, p)$data
```

Throughout this section, we use the generated dataset for demonstrating the use of **fnets**, unless specified otherwise. There also exists `sim.restricted` which generates the factor-driven component under the restricted factor model in (3). For all data simulation functions, the default is to use the standard normal distribution when generating $\mathbf{u}_t$ and ε_t. However, by specifying the argument `heavy = TRUE`, the innovations are generated from $\sqrt{3/5} \cdot t_5$, the t-distribution with 5 degrees of freedom scaled to have unit variance. The package also comes attached with pre-generated datasets `data.restricted` and `data.unrestricted`.

4.2 Calling `fnets` with default parameters

The function `fnets` can be called with the $n \times p$ data matrix `x` as the only input, which sets all other arguments to their default choices. It then performs the factor-adjustment under the unrestricted model in (2) with q estimated by minimising the IC in (19). The VAR parameter matrix is estimated via the Lasso estimator in (11), with $d = 1$ as the VAR order, and the tuning parameters λ and η chosen via CV, without any thresholding step. This returns an object of class `fnets` whose entries are described in Table 1.

```
fnets(x)

Factor-adjusted vector autoregressive model with
n: 500, p: 50
Factor-driven common component ---------
Factor model: unrestricted
Factor number: 2
Factor number selection method: ic
Information criterion: IC5
Idiosyncratic VAR component ---------
VAR order: 1
VAR estimation method: lasso
Tuning method: cv
Threshold: FALSE
Non-zero entries: 95/2500
Long-run partial correlations ---------
LRPC: TRUE
```

4.3 Calling `fnets` with optional parameters

We can also specify the arguments of `fnets` to control how Steps 1–3 of FNETS are to be performed. The full model call is as follows:

```
out <- fnets(x, center = TRUE, fm.restricted = FALSE,
  q = c("ic", "er"), ic.op = NULL, kern.bw = NULL,
  common.args = list(factor.var.order = NULL, max.var.order = NULL, trunc.lags = 20,
  n.perm = 10), var.order = 1, var.method = c("lasso", "ds"),
  var.args = list(n.iter = NULL, n.cores = min(parallel::detectCores() - 1, 3)),
  do.threshold = FALSE, do.lrpc = TRUE, lrpc.adaptive = FALSE,
  tuning.args = list(tuning = c("cv", "bic"), n.folds = 1, penalty = NULL,
  path.length = 10)
)
```

Table 1: Entries of S3 objects of class `fnets`

Name	Description	Type
`q`	Factor number	integer
`spec`	Spectral density matrices for $\mathbf{X}_t$, χ_t and ξ_t (when `fm.restricted = FALSE`)	list
`acv`	Autocovariance matrices for $\mathbf{X}_t$, χ_t and ξ_t	list
`loadings`	Estimates of $\mathbf{B}_\ell$, $0 \le \ell \le K$ (when `fm.restricted = FALSE`) or $\mathbf{\Lambda}$ (when `fm.restricted = TRUE`)	array
`factors`	Estimates of $\{\mathbf{u}_t\}$ (when `fm.restricted = FALSE`) or $\{\mathbf{F}_t\}$ (when `fm.restricted = TRUE`)	array
`idio.var`	Estimates of $\mathbf{A}_\ell$, $1 \le \ell \le d$, and $\mathbf{\Gamma}$, and d and λ used	list
`lrpc`	Estimates of $\mathbf{\Delta}$, $\mathbf{\Omega}$, (long-run) partial correlations and η used	list
`mean.x`	Sample mean vector	vector
`var.method`	Estimation method for $\mathbf{A}_\ell$ (input parameter)	string
`do.lrpc`	Whether to estimate the long-run partial correlations (input parameter)	Boolean
`kern.bw`	Kernel bandwidth (when `fm.restricted = FALSE`, input parameter)	double

Here, we discuss a selection of input arguments. The `center` argument will de-mean the input. `fm.restricted` determines whether to perform the factor-adjustment under the restricted factor model in (3) or not. If the number of factors is known, we can specify `q` with a non-negative integer. Otherwise, it can be set as `"ic"` or `"er"`, which specifies either (19) or (20) as the factor number estimator, respectively. When `q = "ic"`, setting the argument `ic.op` as an integer between 1 and 6 specifies the choice of the IC (see Appendix A) where the default is `ic.op = 5`. `kern.bw` takes a positive integer which specifies the bandwidth to be used in Step 1 of FNETS. The list `common.args` specifies arguments for estimating $\mathbf{B}_\ell$ and $\mathbf{u}_t$ under (2), and relates to the low-rank VAR representation of χ_t under the unrestricted factor model. `var.order` specifies a vector of positive integers to be considered in VAR order selection. `var.method` determines the method for VAR parameter estimation, which can be either `"lasso"` (for the estimator in (11)) or `"ds"` (for that in (12)). The list `var.args` takes additional parameters for Step 2 of FNETS, such as the number of gradient descent steps (`n.iter`, when `var.method = "lasso"`) or the number of cores to use for parallel computing (`n.cores`, when `var.method = "ds"`). `do.threshold` specifies whether to threshold the estimators of $\mathbf{A}_\ell$, $1 \le \ell \le d$, $\mathbf{\Delta}$ and $\mathbf{\Omega}$. It is possible to perform Steps 1–2 of FNETS only without estimating $\mathbf{\Delta}$ and $\mathbf{\Omega}$ by setting `do.lrpc = FALSE`. If `do.lrpc = TRUE`, `lrpc.adaptive` specifies whether to use the non-adaptive estimator in (14) or the ACLIME estimator. The list `tuning.args` supplies arguments to the CV or eBIC procedures, including the number of folds L (`n.folds`), the eBIC parameter α (`penalty`, see (21)) and the length of the grid of values for λ and/or η (`path.length`). Finally, it is possible to set only a subset of the arguments of `common.args`, `var.args` and `tuning.args` whereby the unspecified arguments are set to their default values.

The factor adjustment (Step 1) and VAR parameter estimation (Step 2) functionalities can be accessed individually by calling `fnets.factor.model` and `fnets.var`, respectively. The latter is equivalent to calling `fnets` with `q = 0` and `do.lrpc = FALSE`. The former returns an object of class `fm` which contains the entries of the `fnets` object in Table 1 that relate to the factor-driven component only.

4.4 Network visualisation

Using the `plot` method available for the objects of class `fnets`, we can visualise the Granger network $\mathcal{N}^{\mathrm{G}}$ induced by the estimated VAR parameter matrices (see the left panel of Figure 5):

```
plot(out, type = "granger", display = "network")
```

With `display = "network"`, the function plots an `igraph` object from the **igraph** package (Csardi et al., 2006). Setting the argument `type` to `"pc"` or `"lrpc"`, we can visualise $\mathcal{N}^{\mathrm{C}}$ given by the partial correlations of VAR innovations or $\mathcal{N}^{\mathrm{L}}$ given by the long-run partial correlations of ξ_t. By setting `display = "heatmap"`, we can visualise the networks as a heat map instead, with colour indicating edge weights. This plot relies on the **fields** package (Douglas Nychka et al., 2021) and **RColorBrewer** (Neuwirth, 2022). We plot $\mathcal{N}^{\mathrm{L}}$ as a heat map in the right panel of Figure 5 using the following command:

```
plot(out, type = "lrpc", display = "heatmap")
```

It is also possible to directly produce an `igraph` object from the objects of class `fnets` via the `network` method as:

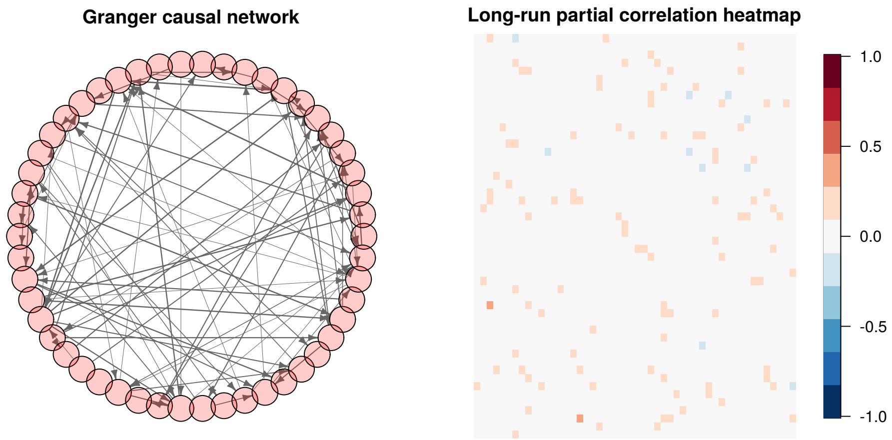

Figure 5: Estimated networks for simulated data described in Data simulation. Left: Granger causal network $\mathcal{N}^{\mathrm{G}}$. A directed arrow from node i to node i' indicates that variable i Granger causes node i', and the width of the arrow indicates the edge weight or estimated coefficient. Right: Long-run partial correlation network $\mathcal{N}^{\mathrm{L}}$ with edge weights (i.e. partial correlations) visualised by the colour.

```
g <- network(out, type = "granger")$network
plot(g, layout = igraph::layout_in_circle(g),
     vertex.color = grDevices::rainbow(1, alpha = 0.2), vertex.label = NA,
     main = "Granger causal network")
```

This produces a plot identical to the left panel of Figure 5 using the `igraph` object `g`.

4.5 Forecasting

The `fnets` objects also implement the `predict` method with which we can forecast the input data `n.ahead` steps. For example, we can produce a one-step ahead forecast of $\mathbf{X}_{n+1}$ as

```
pr <- predict(out, n.ahead = 1, fc.restricted = TRUE)
pr$forecast
```

The argument `fc.restricted` specifies whether to use the estimator $\widehat{\chi}^{\mathrm{res}}_{n+h|n}$ in (17) generated under a restricted factor model (3), or $\widehat{\chi}^{\mathrm{unr}}_{n+h|n}$ in (16) generated without such a restriction. Table 2 lists the entries from the output from `predict.fnets`. We can similarly produce forecasts from `fnets` objects output from `fnets.var`, or `fm` objects from `fnets.factor.model`.

Table 2: Entries of the output from `predict.fnets`

Name	Description	Type
`forecast`	$h \times p$ matrix containing the h-step ahead forecasts of $\mathbf{X}_t$	matrix
`common.predict`	A list containing	list
`$is`	$n \times p$ matrix containing the in-sample estimator of χ_t	
`$fc`	$h \times p$ matrix containing the h-step ahead forecasts of χ_t	
`$h`	Input parameter	
`$r`	Factor number (only produced when `fc.restricted = TRUE`)	
`idio.predict`	A list containing `is`, `fc` and `h`, see `common.predict`	list
`mean.x`	Sample mean vector	vector

4.6 Factor number estimation

It may be of interest to estimate the number of factors (if any) in the input dataset, independent of any estimation procedure. The function `factor.number` provides access to the two methods for selecting q described in Factor numbers q and r. The following code calls the information criterion-based factor number estimation method in (19), and prints the output:

```
fn <- factor.number(x, fm.restricted = FALSE)
print(fn)

Factor number selection
Factor model: unrestricted
Method: Information criterion
Number of factors:
IC1: 2
IC2: 2
IC3: 3
IC4: 2
IC5: 2
IC6: 2
```

Calling `plot(fn)` returns Figure 3 which visualises the factor number estimators from six information criteria implemented. Alternatively, we call the eigenvalue ratio-based method in (20) as

```
fn <- factor.number(x, method = "er", fm.restricted = FALSE)
```

In this case, `plot(fn)` produces a plot of $\mathrm{ER}(b)$ against the candidate factor number $b \in \{1, \ldots, \bar{q}\}$.

4.7 Visualisation of tuning parameter selection procedures

The method for threshold selection discussed in Threshold t is implemented by the `threshold` function, which returns objects of `threshold` class supported by `print` and `plot` methods.

```
th <- threshold(out$idio.var$beta)
th

Thresholded matrix
Threshold: 0.0297308643
Non-zero entries: 62/2500
```

The call `plot(th)` generates Figure 4. Additionally, we provide tools for visualising the tuning parameter selection results adopted in Steps 2 and 3 of FNETS (see VAR order d, λ and η). These tools are accessible from both `fnets` and `fnets.var` by calling the `plot` method with the argument `display = "tuning"`, e.g.

```
set.seed(111)
n <- 500
p <- 10
x <- sim.var(n, p)$data
out1 <- fnets(x, q = 0, var.order = 1:3, tuning.args = list(tuning = "cv"))
plot(out1, display = "tuning")
```

This generates the two plots reported in Figure 6 which visualise the CV errors computed as described in Cross validation and, in particular, the left plot shows that the VAR order is correctly selected by this approach. When `tuning.args` contains `tuning = "bic"`, the results from the eBIC method described in Extended Bayesian information criterion adopted in Step 2, is similarly visualised in place of the left panel of Figure 6.

5 Simulations

Barigozzi et al. (2023) provide comprehensive simulation results on the estimation and forecasting performance of FNETS in comparison with competing methodologies. Therefore in this paper, we focus on assessing the performance of the methods for selecting tuning parameters such as the threshold and VAR order discussed in Tuning parameter selection. Additionally in Appendix B, we compare the adaptive and the non-adaptive estimators in estimating $\mathbf{\Delta}$ and also investigate how their performance is carried over to estimating $\mathbf{\Omega}$.

5.1 Settings

We consider the following data generating processes for the factor-driven component χ_t:

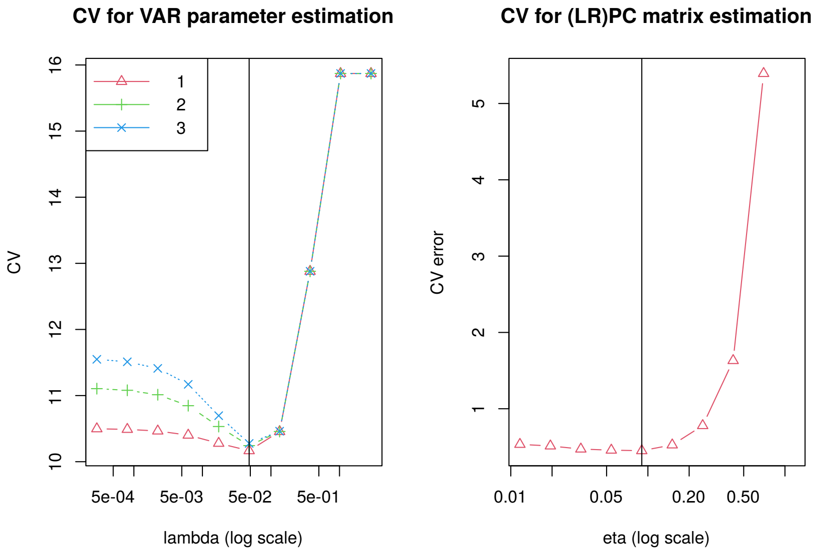

Figure 6: Plots of $\text{CV}(\lambda, b)$ against λ with $b \in \{1, 2, 3\}$ (left) and $\text{CV}(\eta)$ against η (right). Vertical lines denote where the minimum CV measure is attained with respect to λ and η, respectively.

(C1) Taken from Forni et al. (2017), χ_{it} is generated as a sum of AR processes $\chi_{it} = \sum_{j=1}^{q} a_{ij}(1 - \alpha_{ij}L)^{-1}u_{jt}$ with $q = 2$, where $u_{jt} \sim_{\text{iid}} \mathcal{N}(0, 1)$, $a_{ij} \sim_{\text{iid}} \mathcal{U}[-1, 1]$ and $\alpha_{ij} \sim_{\text{iid}} \mathcal{U}[-0.8, 0.8]$ with $\mathcal{U}[a, b]$ denoting a uniform distribution. Then, χ_t does not admit a static representation in (3).

(C2) $\chi_t = \mathbf{0}$, i.e. the VAR process is directly observed as $\mathbf{X}_t = \boldsymbol{\xi}_t$.

For generating a VAR(d) process $\boldsymbol{\xi}_t$, we first generate a directed Erdős-Rényi random graph $\mathcal{N} = (\mathcal{V}, \mathcal{E})$ on $\mathcal{V} = \{1, \ldots, p\}$ with the link probability $1/p$, and set entries of $\mathbf{A}_d$ such that $A_{d,ii'} = 0.275$ when $(i, i') \in \mathcal{E}$ and $A_{d,ii'} = 0$ otherwise. Also, we set $\mathbf{A}_\ell = \mathbf{O}$ for $\ell < d$. The VAR innovations are generated as below.

(E1) Gaussian with the covariance matrix $\boldsymbol{\Gamma} = \boldsymbol{\Delta}^{-1} = \mathbf{I}$.

(E2) Gaussian with the covariance matrix $\boldsymbol{\Gamma} = \boldsymbol{\Delta}^{-1}$ such that $\delta_{ii} = 1$, $\delta_{i,i+1} = \delta_{i+1,i} = 0.6$, $\delta_{i,i+2} = \delta_{i+2,i} = 0.3$, and $\delta_{ii'} = 0$ for $|i - i'| \geq 3$.

For each setting, we generate 100 realisations.

5.2 Results: Threshold selection

We assess the performance of the adaptive threshold. We generate χ_t as in (C1) and fix $d = 1$ for generating $\boldsymbol{\xi}_t$ and further, treat d as known. We consider $(n, p) \in \{(200, 50), (200, 100), (500, 100), (500, 200)\}$. Then we estimate $\boldsymbol{\Omega}$ using the thresholded Lasso estimator of $\mathbf{A}_1$ (see (11) and (13)) with two choices of thresholds, $\mathfrak{t} = \mathfrak{t}_{\text{ada}}$ generated as described in Threshold $\mathfrak{t}$ and $\mathfrak{t} = 0$. To assess the performance of $\widehat{\boldsymbol{\Omega}} = [\widehat{\omega}_{ii'}]$ in recovering the support of $\boldsymbol{\Omega} = [\omega_{ii'}]$, i.e. $\{(i, i') : \omega_{ii'} \neq 0\}$, we plot the receiver operating characteristic (ROC) curves of the true positive rate (TPR) against false positive rate (FPR), where

$$\text{TPR} = \frac{|\{(i, i') : \widehat{\omega}_{ii'} \neq 0 \text{ and } \omega_{ii'} \neq 0\}|}{|\{(i, i') : \omega_{ii'} \neq 0\}|} \quad \text{and} \quad \text{FPR} = \frac{|\{(i, i') : \widehat{\omega}_{ii'} \neq 0 \text{ and } \omega_{ii'} = 0\}|}{|\{(i, i') : \omega_{ii'} = 0\}|}.$$

Figure 7 plots the ROC curves averaged over 100 realisations when $\mathfrak{t} = \mathfrak{t}_{\text{ada}}$ and $\mathfrak{t} = 0$. When $\boldsymbol{\Delta} = \mathbf{I}$ under (E1), we see little improvement from adopting $\mathfrak{t}_{\text{ada}}$ as the support recovery performance is already good even without thresholding. However, when $\boldsymbol{\Delta} \neq \mathbf{I}$ under (E2), the adaptive threshold leads to improved support recovery especially when the sample size is large. Tables 3 and 4 in Appendix C additionally report the errors in estimating $\mathbf{A}_1$ and $\boldsymbol{\Omega}$ with and without thresholding, where we see little change is brought by thresholding. In summary, we conclude that the estimators already perform reasonably well without thresholding, and the adaptive threshold $\mathfrak{t}_{\text{ada}}$ brings marginal improvement in support recovery which is of interest in network estimation.

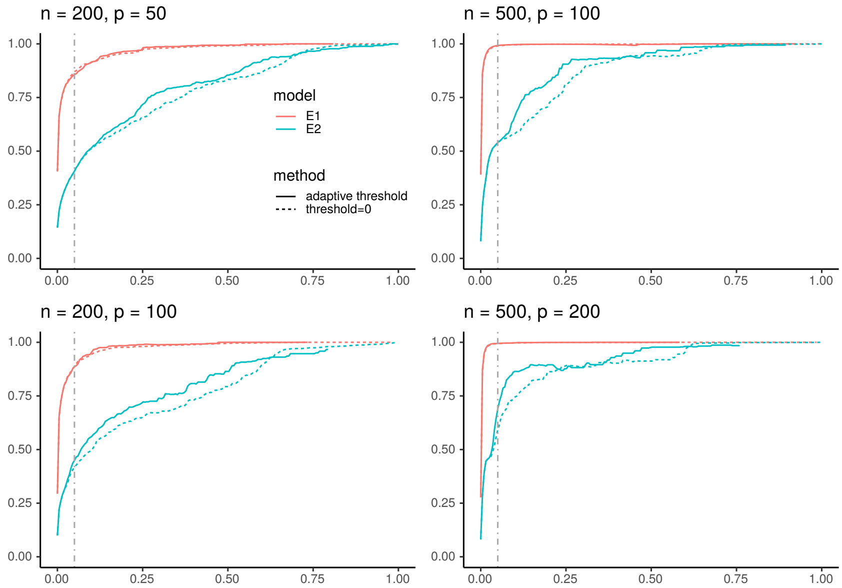

Figure 7: ROC curves of TPR against FPR for $\widetilde{\boldsymbol{\beta}}(\mathfrak{t})$ (13) (with $\widehat{\boldsymbol{\beta}} = \widehat{\boldsymbol{\beta}}^{\text{las}}$) when $\mathfrak{t} = \mathfrak{t}_{ada}$ and $\mathfrak{t} = 0$ in recovering the support of $\boldsymbol{\Omega}$, averaged over 100 realisations. Vertical lines indicate FPR $= 0.05$

5.3 Results: VAR order selection

We compare the performance of the CV and eBIC methods proposed in VAR order d, λ and η for selecting the order of the VAR process. Here, we consider the case when $\chi_t = \mathbf{0}$ (setting (C2)) and when $\boldsymbol{\xi}_t$ is generated under (E1) with $d \in \{1, 3\}$. We set $(n, p) \in \{(200, 10), (200, 20), (500, 10), (500, 20)\}$ where the range of p is in line with the simulation studies conducted in the relevant literature (see e.g. Zheng (2022)). We consider $\{1, 2, 3, 4\}$ as the candidate VAR orders. Figure 8 and Table 5 in Appendix C show that CV works reasonably well regardless of $d \in \{1, 3\}$, with slightly better performance observed together with the DS estimator. On the other hand, eBIC tends to over-estimate the VAR order when $d = 1$ while under-estimating it when $d = 3$, and hence is less reliable compared to the CV method.

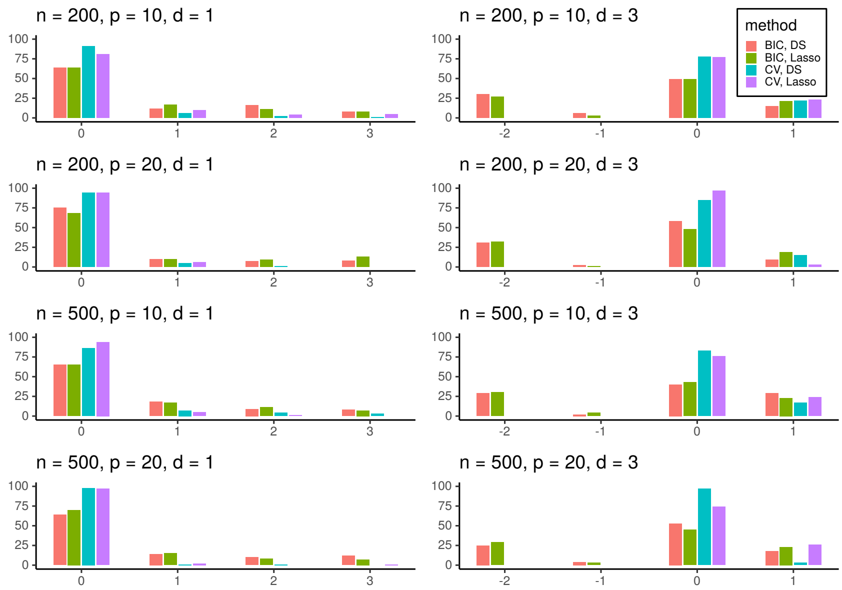

Figure 8: Box plots of $\widehat{d} - d$ over 100 realisations when the VAR order is selected by the CV and eBIC methods in combination with the Lasso (11) and the DS (12) estimators.

6 Real-world data example

6.1 Energy price data

Compared with physical commodities, electricity is more difficult to store, and this results in high volatility and seasonality in spot prices (Han et al., 2022). Global market deregulation has increased the volume of electricity trading, which promotes the development of better forecasting and risk management methods. We analyse a dataset of node-specific prices in the PJM (Pennsylvania, New Jersey and Maryland) power pool area in the United States, accessed using `dataminer2.pjm.com`. There are four node types in the panel, which are Zone, Aggregate, Hub, and Extra High Voltage (EHV) (for definitions, names, and types of the $p = 50$ nodes, see Tables 9 and 8 in Appendix D). The series we model is the sum of the real time congestion price and marginal loss price or, equivalently, the difference between the spot price at a given location and the overall system price, where the latter can be thought of as an observed factor in the local spot price. These are obtained as hourly prices and then averaged over each day as per Maciejowska and Weron (2013). We remove any short-term seasonality by subtracting a separate mean for each day of the week. Since the energy prices may take negative values, we adopt the inverse hyperbolic sine transformation as in Uniejewski et al. (2017) for variance stabilisation.

6.2 Network estimation

We analyse the data collected between 01/01/2021 and 19/07/2021 ($n = 200$). The information criterion in (19) selects a single factor ($\widehat{q} = 1$), and $\widehat{d} = 1$ is selected by CV. See Figure 9 for the heat maps visualising the three networks $\mathcal{N}^{\mathrm{G}}$, $\mathcal{N}^{\mathrm{C}}$ and $\mathcal{N}^{\mathrm{L}}$ described in Networks, which are produced by **fnets**.

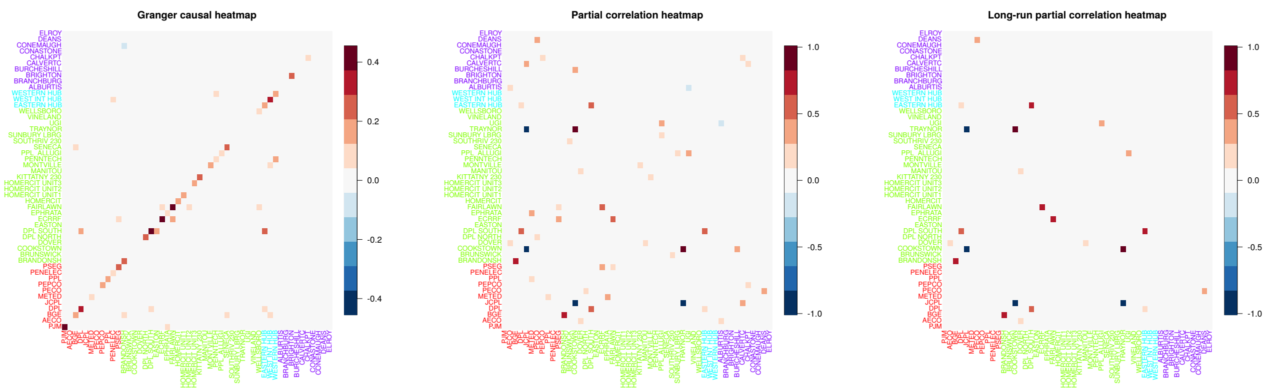

Figure 9: Heat maps of the three networks underlying the energy price data collected over the period 01/01/2021–19/07/2021. Left: $\mathcal{N}^{\mathrm{G}}$ obtained with the Lasso estimator (11) combined with the adaptive threshold $\mathfrak{t}_{\mathrm{ada}}$. Middle: $\mathcal{N}^{\mathrm{C}}$ obtained with the ACLIME estimator of $\mathbf{\Delta}$. Right: $\mathcal{N}^{\mathrm{L}}$ obtained by combining the estimators of VAR parameters and $\mathbf{\Delta}$. In the axis labels, Zone-type nodes are coloured in red, Aggregate-types in green, Hub-types in blue and EHV-types in purple.

The non-zero entries of the VAR parameter matrix estimates tend to take positive values, indicating that high energy prices are persistent and spill over to other nodes. Considering the node types, Hub-type nodes (blue) tend to have out-going edges to nodes of different types, which reflects the behaviour of the electrical transmission system. Some Zone-type nodes (red) have several in-coming edges from Aggregate-types (green) and Hub-types, while EHV-types (purple) have few edges in $\mathcal{N}^{\mathrm{G}}$, which carries forward to $\mathcal{N}^{\mathrm{L}}$ where we observe that those Zone-type nodes have strong long-run correlations with other nodes while EHV-types do not.

7 Summary

We introduce the R package **fnets** which implements the FNETS methodology proposed by Barigozzi et al. (2023) for network estimation and forecasting of high-dimensional time series exhibiting strong correlations. The package further implements several data-driven methods for selecting tuning parameters, and provides tools for high-dimensional time series factor modelling under the GDFM. The efficacy of our package is demonstrated on both real and simulated datasets.

1 Appendix A: Information criteria for factor number selection

Here we list information criteria for factor number estimation which are implemented in **fnets** and accessible by the functions `fnets`, `fnets.factor.model` and `factor.number` by setting the argument `ic.op` at an integer belonging to $\{1, \ldots, 6\}$. When `fm.restricted = FALSE`, we have

IC$_1$: $\left(\frac{1}{p}\sum_{j=b+1}^{p}\frac{1}{2m+1}\sum_{k=-m}^{m}\widehat{\mu}_{x,j}(\omega_k)\right) + b \cdot c \cdot (m^{-2} + \sqrt{m/n} + p^{-1}) \cdot \log(\min(p, m^2, \sqrt{n/m}))$,

IC$_2$: $\left(\frac{1}{p}\sum_{j=b+1}^{p}\frac{1}{2m+1}\sum_{k=-m}^{m}\widehat{\mu}_{x,j}(\omega_k)\right) + b \cdot c \cdot (\min(p, m^2, \sqrt{n/m}))^{-1/2}$,

IC$_3$: $\left(\frac{1}{p}\sum_{j=b+1}^{p}\frac{1}{2m+1}\sum_{k=-m}^{m}\widehat{\mu}_{x,j}(\omega_k)\right) + b \cdot c \cdot (\min(p, m^2, \sqrt{n/m}))^{-1} \cdot \log(\min(p, m^2, \sqrt{n/m}))$,

IC$_4$: $\log\left(\frac{1}{p}\sum_{j=b+1}^{p}\frac{1}{2m+1}\sum_{k=-m}^{m}\widehat{\mu}_{x,j}(\omega_k)\right) + b \cdot c \cdot (m^{-2} + \sqrt{m/n} + p^{-1}) \cdot \log(\min(p, m^2, \sqrt{n/m}))$,

IC$_5$: $\log\left(\frac{1}{p}\sum_{j=b+1}^{p}\frac{1}{2m+1}\sum_{k=-m}^{m}\widehat{\mu}_{x,j}(\omega_k)\right) + b \cdot c \cdot (\min(p, m^2, \sqrt{n/m}))^{-1/2}$,

IC$_6$: $\log\left(\frac{1}{p}\sum_{j=b+1}^{p}\frac{1}{2m+1}\sum_{k=-m}^{m}\widehat{\mu}_{x,j}(\omega_k)\right) + b \cdot c \cdot (\min(p, m^2, \sqrt{n/m}))^{-1} \cdot \log(\min(p, m^2, \sqrt{n/m}))$.

When `fm.restricted = TRUE`, we use one of

IC$_1$: $\left(\frac{1}{p}\sum_{j=b+1}^{p}\widehat{\mu}_{x,j}\right) + b \cdot c \cdot (n+p)/(np) \cdot \log(np/(n+p))$,

IC$_2$: $\left(\frac{1}{p}\sum_{j=b+1}^{p}\widehat{\mu}_{x,j}\right) + b \cdot c \cdot (n+p)/(np) \cdot \log(np/(n+p))$,

IC$_3$: $\left(\frac{1}{p}\sum_{j=b+1}^{p}\widehat{\mu}_{x,j}\right) + b \cdot c \cdot \log(\min(n,p))/(\min(n,p))$,

IC$_4$: $\log\left(\frac{1}{p}\sum_{j=b+1}^{p}\widehat{\mu}_{x,j}\right) + b \cdot c \cdot (n+p)/(np) \cdot \log(np/(n+p))$,

IC$_5$: $\log\left(\frac{1}{p}\sum_{j=b+1}^{p}\widehat{\mu}_{x,j}\right) + b \cdot c \cdot (n+p)/(np) \cdot \log(np/(n+p))$,

IC$_6$: $\log\left(\frac{1}{p}\sum_{j=b+1}^{p}\widehat{\mu}_{x,j}\right) + b \cdot c \cdot \log(\min(n,p))/(\min(n,p))$.

Whether `fm.restricted = FALSE` or not, the default choice is `ic.op = 5`.

2 Appendix B: ACLIME estimator

We provide a detailed description of the adaptive extension of the CLIME estimator of $\mathbf{\Delta}$ in (14), extending the methodology proposed in Cai et al. (2016) for precision matrix estimation in the independent setting. Let $\widehat{\mathbf{\Gamma}}^* = \widehat{\mathbf{\Gamma}} + n^{-1}\mathbf{I}$ and $\eta_1 = 2\sqrt{\log(p)/n}$.

Step 1: Let $\check{\mathbf{\Delta}}^{(1)} = [\check{\delta}_{ii'}^{(1)}]$ be the solution to

$$\begin{aligned} &\check{\mathbf{\Delta}}_{\cdot i'}^{(1)} = \arg\min_{\mathbf{m} \in R^p} |\mathbf{m}|_1 \quad \text{subject to} \\ &\left|(\widehat{\mathbf{\Gamma}}^*\mathbf{m} - \mathbf{e}_{i'})_i\right| \le \eta_1(\widehat{\gamma}_{ii} \vee \widehat{\gamma}_{i'i'})m_{i'} \ \forall\, 1 \le i \le p \ \text{ and } \ m_{i'} > 0, \end{aligned} \tag{22}$$

for $i' = 1, \ldots, p$. Then we obtain truncated estimates

$$\widehat{\delta}_{ii}^{(1)} = \check{\delta}_{ii}^{(1)} \cdot \mathbb{I}_{\{|\widehat{\gamma}_{ii}| \le \sqrt{n/\log(p)}\}} + \sqrt{\frac{\log(p)}{n}} \cdot \mathbb{I}_{\{|\widehat{\gamma}_{ii}| > \sqrt{n/\log(p)}\}}.$$

Step 2: We obtain

$$\check{\mathbf{\Delta}}_{\cdot i'}^{(2)} = \arg\min_{\mathbf{m} \in \mathbb{R}^p} |\mathbf{m}|_1 \quad \text{subject to} \quad \left|(\widehat{\mathbf{\Gamma}}^*\mathbf{m} - \mathbf{e}_{i'})_i\right| \le \eta_2\sqrt{\widehat{\gamma}_{ii}\widehat{\delta}_{i'i'}^{(1)}} \quad \forall\, 1 \le i \le p,$$

where $\eta_2 > 0$ is a tuning parameter. Since $\check{\mathbf{\Delta}}^{(2)}$ is not guaranteed to be symmetric, the final estimator is obtained after a symmetrisation step:

$$\widehat{\mathbf{\Delta}}_{ada} = [\widehat{\delta}_{ii'},\, 1 \le i, i' \le p] \text{ with } \widehat{\delta}_{ii'}^{(2)} = \check{\delta}_{ii'}^{(2)} \cdot \mathbb{I}_{\{|\check{\delta}_{ii'}^{(2)}| \le |\check{\delta}_{i'i}^{(2)}|\}} + \check{\delta}_{i'i}^{(2)} \cdot \mathbb{I}_{\{|\check{\delta}_{i'i}^{(2)}| < |\check{\delta}_{ii'}^{(2)}|\}}. \tag{23}$$

The constraints in (22) incorporate the parameter in the right-hand side. To use linear programming software to solve this, we formulate the constraints for each $1 \leq i' \leq p$ as

$$\begin{aligned} \forall 1 \leq i \leq p, \quad & ((\widehat{\boldsymbol{\Gamma}}^* - \boldsymbol{Q}^{i'})\mathbf{m} - \mathbf{e}_{i'})_i \leq 0, \\ \forall 1 \leq i \leq p, \quad & -((\widehat{\boldsymbol{\Gamma}}^* + \boldsymbol{Q}^{i'})\mathbf{m} - \mathbf{e}_{i'})_i \leq 0, \\ & m_{i'} > 0. \end{aligned}$$

where $\boldsymbol{Q}^{i'}$ has entries $q_{ii'} = \eta_1(\widehat{\gamma}_{ii} \vee \widehat{\gamma}_{i'i'})$ in column i' and 0 elsewhere.

3 Appendix C: Additional simulation results

3.1 Threshold selection

Tables 3 and 4 report the errors in estimating $\mathbf{A}_1$ and $\boldsymbol{\Omega}$ when the threshold $\mathfrak{t} = \mathfrak{t}_{\text{ada}}$ or $\mathfrak{t} = 0$ is applied to the estimator of $\mathbf{A}_1$ obtained by either the Lasso (11) or the DS (12) estimators. With a matrix $\boldsymbol{\gamma}$ as an estimand we measure the estimation error of its estimator $\widehat{\boldsymbol{\gamma}}$ using the following (scaled) matrix norms:

$$L_F = \frac{\|\widehat{\boldsymbol{\gamma}} - \boldsymbol{\gamma}\|_F}{\|\boldsymbol{\gamma}\|_F} \quad \text{and} \quad L_2 = \frac{\|\widehat{\boldsymbol{\gamma}} - \boldsymbol{\gamma}\|}{\|\boldsymbol{\gamma}\|}.$$

Table 3: Errors in estimating $\mathbf{A}_1$ with $\mathfrak{t} \in \{0, \mathfrak{t}_{\text{ada}}\}$ in combination with the Lasso (11) and the DS (12) estimators, measured by L_F and L_2, averaged over 100 realisations (with standard errors reported in brackets). We also report the average TPR when FPR $= 0.05$ and the corresponding standard error. See Results: Threshold selection in the main text for further information.

			$\mathfrak{t} = 0$						$\mathfrak{t} = \mathfrak{t}_{\text{ada}}$					
			$\widehat{\boldsymbol{\beta}}^{\text{las}}$			$\widehat{\boldsymbol{\beta}}^{\text{DS}}$			$\widehat{\boldsymbol{\beta}}^{\text{las}}$			$\widehat{\boldsymbol{\beta}}^{\text{DS}}$		
Model	n	p	TPR	L_F	L_2	TPR	L_F	L_2	TPR	L_F	L_2	TPR	L_F	L_2
(E1)	200	50	0.9681	0.6234	0.7204	0.8991	0.4299	0.3747	0.9413	0.6226	0.7204	0.6932	0.4487	0.3960
			(0.050)	(0.081)	(0.118)	(0.096)	(0.280)	(0.225)	(0.112)	(0.088)	(0.121)	(0.216)	(0.256)	(0.206)
	200	100	0.9398	0.6696	0.8113	0.8810	0.5772	0.4362	0.8832	0.6710	0.8132	0.6491	0.6025	0.4642
			(0.091)	(0.096)	(0.096)	(0.094)	(0.449)	(0.271)	(0.182)	(0.108)	(0.100)	(0.246)	(0.418)	(0.250)
	500	100	0.9990	0.4648	0.6682	0.9304	0.2740	0.2604	0.9971	0.4608	0.6645	0.7237	0.2806	0.2699
			(0.003)	(0.054)	(0.094)	(0.065)	(0.158)	(0.138)	(0.010)	(0.056)	(0.095)	(0.199)	(0.133)	(0.111)
	500	200	0.9986	0.5068	0.7729	0.9167	0.3680	0.3882	0.9964	0.5023	0.7637	0.7095	0.3889	0.4014
			(0.003)	(0.058)	(0.081)	(0.076)	(0.196)	(0.134)	(0.006)	(0.061)	(0.082)	(0.256)	(0.187)	(0.126)
(E2)	200	50	0.9595	0.6375	0.7075	0.8828	0.4673	0.4280	0.9442	0.6356	0.7079	0.6720	0.4835	0.4433
			(0.053)	(0.077)	(0.094)	(0.107)	(0.324)	(0.255)	(0.064)	(0.079)	(0.096)	(0.212)	(0.303)	(0.241)
	200	100	0.9624	0.6200	0.6909	0.8093	0.4519	0.4090	0.9435	0.6175	0.6913	0.5903	0.4765	0.4324
			(0.072)	(0.079)	(0.089)	(0.100)	(0.385)	(0.251)	(0.093)	(0.082)	(0.090)	(0.182)	(0.371)	(0.243)
	500	100	0.9970	0.4657	0.5533	0.9304	0.3434	0.3621	0.9958	0.4638	0.5525	0.8384	0.3370	0.3634
			(0.006)	(0.056)	(0.076)	(0.089)	(0.158)	(0.153)	(0.008)	(0.058)	(0.077)	(0.182)	(0.140)	(0.144)
	500	200	0.9981	0.4702	0.5658	0.9205	0.3684	0.3740	0.9945	0.4686	0.5665	0.8154	0.3663	0.3803
			(0.003)	(0.065)	(0.091)	(0.088)	(0.182)	(0.162)	(0.014)	(0.068)	(0.093)	(0.205)	(0.159)	(0.145)

Table 4: Errors in estimating $\boldsymbol{\Omega}$ with $\mathfrak{t} \in \{0, \mathfrak{t}_{\text{ada}}\}$ applied to the estimator of $\mathbf{A}_1$ in combination with the Lasso (11) and the DS (12) estimators, measured by L_F and L_2, averaged over 100 realisations (with standard errors reported in brackets). We also report the average TPR when FPR $= 0.05$ and the corresponding standard error. See Results: Threshold selection in the main text for further information.

			$\mathfrak{t} = 0$						$\mathfrak{t} = \mathfrak{t}_{\text{ada}}$					
			$\widehat{\boldsymbol{\beta}}^{\text{las}}$			$\widehat{\boldsymbol{\beta}}^{\text{DS}}$			$\widehat{\boldsymbol{\beta}}^{\text{las}}$			$\widehat{\boldsymbol{\beta}}^{\text{DS}}$		
Model	n	p	TPR	L_F	L_2	TPR	L_F	L_2	TPR	L_F	L_2	TPR	L_F	L_2
(E1)	200	50	0.8714	0.4143	0.5553	0.8622	0.4217	0.5691	0.8685	0.4145	0.5559	0.8640	0.4217	0.5695
			(0.108)	(0.048)	(0.066)	(0.119)	(0.054)	(0.070)	(0.118)	(0.049)	(0.067)	(0.121)	(0.055)	(0.070)
	200	100	0.8827	0.4320	0.5890	0.8961	0.4379	0.5949	0.8684	0.4326	0.5892	0.8867	0.4386	0.5960
			(0.084)	(0.050)	(0.072)	(0.080)	(0.046)	(0.065)	(0.139)	(0.052)	(0.074)	(0.120)	(0.048)	(0.066)
	500	100	0.9909	0.3311	0.4916	0.9886	0.3391	0.4989	0.9928	0.3303	0.4901	0.9901	0.3380	0.4975
			(0.016)	(0.031)	(0.069)	(0.021)	(0.036)	(0.065)	(0.015)	(0.032)	(0.069)	(0.018)	(0.037)	(0.066)
	500	200	0.9942	0.3520	0.5287	0.9916	0.3511	0.5400	0.9954	0.3512	0.5273	0.9672	0.3528	0.5399
			(0.009)	(0.038)	(0.054)	(0.018)	(0.045)	(0.065)	(0.008)	(0.039)	(0.055)	(0.129)	(0.055)	(0.072)
(E2)	200	50	0.4074	0.7831	0.8353	0.4027	0.7942	0.8335	0.4063	0.7832	0.8353	0.4045	0.7943	0.8336
			(0.073)	(0.089)	(0.072)	(0.087)	(0.079)	(0.034)	(0.072)	(0.089)	(0.072)	(0.089)	(0.079)	(0.034)
	200	100	0.4178	0.8406	0.8690	0.3541	0.9119	0.8879	0.4486	0.8407	0.8690	0.4038	0.9120	0.8880
			(0.091)	(0.108)	(0.036)	(0.107)	(0.126)	(0.045)	(0.091)	(0.108)	(0.036)	(0.123)	(0.126)	(0.045)
	500	100	0.5405	0.8267	0.8118	0.5632	0.7910	0.7953	0.5406	0.8267	0.8117	0.5628	0.7910	0.7951
			(0.111)	(0.125)	(0.047)	(0.122)	(0.166)	(0.062)	(0.111)	(0.125)	(0.047)	(0.123)	(0.166)	(0.062)
	500	200	0.5951	0.8713	0.8519	0.6487	0.8184	0.8259	0.6918	0.8713	0.8519	0.7101	0.8184	0.8258
			(0.175)	(0.165)	(0.088)	(0.159)	(0.182)	(0.090)	(0.148)	(0.165)	(0.088)	(0.122)	(0.182)	(0.090)

3.2 VAR order selection

Table 5 reports the results of VAR order estimation over 100 realisations.

Table 5: Distribution of $\widehat{d} - d$ over 100 realisations when the VAR order is selected by the CV and eBIC methods in combination with the Lasso (11) and the DS (12) estimators, see Results: VAR order selection in the main text for further information.

			CV								eBIC							
			$\widehat{\beta}^{\text{las}}$				$\widehat{\beta}^{\text{DS}}$				$\widehat{\beta}^{\text{las}}$				$\widehat{\beta}^{\text{DS}}$			
d	n	p	0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3
1	200	10	81	10	4	5	91	6	2	1	64	17	11	8	64	12	16	8
	200	20	94	6	0	0	94	5	1	0	68	10	9	13	75	10	7	8
	500	10	94	5	1	0	86	7	4	3	65	17	11	7	65	18	9	8
	500	20	97	2	0	1	98	1	1	0	70	15	8	7	64	14	10	12
			-2	-1	0	1	-2	-1	0	1	-2	-1	0	1	-2	-1	0	1
3	200	10	0	0	77	23	0	0	78	22	27	3	49	21	30	6	49	15
	200	20	0	0	97	3	0	0	85	15	32	1	48	19	31	2	58	9
	500	10	0	0	76	24	0	0	83	17	30	4	43	23	29	2	40	29
	500	20	0	0	74	26	0	0	97	3	29	3	45	23	25	4	53	18

3.3 CLIME vs. ACLIME estimators

We compare the performance of the adaptive and non-adaptive estimators for the VAR innovation precision matrix $\mathbf{\Delta}$ and its impact on the estimation of $\mathbf{\Omega}$, the inverse of the long-run covariance matrix of the data (see Step 3). We generate χ_t as in (C1), fix $d = 1$ and treat it as known and consider $(n, p) \in \{(200, 50), (200, 100), (500, 100), (500, 200)\}$.

In Tables 6 and 7, we report the errors of $\mathbf{\Delta}$ and $\mathbf{\Omega}$. We consider both the Lasso (11) and DS (12) estimators of VAR parameters, and CLIME and ACLIME estimators for $\mathbf{\Delta}$, which lead to four different estimators for $\mathbf{\Delta}$ and $\mathbf{\Omega}$, respectively. Overall, we observe that with increasing n, the performance of all estimators improve according to all metrics regardless of the scenarios (E1) or (E2), while increasing p has an adverse effect. The two methods perform similarly in setting (E1) when $\mathbf{\Delta} = \mathbf{I}$. There is marginal improvement for adopting the ACLIME estimator noticeable under (E2), particularly in TPR. Figures 10 and 11 shows the ROC curves for the support recovery of $\mathbf{\Delta}$ and $\mathbf{\Omega}$ when the Lasso estimator is used.

Table 6: Errors in estimating $\mathbf{\Delta}$ using CLIME and ACLIME estimators, measured by L_F and L_2, averaged over 100 realisations (with standard errors reported in brackets). We also report the average TPR when FPR $= 0.05$ and the corresponding standard errors.

			CLIME						ACLIME					
			$\widehat{\beta}^{\text{las}}$			$\widehat{\beta}^{\text{DS}}$			$\widehat{\beta}^{\text{las}}$			$\widehat{\beta}^{\text{DS}}$		
Model	n	p	TPR	L_F	L_2	TPR	L_F	L_2	TPR	L_F	L_2	TPR	L_F	L_2
(E1)	200	50	1.000	0.215	0.489	1.000	0.220	0.497	1.000	0.207	0.472	1.000	0.209	0.469
			(0.000)	(0.047)	(0.223)	(0.000)	(0.047)	(0.182)	(0.002)	(0.043)	(0.173)	(0.000)	(0.041)	(0.116)
	200	100	1.000	0.235	0.513	1.000	0.241	0.521	1.000	0.223	0.507	1.000	0.228	0.518
			(0.000)	(0.036)	(0.089)	(0.000)	(0.036)	(0.107)	(0.000)	(0.033)	(0.084)	(0.000)	(0.034)	(0.099)
	500	100	1.000	0.181	0.458	1.000	0.183	0.466	1.000	0.176	0.452	1.000	0.178	0.458
			(0.000)	(0.022)	(0.062)	(0.000)	(0.029)	(0.087)	(0.000)	(0.022)	(0.052)	(0.000)	(0.028)	(0.069)
	500	200	1.000	0.198	0.510	1.000	0.193	0.492	1.000	0.187	0.505	1.000	0.182	0.489
			(0.000)	(0.027)	(0.066)	(0.000)	(0.035)	(0.065)	(0.000)	(0.026)	(0.056)	(0.000)	(0.033)	(0.057)
(E2)	200	50	0.659	0.422	0.816	0.662	0.391	0.608	0.682	0.397	0.706	0.687	0.380	0.600
			(0.058)	(0.101)	(0.654)	(0.057)	(0.031)	(0.144)	(0.055)	(0.056)	(0.351)	(0.054)	(0.030)	(0.176)
	200	100	0.639	0.417	0.695	0.637	0.420	0.720	0.669	0.404	0.663	0.668	0.405	0.684
			(0.044)	(0.039)	(0.205)	(0.042)	(0.043)	(0.249)	(0.041)	(0.037)	(0.162)	(0.039)	(0.037)	(0.193)
	500	100	0.730	0.372	0.764	0.726	0.499	1.708	0.735	0.358	0.650	0.734	0.361	0.718
			(0.035)	(0.097)	(0.828)	(0.039)	(1.101)	(7.586)	(0.032)	(0.038)	(0.322)	(0.031)	(0.056)	(0.517)
	500	200	0.729	0.370	0.711	0.728	0.362	0.736	0.737	0.363	0.647	0.737	0.354	0.673
			(0.028)	(0.035)	(0.355)	(0.028)	(0.035)	(0.384)	(0.023)	(0.026)	(0.239)	(0.024)	(0.028)	(0.279)

Table 7: Errors in estimating $\mathbf{\Omega}$ using CLIME and ACLIME estimators of $\mathbf{\Delta}$, measured by L_F and L_2, averaged over 100 realisations (with standard errors reported in brackets). We also report the average TPR when FPR $= 0.05$ and the corresponding standard errors.

			CLIME						ACLIME					
			$\widehat{\beta}^{\text{las}}$			$\widehat{\beta}^{\text{DS}}$			$\widehat{\beta}^{\text{las}}$			$\widehat{\beta}^{\text{DS}}$		
Model	n	p	TPR	L_F	L_2	TPR	L_F	L_2	TPR	L_F	L_2	TPR	L_F	L_2
(E1)	200	50	0.871	0.415	0.557	0.862	0.422	0.571	0.867	0.411	0.558	0.856	0.417	0.570
			(0.108)	(0.050)	(0.070)	(0.119)	(0.055)	(0.080)	(0.106)	(0.051)	(0.088)	(0.114)	(0.053)	(0.083)
	200	100	0.883	0.432	0.589	0.896	0.438	0.595	0.868	0.423	0.583	0.883	0.429	0.587
			(0.084)	(0.050)	(0.072)	(0.080)	(0.046)	(0.065)	(0.088)	(0.048)	(0.077)	(0.085)	(0.045)	(0.061)
	500	100	0.991	0.331	0.492	0.989	0.339	0.499	0.991	0.328	0.490	0.989	0.337	0.498
			(0.016)	(0.031)	(0.069)	(0.021)	(0.036)	(0.065)	(0.015)	(0.033)	(0.070)	(0.019)	(0.036)	(0.067)
	500	200	0.994	0.352	0.529	0.992	0.351	0.540	0.994	0.344	0.525	0.990	0.342	0.537
			(0.009)	(0.038)	(0.054)	(0.018)	(0.045)	(0.065)	(0.009)	(0.038)	(0.056)	(0.014)	(0.044)	(0.068)
(E2)	200	50	0.509	0.532	0.724	0.510	0.514	0.664	0.504	0.518	0.679	0.507	0.506	0.658
			(0.078)	(0.071)	(0.243)	(0.068)	(0.043)	(0.137)	(0.071)	(0.055)	(0.162)	(0.063)	(0.043)	(0.141)
	200	100	0.511	0.541	0.683	0.513	0.542	0.695	0.509	0.531	0.674	0.504	0.531	0.679
			(0.059)	(0.047)	(0.082)	(0.065)	(0.051)	(0.093)	(0.062)	(0.045)	(0.084)	(0.061)	(0.046)	(0.084)
	500	100	0.640	0.450	0.655	0.624	0.544	1.099	0.642	0.441	0.597	0.637	0.440	0.617
			(0.066)	(0.072)	(0.402)	(0.079)	(0.866)	(3.714)	(0.059)	(0.036)	(0.118)	(0.060)	(0.047)	(0.204)
	500	200	0.670	0.461	0.630	0.658	0.450	0.630	0.677	0.456	0.612	0.661	0.445	0.605
			(0.045)	(0.041)	(0.116)	(0.043)	(0.040)	(0.117)	(0.041)	(0.036)	(0.075)	(0.037)	(0.037)	(0.082)

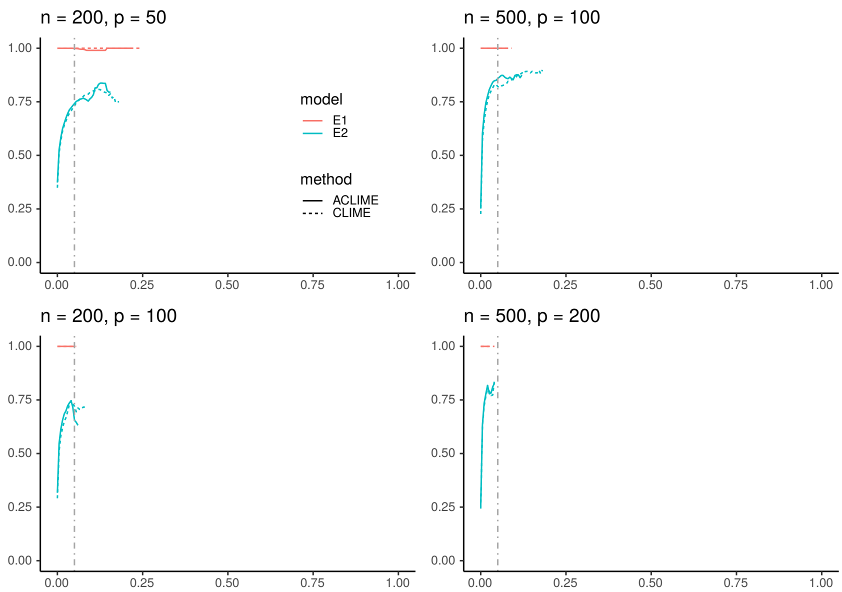

Figure 10: ROC curves of TPR against FPR for $\widehat{\mathbf{\Delta}}$ with CLIME and ACLIME estimators in recovering the support of $\mathbf{\Delta}$, averaged over 100 realisations. Vertical lines indicate FPR $= 0.05$.

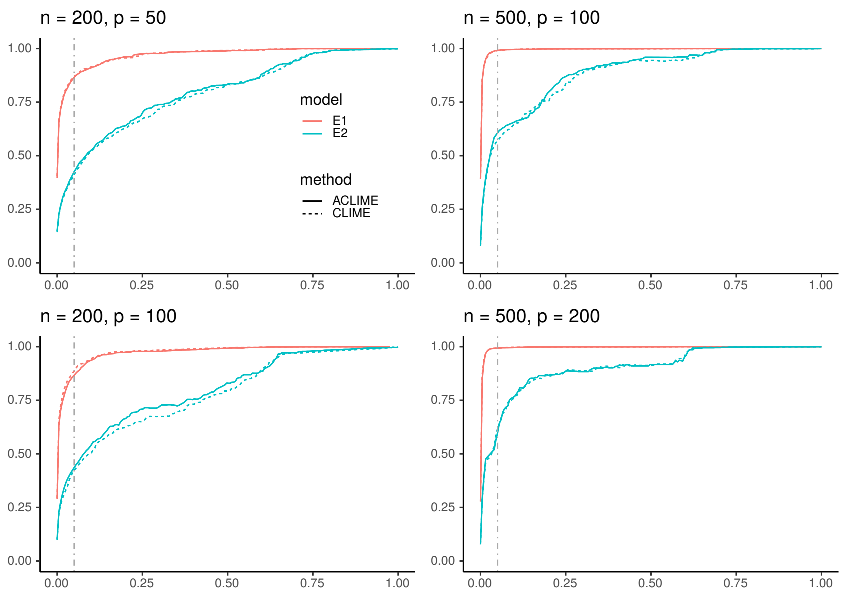

Figure 11: ROC curves of TPR against FPR for $\widehat{\mathbf{\Omega}}$ with CLIME and ACLIME estimators in recovering the support of $\mathbf{\Omega}$, averaged over 100 realisations. Vertical lines indicate FPR $= 0.05$.

4 Appendix D: Dataset information

Table 8 defines the four node types in the panel. Table 9 describes the dataset analysed in Real data example.

Table 8: Node type definitions for energy price data.

Name	Definition
Zone	A transmission owner's area within the PJM Region.
Aggregate	A group of more than one individual bus into a pricing node (pnode) that is considered as a whole in the Energy Market and other various systems and Markets within PJM.
Hub	A group of more than one individual bus into a regional pricing node (pnode) developed to produce a stable price signal in the Energy Market and other various systems and Markets within PJM.
Extra High Voltage (EHV)	Nodes at 345kV and above on the PJM system.

References

S. C. Ahn and A. R. Horenstein. Eigenvalue ratio test for the number of factors. *Econometrica*, 81(3): 1203–1227, 2013. URL https://doi.org/10.3982/ECTA8968. [p220]

L. Alessi, M. Barigozzi, and M. Capasso. Improved penalization for determining the number of factors in approximate factor models. *Statistics & Probability Letters*, 80(23-24):1806–1813, 2010. URL https://doi.org/10.1016/j.spl.2010.08.005. [p220]

M. Avarucci, M. Cavicchioli, M. Forni, and P. Zaffaroni. The main business cycle shock(s): Frequency-band estimation of the number of dynamic factors. CEPR Discussion Paper No. DP17281, 2022. URL https://dx.doi.org/10.2139/ssrn.3970658. [p220]

J. Bai. Inferential theory for factor models of large dimensions. *Econometrica*, 71(1):135–171, 2003. URL https://doi.org/10.1111/1468-0262.00392. [p216]

J. Bai and S. Ng. Determining the number of factors in approximate factor models. *Econometrica*, 70: 191–221, 2002. URL https://doi.org/10.1111/1468-0262.00273. [p220]

P. Bai. *LSVAR: Estimation of low rank plus sparse structured vector auto-regressive (VAR) model*, 2021. URL https://CRAN.R-project.org/package=LSVAR. R package version 1.2. [p214]

M. Barigozzi and C. Brownlees. Nets: Network estimation for time series. *Journal of Applied Econometrics*, 34:347–364, 2019. URL https://doi.org/10.1002/jae.2676. [p214]

M. Barigozzi, H. Cho, and D. Owens. Fnets: Factor-adjusted network estimation and forecasting for high-dimensional time series. *Journal of Business & Economic Statistics*, pages 1–13, 2023. URL https://doi.org/10.1080/07350015.2023.2257270. [p214, 215, 216, 218, 219, 222, 226, 229]

S. Basu and G. Michailidis. Regularized estimation in sparse high-dimensional time series models. *The Annals of Statistics*, 43:1535–1567, 2015. URL 10.1214/15-AOS1315. [p214]

A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202, 2009. URL https://doi.org/10.1137/080716542. [p218]

C. Bergmeir, R. J. Hyndman, and B. Koo. A note on the validity of cross-validation for evaluating autoregressive time series prediction. *Computational Statistics & Data Analysis*, 120:70–83, 2018. [p221]

M. Berkelaar et al. *lpSolve: Interface to 'Lpsolve' v. 5.5 to solve linear/integer programs*, 2020. URL https://CRAN.R-project.org/package=lpSolve. R package version 5.6.15. [p218]

B. S. Bernanke, J. Boivin, and P. Eliasz. Measuring the effects of monetary policy: a factor-augmented vector autoregressive (favar) approach. *The Quarterly journal of economics*, 120(1):387–422, 2005. URL https://doi.org/10.1162/0033553053327452. [p215]

M. Billio, M. Getmansky, A. W. Lo, and L. Pelizzon. Econometric measures of connectedness and systemic risk in the finance and insurance sectors. *Journal of Financial Economics*, 104(3):535–559, 2012. URL https://doi.org/10.1016/j.jfineco.2011.12.010. [p214]

C. Brownlees. *nets: Network estimation for time series*, 2020. URL https://CRAN.R-project.org/package=nets. R package version 0.9.1. [p214]

T. Cai, W. Liu, and X. Luo. A constrained ℓ_1 minimization approach to sparse precision matrix estimation. *Journal of the American Statistical Association*, 106(494):594–607, 2011. URL https://doi.org/10.1198/jasa.2011.tm10155. [p218]

T. T. Cai, W. Liu, and H. H. Zhou. Estimating sparse precision matrix: Optimal rates of convergence and adaptive estimation. *The Annals of Statistics*, 44(2):455–488, 2016. URL 10.1214/13-AOS1171. [p218, 230]

E. Candes and T. Tao. The dantzig selector: Statistical estimation when p is much larger than n. *The Annals of Statistics*, 35(6):2313–2351, 2007. URL 10.1214/009053606000001523. [p218]

J. Chen and Z. Chen. Extended bayesian information criteria for model selection with large model spaces. *Biometrika*, 95(3):759–771, 2008. URL https://doi.org/10.1093/biomet/asn034. [p222]

G. Csardi, T. Nepusz, et al. The igraph software package for complex network research. *InterJournal, complex systems*, 1695(5):1–9, 2006. [p224]

R. Dahlhaus. Graphical interaction models for multivariate time series. *Metrika*, 51(2):157–172, 2000. URL https://doi.org/10.1007/s001840000055. [p214, 216, 217]

Douglas Nychka, Reinhard Furrer, John Paige, and Stephan Sain. fields: Tools for spatial data, 2021. URL https://github.com/dnychka/fieldsRPackage. R package version 14.1. [p224]

M. Eichler. Granger causality and path diagrams for multivariate time series. *Journal of Econometrics*, 137(2):334–353, 2007. URL http://dx.doi.org/10.1016/j.jeconom.2005.06.032. [p214]

S. Epskamp, L. J. Waldorp, R. Mõttus, and D. Borsboom. The gaussian graphical model in cross-sectional and time-series data. *Multivariate Behavioral Research*, 53(4):453–480, 2018. URL https://doi.org/10.1080/00273171.2018.1454823. [p215]

J. Fan, Y. Liao, and M. Mincheva. Large covariance estimation by thresholding principal orthogonal complements. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 75(4), 2013. URL https://doi.org/10.1111/rssb.12016. [p216]

M. Forni, M. Hallin, M. Lippi, and L. Reichlin. The generalized dynamic-factor model: Identification and estimation. *Review of Economics and Statistics*, 82(4):540–554, 2000. URL http://dx.doi.org/10.1162/003465300559037. [p214, 215, 216]

M. Forni, M. Hallin, M. Lippi, and L. Reichlin. The generalized dynamic factor model: one-sided estimation and forecasting. *Journal of the American statistical association*, 100(471):830–840, 2005. URL https://doi.org/10.1198/016214504000002050. [p219]

M. Forni, M. Hallin, M. Lippi, and P. Zaffaroni. Dynamic factor models with infinite-dimensional factor spaces: One-sided representations. *Journal of Econometrics*, 185(2):359–371, 2015. URL https://doi.org/10.1016/j.jeconom.2013.10.017. [p215, 219]

M. Forni, M. Hallin, M. Lippi, and P. Zaffaroni. Dynamic factor models with infinite-dimensional factor space: Asymptotic analysis. *Journal of Econometrics*, 199(1):74–92, 2017. URL https://doi.org/10.1016/j.jeconom.2017.04.002. [p219, 227]

M. Hallin and R. Liška. Determining the number of factors in the general dynamic factor model. *Journal of the American Statistical Association*, 102(478):603–617, 2007. URL https://doi.org/10.1198/016214506000001275. [p219, 220]

F. Han, H. Lu, and H. Liu. A direct estimation of high dimensional stationary vector autoregressions. *Journal of Machine Learning Research*, 16(97):3115–3150, 2015. URL http://jmlr.org/papers/v16/han15a.html. [p214]

L. Han, I. Cribben, and S. Trueck. Extremal dependence in australian electricity markets. *arXiv preprint arXiv:2202.09970*, 2022. URL https://doi.org/10.48550/arXiv.2202.09970. [p229]

J. M. Haslbeck and L. J. Waldorp. mgm: Estimating time-varying mixed graphical models in high-dimensional data. *Journal of Statistical Software*, 93:1–46, 2020. URL https://doi.org/10.18637/jss.v093.i08. [p214]

C. Kirch, B. Muhsal, and H. Ombao. Detection of changes in multivariate time series with application to eeg data. *Journal of the American Statistical Association*, 110:1197–1216, 2015. URL https://doi.org/10.1080/01621459.2014.957545. [p214]

M. Knight, K. Leeming, G. Nason, and M. Nunes. Generalized network autoregressive processes and the gnar package. *Journal of Statistical Software*, 96:1–36, 2020. [p215]

A. B. Kock and L. Callot. Oracle inequalities for high dimensional vector autoregressions. *Journal of Econometrics*, 186(2):325–344, 2015. URL https://doi.org/10.1016/j.jeconom.2015.02.013. [p214]

G. M. Koop. Forecasting with medium and large bayesian vars. *Journal of Applied Econometrics*, 28(2): 177–203, 2013. URL https://doi.org/10.1002/jae.1270. [p214]

J. Krampe and L. Margaritella. Dynamic factor models with sparse var idiosyncratic components. *arXiv preprint arXiv:2112.07149*, 2021. URL https://doi.org/10.48550/arXiv.2112.07149. [p221]

S. Krantz and R. Bagdziunas. *dfms: Dynamic Factor Models*, 2023. URL https://sebkrantz.github.io/dfms/. R package version 0.2.1. [p215]

B. Liu, X. Zhang, and Y. Liu. Simultaneous change point inference and structure recovery for high dimensional gaussian graphical models. *Journal of Machine Learning Research*, 22(274):1–62, 2021. URL https://www.jmlr.org/papers/volume22/20-327/20-327.pdf. [p220]

L. Liu and D. Zhang. Robust estimation of high-dimensional vector autoregressive models. *arXiv preprint arXiv:2109.10354*, 2021. [p214]

H. Lütkepohl. *New introduction to multiple time series analysis*. Springer Science & Business Media, 2005. [p221]

K. Maciejowska and R. Weron. Forecasting of daily electricity spot prices by incorporating intra-day relationships: Evidence from the uk power market. In *2013 10th International Conference on the European Energy Market (EEM)*, pages 1–5. IEEE, 2013. URL 10.1109/EEM.2013.6607314. [p229]

M. C. Medeiros and E. F. Mendes. ℓ_1-regularization of high-dimensional time-series models with non-gaussian and heteroskedastic errors. *Journal of Econometrics*, 191(1):255–271, 2016. URL https://doi.org/10.1016/j.jeconom.2015.10.011. [p214]

Microsoft and S. Weston. *doParallel: Foreach parallel adaptor for the 'parallel' package*, 2022a. URL https://CRAN.R-project.org/package=doParallel. R package version 1.0.17. [p218]

Microsoft and S. Weston. *foreach: Provides foreach looping construct*, 2022b. URL https://CRAN.R-project.org/package=foreach. R package version 1.5.2. [p218]

L. Mosley, T.-S. Chan, and A. Gibberd. sparseDFM: An R package to estimate dynamic factor models with sparse loadings. *arXiv preprint arXiv:2303.14125*, 2023. URL https://doi.org/10.48550/arXiv.2303.14125. [p215]

E. Neuwirth. *RColorBrewer: ColorBrewer palettes*, 2022. URL https://CRAN.R-project.org/package=RColorBrewer. R package version 1.1-3. [p224]

W. Nicholson, D. Matteson, and J. Bien. Bigvar: Tools for modeling sparse high-dimensional multivariate time series. *arXiv preprint arXiv:1702.07094*, 2017. URL https://doi.org/10.48550/arXiv.1702.07094. [p215]

W. B. Nicholson, I. Wilms, J. Bien, and D. S. Matteson. High dimensional forecasting via interpretable vector autoregression. *Journal of Machine Learning Research*, 21(166):1–52, 2020. URL https://jmlr.org/papers/v21/19-777.html. [p214, 221]

J. Peng, P. Wang, N. Zhou, and J. Zhu. Partial correlation estimation by joint sparse regression models. *Journal of the American Statistical Association*, 104(486):735–746, 2009. URL 10.32614/RJ-2021-023. [p216]

A. Shojaie and G. Michailidis. Discovering graphical granger causality using the truncating lasso penalty. *Bioinformatics*, 26(18):i517–i523, 2010. URL https://doi.org/10.1093/bioinformatics/btq377. [p214]

J. H. Stock and M. W. Watson. Forecasting using principal components from a large number of predictors. *Journal of the American Statistical Association*, 97(460):1167–1179, 2002. URL https://doi.org/10.1198/016214502388618960. [p216]

B. Uniejewski, R. Weron, and F. Ziel. Variance stabilizing transformations for electricity spot price forecasting. *IEEE Transactions on Power Systems*, 33(2):2219–2229, 2017. URL `http://dx.doi.org/10.1109/TPWRS.2017.2734563`. [p229]

S. Vazzoler. *sparsevar: Sparse VAR/VECM Models Estimation*, 2021. URL `https://CRAN.R-project.org/package=sparsevar`. R package version 0.1.0. [p214]

D. Wang and R. S. Tsay. Rate-optimal robust estimation of high-dimensional vector autoregressive models. *arXiv preprint arXiv:2107.11002*, 2021. URL `https://doi.org/10.48550/arXiv.2107.11002`. [p221]

I. Wilms, S. Basu, J. Bien, and D. Matteson. bigtime: Sparse estimation of large time series models, 2021. URL `https://cran.r-project.org/package=bigtime`. R package version 0.2.1. [p215]

Y. Zheng. An interpretable and efficient infinite-order vector autoregressive model for high-dimensional time series. *arXiv preprint arXiv:2209.01172*, 2022. URL `https://doi.org/10.48550/arXiv.2209.01172`. [p221, 228]

Dom Owens
School of Mathematics, University of Bristol
Supported by EPSRC Centre for Doctoral Training (EP/S023569/1)
`domowens1@gmail.com`

Haeran Cho
School of Mathematics, University of Bristol
Supported by the Leverhulme Trust (RPG-2019-390)
`haeran.cho@bristol.ac.uk`

Matteo Barigozzi
Department of Economics, Università di Bologna
Supported by MIUR (PRIN 2017, Grant 2017TA7TYC)
`matteo.barigozzi@unibo.it`

Table 9: Names, IDs and Types for the 50 power nodes in the energy price dataset.

Name	Node ID	Node Type
PJM	1	ZONE
AECO	51291	ZONE
BGE	51292	ZONE
DPL	51293	ZONE
JCPL	51295	ZONE
METED	51296	ZONE
PECO	51297	ZONE
PEPCO	51298	ZONE
PPL	51299	ZONE
PENELEC	51300	ZONE
PSEG	51301	ZONE
BRANDONSH	51205	AGGREGATE
BRUNSWICK	51206	AGGREGATE
COOKSTOWN	51211	AGGREGATE
DOVER	51214	AGGREGATE
DPL NORTH	51215	AGGREGATE
DPL SOUTH	51216	AGGREGATE
EASTON	51218	AGGREGATE
ECRRF	51219	AGGREGATE
EPHRATA	51220	AGGREGATE
FAIRLAWN	51221	AGGREGATE
HOMERCIT	51229	AGGREGATE
HOMERCIT UNIT1	51230	AGGREGATE
HOMERCIT UNIT2	51231	AGGREGATE
HOMERCIT UNIT3	51232	AGGREGATE
KITTATNY 230	51238	AGGREGATE
MANITOU	51239	AGGREGATE
MONTVILLE	51241	AGGREGATE
PENNTECH	51246	AGGREGATE
PPL_ALLUGI	51252	AGGREGATE
SENECA	51255	AGGREGATE
SOUTHRIV 230	51261	AGGREGATE
SUNBURY LBRG	51270	AGGREGATE
TRAYNOR	51277	AGGREGATE
UGI	51279	AGGREGATE
VINELAND	51280	AGGREGATE
WELLSBORO	51285	AGGREGATE
EASTERN HUB	51217	HUB
WEST INT HUB	51287	HUB
WESTERN HUB	51288	HUB
ALBURTIS	52443	EHV
BRANCHBURG	52444	EHV
BRIGHTON	52445	EHV
BURCHESHILL	52446	EHV
CALVERTC	52447	EHV
CHALKPT	52448	EHV
CONASTONE	52449	EHV
CONEMAUGH	52450	EHV
DEANS	52451	EHV
ELROY	52452	EHV