Journal of Machine Learning Research 21 (2020) 1-52 Submitted 9/19; Revised 4/20; Published 9/20

High Dimensional Forecasting via Interpretable Vector Autoregression

William B. Nicholson* WBN8@CORNELL.EDU
Point72 Asset Management, L.P.
New York, USA

Ines Wilms I.WILMS@MAASTRICHTUNIVERSITY.NL
Department of Quantitative Economics
Maastricht University
Maastricht, The Netherlands

Jacob Bien JBIEN@USC.EDU
Department of Data Sciences and Operations
Marshall School of Business, University of Southern California
California, USA

David S. Matteson MATTESON@CORNELL.EDU
Department of Statistics and Data Science
Cornell University
Ithaca, USA

Editor: Julien Mairal

Abstract

Vector autoregression (VAR) is a fundamental tool for modeling multivariate time series. However, as the number of component series is increased, the VAR model becomes overparameterized. Several authors have addressed this issue by incorporating regularized approaches, such as the lasso in VAR estimation. Traditional approaches address overparameterization by selecting a low lag order, based on the assumption of short range dependence, assuming that a universal lag order applies to all components. Such an approach constrains the relationship between the components and impedes forecast performance. The lasso-based approaches perform much better in high-dimensional situations but do not incorporate the notion of lag order selection. We propose a new class of hierarchical lag structures (HLag) that embed the notion of lag selection into a convex regularizer. The key modeling tool is a group lasso with nested groups which guarantees that the sparsity pattern of lag coefficients honors the VAR's ordered structure. The proposed HLag framework offers three basic structures, which allow for varying levels of flexibility, with many possible generalizations. A simulation study demonstrates improved performance in forecasting and lag order selection over previous approaches, and macroeconomic, financial, and energy applications further highlight forecasting improvements as well as HLag's convenient, interpretable output.

Keywords: forecasting, group lasso, multivariate time series, variable selection, vector autoregression

∗. Mr. Nicholson contributed to this article in his personal capacity. The information, views, and opinions expressed herein are solely his own and do not necessarily represent the views of Point72. Point72 is not responsible for, and did not verify for accuracy, any of the information contained herein.

1. Introduction

Vector autoregression (VAR) has emerged as the standard-bearer for macroeconomic forecasting since the seminal work of Sims (1980). VAR is also widely applied in numerous fields, including finance (e.g., Han et al., 2015), neuroscience (e.g, Hyvärinen et al., 2010), and signal processing (e.g., Basu et al., 2019). The number of VAR parameters grows quadratically with the the number of component series, and, in the words of Sims, this "profligate parameterization" becomes intractable for large systems. Without further assumptions, VAR modeling is infeasible except in limited situations with small number of components and lag order.

Many approaches have been proposed for reducing the dimensionality of vector time series models, including canonical correlation analysis (Box and Tiao, 1977), factor models (e.g., Forni et al., 2000, Stock and Watson, 2002, Bernanke et al., 2005), Bayesian models (e.g., Banbura et al., 2010; Koop, 2013), scalar component models (Tiao and Tsay, 1989), independent component analysis (Hyvärinen et al., 2010), and dynamic orthogonal component models (Matteson and Tsay, 2011). Recent approaches have focused on imposing sparsity in the estimated coefficient matrices through the use of convex regularizers such as the lasso (Tibshirani, 1996). Most of these methods are, however, adapted from the standard regression setting and do not specifically leverage the ordered structure inherent to the lag coefficients in a VAR.

This paper contributes to the lasso-based regularization literature on VAR estimation by proposing a new class of regularized hierarchical lag structures (HLag), that embed lag order selection into a convex regularizer to simultaneously address the dimensionality and lag selection issues. HLag thus shifts the focus from obtaining estimates that are generally sparse (as measured by the number of nonzero autoregressive coefficients) to attaining estimates with *low maximal lag order*. As such, it combines several important advantages: It produces interpretable models, provides a flexible, computationally efficient method for lag order selection, and offers practitioners the ability to fit VARs in situations where various components may have highly varying maximal lag orders.

Like other lasso-based methods, HLag methods have an interpretability advantage over factor and Bayesian models. They provide direct insight into the series contributing to the forecasting of each individual component. HLag has further exploratory uses relevant for the study of different economic applications, as we find our estimated models on the considered macroeconomic data sets to have an underlying economic interpretation. Comparable Bayesian methods, in contrast, primarily perform shrinkage making the estimated models more difficult to interpret, although they can be extended to include variable selection (e.g., stochastic search). Furthermore, factor models that are combinations of all the component series can greatly reduce dimensionality but forecast contributions from the original series are only implicit. By contrast, the sparse structure imposed by the HLag penalty explicitly identifies which components are contributing to model forecasts.

While our motivating goal is to produce interpretable models with improved point forecast performance, a convenient byproduct of the HLag framework is a flexible and computationally efficient method for *lag order selection*. Depending on the proposed HLag structure choice, each equation row in the VAR will either entirely truncate at a given lag ("componentwise HLag"), or allow the series's own lags to truncate at a different order than

those of other series ("own/other HLag"), or allow every (cross) component series to have its own lag order ("elementwise HLag"). Such lag structures are conveniently depicted in a "Maxlag matrix" which we introduce and use throughout the paper.

Furthermore, HLag penalties are unique in providing a computationally tractable way to fit high order VARs, i.e., those with a *large* maximal lag order ($pmax$). They allow the possibility of certain components requiring large max-lag orders without having to enumerate over all combinations of choices. Practitioners, however, typically choose a relatively small $pmax$. We believe that this practice is in part due to the limitations of current methods: information criteria make it impossible to estimate VARs with large $pmax$ by least squares as the number of candidate lag orders scales exponentially with the number of components k. Not only is it computationally demanding to estimate so many models, overfitting also becomes a concern. Likewise, traditional lasso VAR forecasting performance degrades when $pmax$ is too large, and many Bayesian approaches, while statistically viable, are computationally infeasible or prohibitive, as we will illustrate through simulations and applications.

In Section 2 we review the literature on dimension reduction methods to address the VAR's overparametrization problem. In Section 3 we introduce the HLag framework. The three aforementioned hierarchical lag structures are proposed in Section 3.1. As detailed above, these structures vary in the degree to which lag order selection is common across different components. For each lag structure, a corresponding HLag model is detailed in Section 3.2 for attaining that sparsity structure. Theoretical properties of high-dimensional VARs estimated by HLag are analyzed in Section 3.3. The proposed methodology allows for flexible estimation in high dimensional settings with a single tuning parameter. We develop algorithms in Section 4 that are computationally efficient and parallelizable across components. Simulations in Section 5 and applications in Section 6 highlight HLag's advantages in forecasting and lag order selection.

2. Review of Mitigating VAR Overparametrization

We summarize the most popular approaches to address the VAR's overparametrization problem and discuss their link to the HLag framework.

2.1 Information Criteria

Traditional approaches address overparametrization by selecting a low lag order. Early attempts utilize least squares estimation with an information criterion or hypothesis testing (Lütkepohl, 1985). The asymptotic theory of these approaches is well developed in the fixed-dimensional setting, in which the time series length T grows while the number of components k and maximal lag order $pmax$ are held fixed (White, 2001). However, for small T, it has been observed that no criterion works well (Nickelsburg, 1985). Gonzalo and Pitarakis (2002) find that for fixed k and $pmax$, when T is relatively small, Akaike's Information Criterion (AIC) tends to overfit whereas Schwarz's Information Criterion (BIC) tends to severely underfit. Despite their shortcomings, AIC, BIC, and corrected AIC (Hurvich and Tsai 1989) are still the preferred lag order selection tools by most practitioners (Lütkepohl, 2007; Tsay, 2013).

A drawback with such approaches is, however, that they typically require the strong assumption of a single, universal lag order that applies across all components. While this reduces the computational complexity of model selection, it has little statistical or economic justification, unnecessarily constrains the dynamic relationship between the components, and impedes forecast performance. An important motivating goal of the HLag framework is to relax this strong assumption. Gredenhoff and Karlsson (1999) show that violation of the universal lag order assumption can lead to overparameterized models or the imposition of false zero restrictions. They instead suggest considering *componentwise* specifications that allow each marginal regression to have a different lag order (sometimes referred to as an *asymmetric VAR*). One such procedure (Hsiao, 1981) starts from univariate autoregressions and sequentially adds lagged components according to Akaike's "Final Prediction Error" (Akaike, 1969). However, this requires an *a priori* ranking of components based on their perceived predictive power, which is inherently subjective. Keating (2000) offers a more general method which estimates all potential $pmax^k$ componentwise VARs and utilizes AIC/BIC for lag order selection. Such an approach is computationally intractable and standard asymptotic justifications are inapplicable if the number of components k is large. Ding and Karlsson (2014) present several specifications which allow for varying lag order within a Bayesian framework. Markov chain Monte Carlo estimation methods with spike and slab priors are proposed, but these are computationally intensive, and estimation becomes intractable in high dimensions though recent advances have been made by Giannone et al. (2017).

Given the difficulties with lag order selection in VARs, many authors have turned instead to shrinkage-based approaches, which impose sparsity, or other economically-motivated restrictions, on the parameter space to make reliable estimation tractable, and are discussed below.

2.2 Bayesian Shrinkage

Early shrinkage methods, such as Litterman (1979), take a pragmatic Bayesian perspective. Many of them (e.g., Banbura et al., 2010; Koop, 2013) apply the *Minnesota prior,* which uses natural conjugate priors to shrink the VAR toward either an intercept-only model or a vector random walk, depending on the context. The prior covariance is specified so as to incorporate the belief that a series' *own* lags are more informative than *other* lags and that lower lags are more informative than higher lags. With this prior structure, coefficients at high lags will have a prior mean of zero and a prior variance that decays with the lag. Hence, coefficients with higher lags are shrunk more toward zero. However, unlike the HLag methods but similar to ridge regression, coefficients will not be estimated as exactly zero.

The own/other HLag penalty proposed below is inspired by this Minnesota prior. It also has the propensity to prioritize own lags over other lags and to assign a greater penalty to distant lags, but it formalizes these relationships by embedding two layers of hierarchy into a convex regularization framework. One layer (within each lag vector) prioritizes own lags before other lags. Another layer (across lag vectors) penalizes distant lags more than recent lags since the former can only be included in the model if the latter are selected.

The Bayesian literature on dealing with overparametrization of VARs is rapidly growing, with many recent advances on, amongst others, improved prior choices (e.g., Carriero

et al., 2012, Giannone et al., 2015), stochastic volatility (e.g., Carriero et al., 2019), time-varying parameter estimation (e.g., Koop and Korobilis, 2013), and dimension reduction via compressing (Koop et al., 2019).

2.3 Factor Models

Factor models form another widely used class to overcome the VAR's overparameterization and have been used extensively for macroeconomic forecasting (e.g., Stock and Watson, 2002). Here, the factors serve the purpose of dimension reduction since the information contained in the original high dimensional data set is summarized—often using principal component analysis—in a small number of factors. While Factor Augmented VARs (FAVAR) (e.g., Bernanke et al., 2005) include one or more factors in addition to the observables, all observables are expressed as a weighted average of factors in Dynamic Factor Models (e.g., Forni et al., 2000).

2.4 Lasso-based Regularization

Other shrinkage approaches have incorporated the lasso (Tibshirani, 1996). Hsu et al. (2008) consider the lasso with common information criterion methods for model selection. The use of the lasso mitigates the need to conduct an exhaustive search over the space of all $2^{k^2 pmax}$ possible models but does not explicitly encourage lags to be small. HLag, in contrast, forces low lag coefficients to be selected before corresponding high lag coefficients, thereby specifically shrinking toward low lag order solutions. As will be illustrated through simulations and empirical applications, this often improves forecast performance.

To account for the VAR's inherent ordered structure, Lozano et al. (2009) use a group lasso (Yuan and Lin, 2006) penalty to group together coefficients within a common component. Song and Bickel (2011) treat each variable's own lags differently from other variables' lags (similar to the own/other Hlag penalty we propose), consider a group lasso structure and additionally down-weight higher lags via scaling the penalty parameter by an increasing function of the coefficients' lag. The authors note that the functional form of these weights is arbitrary, but the estimates are sensitive to the choice of weights. A similar *truncating lasso* penalty is proposed by Shojaie and Michailidis (2010) and refined by Shojaie et al. (2012) in the context of graphical Granger causality. However, unlike HLag, this framework requires a functional form assumption on the decay of the weights as well as a two-dimensional penalty parameter search which generally squares the computational burden.

3. Methodology

Let $\{\mathbf{y}_t \in \mathbb{R}^k\}_{t=1}^T$ denote a k-dimensional vector time series of length T. A pth order vector autoregression $\text{VAR}_k(p)$ may be expressed as a multivariate regression

$$\mathbf{y}_t = \boldsymbol{\nu} + \boldsymbol{\Phi}^{(1)}\mathbf{y}_{t-1} + \cdots + \boldsymbol{\Phi}^{(p)}\mathbf{y}_{t-p} + \mathbf{u}_t, \quad \text{for } t = 1, \ldots, T, \tag{1}$$

conditional on initial values $\{\mathbf{y}_{-(p-1)}, \ldots, \mathbf{y}_0\}$, where $\boldsymbol{\nu} \in \mathbb{R}^k$ denotes an intercept vector, $\{\boldsymbol{\Phi}^{(\ell)} \in \mathbb{R}^{k\times k}\}_{\ell=1}^p$ are lag-ℓ coefficient matrices, and $\{\mathbf{u}_t \in \mathbb{R}^k\}_{t=1}^T$ is a mean zero white noise vector time series with unspecified $k \times k$ nonsingular contemporaneous covariance matrix $\boldsymbol{\Sigma}_u$.

In the classical low-dimensional setting in which $T > kp$, one may perform least squares to fit the $\text{VAR}_k(p)$ model, minimizing

$$\sum_{t=1}^{T} \|\mathbf{y}_t - \boldsymbol{\nu} - \sum_{\ell=1}^{p} \boldsymbol{\Phi}^{(\ell)} \mathbf{y}_{t-\ell}\|_2^2 \tag{2}$$

over $\boldsymbol{\nu}$ and $\{\boldsymbol{\Phi}^{(\ell)}\}$, where $\|\mathbf{a}\|_2 = (\sum_i \mathbf{a}_i^2)^{1/2}$ denotes the Euclidean norm of a vector $\mathbf{a}$. We will find it convenient to express the VAR using compact matrix notation:

$$\begin{array}{llll} \mathbf{Y} = [\mathbf{y}_1 \ \cdots \ \mathbf{y}_T] & (k \times T); & \boldsymbol{\Phi} = [\boldsymbol{\Phi}^{(1)} \ \cdots \ \boldsymbol{\Phi}^{(p)}] & (k \times kp); \\ \mathbf{z}_t = [\mathbf{y}_{t-1}^\top \ \cdots \ \mathbf{y}_{t-p}^\top]^\top & (kp \times 1); & \mathbf{Z} = [\mathbf{z}_1 \ \cdots \ \mathbf{z}_T] & (kp \times T); \\ \mathbf{U} = [\mathbf{u}_1 \ \cdots \ \mathbf{u}_T] & (k \times T); & \mathbf{1} = [1 \ \cdots \ 1]^\top & (T \times 1). \end{array} \tag{3}$$

Equation (1) is then simply

$$\mathbf{Y} = \boldsymbol{\nu}\mathbf{1}^\top + \boldsymbol{\Phi}\mathbf{Z} + \mathbf{U},$$

and the least squares procedure (2) can be expressed as minimizing

$$\|\mathbf{Y} - \boldsymbol{\nu}\mathbf{1}^\top - \boldsymbol{\Phi}\mathbf{Z}\|_2^2$$

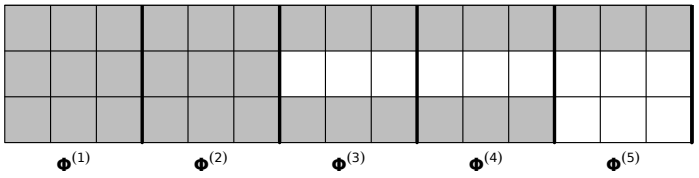
over $\boldsymbol{\nu}$ and $\boldsymbol{\Phi}$, where $\|\mathbf{A}\|_2$ denotes the Frobenius norm of the matrix $\mathbf{A}$, that is the Euclidean norm of $\text{vec}(\mathbf{A})$ (not to be mistaken for the operator norm, which does not appear in this paper).

Estimating the parameters of this model is challenging unless T is sufficiently large. Indeed, when $T > kp$ but $kp/T \approx 1$, estimation by least squares becomes imprecise. We therefore seek to incorporate reasonable structural assumptions on the parameter space to make estimation tractable for moderate to small T. Multiple authors have considered using the lasso penalty, building in the assumption that the lagged coefficient matrices $\boldsymbol{\Phi}^{(\ell)}$ are sparse (e.g., Song and Bickel, 2011; Davis et al., 2016; Hsu et al., 2008); theoretical work has elucidated how such structural assumptions can lead to better estimation performance even when the number of parameters is large (e.g., Basu and Michailidis, 2015, Melnyk and Banerjee, 2016, Lin and Michailidis, 2017). In what follows, we define a class of sparsity patterns, which we call hierarchical lag or HLag structures, that arises in the context of multivariate time series.

3.1 HLag: Hierarchical Lag Structures

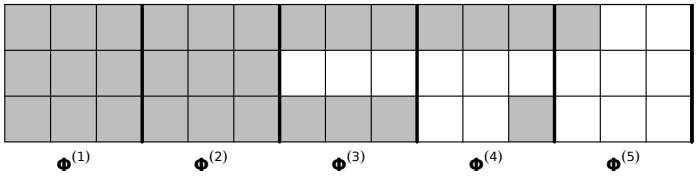
In Equation (1), the parameter $\boldsymbol{\Phi}_{ij}^{(\ell)}$ controls the dynamic dependence of the ith component of $\mathbf{y}_t$ on the jth component of $\mathbf{y}_{t-\ell}$. In describing HLag structures, we will use the following notational convention: for $1 \leq \ell \leq p$, let

$$\begin{aligned} \boldsymbol{\Phi}^{(\ell:p)} &= [\boldsymbol{\Phi}^{(\ell)} \ \cdots \ \boldsymbol{\Phi}^{(p)}] \in \mathbb{R}^{k \times k(p-\ell+1)} \\ \boldsymbol{\Phi}_i^{(\ell:p)} &= [\boldsymbol{\Phi}_i^{(\ell)} \ \cdots \ \boldsymbol{\Phi}_i^{(p)}] \in \mathbb{R}^{1 \times k(p-\ell+1)} \\ \boldsymbol{\Phi}_{ij}^{(\ell:p)} &= [\boldsymbol{\Phi}_{ij}^{(\ell)} \ \cdots \ \boldsymbol{\Phi}_{ij}^{(p)}] \in \mathbb{R}^{1 \times (p-\ell+1)}. \end{aligned}$$



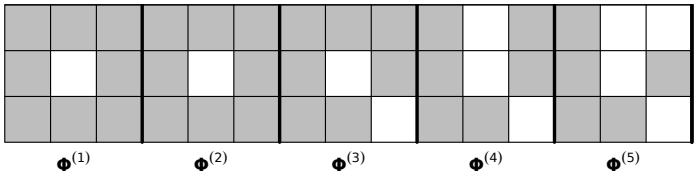
$$\mathbf{L}^C = \begin{pmatrix} 5 & 5 & 5 \\ 2 & 2 & 2 \\ 4 & 4 & 4 \end{pmatrix}$$

Figure 1: A componentwise (C) HLag active set structure (shaded): $\text{HLag}_3^C(5)$.



$$\mathbf{L}^O = \begin{pmatrix} 5 & 4 & 4 \\ 2 & 2 & 2 \\ 3 & 3 & 4 \end{pmatrix}$$

Figure 2: An own-other (O) HLag active set structure (shaded): $\text{HLag}_3^O(5)$.



$$\mathbf{L}^E = \begin{pmatrix} 5 & 3 & 4 \\ 5 & 0 & 5 \\ 5 & 5 & 2 \end{pmatrix}$$

Figure 3: An elementwise (E) HLag active set structure (shaded): $\text{HLag}_3^E(5)$.

Consider the $k \times k$ matrix of *elementwise* coefficient lags $\mathbf{L}$ defined by

$$\mathbf{L}_{ij} = \max\{\ell : \mathbf{\Phi}_{ij}^{(\ell)} \neq 0\},$$

in which we define $\mathbf{L}_{ij} = 0$ if $\mathbf{\Phi}_{ij}^{(\ell)} = 0$ for all $\ell = 1, \ldots, p$. Therefore, each $\mathbf{L}_{ij}$ denotes the maximal coefficient lag (maxlag) for component j in the regression model for component i. In particular, $\mathbf{L}_{ij}$ is the smallest ℓ such that $\mathbf{\Phi}_{ij}^{([\ell+1]:p)} = \mathbf{0}$. Note that the maxlag matrix $\mathbf{L}$ is not symmetric, in general. There are numerous HLag structures that one can consider within the context of the $\text{VAR}_k(p)$ model. The simplest such structure is that $\mathbf{L}_{ij} = L$ for all i and j, meaning that there is a *universal* (U) maxlag that is shared by every pair of components. Expressed in terms of Equation (1), this would say that $\mathbf{\Phi}^{([L+1]:p)} = \mathbf{0}$ and that $\mathbf{\Phi}_{ij}^{(L)} \neq 0$ for all $1 \leq i, j \leq k$. While the methodology we introduce can be easily extended to this and many other potential HLag structures, in this paper we focus on the following three fundamental structures.

1. **Componentwise (C).** A componentwise HLag structure allows each of the k marginal equations from (1) to have its own maxlag, but all components within each equation must share the same maximal lag:

   $$\mathbf{L}_{ij} = L_i \ \ \forall j, \ \text{ for } \ i = 1, \ldots k.$$

   Hence in Equation (1), this implies $\mathbf{\Phi}_i^{([L_i+1]:p)} = \mathbf{0}$ and $\mathbf{\Phi}_{ij}^{(L_i)} \neq 0$ for all i and j. This componentwise HLag active set structure (shaded) is illustrated in Figure 1.

2. **Own-Other (O).** The own-other HLag structure is similar to the componentwise one, but with an added within-lag hierarchy that imposes the mild assumption that a series' own lags ($i = j$) are more informative than other lags ($i \neq j$). Thus, diagonal elements are prioritized before off-diagonal elements within each lag, componentwise (i.e., row-wise). In particular,

$$\mathbf{L}_{ij} = L_i^{other} \text{ for } i \neq j \text{ and } \mathbf{L}_{ii} \in \{L_i^{other}, L_i^{other} + 1\}, \text{ for } i = 1, \dots k.$$

   This HLag structure allows each component of $\mathbf{y}_t$ to have longer range lagged self-dependence than lagged cross-dependencies. This own-other HLag structure is illustrated in Figure 2.

3. **Elementwise (E).** Finally, we consider a completely flexible structure in which the elements of $\mathbf{L}$ have no stipulated relationships. Figure 3 illustrates this elementwise HLag structure.

In the next section, we introduce the proposed class of HLag estimators aimed at estimating $\text{VAR}_k(p)$ models while shrinking the elements of $\mathbf{L}$ towards zero by incorporating the three HLag structures described above.

3.2 HLag: Hierarchical Group Lasso for Lag Structured VARs

In this section, we introduce convex penalties specifically tailored for attaining the three lag structures presented in the previous section. Our primary modeling tool is the hierarchical group lasso (Zhao et al., 2009; Yan and Bien, 2017), which is a group lasso (Yuan and Lin, 2006) with a nested group structure. The group lasso is a sum of (unsquared) Euclidean norms and is used in statistical modeling as a penalty to encourage groups of parameters to be set to zero simultaneously. Using nested groups leads to hierarchical sparsity constraints in which one set of parameters being zero implies that another set is also zero. This penalty has been applied to multiple statistical problems including regression models with interactions (Zhao et al., 2009; Jenatton et al., 2010; Radchenko and James, 2010; Bach et al., 2012; Bien et al., 2013; Lim and Hastie, 2015; Haris et al., 2016; She et al., 2018), covariance estimation (Bien et al., 2016), additive modeling (Lou et al., 2016), and time series (Tibshirani and Suo, 2016). This last work focuses on transfer function estimation, in this case scalar regression with multiple time-lagged covariates whose coefficients decay with lag.

For each hierarchical lag structure presented above, we propose an estimator based on a convex optimization problem:

$$\min_{\boldsymbol{\nu}, \boldsymbol{\Phi}} \left\{ \frac{1}{2T} \|\mathbf{Y} - \boldsymbol{\nu}\mathbf{1}^\top - \boldsymbol{\Phi}\mathbf{Z}\|_2^2 + \lambda \mathcal{P}_{\text{HLag}}(\boldsymbol{\Phi}) \right\}, \tag{4}$$

in which $\mathcal{P}_{\text{HLag}}$ denotes a hierarchical lag group (HLag) penalty function. We propose three such penalty functions: componentwise; own-other; and elementwise; and discuss their relative merits.

1. **HLag**^{C} aims for a *componentwise* hierarchical lag structure and is defined by

$$\mathcal{P}^C_{\text{HLag}}(\mathbf{\Phi}) = \sum_{i=1}^{k}\sum_{\ell=1}^{p} \|\mathbf{\Phi}_i^{(\ell:p)}\|_2, \tag{5}$$

in which $\|\mathbf{A}\|_2$ denotes the Euclidean norm of vec($\mathbf{A}$), for a matrix $\mathbf{A}$. As the penalty parameter $\lambda \geq 0$ is increased, we have $\hat{\mathbf{\Phi}}_i^{(\ell:p)} = \mathbf{0}$ for more i, and for smaller ℓ. This componentwise HLag structure builds in the condition that if $\hat{\mathbf{\Phi}}_i^{(\ell)} = 0$, then $\hat{\mathbf{\Phi}}_i^{(\ell')} = 0$ for all $\ell' > \ell$, for each $i = 1, \ldots, k$. This structure favors lower maxlag models componentwise, rather than simply giving sparse $\mathbf{\Phi}$ estimates with no particular structure.

2. **HLag**^{O} aims for a *own-other* hierarchical lag structure and is defined by

$$\mathcal{P}^O_{\text{HLag}}(\mathbf{\Phi}) = \sum_{i=1}^{k}\sum_{\ell=1}^{p} \left[\|\mathbf{\Phi}_i^{(\ell:p)}\|_2 + \|(\mathbf{\Phi}_{i,-i}^{(\ell)}, \mathbf{\Phi}_i^{([\ell+1]:p)})\|_2\right], \tag{6}$$

in which $\mathbf{\Phi}_{i,-i}^{(\ell)} = \{\mathbf{\Phi}_{ij}^{(\ell)} : j \neq i\}$, and where we adopt the convention that $\mathbf{\Phi}_i^{([p+1]:p)} = \mathbf{0}$. The first term in this penalty is identical to that of (5). The difference is the addition of the second penalty term, which is just like the first except that it omits $\mathbf{\Phi}_{ii}^{(\ell)}$. This penalty allows sparsity patterns in which the influence of component i on itself may be nonzero at lag ℓ even though the influence of other components is thought to be zero at that lag. This model ensures that, for all $\ell' > \ell$, $\hat{\mathbf{\Phi}}_i^{(\ell)} = \mathbf{0}$ implies $\hat{\mathbf{\Phi}}_i^{(\ell')} = \mathbf{0}$ and $\hat{\mathbf{\Phi}}_{ii}^{(\ell)} = \mathbf{0}$ implies $\hat{\mathbf{\Phi}}_{i,-i}^{(\ell'+1)} = \mathbf{0}$. This accomplishes the desired own-other HLag structure such that $\mathbf{L}_{i,-i} = L_i^{other}\mathbf{1}_{k-1}$ and $\mathbf{L}_{ii} \in \{L_i^{other}, L_i^{other}+1\}$, componentwise.

3. **HLag**^{E} aims for an *elementwise* hierarchical lag structure and is defined by

$$\mathcal{P}^E_{\text{HLag}}(\mathbf{\Phi}) = \sum_{i=1}^{k}\sum_{j=1}^{k}\sum_{\ell=1}^{p} \|\mathbf{\Phi}_{ij}^{(\ell:p)}\|_2. \tag{7}$$

Here, each of the k^2 pairs of components can have its own maxlag, such that $\mathbf{\Phi}_{ij}^{(\ell:p)} = \mathbf{0}$ may occur for different values of ℓ for each pair i and j. While this model is the most flexible of the three, it also borrows the least strength across the different components. When $\mathbf{L}_{ij}$ differ for all i and j, we expect this method to do well, whereas when, for example $\mathbf{L}_{ij} = L_i$, we expect it to be inefficient relative to (5).

Since all three penalty functions are based on hierarchical group lasso penalties, a unified computational approach to solve each is detailed in Section 4. First, we discuss theoretical properties of HLag.

3.3 Theoretical Properties

We build on Basu and Michailidis (2015) to analyze theoretical properties of high-dimensional VARs estimated by HLag. Consider a fixed realization of $\{\mathbf{y}_t\}_{t=-(p-1)}^{T}$ generated from the VAR model (1) with fixed autoregressive order p and $\mathbf{u}_t \overset{iid}{\sim} N(\mathbf{0}, \mathbf{\Sigma}_u)$. Denote the corresponding true maxlag matrix by $\mathbf{L}$. We make the following assumptions.

Assumption 1 *The VAR model is stable, such that* $det\{\mathbf{\Phi}(z)\} \neq 0$ *for all* $\{z \in \mathbb{C} : |z| \leq 1\}$, *where*

$$\mathbf{\Phi}(z) = \mathbf{I} - \mathbf{\Phi}^{(1)}z - \mathbf{\Phi}^{(2)}z^2 - \ldots - \mathbf{\Phi}^{(p)}z^p;$$

and the error covariance matrix $\mathbf{\Sigma}_u$ *is positive definite such that its minimum eigenvalue* $\Lambda_{min}(\mathbf{\Sigma}_u) > 0$ *and its maximum eigenvalue* $\Lambda_{max}(\mathbf{\Sigma}_u) < \infty$.

These assumptions are standard in the time series literature. Define the following two measures of stability of the VAR process, which will be useful for our theoretical analysis (see Basu and Michailidis, 2015 for more detail)

$$\mu_{\min}(\mathbf{\Phi}) = \min_{|z|=1} \Lambda_{\min}(\mathbf{\Phi}^*(z)\mathbf{\Phi}(z)), \text{ and } \mu_{\max}(\mathbf{\Phi}) = \max_{|z|=1} \Lambda_{\max}(\mathbf{\Phi}^*(z)\mathbf{\Phi}(z)),$$

where $\mathbf{\Phi}^*(\cdot)$ denotes the conjugate transpose of a complex matrix.

We derive a bound on the in-sample prediction error. Define the in-sample, one-step-ahead mean squared forecast error to be

$$\text{MSFE}_{\text{in}} = \mathbb{E}\left[\frac{1}{T}\|\mathbf{Y} - \hat{\mathbf{\Phi}}\mathbf{Z}\|_2^2 \mid \mathbf{Z}\right] = \text{tr}(\mathbf{\Sigma}_u) + \frac{1}{T}\sum_{t=1}^{T}\left\|\sum_{\ell=1}^{p}(\hat{\mathbf{\Phi}}^{(\ell)} - \mathbf{\Phi}^{(\ell)})\mathbf{y}_{t-\ell}\right\|_2^2,$$

with $\mathbf{Y}, \mathbf{\Phi}$ and $\mathbf{Z}$ as defined in equation (3). While $\text{tr}(\mathbf{\Sigma}_u)$ is the irreducible error, an unavoidable part of the forecast error, a good estimator of the autoregressive parameters should allow us to control the size of the second term. In Theorem 1, we provide such a bound on the in-sample prediction error for the most flexible HLag method, namely elementwise HLag.

Theorem 1 *Suppose* $T > \max\{25\log(pk^2), 4\}$ *and* $pk^2 \gg 1$. *Under Assumption 1 and taking all lag coefficients to be bounded in absolute value by* M, *we choose* $\lambda \asymp v(\mathbf{\Phi}, \mathbf{\Sigma}_u)\sqrt{\log(pk^2)/T}$, *where* $v(\mathbf{\Phi}, \mathbf{\Sigma}_u) = \Lambda_{max}(\mathbf{\Sigma}_u)\left(1 + \frac{1+\mu_{max}(\mathbf{\Phi})}{\mu_{min}(\mathbf{\Phi})}\right)$. *Then, with probability at least* $1 - \frac{12}{(pk^2)^{23/2}}$,

$$\frac{1}{T}\sum_{t=1}^{T}\left\|\sum_{\ell=1}^{p}(\hat{\mathbf{\Phi}}^{(\ell)} - \mathbf{\Phi}^{(\ell)})\mathbf{y}_{t-\ell}\right\|_2^2 \lesssim Mv(\mathbf{\Phi}, \mathbf{\Sigma}_u)\sqrt{\frac{\log(pk^2)}{T}}\sum_{i=1}^{k}\sum_{j=1}^{k}L_{ij}^{3/2},$$

where $\hat{\mathbf{\Phi}}$ *is the elementwise HLag estimator with* $pmax = p$.

The proof of Theorem 1 is included in Section A of the appendix. Theorem 1 establishes in-sample prediction consistency in the high-dimensional regime $\log(pk^2)/T \to 0$. Hence, the same rate is obtained as for i.i.d. data, modulo a "price" paid for dependence. The temporal and cross-sectional dependence affects the rate through the internal parameters $\Lambda_{\max}(\mathbf{\Sigma}_u), \mu_{\min}(\mathbf{\Phi})$ and $\mu_{\max}(\mathbf{\Phi})$.

While Theorem 1 is derived under the assumption that p is the true order of the VAR, the results hold even if p is replaced by any upper bound $pmax$ on the true order since the $\text{VAR}_k(p)$ can be viewed as a $\text{VAR}_k(pmax)$ with $\mathbf{\Phi}^{(\ell)} = \mathbf{0}$ for $\ell > p$, see Basu and Michailidis (2015). The convergence rate then becomes $\sqrt{\log(pmax \cdot k^2)/T}$ instead of $\sqrt{\log(pk^2)/T}$.

The bound includes terms of the form $L_{ij}^{3/2}$. The 3/2 exponent can be removed if one adopts a more complicated weighting scheme (see e.g., Jenatton et al., 2011a, Bien et al., 2016), which would avoid high order lag coefficients from being aggressively shrunken. However, in the context of VAR estimation, we find through simulation experiments that this aggressive shrinkage is in fact beneficial (see Section C.3 of the appendix).

4. Optimization Algorithm

We begin by noting that since the intercept $\boldsymbol{\nu}$ does not appear in the penalty terms, it can be removed if we replace $\mathbf{Y}$ by $\mathbf{Y}(\mathbf{I}_T - \frac{1}{T}\mathbf{1}\mathbf{1}^\top)$ and $\mathbf{Z}$ by $\mathbf{Z}(\mathbf{I}_T - \frac{1}{T}\mathbf{1}\mathbf{1}^\top)$. All three optimization problems are of the form

$$\min_{\boldsymbol{\Phi}} \left\{ \frac{1}{2T}\|\mathbf{Y} - \boldsymbol{\Phi}\mathbf{Z}\|_2^2 + \lambda \sum_{i=1}^{k}\sum_{\ell=1}^{p} \Omega_i(\boldsymbol{\Phi}_i^{(\ell:p)}) \right\}, \tag{8}$$

and (5), (6), and (7) only differ by the form of the norm Ω_i. A key simplification is possible by observing that the objective above decouples across the rows of $\boldsymbol{\Phi}$:

$$\min_{\boldsymbol{\Phi}} \sum_{i=1}^{k} \left[\frac{1}{2T}\|\mathbf{Y}_i - \boldsymbol{\Phi}_i\mathbf{Z}\|_2^2 + \lambda \sum_{\ell=1}^{p} \Omega_i(\boldsymbol{\Phi}_i^{(\ell:p)}) \right],$$

in which $\mathbf{Y}_i \in \mathbb{R}^{1\times T}$ and $\boldsymbol{\Phi}_i = \boldsymbol{\Phi}_i^{(1:p)} \in \mathbb{R}^{1\times kp}$. Hence, Equation (8) can be solved in parallel by solving the "one-row" subproblem

$$\min_{\boldsymbol{\Phi}_i} \left\{ \frac{1}{2T}\|\mathbf{Y}_i - \boldsymbol{\Phi}_i\mathbf{Z}\|_2^2 + \lambda \sum_{\ell=1}^{p} \Omega_i(\boldsymbol{\Phi}_i^{(\ell:p)}) \right\}.$$

Jenatton et al. (2011b) show that hierarchical group lasso problems can be efficiently solved via the proximal gradient method. This procedure can be viewed as an extension of traditional gradient descent methods to nonsmooth objective functions. Given a convex objective function of the form $f_i(\boldsymbol{\Phi}_i) = \mathcal{L}_i(\boldsymbol{\Phi}_i) + \lambda\Omega_i^*(\boldsymbol{\Phi}_i)$, where $\mathcal{L}_i$ is differentiable with a Lipschitz continuous gradient, the proximal gradient method produces a sequence $\hat{\boldsymbol{\Phi}}_i[1], \hat{\boldsymbol{\Phi}}_i[2], \ldots$ with the guarantee that

$$f_i(\hat{\boldsymbol{\Phi}}_i[m]) - \min_{\boldsymbol{\Phi}_i} f_i(\boldsymbol{\Phi}_i)$$

is $O(1/m)$ (cf. Beck and Teboulle 2009). For $m = 1, 2, \ldots$, its update is given by

$$\hat{\boldsymbol{\Phi}}_i[m] = \mathrm{Prox}_{s_m\lambda\Omega_i^*}\left(\hat{\boldsymbol{\Phi}}_i[m-1] - s_m\nabla\mathcal{L}(\hat{\boldsymbol{\Phi}}_i[m-1])\right),$$

where s_m is an appropriately chosen step size and $\mathrm{Prox}_{s_m\lambda\Omega_i^*}$ is the proximal operator of the function $s_m\lambda\Omega_i^*(\cdot)$, which is evaluated at the gradient step we would take if we were minimizing $\mathcal{L}_i$ alone. The proximal operator is defined as the unique solution of a convex optimization problem involving Ω_i^* but not $\mathcal{L}_i$:

$$\mathrm{Prox}_{s_m\lambda\Omega_i^*}(u) = \operatorname*{argmin}_{v} \left\{ \frac{1}{2}\|u - v\|_2^2 + s_m\lambda\Omega_i^*(v) \right\}. \tag{9}$$

The proximal gradient method is particularly effective when the proximal operator can be evaluated efficiently. In our case, $\Omega_i^*(\mathbf{\Phi}_i) = \sum_{\ell=1}^{p} \Omega_i(\mathbf{\Phi}_i^{(\ell:p)})$ is a sum of hierarchically nested Euclidean norms. Jenatton et al. (2011b) show that for such penalties, the proximal operator has essentially a closed form solution, making it extremely efficient. It remains to note that $\mathcal{L}_i(\mathbf{\Phi}_i) = \frac{1}{2T}\|\mathbf{Y}_i - \mathbf{\Phi}_i\mathbf{Z}\|_2^2$ has gradient $\nabla\mathcal{L}_i(\mathbf{\Phi}_i) = -\frac{1}{T}(\mathbf{Y}_i - \mathbf{\Phi}_i\mathbf{Z})\mathbf{Z}^\top$ and that the step size s_m can be determined adaptively through a backtracking procedure or it can be set to the Lipschitz constant of $\nabla\mathcal{L}_i(\mathbf{\Phi}_i)$, which in this case is $\sigma_1(\mathbf{Z})^{-2}$ (where $\sigma_1(\mathbf{Z})$ denotes the largest singular value of $\mathbf{Z}$).

We use an accelerated version of the proximal gradient method which leads to a faster convergence rate and improved empirical performance with minimal additional overhead. Our particular implementation is based on Algorithm 2 of Tseng (2008). It repeats, for $m = 1, 2, \ldots$ to convergence,

$$\hat{\phi} \leftarrow \hat{\mathbf{\Phi}}_i[m-1] + \theta_{m-1}(\theta_{m-2}^{-1} - 1)\left(\hat{\mathbf{\Phi}}_i[m-1] - \hat{\mathbf{\Phi}}_i[m-2]\right)$$
$$\hat{\mathbf{\Phi}}_i[m] \leftarrow \mathrm{Prox}_{s_m\lambda\Omega_i^*}\left(\hat{\phi} - s_m\nabla\mathcal{L}_i(\hat{\phi})\right),$$

with $\theta_m = 2/(m+2)$ as in Tseng (2008) and converges at rate $1/m^2$ (compared to the unaccelerated proximal gradient method's $1/m$ rate). Alternatively, one could set $\theta_m = \frac{1}{2}\left(\sqrt{\theta_{m-1}^4 + 4\theta_{m-1}^2} - \theta_{m-1}^2\right)$ which is essentially the Fast Iterative Soft-Thresholding Algorithm developed by Beck and Teboulle (2009). We verified that our findings in the simulation study are unaffected by this choice.

Our full procedure is detailed in Algorithm 1 and is applicable to all three HLag estimators. Note that the algorithm requires an initial value $\hat{\mathbf{\Phi}}[0]$. As is standard in the regularization literature (e.g., Friedman et al., 2017), we use "warm starts". We solve Algorithm 1 for a grid of penalty values starting at $\lambda_{\max}$, the smallest value of the regularization parameter in which all coefficients will be zero. For each smaller value of λ along this grid, we use the previous solution as a "warm start" ($\hat{\mathbf{\Phi}}[0]$) to run Algorithm 1 with the new λ-value. A key advantage of our HLag estimates being solutions to a convex optimization problem is that the algorithms are stable and not sensitive to the choice of initialization (Beck and Teboulle, 2009). As stopping criterion, we use $||\hat{\phi} - \hat{\mathbf{\Phi}}_i[m]||_\infty \leq \epsilon$, while one could also use $||\hat{\mathbf{\Phi}}_i[m] - \hat{\mathbf{\Phi}}_i[m-1]||_\infty \leq \epsilon$. We opt for the former since we have numerically observed in our simulation experiments that considerably less iterations are needed without affecting accuracy.

The algorithms for these methods differ only in the evaluation of their proximal operators (since each method has a different penalty Ω_i^*). However, all three choices of Ω_i^* correspond to hierarchical group lasso penalties, allowing us to use the result of Jenatton et al. (2011b), which shows that the proximal operator has a remarkably simple form. We write these three problems generically as

$$\hat{\mathbf{x}} = \underset{\mathbf{x}}{\mathrm{argmin}} \left\{ \frac{1}{2}\|\mathbf{x} - \tilde{\mathbf{x}}\|_2^2 + \lambda \sum_{h=1}^{H} w_h \|\mathbf{x}_{g_h}\|_2 \right\}, \tag{10}$$

where $g_1 \subset \cdots \subset g_H$. The key observation in Jenatton et al. (2011b) is that the dual of the proximal problem (9) can be solved exactly in a *single pass* of blockwise coordinate descent.

Algorithm 1 General algorithm for HLag with penalty Ω_i^*

Require: $\mathbf{Y}, \mathbf{Z}, \hat{\mathbf{\Phi}}[0], \lambda, \epsilon = 10^{-4}$
  $\hat{\mathbf{\Phi}}[1] \leftarrow \hat{\mathbf{\Phi}}[0]$; $\hat{\mathbf{\Phi}}[2] \leftarrow \hat{\mathbf{\Phi}}[0]$
  $s \leftarrow \sigma_1(Z)^{-2}$
  for $i = 1, \ldots, k$ **do**
    for $m = 3, 4, \ldots$ **do**
      $\hat{\boldsymbol{\phi}} \leftarrow \hat{\mathbf{\Phi}}_i[m-1] + \frac{m-2}{m+1}\left(\hat{\mathbf{\Phi}}_i[m-1] - \hat{\mathbf{\Phi}}_i[m-2]\right)$
      $\hat{\mathbf{\Phi}}_i[m] \leftarrow \text{Prox}_{s\lambda\Omega_i^*}\left(\hat{\boldsymbol{\phi}} + \frac{s}{T} \cdot (\mathbf{Y}_i - \hat{\boldsymbol{\phi}}\mathbf{Z})\mathbf{Z}^\top\right)$
      if $\|\hat{\boldsymbol{\phi}} - \hat{\mathbf{\Phi}}_i[m]\|_\infty \leq \epsilon$ **then**
        break
      end if
    end for
  end for
  return $\hat{\mathbf{\Phi}}[m]$

Algorithm 2 Solving Problem (10)

Require: $\tilde{\mathbf{x}}, \lambda, w_1, \ldots, w_H$
  $\mathbf{r} \leftarrow \tilde{\mathbf{x}}$
  for $h = 1, \ldots, H$ **do**
    $\mathbf{r}_{g_h} \leftarrow (1 - \lambda w_h / \|\mathbf{r}_{g_h}\|_2)_+ \mathbf{r}_{g_h}$
  end for
  return $\mathbf{r}$ as the solution $\hat{\mathbf{x}}$.

By strong duality, this solution to the dual provides us with a solution to problem (9). The updates of each block are extremely simple, corresponding to a groupwise-soft-thresholding operation. Algorithm 2 shows the solution to (10), which includes all three of our penalties as special cases.

Selection of the penalty parameters. While some theoretical results on the choice of penalty parameters are available in the literature (Basu and Michailidis, 2015), such theoretical results can not be used in practice since the penalty parameter's value depends on properties of the underlying model that are not observable. For this reason, we use cross validation, one of the standard approaches to penalty parameter selection.

Following Friedman et al. (2010), the grid of penalty values is constructed by starting with $\lambda_{\max}$, an estimate of the smallest value in which all coefficients are zero, then decrementing in log linear increments. The grid bounds are detailed in the appendix of Nicholson et al. (2017). The HLag methods rely on a single tuning parameter λ in equation (8). Our penalty parameter search over a one-dimensional grid is much less expensive than the search over a multi-dimensional grid as needed for the lag-weighted lasso (Song and Bickel, 2011). To accommodate the time series nature of our data, we select the penalty parameters using the cross-validation approach utilized by Song and Bickel (2011) and Banbura et al. (2010). Given an evaluation period $[T_1, T_2]$, we use one-step-ahead *mean-squared forecast*

error (MSFE) as a cross-validation score:

$$MSFE(T_1, T_2) = \frac{1}{k(T_2 - T_1)} \sum_{i=1}^{k} \sum_{t=T_1}^{T_2-1} (\hat{y}_{i,t+1} - y_{i,t+1})^2, \tag{11}$$

with $\hat{y}_{i,t+1}$ representing the forecast for time $t+1$ and component i based on observing the series up to time t. If multi-step ahead forecast horizons are desired, we can simply substitute (11) with our desired forecast horizon h. Since this penalty search requires looping over many time points, we have coded most of the HLag methods in `C++` to increase computational efficiency.

5. Simulation Study

We compare the proposed HLag methods with 13 competing approaches: (i) AIC-VAR: least squares estimation of the VAR and selection of a universal lag order ℓ using AIC, (ii) BIC-VAR: same as in (i) but lag order selection using BIC, (iii) Lasso-VAR: estimation of the VAR using an L_1-penalty, (iv) Lag-weighted (LW) Lasso-VAR: estimation of the VAR using a weighted L_1-penalty, which applies greater regularization to higher order lags, (v) BGR-BVAR: Bayesian VAR of Banbura et al. (2010), (vi) GLP-BVAR: Bayesian VAR of Giannone et al. (2015), (vii) CCM-BVAR: Bayesian VAR of Carriero et al. (2019) (viii) DFM: Dynamic Factor Model (see e.g., Forni et al., 2000), (ix) FAVAR: Factor Augmented VAR (Bernanke et al., 2005) (x) VAR(1): least squares estimation of a VAR(1) (xi) AR: univariate autoregressive model, (xii) Sample mean: intercept-only model, (xiii) Random walk: vector random walk model. The comparison methods are detailed in Section B of the appendix.

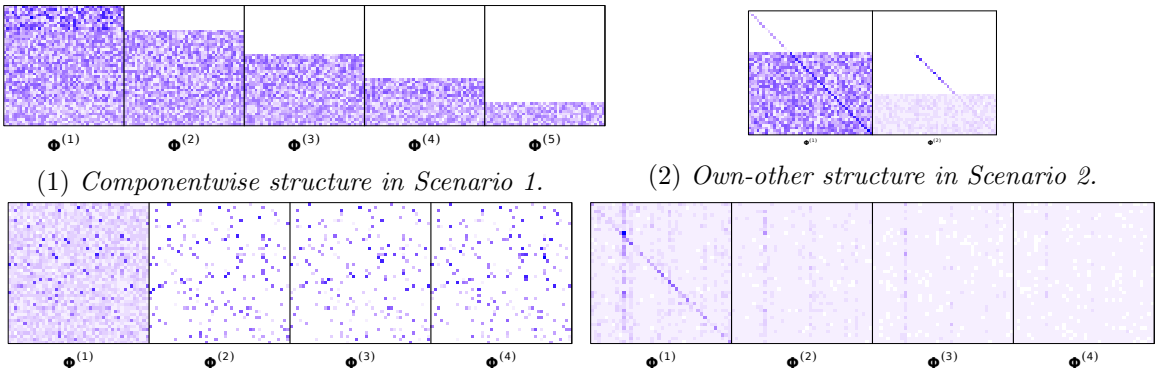
5.1 Forecast Comparisons

To demonstrate the efficacy of the HLag methods in applications with various lag structures, we evaluate the proposed methods under four simulation scenarios.

In Scenarios 1-3, we take $k = 45$ components, a series length of $T = 100$ and simulate from a VAR with the respective HLag structures: componentwise, own-other, and elementwise. In this section, we focus on simulation scenarios where the sample size T is small to moderate compared to the number of parameters to be estimated ($pmax \cdot k^2 + k$). We investigate the impact of increasing the time series length in Section C.4 of the appendix. The coefficient matrices used in these scenarios are depicted in Figure 4, panel (1)-(3) respectively.

In Scenario 4, we consider a data generating process (DGP) with $k = 40$ and $T = 195$ that does not a priori favor the HLag approaches vis-a-vis the competing approaches but follows the "data-based Monte Carlo method" (Ho and Sorensen, 1996) to make the simulation setting robust to arbitrary DGPs. This DGP does not have any special lag structure; all variables in all equations have $p = 4$ non-zero lags, as can be seen from Figure 4, panel (4).

All simulations are generated from stationary coefficient matrices. Full details on each simulation design together with the steps taken to ensure the stationarity of the simulation structures are given in Sections C.1 and C.2 of the appendix. In each scenario, the error



(1) *Componentwise structure in Scenario 1.*

(2) *Own-other structure in Scenario 2.*

(3) *Elementwise structure in Scenario 3.*

(4) *Data-based structure in Scenario 4.*

Figure 4: Sparsity patterns (and magnitudes) of the HLag based simulation scenarios. Darker shading indicates coefficients that are larger in magnitude.

covariance is taken to be $\mathbf{\Sigma}_u = 0.01 \cdot \mathbf{I}_k$. We investigate the sensitivity of our results to various choices of error covariance in Section C.5 of the appendix. To reduce the influence of initial conditions on the DGPs, the first 500 observations were discarded as burn-in for each simulation run. We run $M = 500$ simulations in each scenario.

Forecast performance measure. We focus on the problem of obtaining reliable point forecasts. To evaluate how well our methods and their competitors do in the context of providing such point forecasts, we measure their performance in terms of out-of-sample point forecast accuracy and choose mean squared forecast error as our main measure of performance. We generate time series of length T, fit the models to the first $T-1$ observations and use the last observation to compute the one-step-ahead *mean squared forecast error*

$$MSFE = \frac{1}{kM} \sum_{s=1}^{M} \sum_{i=1}^{k} (y_{i,T}^{(s)} - \widehat{y}_{i,T}^{(s)})^2,$$

with $y_{i,T}^{(s)}$ the value of component time series i at the time point T in the s^{th} simulation run, and $\widehat{y}_{i,T}^{(s)}$ is its predicted value.

Figure 5 gives the forecast performance of the methods in Scenarios 1-4. Concerning the VAR-based methods, we report the results for known ($p = 5$ in Scenario 1, $p = 2$ in Scenario 2 and $p = 4$ in Scenario 3 and 4) maximal lag order. We first discuss these results and then summarize the differences in results when the maximal lag order is unknown, for which we take $pmax = 12$.

Scenario 1: Componentwise HLag. Componentwise and own-other HLag perform best, which is to be expected since both are geared explicitly toward Scenario 1's lag structure. Elementwise HLag outperforms the lag-weighted lasso, and both do better than the lasso. Among the Bayesian methods, the BGR and CCM approaches are competitive to elementwise HLag, whereas the GLP approach is not. All Bayesian methods perform significantly worse (as confirmed with paired t-tests) than componentwise and own-other HLag. The factor models are not geared towards the DGP of Scenario 1: They select around five factors,

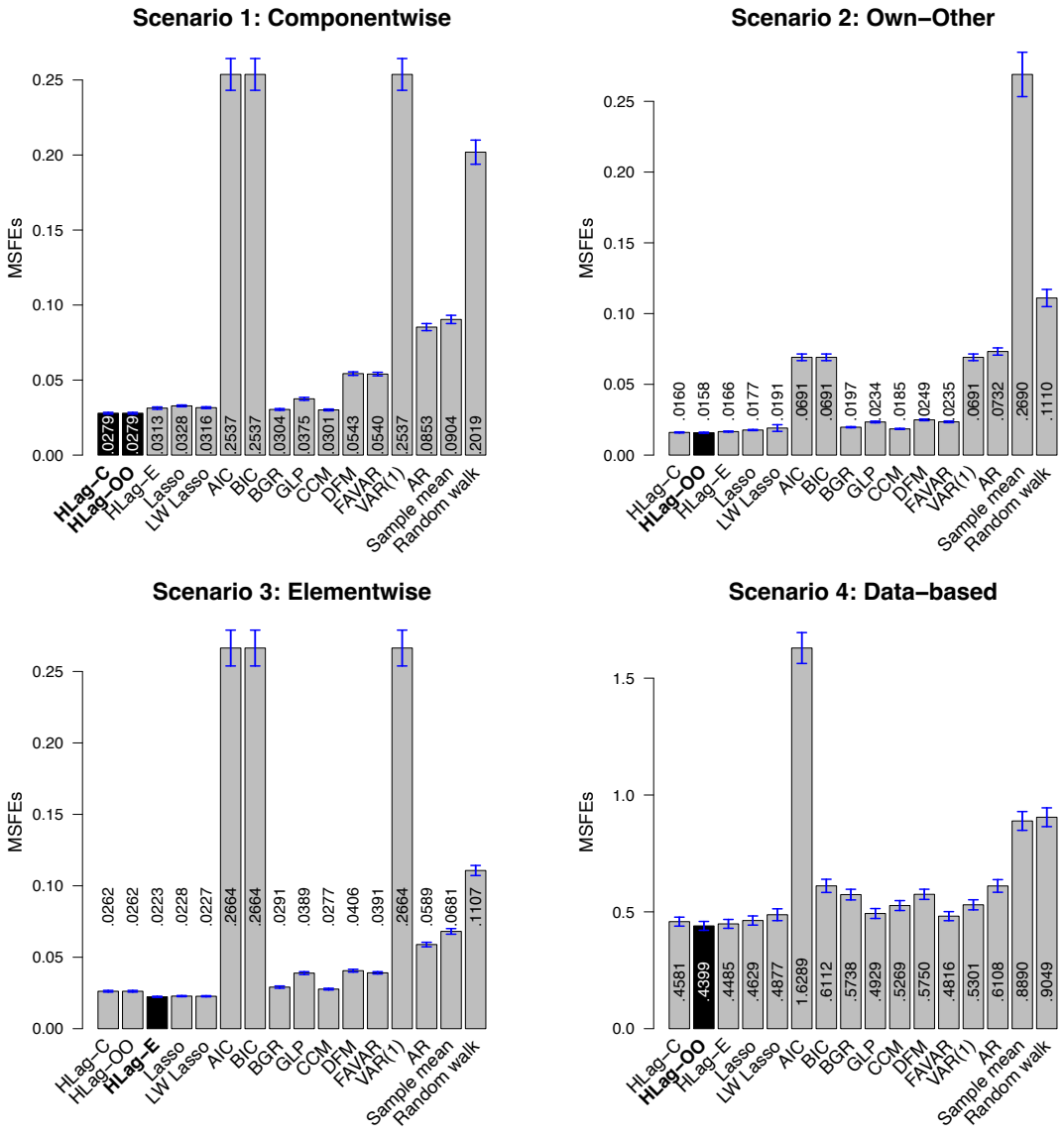


Figure 5: Out-of-sample mean squared forecast error for VARs in Scenario 1 to 4. Error bars of length two standard errors are in blue; the best performing method is in black.

on average, in their attempt to capture the time series dynamics and are not competitive to HLag. Regarding lag order selection with AIC/BIC, we can not estimate the VAR model for $\ell > 1$ with least squares, thus for a simple benchmark we instead estimate a $VAR_k(1)$ by least squares. Despite the explicit orientation toward modeling recent behavior in the $\text{VAR}_{45}(1)$ model, it suffers both because it misses important longer range lag coefficients and because it is an unregularized estimator of $\mathbf{\Phi}^{(1)}$ and therefore has high variance. The

univariate AR benchmark also suffers because it misses the dynamics among the time series: its MSFE is more than twice as large as the MSFEs of the HLag methods.

Scenario 2: Own-other HLag. All three HLag methods perform significantly better than the competing methods. As one would expect, own-other HLag achieves the best forecasting performance, with componentwise and elementwise HLag performing only slightly worse. As with the previous scenario, the least-squares approaches are not competitive.

Scenario 3: Elementwise HLag. As expected, elementwise HLag outperforms all others. The lag-weighted lasso outperforms componentwise and own-other HLag, which is not surprising as it is designed to accommodate this type of structure in a more crude manner than elementwise HLag. The relatively poor performance of componentwise and own-other HLag is likely due to the coefficient matrix explicitly violating the structures in all 45 rows. However, both still significantly outperform the Bayesian methods, factor-based methods and univariate benchmarks.

Scenario 4: Data-based. Though all true parameters are non-zero, the HLag approaches perform considerably better than the lasso, lag-weighted lasso, Bayesian, factor-based and univariate approaches. HLag achieves variance reduction by enforcing sparsity and low max-lag orders. This, in turn, helps to improve forecast accuracy even for non-sparse DGPs where many of the coefficients are small in magnitude, as in Figure 4, panel (4).

Unknown maximal lag order. In Figure 6, we compare the performance of the VAR-based methods for known and unknown maximal lag order. For all methods in all considered scenarios, the MSFEs are, overall larger when the true maximal lag order is unknown since now the true lag order of each time series in each equation of the VAR can be overestimated. With a total of $pmax \cdot k^2 = 12 \times 45^2$ autoregressive parameters to estimate, the methods that assume an ordering, like HLag, are greatly advantaged over a method like the lasso that does not exploit this knowledge. Indeed, in Scenario 3 with unknown order, componentwise and own-other HLag outperform the lasso.

Computation time. Average computation times, in seconds on an Intel Core i7-6820HQ 2.70GHz machine including the penalty parameter search, for Scenario 1 and known order are reported in Table 1 for comparison. The relative performance of the methods with regard to average computation time in the other scenarios was very similar. The HLag methods have a clear advantage over the Bayesian methods of Giannone et al. (2015), Carriero et al. (2019) and the lag-weighted lasso. The latter minimally requires specifying a weight function, and a two-dimensional penalty parameter search in our implementation, which is much more time intensive than a one-dimensional search, as required for HLag. The Bayesian method of Banbura et al. (2010) is fast to compute since there is a closed-form expression for the mean of the posterior distribution of the autoregressive parameters conditional on the error variance-covariance matrix. While the Bayesian method of Banbura et al. (2010) and lasso require, in general, less computation time, HLag has clear advantages over the former two in terms of forecast accuracy, especially when the maximal lag length $pmax$ is large, but also in terms of lag order selection, as discussed in the following sections.

5.2 Robustness of HLag as $pmax$ Increases

We examine the impact of the maximal lag order $pmax$ on HLag's performance. Ideally, provided that $pmax$ is large enough to capture the system dynamics, its choice should

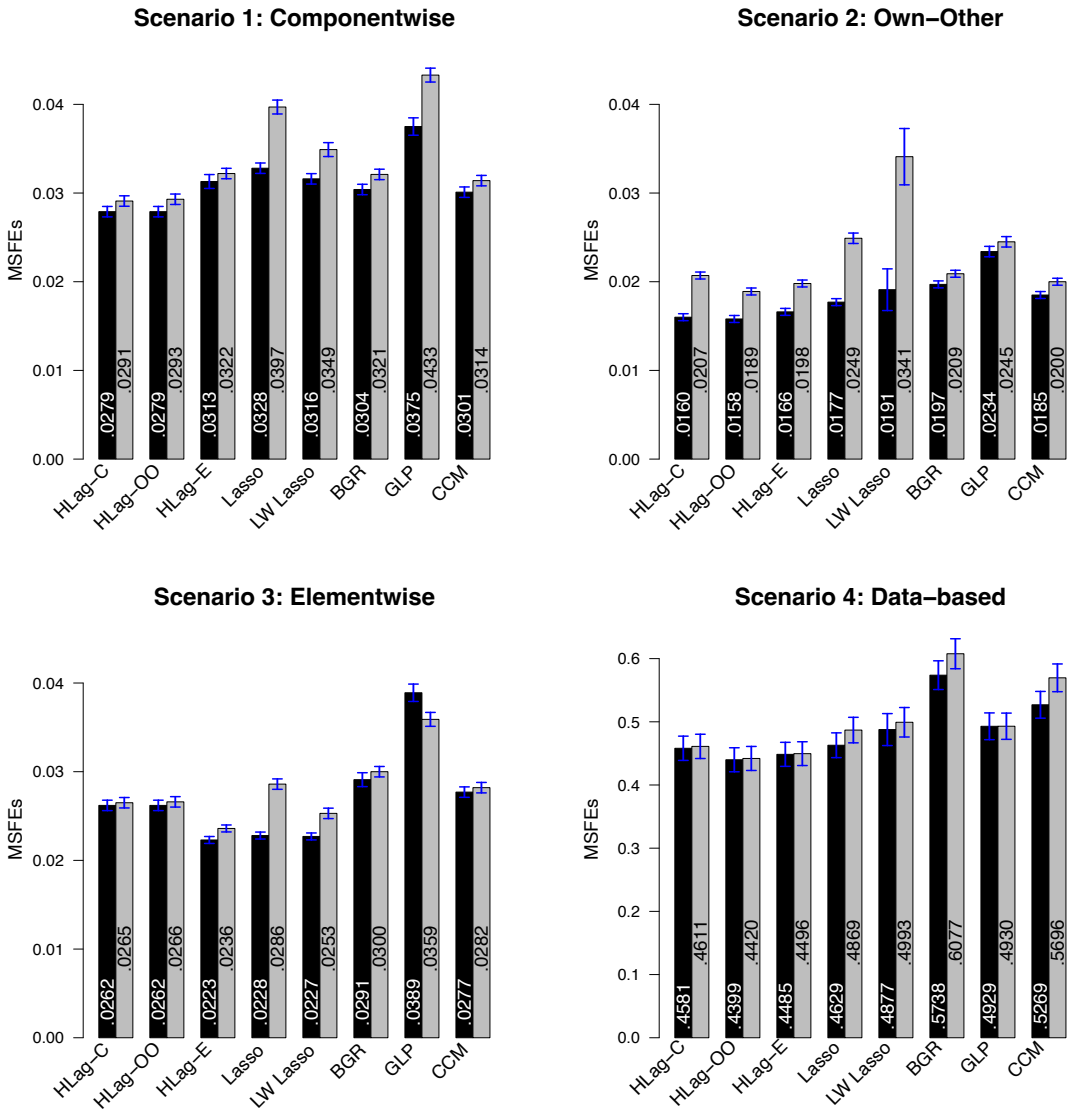


Figure 6: Out-of-sample mean squared forecast error for VARs in Scenario 1 to 4 for known (black) and unknown (gray) order. Error bars of length two standard errors are in blue.

have little impact on forecast performance. However, we expect regularizers that treat each coefficient democratically, like the lasso, to experience degraded forecast performance as $pmax$ increases.

As an experiment, we simulate from an $\text{HLag}^{C}_{10}(5)$ while increasing $pmax$ to substantially exceed the true $\mathbf{L}$. Figure 7 depicts the coefficient matrices and its magnitudes in what we will call Scenario 5. All series in the first 4 rows have $\mathbf{L} = 2$, the next 3 rows have $\mathbf{L} = 5$, and the final 3 rows have $\mathbf{L} = 0$. We consider varying $pmax \in \{1, 5, 12, 25, 50\}$ and show

Class	Method	Computation time (in seconds)
HLag	Componentwise	17.1
	Own-other	6.5
	Elementwise	10.9
VAR	Lasso	8.4
	Lag-weighted lasso	154.2
BVAR	BGR	0.4
	GLP	348.8
	CCM	79.5
Factor	DFM	3.5
	FAVAR	3.1

Table 1: Average computation times (in seconds), including the penalty parameter search, for the different methods in Scenario 1 ($T = 100,\ k = 45,\ p = 5$). The results for the least squares, sample mean, VAR(1), AR model and random walk are omitted as their computation time is negligible.

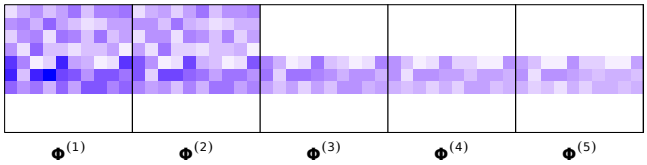


Figure 7: Componentwise structure in the Robustness simulation Scenario 5.

the MSFEs of all VAR-based methods requiring a maximal lag order in Figure 8. As *pmax* increases, we expect the performance of HLag to remain relatively constant whereas the lasso and information-criterion based methods should return worse forecasts.

At $pmax = 1$ all models are misspecified. Since no method is capable of capturing the true dynamics of series 1-7 in Figure 7, all perform poorly. As expected, after ignoring $pmax = 1$, componentwise HLag achieves the best performance across all other choices for *pmax*, but is very closely followed by the own-other and elementwise HLag methods. Among the information-criterion based methods, AIC performs substantially worse than BIC as *pmax* increases. This is likely the result of BIC assigning a larger penalty on the number of coefficients than AIC. The lasso's performance degrades substantially as the lag order increases, while the lag-weighted lasso and Bayesian methods are somewhat more robust to the lag order, but still achieve worse forecasts than every HLag procedure under all choices for *pmax*.

5.3 Lag Order Selection

While our primary intent in introducing the HLag framework is better point forecast performance and improved interpretability, one can also view HLag as an approach for selecting lag order. Below, we examine the performance of the proposed methods in estimating the maxlag matrix $\mathbf{L}$ defined in Section 3.1. Based on an estimate $\hat{\mathbf{\Phi}}$ of the autoregressive

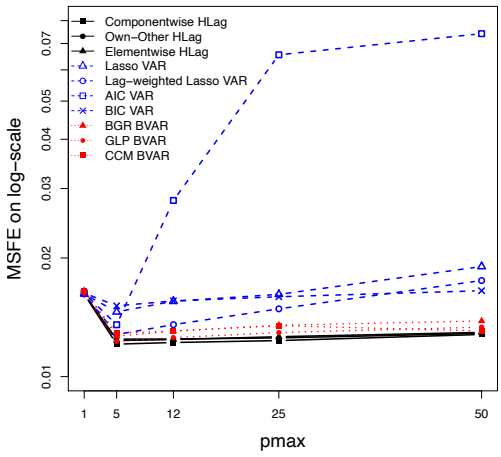


Figure 8: Robustness simulation scenario: Out-of-sample mean squared forecast errors, for different values of the maximal lag order $pmax$.

coefficients, we can likewise define a matrix of estimated lag orders:

$$\hat{\mathbf{L}}_{ij} = \max\{\ell : \hat{\mathbf{\Phi}}_{ij}^{(\ell)} \neq 0\},$$

where we define $\hat{\mathbf{L}}_{ij} = 0$ if $\hat{\mathbf{\Phi}}_{ij}^{(\ell)} = 0$ for all ℓ. It is well known in the regularized regression literature (cf., Leng et al. 2006) that the optimal tuning parameter for prediction is different from that for support recovery. Nonetheless, in this section we will proceed with the cross-validation procedure used previously with only two minor modifications intended to ameliorate the tendency of cross-validation to select a value of λ that is smaller than optimal for support recovery. First, we cross-validate a relaxed version of the regularized methods in which the estimated nonzero coefficients are refit using ridge regression, as detailed in Section C.6 of the appendix. This modification makes the MSFE more sensitive to $\hat{\mathbf{L}}_{ij}$ being larger than necessary. Second, we use the "one-standard-error rule" discussed in Hastie et al. (2009), in which we select the largest value of λ whose MSFE is no more than one standard error above that of the best performing model (since we favor the most parsimonious model that does approximately as well as any other).

We consider Scenario 1 to 5 and estimate a $VAR_k(12)$. A procedure's lag order selection accuracy is measured based on the sum of absolute differences between $\mathbf{L}$ and $\hat{\mathbf{L}}$ and the maximum absolute differences between $\mathbf{L}$ and $\hat{\mathbf{L}}$:

$$\|\hat{\mathbf{L}} - \mathbf{L}\|_1 = \sum_{ij} |\hat{\mathbf{L}}_{ij} - \mathbf{L}_{ij}| \text{ and } \|\hat{\mathbf{L}} - \mathbf{L}\|_\infty = \max_{i,j} |\hat{\mathbf{L}}_{ij} - \mathbf{L}_{ij}|. \tag{12}$$

The former can be seen as an overall measure of lag order error, the latter as a "worst-case" measure. We present the values on both measures relative to that of the sample mean (which chooses $\hat{\mathbf{L}}_{ij} = 0$ for all i and j). Figure 9 gives the results on the L_1-based measure.

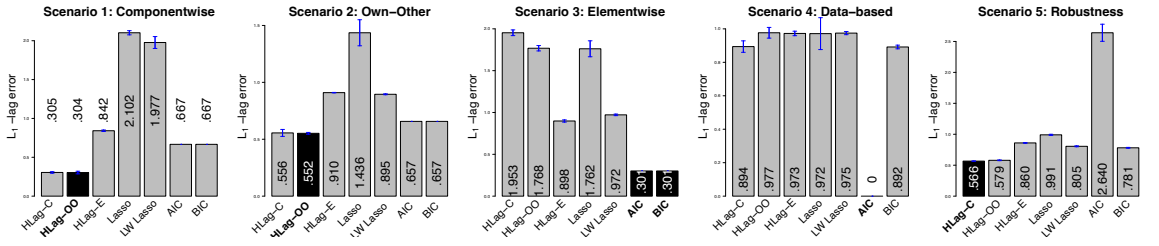


Figure 9: L_1-lag selection performance for Scenario 1 to 5. Error bars of length two standard errors are in blue; the best performing method is in black.

We focus our discussion on the VAR-methods performing actual lag order selection. We first discuss these results then summarize the differences in results for the L_∞-based measure.

L_1-lag selection performance. In Scenarios 1-3, the HLag methods geared towards the design-specific lag structure perform best, as expected. Least squares AIC/BIC always estimates a $VAR_k(1)$ and performs considerably worse than the best performing HLag method in Scenarios 1-2. In Scenario 3, they attain the best performance since around 82% of the elements in the true maxlag matrix are equal to one, and hence correctly recovered. However, the higher order dynamics of the remaining 18% of the elements are ignored, while elementwise HLag—which performs second best—better captures these dynamics. This explains why in terms of MSFE, elementwise HLag outperforms the $VAR_k(1)$ by a factor of 10.

In Scenario 4, least squares AIC consistently recovers the true universal order $p = 4$. Nevertheless, it has, in general, a tendency to select the highest feasible order, which happens to coincide here with the true order. Its overfitting tendency generally has more negative repercussions, as can be seen from Scenario 5, and even more importantly from its poor forecast performance. Componentwise HLag and least squares BIC perform similarly and are second best. Own-other, elementwise HLag, lasso and lag-weighted lasso perform similarly but underestimate the lag order of the component series with small non-zero values at higher order lags. While this negatively affects their lag order selection performance, it helps for forecast performance as discussed in Section 5.1.

In Scenario 5, componentwise and own-other HLag achieve the best performance. Their performance is five times better than the least squares AIC, and roughly 1.5 times better than the lasso, lag-weighted lasso and least squares BIC. Elementwise HLag substantially outperforms the lasso and least squares AIC, which consistently severely overestimates the true lag order. The least squares BIC, on the other hand, performs similarly to elementwise HLag on the lag selection criterion but selects the universal lag order at either 1 or 2 and thus does not capture the true dynamics of series 5-7 in Figure 7.

In Figure 10, we examine the impact of the maximal lag order *pmax* on a method's lag order error. At the true order ($pmax = 5$), all methods achieve their best performance. As *pmax* increases, we find the methods' performance to decrease, in line with the findings by Percival (2012). Yet, the HLag methods and lag-weighted lasso remain much more robust than the AIC and lasso, whose performance degrade considerably.

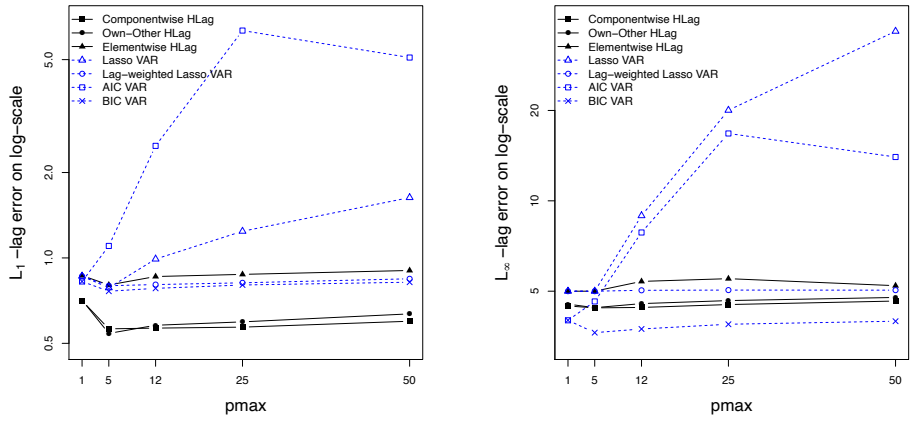


Figure 10: Robustness simulation scenario: Lag order error measures, for different values of the maximal lag order $pmax$.

L_∞-lag selection performance. Results on the "worst-case" L_∞-measure are presented in Figure 11. Differences compared to the L_1-measure are: (i) Least squares AIC/BIC are the best performing. This occurs since the true maximal lag orders are small, as well as the estimated lag orders by AIC/BIC due to the maximum number of parameters that least squares can take. Hence, the maximal difference between both is, overall, small. Their negative repercussions are better reflected through the overall L_1-measure, or in case of the AIC as $pmax$ increases (see Figure 10). (ii) Componentwise and own-other HLag are more robust with respect to the L_∞-measure than elementwise HLag. The former two either add an additional lag for all time series or for none, thereby encouraging low lag order solutions—and thus controlling the maximum difference with the small true orders—even more than elementwise HLag. The latter (and the lag-weighted lasso) can flexibly add an additional lag for each time series separately. Their price to pay for this flexibility becomes apparent through the L_∞-measure. (iii) A noticeable difference occurs between the methods that assume an ordering, like HLag and the lag-weighted lasso, and methods, like the lasso, that do not encourage low maximal lag orders. The lasso often picks up at least one lag close to the maximally specified order, thereby explaining its bad performance in terms of the L_∞-measure. As $pmax$ increases, its performance deteriorates even more, see Figure 10.

Stability across time. We verified the stability in lag order selection across time with a rolling window approach. We estimate the different models for the last 40 time points (20%), each time using the most recent 160 observations. For each of these time points, the lag matrices are obtained and the lag selection accuracy measures in equation (12) are computed. For all methods, we find the lag order selection to be very stable across time with no changes in their relative performance.

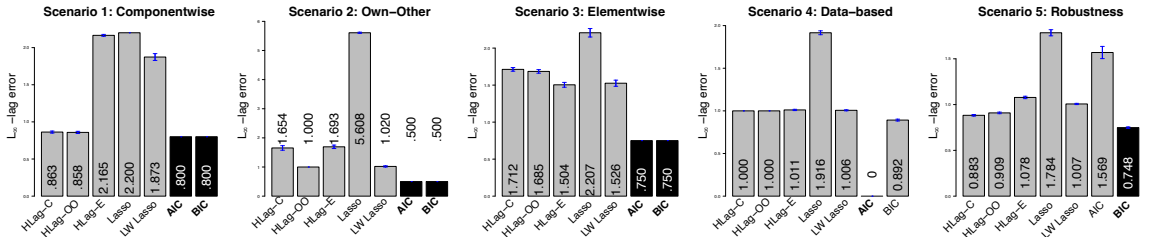


Figure 11: L_∞-lag selection performance for Scenario 1 to 5. Error bars of length two standard errors are in blue; the best performing method is in black.

6. Data Analysis

We demonstrate the usefulness of the proposed HLag methods for various applications. Our first and main application is macroeconomic forecasting (Section 6.1). We investigate the performance of the HLag methods on several VAR models where the number of time series is varied relative to the fixed sample size. Secondly, we use the HLag methods for forecast applications with high sampling rates (Section 6.2).

For all applications, we compare the forecast performance of the HLag methods to their competitors. We use the cross-validation approach from Section 4 for penalty parameter selection on time points T_1 to T_2: At each time point $t = T_1 - h, \ldots, T_2 - h$ (with h the forecast horizon), we first standardize each series to have sample mean zero and variance one using the most recent $T_1 - h$ observations. We do this to account for possible time variation in the first and second moment of the data. Then, we estimate the VAR with $pmax$ and compute the weighted Mean Squared Forecast Error

$$wMSFE = \frac{1}{k(T_2 - T_1 + 1)} \sum_{i=1}^{k} \sum_{t=T_1-h}^{T_2-h} \left(\frac{y_{i,t+h}^{(s)} - \hat{y}_{i,t+h}^{(s)}}{\widehat{\sigma}_i} \right)^2,$$

where $\widehat{\sigma}_i$ is the standard deviation of the i^{th} to be forecast series, computed over the forecast evaluation period $[T_1, T_2]$ for each penalty parameter. We use a *weighted* MSFE to account for the different volatilities and predictabilities of the different series when computing an overall forecast error measure (Carriero et al., 2011). The selected penalty parameter is the one giving the lowest $wMSFE$.

After penalty parameter selection, time points T_3 to T_4 are used for out-of-sample rolling window forecast comparisons. Again, we standardize each series separately in each rolling window, estimate a VAR on the most recent $T_3 - h$ observations and evaluate the overall forecast accuracy with the $wMSFE$ of equation (13), averaged over all k time series and time points of the forecast evaluation period. Similar results are obtained with an expanding window forecast exercise and available from the authors upon request.

Finally, to assess the statistical significance of the results, we use the Model Confidence Set (MCS) procedure of Hansen et al. (2011). It separates the best forecast methods with equal predictive ability from the others, who perform significantly worse. We use the `MCSprocedure` function in `R` to obtain a MCS that contains the best model with 75% confidence as done in Hansen et al. (2011).

6.1 Macroeconomic Forecasting

We apply the proposed HLag methods to a collection of US macroeconomic time series compiled by Stock and Watson (2005) and augmented by Koop (2013). The full data set, publicly available at The Journal of Applied Econometrics Data Archive, contains 168 quarterly macroeconomic indicators over 45 years: Quarter 2, 1959 to Quarter 4, 2007, hence $T = 195$. Following Stock and Watson (2012), we classify the series into 13 categories, listed in Table 6 of the appendix. Further details can be found in Section D of the appendix.

Following Koop (2013), we estimate four VAR models on this data set: The *Small-Medium VAR ($k = 10$)* which consists of GDP growth rate, the Federal Funds Rate, and CPI plus 7 additional variables, including monetary variables. The *Medium* VAR ($k = 20$) which contains the Small-Medium group plus 10 additional variables containing aggregated information on several aspects of the economy. The *Medium-Large VAR ($k = 40$)* which contains the Medium group plus 20 additional variables, including most of the remaining aggregate variables in the data set. The *Large VAR ($k = 168$)* which contains the Medium-Large group plus 128 additional variables, consisting primarily of the components that make up the aggregated variables. Note that the number of parameters quickly increases from $4 \times 10^2 + 10 = 410$ (*Small-Medium* VAR) over $4 \times 20^2 + 20 = 1{,}620$ (*Medium* VAR), $4 \times 40^2 + 40 = 6{,}440$ (*Medium-Large* VAR), to $4 \times 168^2 + 168 = 113{,}064$ (*Large* VAR).

6.1.1 Forecast Comparisons

We compare the forecast performance of the HLag methods to their competitors on the four VAR models with $pmax = 4$, following the convention from Koop (2013). Quarter 3, 1977 (T_1) to Quarter 3, 1992 (T_2) is used for penalty parameter selection; Quarter 4, 1992 (T_3) to Quarter 4, 2007 (T_4) are used for out-of-sample rolling window forecast comparisons. We start with a discussion on the forecast accuracy for all series combined, then break down the results across different VAR sizes for specific variables.

Forecast performance across all series. We report the out-of-sample one-step-ahead weighted mean squared forecast errors for the four VAR groups with forecast horizon $h = 1$ in Figure 12. We discuss the results for each VAR group separately since the $wMSFE$ are not directly comparable across the panels of Figure 12, as an average is taken over different component series which might be more or less difficult to predict.

With only a limited number of component series k included in the *Small* VAR, the univariate AR attains the lowest $wMSFE$, but own-other HLag, the lasso and FAVAR have equal predictive ability since they are included in the MCS. As more component series are added in the *Medium* and *Medium-Large* VAR, own-other and elementwise HLag outperform all other methods. The more flexible own-other and elementwise structures perform similarly, and better than the componentwise structure. While the MCS includes own-other HLag, elementwise HLag and the lasso for the *Medium* VAR, only own-other HLag survives for the *Medium-Large* VAR. This supports the widely held belief that in economic applications, a components' *own* lags are likely more informative than *other* lags and that maxlag varies across components. Furthermore, the Bayesian and factor models are never included in the MCS, nor are the least squares methods, or univariate methods. For the *Medium* VAR, the information criteria AIC and BIC always select three lags. Since

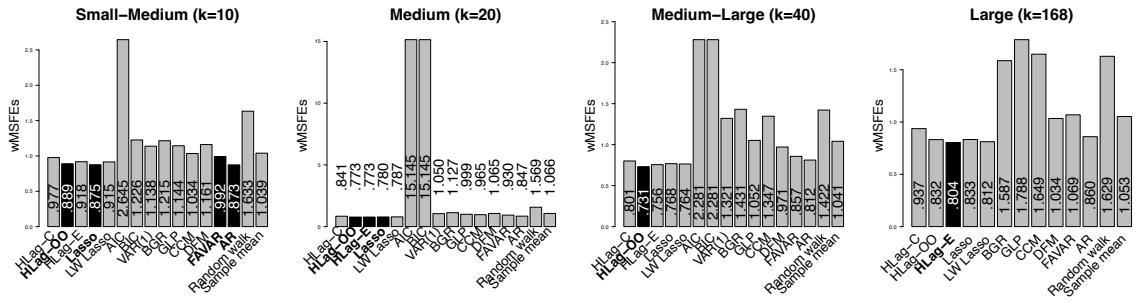


Figure 12: Rolling out-of-sample one-step-ahead $wMSFE$ for the four VAR sizes. For each VAR size, forecast methods in the 75% Model Confidence Set (MCS) are in black.

a relatively large number of parameters need to be estimated, their estimation error becomes large, and this, in turn, severely impacts their forecast accuracy.

Next, consider the *Large* VAR, noting that the VAR by AIC, BIC and VAR(1) are overparametrized and not included. As the number of component series k further increases, the componentwise HLag structure becomes less realistic. This is especially true in high-dimensional economic applications, in which a core subset of the included series is typically most important in forecasting. In Figure 12 we indeed see that the more flexible own-other and elementwise HLag perform considerably better than the componentwise HLag. The MCS confirms the strong performance of elementwise HLag.

HLag's good performance across all series is confirmed by forecast accuracy results broken down by macroeconomic category. The flexible elementwise HLag is the best performing method; for almost all categories, it is included in the MCS, which is not the case for any other forecasting method. Detailed results can be found in Figure 18 of the appendix.

Furthermore, our findings remain stable when we increase the maximal lag order $pmax$. In line with Banbura et al. (2010), we re-estimated all models with $pmax = 13$. Detailed results are reported in Figure 19 of the appendix. For the *Small-Medium* VAR, own-other HLag performs comparable to the AR benchmark, while it outperforms all other methods for larger VARs. The lasso (and to a lesser extent the lag-weighted lasso) loses its competitiveness vis-a-vis the HLag approaches as soon as the maximal lag order $pmax$ increases, in line with the results of Section 5.2.

Finally, we re-did our forecast exercise for longer forecast horizons $h = 4$ and $h = 8$. Detailed results are reported in Figure 20 of the appendix. All forecast errors increase with distant forecast horizons. Nonetheless, own-other HLag remains among the best forecast methods: it is the only method that is always included in the MCS. Its performance gets closer to the sample mean as the forecast horizon increases.

Comparing forecast performance across different VAR sizes. To investigate whether large VARs improve forecast accuracy over smaller VARs, we turn to the MSFEs of the individual component series obtained with the multivariate forecast methods. We focus on Real Gross Domestic Product (*GDP251*), Consumer Price Index (*CPIAUSL*), and the Federal Funds Rate (*FYFF*) which are generally of primary interest to forecasters and

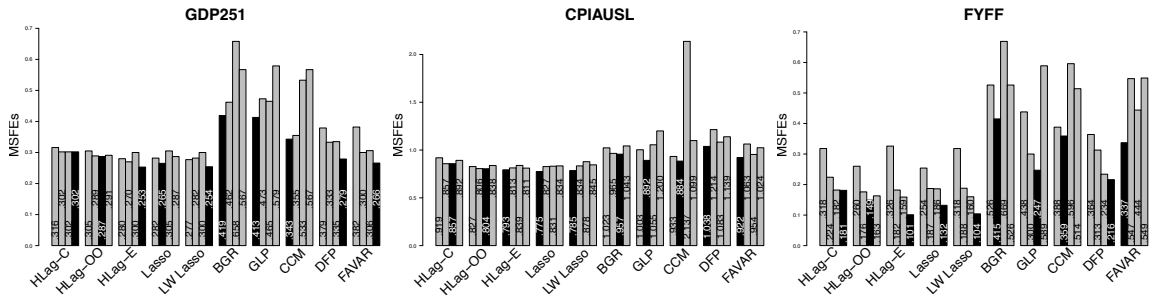


Figure 13: Rolling out-of-sample one-step ahead mean squared forecast error of *GDP251*, *CPIAUSL* and *FYFF* for the different VAR sizes (bars from left to right: Small-Medium, Medium, Medium-Large, Large). For each method, the lowest MSFE is indicated in black.

policymakers. Figure 13 gives the MSFEs of these three component series in the four VAR models.

Despite the fact that the *Small-Medium* VAR forecasts well for some component series, like *CPIAUSL*, we often find, similar to Koop (2013), that moving away from small VARs leads to improved forecast performance. Consider, for instance, *GDP251* and *FYFF* where half of the forecast methods give the best MSFE in the *Large* VAR. Across the $k = 10$ component series included in all four VARs, HLag, the lasso and factor methods produce the best MSFEs mainly for the *Medium-Large* or *Large* VARs; the Bayesian methods mainly for the *Small-Medium* or *Medium* VARs.

Furthermore, the loss in forecast accuracy when adding variables to the VAR, if it occurs, remains relatively limited for HLag methods (on average, only 5%) but is severe for Bayesian methods (on average, 46%). Although Bayesian methods perform shrinkage, all component series remain included in the larger VARs, which can severely impede forecast accuracy. HLag methods, in contrast, do not use all component series but offer the possibility to exclude possibly irrelevant or redundant variables from the forecast model.

While factor-based models produce good forecasts for larger VARs, as the factors can be estimated more precisely as the number of component series increases, the factors themselves do not carry, in many cases, economic interpretation. The HLag methods, in contrast, facilitate interpretation by providing direct insight into the component series that contribute to the good forecast performance, as discussed next.

6.1.2 Lag Order Selection

The HLag methods provide direct insight into the series contributing to the forecasting of each individual component. As an example, consider the estimated lag orders of the three main component series (*GDP251*, *CPIAUSL* and *FYFF*) from a fitted HLag_{40}^{E} model of the *Medium-Large* group in Figure 14. Elementwise HLag finds, for instance, that the Federal Funds Rate *FYFF* is an important predictor of Gross Domestic Product since two of its lagged components are included in the equation for forecasting *GDP251*.

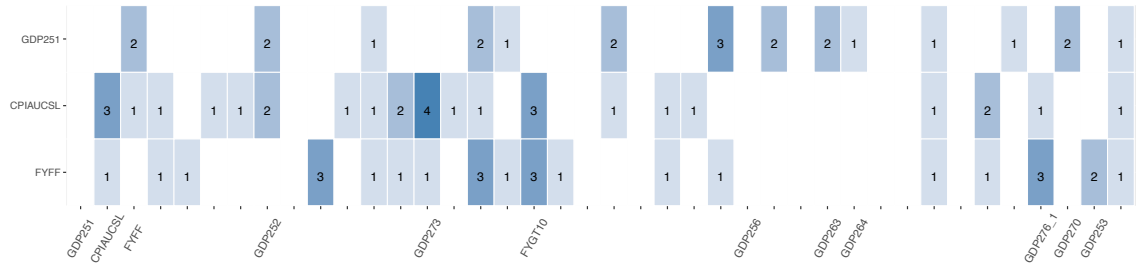


Figure 14: The first three rows of $\hat{L}^E$, the estimated elementwise maxlag matrix in the *Medium-Large* VAR for the $HLag^E$ method. Components with zero maxlag are left empty.

Generally speaking, the lag selection results are considerably stable across time. Figure 21 in Section D of the appendix gives, for each end point of the rolling window, the fraction of non-zero coefficients in each of the 13 macroeconomic categories when forecasting *GDP251*, *CPIAUSL*, and *FYFF*. To forecast GDP growth, for instance, GDP components, employment, interest rates and stock prices have a stable and important contribution throughout the entire forecast evaluation period.

6.2 Applications with High Sampling Rates

The HLag methods can also be used for applications with high sampling rates. To illustrate this, we consider a financial and energy data set.

6.2.1 Financial Application

We apply the HLag methods to a financial data set containing realized variances for $k = 16$ stock market indices, listed in Table 7 of the appendix. Daily realized variances based on five minute returns are taken from Oxford-Man Institute of Quantitative Finance (publicly available on http://realized.oxford-man.ox.ac.uk/data/download). Our data set consists of $T = 4{,}163$ trading days between January 4, 2000 and December 30, 2019.

We compare the HLag methods to their competitors on estimated VARs with $pmax = 22$ (one trading month). The number of parameters is thus $22 \times 16^2 + 16 = 5,648$. December 7, 2018 to June 26, 2019 (104 observations) are used for penalty parameter selection; June 27, 2019 to December 30, 2019 (104 observations) for forecast comparisons.

Figure 15, panel (a) presents the one-step-ahead weighted mean squared forecast errors.[1] All three HLag methods are, together with the lasso, among the best performing methods, as confirmed through the MCS. The HLag methods and lasso attain considerable forecast gains over all other methods. The HLag methods' performance remains stable across different values of the maximal lag order, unlike the performance of the lasso. Furthermore, elementwise HLag achieves its good forecast accuracy using a more parsimonious, more in-

1. We excluded the BVAR methods GLP and CCM as they are too time consuming for large-scale VARs.

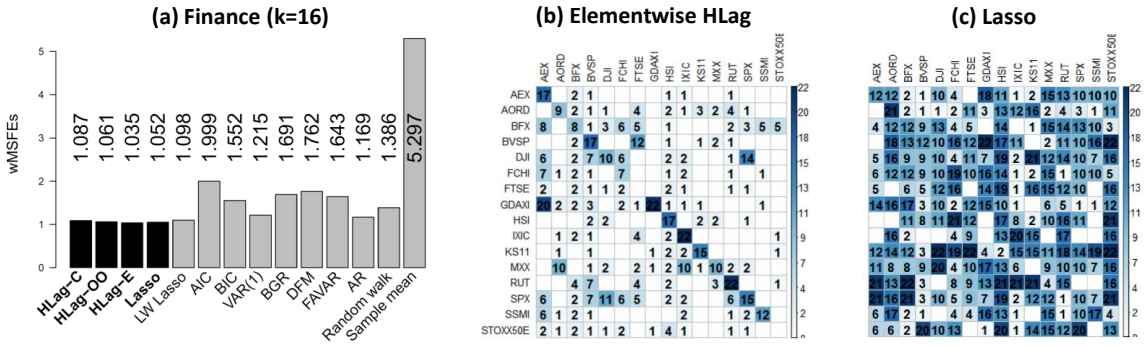


Figure 15: Financial application. Panel (a): Rolling out-of-sample one-step-ahead $wMSFE$ with the forecast methods in the 75% MCS in black. Panel (b): Estimated maxlag matrix for elementwise HLag and Panel (c): for the lasso.

terpretable description of the data than the lasso as can be seen from the estimated maxlag matrices in Figure 15 panel (b) and (c) respectively.

6.2.2 Energy Application

We apply the HLag methods to an energy data set (Candanedo et al., 2017) containing information on $k = 26$ variables related to in-house energy usage, temperature and humidity conditions. The energy data was logged every 10 minutes for about 4.5 months, giving $T = 19{,}735$ observations in total. A list of all variables and a short description is provided in Table 8 of the appendix. Data are taken from the publicly available UCI Machine Learning Repository (https://archive.ics.uci.edu/ml/data sets/Appliances+energy+prediction).

To evaluate the forecast performance of HLag, we estimate VAR models with $pmax = 6$ (one hour), thus containing $6 \times 26^2 + 26 = 4{,}082$ parameters. May 16, 18:10 to May 17, 2016 18:00 (144 observations) are used for penalty parameter selection; May 17, 18:10 to May 27, 2016 18:00 (1440 observations) for forecast comparisons.

Figure 16 presents the one-step-ahead weighted mean squared forecast errors.[2] As the sample size is large, the least squares VAR-based methods do not suffer as much from the curse of dimensionality. Still, HLag has an advantage by not imposing a universal maximal lag order. On the whole data set (panel a), componentwise and elementwise HLag outperform all other methods apart from the lasso and lag-weighted lasso. Yet, a subsample analysis reveals the dominance of elementwise HLag. We split the data set into ten consecutive subperiods of equal length and repeated the same forecast exercise. Results are displayed in panels (b)-(k). Elementwise HLag maintains its good performance across the subperiods and performs best. It is included in the MCS for all subperiods except for the second, making it a valuable addition to a forecaster's toolbox.

2. We excluded the BVAR methods GLP and CCM as they are too time consuming for large-scale VARs.

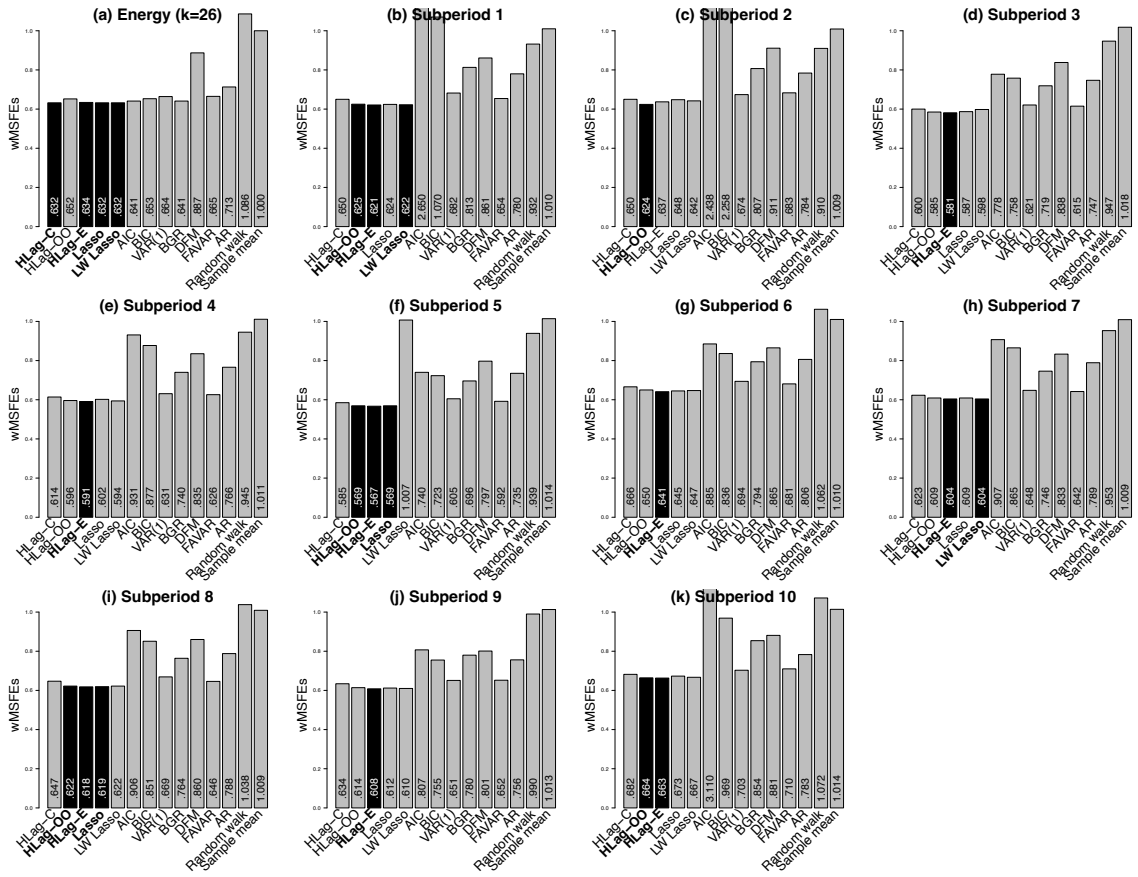


Figure 16: Energy application. Rolling out-of-sample one-step-ahead $wMSFE$ on the data set and on ten subperiods. Forecast methods in the 75% MCS are indicated in black in each panel.

7. Discussion

By incorporating the property that more recent lags convey more information than distant lags, the HLag framework offers substantial forecast improvements as well as greater insight into lag order selection than existing methods. In addition, throughout our simulation scenarios, we see that each method is fairly robust to deviations from its particular hierarchical structure. The substantial improvements in forecasting accuracy in data applications provide justification for the widely held belief that as the number of component series included in a model increases, the maximal lag order is not symmetric across series.

To enforce the hierarchical lag structures, we use the nested group structure of Zhao et al. (2009). Alternatively, one could leverage the latent overlapping group lasso (LOG) proposed by Jacob et al. (2009). While Yan and Bien (2017) indicate that the nested group structures might suffer from a more aggressive shrinkage of parameters deep in the hierarchy (i.e. higher-order autoregressive coefficients), in the VAR model, large amounts of shrinkage on the more distant lags versus small amounts of shrinkage on the more recent lags may

be desirable (Song and Bickel, 2011). In our simulation studies, the nested group lasso structures significantly outperformed the LOG structures in the large majority of cases. Especially as the maximal lag order increases, the nested group lasso turned out to be more robust. Detailed results are available in Section C.3 of the appendix.

Implementations of our methods are available in the R package `BigVAR`, which is hosted on the Comprehensive R Archive Network (cran). Despite the more challenging computational nature of overlapping group lasso problems compared to conventional sparsity or non-overlapping group sparsity problems (e.g., Chen et al., 2014, Yuan et al., 2011, Mairal et al., 2010), our methods scale well and are computationally feasible in high dimensions. For instance, for the *Large* VAR ($k = 168, T = 195$, and 113,064 parameters) estimated on the Stock and Watson data, the HLag methods only require (on an Intel Xean Gold 6126 CPU @ 2.60GHz machine) around 1.5 (Own-Other), 2 (Componentwise) and 3.5 minutes (Elementwise), including penalty parameter selection. This requires estimating the VAR 610 times (61 time points $\times$ 10 penalty parameters). For fixed penalty parameter, the HLag methods can be computed in less than a second. The computational bottleneck of our implementation thus concerns the penalty parameter selection. Alternatives (information criteria or a time series cross-validation search where the models are not re-estimated every single time point but at a lower sampling frequency) can be considered to reduce the penalty parameter search for applications with high sampling rates. To be widely adopted by practitioners, we do think that our methods have a considerable advantage compared to more computationally intensive methods such as the lag-weighted lasso, the Bayesian CCM and GLP approaches requiring around 33 minutes (Lag-weighted lasso) or even more than 2 hours (Bayesian methods) for one model fit of the *Large* Stock and Watson VAR. At the very least, one of the proposed HLag approaches can be quickly run to provide numerous insights before a more computationally demanding method is adopted.

The HLag framework is quite flexible and can be extended in various ways. For example, more complicated weighting schemes (see e.g., Jenatton et al., 2011a, Bien et al., 2016) could be adopted to address the more aggressive shrinkage of parameters deep in the hierarchy, but these make computation more involved (Yan and Bien, 2017) and our simulations in Section C.3 of the appendix indicate that this may not be beneficial in the VAR setting. Furthermore, if the practitioner prefers to summarize the information content in large data sets by constructing few factors, HLag penalties can, for instance, be applied with minimal adaption to the factors augmenting the VAR in a FAVAR. The HLag framework would allow one to flexibly vary the number of factors in each marginal equation of the FAVAR and to automatically determine the lag order of the factors, in addition to the lag structure of the autoregressive components. Finally, building on Basu and Michailidis (2015), we derive preliminary theoretical results on prediction consistency for HLag in a high-dimensional regime. Given the complicated nested group structure of the HLag penalty, work is needed to further explore its theoretical properties. To this end, recent advances in the theory of the hierarchical group lasso (e.g., Yan and Bien, 2017; Yu and Bien, 2017) could be leveraged.

Acknowledgments

We thank the reviewers for their thorough review and highly appreciate their comments and suggestions which substantially improved the quality of the manuscript. The authors thank Gary Koop for providing his data transformation script. This research was supported by an Amazon Web Services in Education Research Grant. IW was supported by the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 832671. JB was supported by NSF DMS-1405746 and NSF DMS-1748166 and DSM was supported by NSF (1455172, 1934985, 1940124, 1940276), Xerox PARC, the Cornell University Atkinson Center for a Sustainable Future, USAID, and the Cornell University Institute of Biotechnology & NYSTAR.

Appendix A. Theoretical properties: Proofs

We start by proving two auxiliary results, then we combine these in the proof of Theorem 1. For ease of notation and without loss of generality, we omit the intercept vector $\boldsymbol{\nu}$ from the VAR model (1).

Lemma 1 *If* $\lambda \geq \max_\ell \|\frac{1}{T}\sum_{t=1}^T \mathbf{y}_{t-\ell}\mathbf{u}_t^\top\|_\infty$, *then*

$$\frac{1}{2T}\sum_{t=1}^T \|\sum_{\ell=1}^p (\boldsymbol{\Phi}^{(l)} - \hat{\boldsymbol{\Phi}}^{(l)})\mathbf{y}_{t-\ell}\|_2^2 \leq 2\lambda \mathcal{P}^E_{HLag}(\boldsymbol{\Phi}).$$

Proof of Lemma 1. Since $\hat{\boldsymbol{\Phi}}$ is a minimizer of (4), we have that

$$\frac{1}{2T}\sum_{t=1}^T \|\mathbf{y}_t - \sum_{\ell=1}^p \hat{\boldsymbol{\Phi}}^{(l)}\mathbf{y}_{t-\ell}\|_2^2 + \lambda\mathcal{P}^E_{\text{HLag}}(\hat{\boldsymbol{\Phi}}) \leq \frac{1}{2T}\sum_{t=1}^T \|\mathbf{y}_t - \sum_{\ell=1}^p \boldsymbol{\Phi}^{(\ell)}\mathbf{y}_{t-\ell}\|_2^2 + \lambda\mathcal{P}^E_{\text{HLag}}(\boldsymbol{\Phi}).$$

Substituting the data generating process $\mathbf{y}_t = \boldsymbol{\Phi}^{(1)}\mathbf{y}_{t-1} + \cdots + \boldsymbol{\Phi}^{(p)}\mathbf{y}_{t-p} + \mathbf{u}_t$ into the above, we obtain

$$\frac{1}{2T}\sum_{t=1}^T \|\mathbf{u}_t + \sum_{\ell=1}^p (\boldsymbol{\Phi}^{(l)} - \hat{\boldsymbol{\Phi}}^{(l)})\mathbf{y}_{t-\ell}\|_2^2 + \lambda\mathcal{P}^E_{\text{HLag}}(\hat{\boldsymbol{\Phi}}) \leq \frac{1}{2T}\sum_{t=1}^T \|\mathbf{u}_t\|_2^2 + \lambda\mathcal{P}^E_{\text{HLag}}(\boldsymbol{\Phi}).$$

After re-arranging, we get

$$\frac{1}{2T}\sum_{t=1}^T \|\sum_{\ell=1}^p (\boldsymbol{\Phi}^{(l)} - \hat{\boldsymbol{\Phi}}^{(l)})\mathbf{y}_{t-\ell}\|_2^2 + \lambda\mathcal{P}^E_{\text{HLag}}(\hat{\boldsymbol{\Phi}}) \leq \frac{1}{T}\sum_{t=1}^T\sum_{\ell=1}^p \mathbf{u}_t^\top(\hat{\boldsymbol{\Phi}}^{(l)} - \boldsymbol{\Phi}^{(l)})y_{t-\ell} + \lambda\mathcal{P}^E_{\text{HLag}}(\boldsymbol{\Phi}).$$

Now,

$$\begin{aligned}
\frac{1}{T}\sum_{t=1}^T\sum_{\ell=1}^p \mathbf{u}_t^\top(\hat{\boldsymbol{\Phi}}^{(l)} - \boldsymbol{\Phi}^{(l)})\mathbf{y}_{t-\ell} &= \frac{1}{T}\sum_{\ell=1}^p \langle \hat{\boldsymbol{\Phi}}^{(l)} - \boldsymbol{\Phi}^{(l)}, \sum_{t=1}^T \mathbf{y}_{t-\ell}\mathbf{u}_t^\top\rangle \\
&\leq \frac{1}{T}\|\hat{\boldsymbol{\Phi}} - \boldsymbol{\Phi}\|_1 \max_\ell \|\sum_{t=1}^T \mathbf{y}_{t-\ell}\mathbf{u}_t^\top\|_\infty \\
&\leq \frac{1}{T}\mathcal{P}^E_{\text{HLag}}(\hat{\boldsymbol{\Phi}} - \boldsymbol{\Phi}) \max_\ell \|\sum_{t=1}^T \mathbf{y}_{t-\ell}\mathbf{u}_t^\top\|_\infty
\end{aligned}$$

since $\|\mathbf{\Phi}\|_1 := \sum_{i=1}^k \sum_{j=1}^k \sum_{\ell=1}^p \|\mathbf{\Phi}_{ij}\|_1 \leq \sum_{i=1}^k \sum_{j=1}^k \sum_{\ell=1}^p \|\mathbf{\Phi}_{ij}^{(\ell:p)}\|_2 := \mathcal{P}^E_{\text{HLag}}(\mathbf{\Phi})$. Thus,

$$\frac{1}{2T}\sum_{t=1}^T \|\sum_{\ell=1}^p (\mathbf{\Phi}^{(l)} - \hat{\mathbf{\Phi}}^{(l)})\mathbf{y}_{t-\ell}\|_2^2 + \lambda \mathcal{P}^E_{\text{HLag}}(\hat{\mathbf{\Phi}}) \leq \frac{1}{T}\mathcal{P}^E_{\text{HLag}}(\hat{\mathbf{\Phi}} - \mathbf{\Phi}) \max_\ell \|\sum_{t=1}^T \mathbf{y}_{t-\ell}\mathbf{u}_t^\top\|_\infty + \lambda \mathcal{P}^E_{\text{HLag}}(\mathbf{\Phi}).$$

Under the assumption on λ, we get

$$\frac{1}{2T}\sum_{t=1}^T \|\sum_{\ell=1}^p (\mathbf{\Phi}^{(l)} - \hat{\mathbf{\Phi}}^{(l)})\mathbf{y}_{t-\ell}\|_2^2 + \lambda \mathcal{P}^E_{\text{HLag}}(\hat{\mathbf{\Phi}}) \leq \lambda \mathcal{P}^E_{\text{HLag}}(\hat{\mathbf{\Phi}} - \mathbf{\Phi}) + \lambda \mathcal{P}^E_{\text{HLag}}(\mathbf{\Phi}).$$

The result follows from observing that $\mathcal{P}^E_{\text{HLag}}(\hat{\mathbf{\Phi}} - \mathbf{\Phi}) \leq \mathcal{P}^E_{\text{HLag}}(\hat{\mathbf{\Phi}}) + \mathcal{P}^E_{\text{HLag}}(\mathbf{\Phi})$. ■

Lemma 2 *If $T > 25\log(pk^2)$, $T > 4$, $pk^2 \gg 1$ and we choose $\lambda \geq 30v(\mathbf{\Phi}, \mathbf{\Sigma}_u)\sqrt{\log(pk^2)/T}$, then*

$$\max_{\ell,j,k} |\frac{1}{T}\sum_{t=1}^T y_{t-\ell,j}u_{t,k}| \leq \lambda$$

with probability at least $1 - \frac{12}{(pk^2)^{23/2}}$.

Proof of Lemma 2. In the middle of page 20 of Basu and Michailidis (2013)[3] (a preliminary version of Basu and Michailidis, 2015), it is shown that,

$$\mathbb{P}\left(\frac{1}{T}\max_{\ell,j,k}|\sum_{t=1}^T y_{t-\ell,j}u_{t,k}| > b\right) \leq 12\exp\left\{-\frac{T}{2}\min\left\{1, \left(\frac{b}{6v(\mathbf{\Phi}, \mathbf{\Sigma}_u)} - \frac{2}{\sqrt{T}}\right)^2\right\} + \log(pk^2)\right\}$$

where

$$b = (18 + 6\sqrt{2(A+1)})v(\mathbf{\Phi}, \mathbf{\Sigma}_u)\sqrt{\log(pk^2)/T}$$

for some constant $A > 0$. For simplicity, we take $A = 1$. Note that the exponent can be written as

$$\begin{aligned} -\frac{T}{2}\min\left\{1, \left(\frac{b}{6v(\mathbf{\Phi}, \mathbf{\Sigma}_u)} - \frac{2}{\sqrt{T}}\right)^2\right\} + \log(pk^2) &= -\frac{T}{2}\min\left\{1, \left(\frac{5\sqrt{\log(pk^2)} - 2}{\sqrt{T}}\right)^2\right\} + \log(pk^2) \\ &= -\frac{1}{2}\min\left\{T, \left(5\sqrt{\log(pk^2)} - 2\right)^2\right\} + \log(pk^2). \end{aligned}$$

Since $T > 25\log(pk^2)$ and $T > 4$, it follows that $T > \left(5\sqrt{\log(pk^2)} - 2\right)^2$ and the exponent is

$$-\frac{1}{2}[5\sqrt{\log(pk^2)} - 2]^2 + \log(pk^2) = -(23/2)\log(pk^2) + 10\sqrt{\log(pk^2)} - 2 \approx -(23/2)\log(pk^2),$$

3. S. Basu and G. Michailidis. Estimation in High-dimensional Vector Autoregressive Models. arXiv:1311.4175v1, 2013.

where the last approximation follows from the assumption that $pk^2 \gg 1$.

Thus, for the choice of λ given above, we have that

$$\mathbb{P}\left(\max_{\ell,j,k}|\sum_{t=1}^{T} y_{t-\ell,j}u_{t,k}| \leq \lambda\right) \geq 1 - 12\exp\left\{-(23/2)\log(pk^2)\right\} = 1 - \frac{12}{(pk^2)^{23/2}}.$$

■

Proof of Theorem 1. Combining the results from Lemma 1 and 2, we have for $T > 25\log(pk^2)$, $T > 4$, $pk^2 \gg 1$ and $\lambda \geq 30v(\mathbf{\Phi}, \mathbf{\Sigma}_u)\sqrt{\log(pk^2)/T}$, that

$$\text{MSFE}_{\text{in}} \leq \text{tr}(\mathbf{\Sigma}_u) + 4\lambda\mathcal{P}^E_{\text{HLag}}(\mathbf{\Phi})$$

or alternatively

$$\frac{1}{T}\sum_{t=1}^{T}\left\|\sum_{\ell=1}^{p}(\hat{\mathbf{\Phi}}^{(\ell)} - \mathbf{\Phi}^{(\ell)})\mathbf{y}_{t-\ell}\right\|_2^2 \leq 4\lambda\mathcal{P}^E_{\text{HLag}}(\mathbf{\Phi})$$

with probability at least $1 - \frac{12}{(pk^2)^{23/2}}$.

Assuming that all coefficients are bounded by M, we have that

$$\|\mathbf{\Phi}_{ij}^{(\ell:L_{ij})}\|_2 \leq M\sqrt{L_{ij} - \ell + 1} \leq M\sqrt{L_{ij}},$$

so

$$\mathcal{P}^E_{\text{HLag}}(\mathbf{\Phi}) = \sum_{i=1}^{k}\sum_{j=1}^{k}\sum_{\ell=1}^{L_{ij}}\|\mathbf{\Phi}_{ij}^{(\ell:L_{ij})}\|_2 \leq M\sum_{i=1}^{k}\sum_{j=1}^{k}L_{ij}^{3/2},$$

and finally,

$$\frac{1}{T}\sum_{t=1}^{T}\left\|\sum_{\ell=1}^{p}(\hat{\mathbf{\Phi}}^{(\ell)} - \mathbf{\Phi}^{(\ell)})\mathbf{y}_{t-\ell}\right\|_2^2 \lesssim Mv(\mathbf{\Phi}, \mathbf{\Sigma}_u)\sqrt{\frac{\log(pk^2)}{T}}\sum_{i=1}^{k}\sum_{j=1}^{k}L_{ij}^{3/2}.$$

■

Appendix B. Comparison Methods

B.1 Least Squares VAR

A standard method in lower dimensional settings is to fit a $VAR_k(\ell)$ with least squares for $0 \leq \ell \leq pmax$ and then to select a universal lag order ℓ using AIC or BIC. Per Lütkepohl (2007), the AIC and BIC of a $VAR_k(\ell)$ are defined as

$$\text{AIC}(\ell) = \log\det(\hat{\mathbf{\Sigma}}_u^{\ell}) + \frac{2k^2\ell}{T},$$
$$\text{BIC}(\ell) = \log\det(\hat{\mathbf{\Sigma}}_u^{\ell}) + \frac{\log(T)k^2\ell}{T},$$

in which $\hat{\boldsymbol{\Sigma}}_u^\ell$ is the residual sample covariance matrix having used least squares to fit the $VAR_k(\ell)$. The lag order ℓ that minimizes AIC(ℓ) or BIC(ℓ) is selected. This method of lag order selection is only possible when $k\ell \leq T$ since otherwise least squares is not well-defined. In simulation Scenarios 1-3 ($T = 100$), we cannot use least squares for $\ell > 1$, thus for a simple benchmark we instead estimate a $VAR_k(1)$ by least squares:

$$\min_{\boldsymbol{\nu},\boldsymbol{\Phi}} \left\{ \frac{1}{2T} \|\boldsymbol{Y} - \boldsymbol{\nu}\mathbf{1}^\top - \boldsymbol{\Phi}^{(1)}\boldsymbol{Z}^{(1)}\|_2^2 \right\},$$

where $\boldsymbol{Z}^{(1)} = [\mathbf{y_0} \ \cdots \ \mathbf{y_{T-1}}]$.

B.2 Lasso VAR

We also include two well-known lasso-based VAR regularization approaches. The *lasso* estimates the VAR using an L_1-penalty:

$$\min_{\boldsymbol{\nu},\boldsymbol{\Phi}} \left\{ \frac{1}{2T} \|\boldsymbol{Y} - \boldsymbol{\nu}\mathbf{1}^\top - \boldsymbol{\Phi}\boldsymbol{Z}\|_2^2 + \lambda\|\boldsymbol{\Phi}\|_1 \right\},$$

where $\|\boldsymbol{\Phi}\|_1$ denotes $\|\text{vec}(\boldsymbol{\Phi})\|_1$. The lasso does not intrinsically consider lag order, hence Song and Bickel (2011) propose a *lag-weighted lasso* penalty in which a weighted L_1-penalty is used with weights that increase geometrically with lag order:

$$\min_{\boldsymbol{\nu},\boldsymbol{\Phi}} \left\{ \frac{1}{2T} \|\boldsymbol{Y} - \boldsymbol{\nu}\mathbf{1}^\top - \boldsymbol{\Phi}\boldsymbol{Z}\|_2^2 + \lambda \sum_{\ell=1}^{p} \ell^\alpha \|\boldsymbol{\Phi}^{(\ell)}\|_1 \right\}.$$

The tuning parameter $\alpha \in [0, 1]$ determines how fast the penalty weight increases with lag. While this form of penalty applies greater regularization to higher order lags, it is less structured than our HLag penalties in that it does not necessarily produce sparsity patterns in which all coefficients beyond a certain lag order are zero. The regularization parameters λ and α are jointly selected using a two-dimensional penalty parameter search. We have implemented these methods in `R`, the code is available as Supplementary Material.

B.3 Bayesian VAR

We consider three Bayesian benchmarks: the method of Banbura et al. (2010), Giannone et al. (2015) and Carriero et al. (2019). These approaches are also applicable to a situation like ours where many parameters need to be estimated but the observation period is limited. However, in contrast to the HLag methods, these methods are not sparse (parameter estimates are only shrunken towards zero) and do not perform lag order selection.

Banbura et al. (2010) use a modified Minnesota prior which leads to a posterior for the autoregressive parameters, conditional on the error variance-covariance matrix, that is normal. As we transformed all variables for stationarity, we set all prior means in the BGR implementation to zeros. Following Banbura et al. (2010), we select the hyperparameter that controls the degree of regularization as that which minimizes the h-step ahead MSFE across the k component series. We have implemented this method in `R`, the code is available as Supplementary Material.

Giannone et al. (2015) choose the informativeness of the priors in an "optimal" way by treating the priors as additional parameters, as in hierarchical modeling. We use the authors' replication files (`Matlab`-code) publicly available at `https://www.newyorkfed.org/research/economists/giannone/pub`.

Carriero et al. (2019) use a general Minnesota-based independent prior to allow for a more flexible lag choice. Note that the authors also allow for stochastic volatility, but we compare the HLag methods to their "homoscedastic" BVAR that does not allow for stochastic volatility, in line with the other methods considered in this paper. We adapt the authors' code (publicly available at `http://didattica.unibocconi.eu/mypage/index.php?IdUte=49257&idr=27515&lingua=eng`) to this homoscedastic setting by combining it with `Matlab` code for BVAR using Gibbs sampling available at `https://sites.google.com/site/dimitriskorobilis/matlab/code-for-vars`. For full technical details on the Bayesian methods, we refer the reader to Banbura et al. (2010), Giannone et al. (2015) and Carriero et al. (2019) respectively.

B.4 Factor Models

We consider two factor-based benchmarks: a Dynamic Factor Model (DFM, see e.g. Forni et al., 2000; Stock and Watson (2002)) and a Factor Augmented VAR Model (FAVAR, Bernanke et al., 2005). In contrast to the HLag methods, these methods do not achieve dimension reduction by sparsity. Instead, the information contained in the large predictor set is summarized by few factors. We estimate the factors by Principal Component Analysis and follow McCracken and Ng (2016) in using the PC_{p2} criterion, developed in Bai and Ng (2002), to select the number of factors

Regarding the DFM, the time series are regressed on lagged values of the factors. The factors are obtained from the whole data set and their lag order is determined via AIC. Similar results are obtained with BIC and available from the authors upon request. Regarding the FAVAR model, we regress each time series on its own lagged values and lagged values of the factors. The factors are obtained from the data set of all other variables. Lag selection is done via AIC, while similar results are obtained with BIC. We have implemented both methods in `R`, the code is available as Supplementary Material.

B.5 Other Methods

Finally, we compare against three simple baselines. The unconditional *sample mean* corresponds to the intercept-only model,

$$\min_{\boldsymbol{\nu}} \frac{1}{2T} \|\boldsymbol{Y} - \boldsymbol{\nu}\mathbf{1}^\top\|_2^2,$$

which makes one-step-ahead forecasts of the form $\hat{\mathbf{y}}_{t+1} = \frac{1}{t}\sum_{\ell=1}^{t} \mathbf{y}_\ell$. The vector *random walk* model, which corresponds to

$$\hat{\boldsymbol{\nu}} = \mathbf{0}, \qquad \hat{\boldsymbol{\Phi}}^{(1)} = \mathbf{I}_k, \qquad \hat{\boldsymbol{\Phi}}^{(2:p)} = \mathbf{0},$$

and makes one-step-ahead forecasts of the form $\hat{\mathbf{y}}_{t+1} = \mathbf{y}_t$. Finally, we consider a separate autoregressive model for each time series. To simultaneously obtain parameter estimates

and select the lag order, we use the univariate analogue of equation (4)

$$\min_{\phi_i} \left\{ \frac{1}{2T} \|\mathbf{Y}_i - \boldsymbol{\phi}_i \mathbf{X}\|_2^2 + \lambda_i \sum_{\ell=1}^{p} ||\boldsymbol{\phi}_i^{(\ell:p)}||_2 \right\}.$$

for each component series $i = 1, \ldots, k$ with $\mathbf{X} = [\mathbf{x}_1 \ \cdots \ \mathbf{x}_T] \in \mathbb{R}^{p \times T}$, $\mathbf{x}_t = [y_{i,t-1} \ \cdots \ y_{i,t-p}]^\top \in \mathbb{R}^{p \times 1}$ and $\phi_i \in \mathbb{R}^{1 \times p}$. As such, the univariate AR is a special univariate case of the multivariate elementwise HLag introduced in Section 3.2. For each individual autoregression, we take the maximal autoregressive order equal to the true VAR order p in the simulations. In the empirical application we take four as maximal autoregressive order.

Appendix C. Simulation Study

C.1 Simulation Scenarios

Simulation Scenario 1: Componentwise Lag Structure. In this scenario, we simulate according to an $\text{HLag}_{45}^{C}(5)$ structure. In particular, we choose the maxlag matrix

$$\mathbf{L} = [1, 2, 3, 4, 5]^\top \otimes (\mathbf{1}_9 \mathbf{1}_{45}^\top).$$

This 45×45 maxlag matrix is row-wise constant, meaning that all components *within a row* have the same maxlag; we partition the rows into 5 groups of size 9, each group taking on a distinct maxlag in $\{1, 2, 3, 4, 5\}$. A coefficient matrix $\boldsymbol{\Phi}$ with maxlag matrix $\mathbf{L}$ is used in Scenario 1's simulations and its magnitudes are depicted in Figure 4, panel (1) of the manuscript.

Simulation Scenario 2: Own-Other Lag Structure. In this scenario, we create the matrix $\boldsymbol{\Phi}$ in such a manner that it differentiates between *own* and *other* coefficients. The coefficients of a series' "own lags" (i.e., $\boldsymbol{\Phi}_{ii}^{(\ell)}$) are larger in magnitude than those of "other lags" (i.e., $\boldsymbol{\Phi}_{ij}^{(\ell)}$ with $i \neq j$). The magnitude of coefficients decreases as the lag order increases. The $\text{HLag}_{45}^{O}(2)$ model we simulate is depicted in Figure 4, panel (2) of the manuscript. The first 15 rows can be viewed as univariate autoregressive models in which only the *own* term is nonzero; in the next 15 rows, for the first k coefficients, the coefficient on a series' *own* lags is larger than "other lags," and, for the next k coefficients, only *own* coefficients are nonzero; the final 15 rows have nonzeros throughout the first $2k$ coefficients, with *own* coefficients dominating *other* coefficients in magnitude.

Simulation Scenario 3: Elementwise Lag Structure. In this scenario, we simulate under an $\text{HLag}_{45}^{E}(4)$ model, meaning that the maxlag is allowed to vary not just across rows but also *within* rows. Each marginal series in each row is randomly assigned a maxlag of either 1 (with 90 percent probability) or 4 (with 10 percent probability). The coefficient matrices are depicted in Figure 4, panel (3).

Simulation Scenario 4: Data-based Lag Structure. Similar to Carriero et al. (2012), we carry out a simulation by bootstrapping the actual *Medium-Large* macroeconomic data set with $k = 40$ and $T = 195$ as discussed in Section 6 of the manuscript. We start from the estimates obtained by applying the Bayesian approach of Giannone et al. (2015) to this data set with $pmax = 4$. The obtained estimates of the autoregressive matrices are visualized in Figure 4, panel (4) and the autoregressive matrices verify the VAR stability conditions.

		Simulation Scenarios				
Order	HLag	1. Componentwise	2. Own-Other	3. Elementwise	4. Data	5. Robustness
Known	Componentwise	**1.036**	0.980	0.946	**1.019**	1.004
	Own-other	**1.048**	1.000	0.944	1.006	0.986
	Elementwise	**1.037**	1.001	0.944	**1.010**	1.005
Unkown	Componentwise	**1.138**	**1.235**	**1.094**	**1.051**	**1.030**
	Own-other	**1.119**	**1.171**	**1.064**	**1.031**	**1.014**
	Elementwise	**1.053**	**1.142**	**1.030**	**1.028**	**1.037**

Table 2: Out-of-sample mean squared forecast errors of the LOG relative to that of the nested group lasso in Scenario 1 to 5. Outperformance (as confirmed with paired t-tests) by the nested group lasso is indicated in bold.

We then construct our simulated data using a non-parametric residual bootstrap procedure (e.g., Kreiss and Lahiri, 2012) with bootstrap errors an i.i.d. sequence of discrete random variables uniformly distributed on $\{1, \ldots, T\}$.

C.2 Generation of Simulation Scenarios

All of our simulation structures were generated to ensure a stationary coefficient matrix, $\mathbf{\Phi}$. In order to construct a coefficient matrix for these scenarios, we started by converting the $\text{VAR}_k(p)$ to a $\text{VAR}_k(1)$ as described in equation 2.1.8 of Lütkepohl (2007)

$$\mathbf{A} = \begin{bmatrix} \mathbf{\Phi}^{(1)} & \mathbf{\Phi}^{(2)} & \cdots & \mathbf{\Phi}^{(p-1)} & \mathbf{\Phi}^{(p)} \\ \boldsymbol{I}_k & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{I}_k & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \boldsymbol{I}_k & \mathbf{0} \end{bmatrix} \tag{13}$$

For $\mathbf{A}$ to be stationary, its maximum eigenvalue must be less than 1 in modulus. In general, it is very difficult to generate stationary coefficient matrices. Boshnakov and Iqelan (2009) offer a potentially viable procedure that utilizes the unique structure of equation (13), but it does not allow for structured sparsity. We instead follow the approach put forth by Gilbert (2005) in which structured random coefficient matrices are generated until a stationary matrix is recovered.

C.3 Sensitivity Analysis: Choice of Group Lasso Formulation

The hierarchical lag structures of the HLag methods can either be enforced via the nested group structure of Zhao et al. (2009) or via the latent overlapping group lasso (LOG) proposed by Jacob et al. (2009). We compare the LOG to the nested group structures in our simulation studies.

In Table 2, we present the MSFEs of the LOG structures relative to those of the nested group lasso (for each HLag method). In Table 3 their lag order selection performance is compared. Values above one indicate better performance of the nested group lasso compared to the LOG. In both Tables, the nested group lasso significantly outperforms the LOG in

		Simulation Scenarios				
Measure	HLag	1. Componentwise	2. Own-Other	3. Elementwise	4. Data	5. Robustness
L_1-lag	Componentwise	**4.581**	**7.072**	**3.145**	1.002	**1.211**
error	Own-other	**4.837**	**3.271**	**3.076**	1.001	**1.177**
	Elementwise	**1.529**	**2.342**	**1.383**	1.001	**1.058**
L_∞-lag	Componentwise	**2.418**	**3.150**	**1.752**	**1.281**	**1.203**
error	Own-other	**2.513**	**4.979**	**1.780**	**1.138**	**1.741**
	Elementwise	**1.015**	**3.091**	**1.683**	**1.636**	**1.266**

Table 3: Lag order selection of the LOG relative to that of the nested group lasso in Scenario 1 to 5. Outperformance (as confirmed with paired t-tests) by the nested group lasso is indicated in bold.

Performance		Maximal lag order			
measure	HLag	$pmax = 5$	$pmax = 12$	$pmax = 25$	$pmax = 50$
MSFE	Componentwise	1.004	**1.030**	**1.061**	**1.114**
	Own-other	0.986	**1.014**	**1.049**	**1.105**
	Elementwise	1.005	**1.037**	**1.075**	**1.151**
L_1-lag	Componentwise	0.912	**1.211**	**1.331**	**1.969**
error	Own-other	**1.090**	**1.177**	**1.226**	**1.529**
	Elementwise	0.990	**1.058**	**1.100**	**1.149**
L_∞-lag	Componentwise	0.985	**1.203**	**1.399**	**2.491**
error	Own-other	**1.127**	**1.741**	**2.263**	**4.130**
	Elementwise	1.000	**1.266**	**1.898**	**2.630**

Table 4: Robustness simulation scenario: Forecast performance and lag order selection of the LOG relative to that of the nested group lasso for different values of the maximal lag order $pmax$. Outperformance (as confirmed with paired t-tests) by the nested group lasso is indicated in bold.

the vast majority of cases. Especially when the maximal lag order $pmax$ increases, the nested group lasso structures perform better than the LOG structures.

The finding that the nested group lasso structures are more robust than the LOG structures as $pmax$ increases, is confirmed through the Robustness simulation scenario. In Table 4, we report the MSFEs and lag order measures as $pmax$ increases from its true order (five) to $pmax = 50$. On all performance measures, the nested group lasso structures perform, overall, better than the LOG structures and the margin by which the former outperforms the latter increases with $pmax$.

C.4 Sensitivity Analysis: Impact of Increasing the Time Series Length

We investigate the impact of increasing the time series length on our forecast accuracy results. We use the autoregressive parameter structure of Scenario 5 and increase the time series length from from $T = 200$ over $T = 500$ to $T = 1000$ while keeping the maximal lag

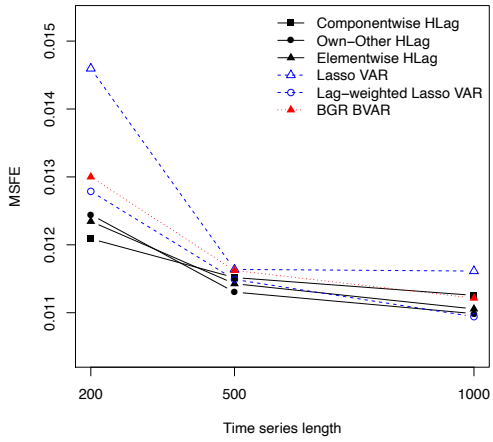


Figure 17: Robustness simulation scenario: Out-of-sample mean squared forecast errors, for different values of the sample size T. Note that we have not included the BVAR methods GLP and CCL as they are too time consuming for large-scale VARs.

order $pmax = 5$. Figure 17 presents the MSFEs. The forecast errors of all methods decrease as T increases, in line with our expectations. While the difference between the methods decreases as the sample size increases, all HLag methods sill significantly outperform the lasso.

C.5 Sensitivity Analysis: Choice of Error Covariance matrix

We investigate the sensitivity of our forecast accuracy results to the choice of error covariance matrix. We start from the autoregressive parameter structure of Scenario 5 ($pmax = 5$) and consider, in turn, robustness to (i) varying the signal-to-noise ratio, (ii) unequal error variances and (iii) time variation in the error covariance matrix (i.e. stochastic volatility).

Signal-to-noise ratio. In the paper, we consider $\mathbf{\Sigma}_u = 0.01 \cdot \mathbf{I}_k$, corresponding to a signal-to-noise ratio[4] of around 100. To investigate the sensitivity of the results to a lower signal-to-noise ratio, we re-ran the simulation study with $\mathbf{\Sigma}_u = 0.1 \cdot \mathbf{I}_k$, corresponding to a signal-to-noise ratio around 10 and $\mathbf{\Sigma}_u = \mathbf{I}_k$, corresponding to a signal-to-noise ratio around one.

Unequal error variances. We investigate whether the HLag methods behave comparably if one group of time series has a large residual variance and another group has a small residual variance. To this end, we consider one group (series 1 to 5) with residual variance one, and the other group (series 6 to 10) with residual variance equal to 0.5.

4. Defined as the maximum eigenvalue of the parameter matrix over the maximum eigenvalue of the error covariance matrix.

Class	Method	SNR≈100 (in paper)	SNR≈10	SNR≈1	Unequal Variances	Stochastic Volatility
HLag	Componentwise	0.0125 (0.0003)	0.1245 (0.0026)	1.1814 (0.0228)	0.9222 (0.0412)	3.5548 (0.2148)
	Own-other	0.0128 (0.0003)	0.1278 (0.0027)	1.2075 (0.0234)	0.9414 (0.0421)	3.6507 (0.2188)
	Elementwise	0.0126 (0.0003)	0.1262 (0.0026)	1.2000 (0.0230)	0.9414 (0.0421)	3.6122 (0.2168)
VAR	Lasso	0.0131 (0.0003)	0.1305 (0.0027)	1.2365 (0.0236)	0.9708 (0.0434)	3.6818 (0.2196)
	Lag-weighted lasso	0.0144 (0.0007)	0.1439 (0.0066)	1.5636 (0.1402)	1.0803 (0.0483)	4.2317 (0.2927)
	Least squares AIC	0.0136 (0.0003)	0.1358 (0.0029)	1.3250 (0.0261)	1.0134 (0.0453)	3.9678 (0.2260)
	Least squares BIC	0.0155 (0.0003)	0.1554 (0.0034)	1.5189 (0.0304)	1.2491 (0.0559)	3.9970 (0.2333)
	VAR(1)	0.0164 (0.0004)	0.1643 (0.0035)	1.5859 (0.0312)	1.2555 (0.0276)	4.4612 (0.2248)
BVAR	BGR	0.0129 (0.0003)	0.1295 (0.0027)	1.2325 (0.0240)	0.9688 (0.0433)	3.6102 (0.2044)
	GLP	0.0125 (0.0003)	0.1253 (0.0026)	1.2054 (0.0232)	0.9312 (0.0194)	3.6572 (0.2282)
	CCM	0.0129 (0.0003)	0.1274 (0.0026)	1.2128 (0.0236)	0.9430 (0.0204)	3.5280 (0.2088)
Factor	DFM	0.0214 (0.0005)	0.2142 (0.0054)	1.9738 (0.0421)	1.5231 (0.0395)	4.6188 (0.2338)
	FAVAR	0.0191 (0.0004)	0.1913 (0.0043)	1.7898 (0.0380)	1.2990 (0.0293)	4.3722 (0.2346)
Other	AR	0.0475 (0.0013)	0.4753 (0.0134)	4.5135 (0.1333)	3.4895 (0.0987)	11.9719 (0.6454)
	Sample mean	0.2067 (0.0083)	2.0675 (0.0826)	20.0255 (0.8383)	14.5514 (0.6508)	69.9780 (7.4943)
	Random walk	0.6268 (0.0256)	6.2679 (0.2561)	61.9335 (2.7009)	44.8748 (2.0068)	223.4107 (26.2932)

Table 5: Robustness to various choices of error covariance matrix: Out-of-sample mean squared forecast error (standard errors are in parentheses).

Stochastic volatility. As stochastic volatility is an important feature for macroeconomic forecasting (Clark and Ravazzolo, 2015), we investigate the performance of all methods in the presence of parametric variation in the error covariance matrix. Note that none of the methods considered in this paper account for stochastic volatility and, hence, their forecast accuracy is expected to suffer. Nevertheless, it remains interesting to investigate their sensitivity to the presence of parametric variation in the VAR errors.

We consider the VAR-SV model of Clark and Ravazzolo (2015) which includes the conventional macroeconomic formulation of a random walk process for log volatility. In particular, we take

$$\mathbf{u}_t = \mathbf{A}^{-1}\mathbf{\Lambda}_t^{0.5}\boldsymbol{\varepsilon}_t,$$

with $\boldsymbol{\varepsilon}_t \sim N(\mathbf{0}, \mathbf{I}_k)$, $\mathbf{A} = \begin{bmatrix} 1 & \mathbf{0} \\ \mathbf{0.5} & \mathbf{I}_{k-1} \end{bmatrix}$ and $\mathbf{\Lambda}_t = \text{diag}(\lambda_{1,t}, \ldots, \lambda_{k,t})$ where

$$\log(\lambda_{i,t}) = \log(\lambda_{i,t-1}) + v_{i,t},$$

with $v_{i,t} = (v_{1,t}, \ldots, v_{k,t})^\top \sim N(\mathbf{0}, 0.01 \cdot \mathbf{I}_k)$.

Table 5 gives the forecast performance of the methods under the various choices of error covariance matrix. When decreasing the signal-to-noise ratio, the forecast accuracy of all methods decreases accordingly, as expected. Similarly, under unequal error variances and in the presence of stochastic volatility, the forecast accuracy of all methods suffers compared to their performance in the original design (column 1). Importantly, the relative performance of the HLag methods to the other methods is, mainly, unaffected. One exception concerns the presence of stochastic volatility where even the homoscedastic BVAR of Carriero et al. (2019), which does not account for stochastic volatility, outperforms the HLag methods. Their heteroskedastic BVAR, which accounts for stochastic volatility, is expected to perform even better in such settings.

C.6 Relaxed VAR Estimation

Since the lasso and its structured counterparts are known to shrink non-zero regression coefficients, in practice, they are often used for model selection, followed by refitting the reduced model using least squares (Meinshausen, 2007). In this section, we detail our approach to refit based on the support selected by our procedures while taking into consideration both numerical stability as well as computational efficiency.

Let $\widehat{\mathbf{\Phi}}$ denote the coefficient matrix recovered from one of our sparsity-imposing algorithms (e.g. HLag, Lasso-VAR) and suppose that it contains r nonzero coefficients. In order to take the support recovered into account we introduce $\mathbf{V}$, a $k^2p \times r$ *restriction matrix* of rank r that denotes the location of nonzero elements in $\hat{\mathbf{\Phi}}$. Defining β as the vec of the nonzero entries of $\widehat{\mathbf{\Phi}}$, we obtain the relationship

$$\text{vec}(\hat{\mathbf{\Phi}}) = \mathbf{V}\beta.$$

We can then express the *Relaxed Least Squares* estimator as:

$$\text{vec}(\widehat{\mathbf{\Phi}}_{\text{Relaxed}}) = \mathbf{V}[\mathbf{V}^\top(\boldsymbol{Z}\boldsymbol{Z}^\top \otimes \boldsymbol{I}_k)\mathbf{V}]^{-1}\mathbf{V}^\top(\boldsymbol{Z} \otimes \boldsymbol{I}_k)\text{vec}(\boldsymbol{Y}), \tag{14}$$

in which $\otimes$ denotes the Kronecker operator. In general, it is ill-advised to directly form equation (14). First, performing matrix operations with $\boldsymbol{Z} \otimes \boldsymbol{I}_k$, which has dimension $kT \times k^2p$, can be very computationally demanding, especially if k is large. Second, in the event that $r \approx T$, the resulting estimator can be very poorly conditioned. To obviate these two concerns, we propose a slight adaptation of the techniques detailed in Neumaier and Schneider (2001) that computes a variant of equation (14) using a QR decomposition to avoid explicit matrix inversion. Additionally, if the resulting matrix is found to be ill-conditioned, a small ridge penalty should be utilized to ensure numerically-stable solutions.

C.7 Refinements

As opposed to performing a Kronecker expansion we instead consider imposing the restrictions by row in $\widehat{\mathbf{\Phi}}$ and define $V_1, \ldots, V_k$ as $kp \times r_i$ restriction matrices of rank $r_1, \ldots, r_k$, denoting the number of nonzero elements in each row of $\mathbf{\Phi}$. We can then calculate each row of $\widehat{\mathbf{\Phi}}_{\text{Relaxed}}$ by

$$\widehat{\mathbf{\Phi}}_{\text{Relaxed}_i} = \left(V_i(V_i^\top \boldsymbol{Z}\boldsymbol{Z}^\top V_i)^{-1}V_i^\top \boldsymbol{Z}\boldsymbol{Y}_i\right)^\top.$$

Now, following Neumaier and Schneider (2001), construct the matrix $\mathbf{K}_i = [(V_i\boldsymbol{Z})^\top, \boldsymbol{Y}_i]$. We then compute a QR factorization of $\mathbf{K}_i$

$$\mathbf{K}_i = QR,$$

in which Q is an orthogonal matrix and R is upper triangular of the form:

$$R = \begin{array}{c} \begin{matrix} r_i & 1 \end{matrix} \\ \begin{bmatrix} R_{11} & R_{12} \\ 0 & R_{22} \end{bmatrix} \end{array} \begin{array}{l} r_i \\ T - r_i \end{array}$$

As expanded upon in Neumaier and Schneider (2001), we can compute

$$\begin{aligned}\widehat{\mathbf{\Phi}}_{\text{Relaxed}_i} &= \left(V_i R_{12}^\top R_{11} (R_{11}^\top R_{11})^{-1}\right)^\top, \\ &= \left(V_i R_{12}^\top R_{11} R_{11}^{-1} (R_{11}^\top)^{-1}\right)^\top, \\ &= \left(V_i R_{12}^\top (R_{11}^\top)^{-1}\right)^\top, \\ &= \left(V_i (R_{11}^{-1} R_{12})^\top\right)^\top,\end{aligned}$$

which can be evaluated with a triangular solver, hence does not require explicit matrix inversion. In the event that $\mathbf{K}$ is poorly conditioned, to improve numerical stability, we add a small ridge penalty. It is suggested by Neumaier and Schneider (2001) to add a penalty corresponding to scaling a diagonal matrix D consisting of the Euclidean norms of the columns of $\mathbf{K}$ by $(r_i^2 + r_i + 1)\epsilon_{\text{machine}}$, in which $\epsilon_{\text{machine}}$ denotes machine precision. The full refitting algorithm is detailed in Algorithm 3.

Algorithm 3 Relaxed Least Squares

Require: $\boldsymbol{Z}, \boldsymbol{Y}, V_1, \ldots, V_k$
  for $i = 1, 2, \ldots, k$ **do**
    $\mathbf{K}_i \leftarrow [(V_i \boldsymbol{Z})^\top, Y_i]$
    $D \leftarrow (r_i^2 + r_i + 1)\epsilon_{\text{machine}} \text{diag}(\|\mathbf{K}_{i\cdot}\|_2)$
    $R, Q \leftarrow QR(\begin{bmatrix}\mathbf{K}_i \\ D\end{bmatrix})$
    $\widehat{\mathbf{\Phi}}_{\text{Relaxed}_i} \leftarrow \left(V_i (R_{11}^{-1} R_{12})^\top\right)^\top$
  end for
  return $\widehat{\mathbf{\Phi}}_{\text{Relaxed}}$.

Appendix D. Stock and Watson Application

To make the $k = 168$ variables of the Stock and Watson data approximately stationary, we apply the transformation codes provided by Stock and Watson (2005). A brief description of each variable, along with the transformation code to make them approximately stationary can be found in the Data Appendix of Koop (2013).

All 168 variables are classified into one of 13 macroeconomic categories, detailed in Table 6. The good performance of the HLag methods across all variables is confirmed by a sub-analysis on the 13 macroeconomic categories. Figure 18 breaks down the results of the *Large* VAR by the 13 macroeconomic categories. Generally speaking, the flexible elementwise HLag is the best performing forecasting method; for 10 out of 13 categories, it is included in the MCS. The second best performing methods are own-other HLag and the lag-weighted lasso (both for 6 out of 13 categories in the MCS).

Upon examination of the different categories, three groups can be distinguished. The first group consists of categories with a single preferred forecast method, always an HLag method. Elementwise HLag is preferred for interest rates and money; own-other HLag for employment series. The second group consists of categories with several, but a limited number (between 2 and 4) of preferred methods. Series in the second group are major

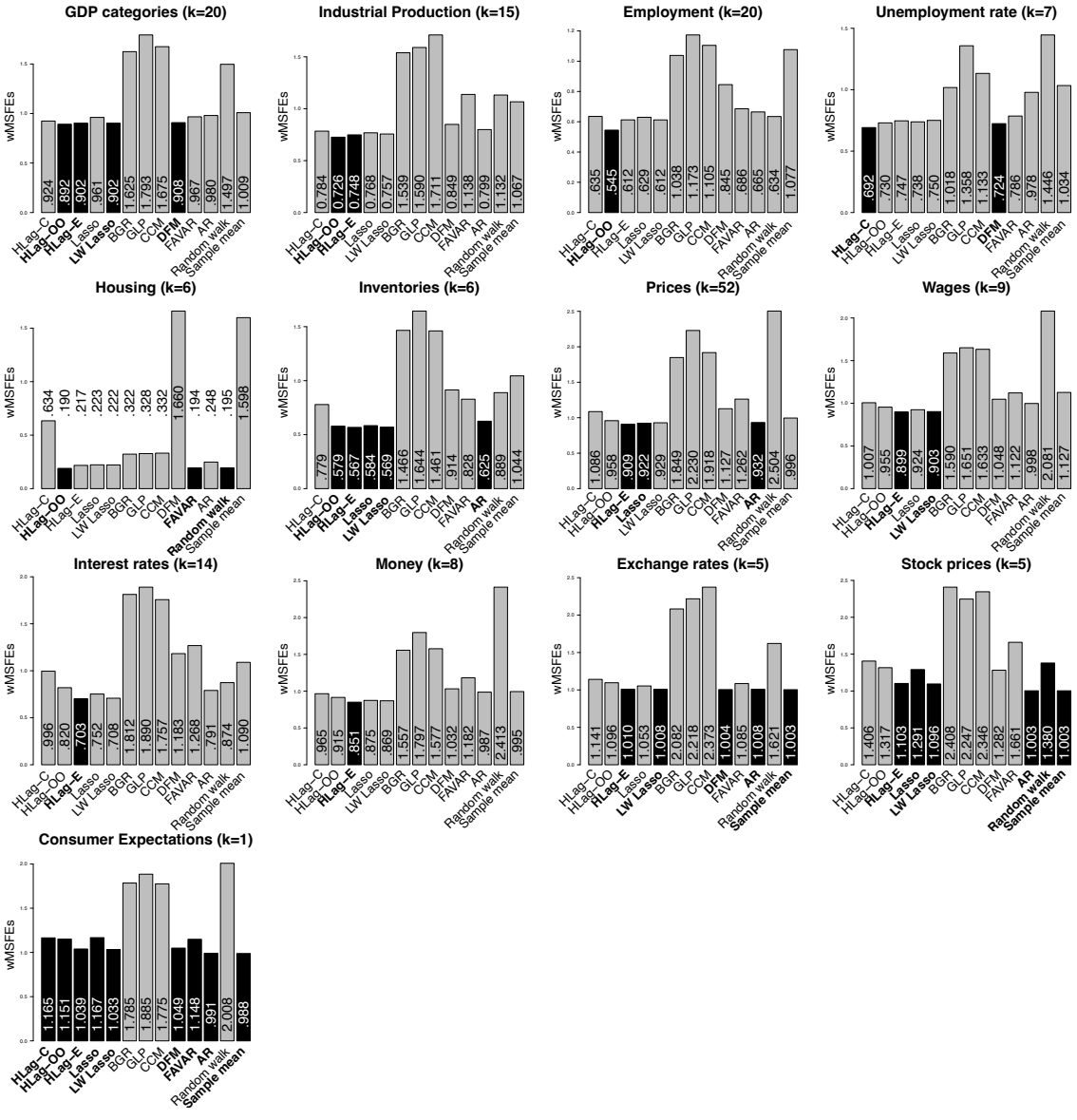


Figure 18: Rolling out-of-sample one-step ahead $wMSFE$ of different categories of macroeconomic indicators in the *Large* VAR. For each category, forecast methods in the 75% MCS are in black.

Group	Brief description	Examples of series	Number of series
1	GDP components	GDP, consumption, investment	20
2	IP	IP, capacity utilization	15
3	Employment	Sectoral and total employment and hours	20
4	Unemployment rate	Unemployment rate, total and by duration	7
5	Housing	Housing starts, total and by region	6
6	Inventories	NAPM inventories, new orders	6
7	Prices	Price indexes, aggregate and disaggregate; commodity prices	52
8	Wages	Average hourly earnings, unit labor cost	9
9	Interest rates	Treasuries, corporate, term spreads, public-private spreads	14
10	Money	M1, M2, business loans, consumer credit	8
11	Exchange rates	Average and selected trading partners	5
12	Stock prices	Various stock price indexes	5
13	Consumer expectations	Michigan consumer expectations	1

Table 6: Macroeconomic categories of series in the 168-variable data set, following the classification of Stock and Watson (2012) their Table 1.

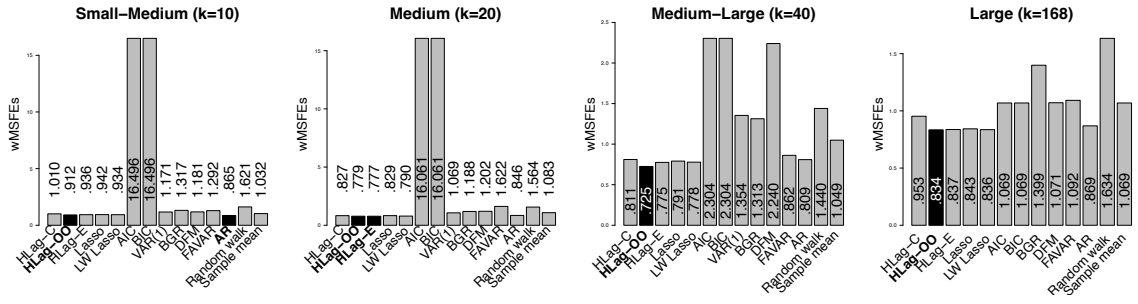


Figure 19: Rolling out-of-sample one-step-ahead $wMSFE$ for the four VAR sizes with $pmax = 13$. For each VAR size, forecast methods in the 75% Model Confidence Set are in black.

measures of real economic activity (GDP components, industrial production, unemployment rate, prices), housing, and wages. The strong performance of elementwise and own-other HLag is re-confirmed in the majority of cases (3 out of 5 categories), but the MCS is extended by the lag-weighted lasso and DFM (2 out of 5 categories), or componentwise HLag, FAVAR and random walk (1 out of 5 categories). The third group consists of categories which have a larger number of preferred forecast methods, like inventories and hard-to-predict series such as exchange rates, stock prices and consumer expectations. For the latter categories, in line with Stock and Watson (2012), we find multivariate forecast methods to provide no meaningful reductions over simple univariate methods (AR or sample mean).

Results on additional sensitivity analyses concerning the choice of the maximal lag order pmax and forecasts horizon are provided in Figures 19 and 20 respectively. Results on the stability of the lag selection results are displayed in Figure 21.

We focus on the Stock and Watson macroeconomic data set since it is readily available and popular in the literature on macroeconomic forecasting. A more recent variant is

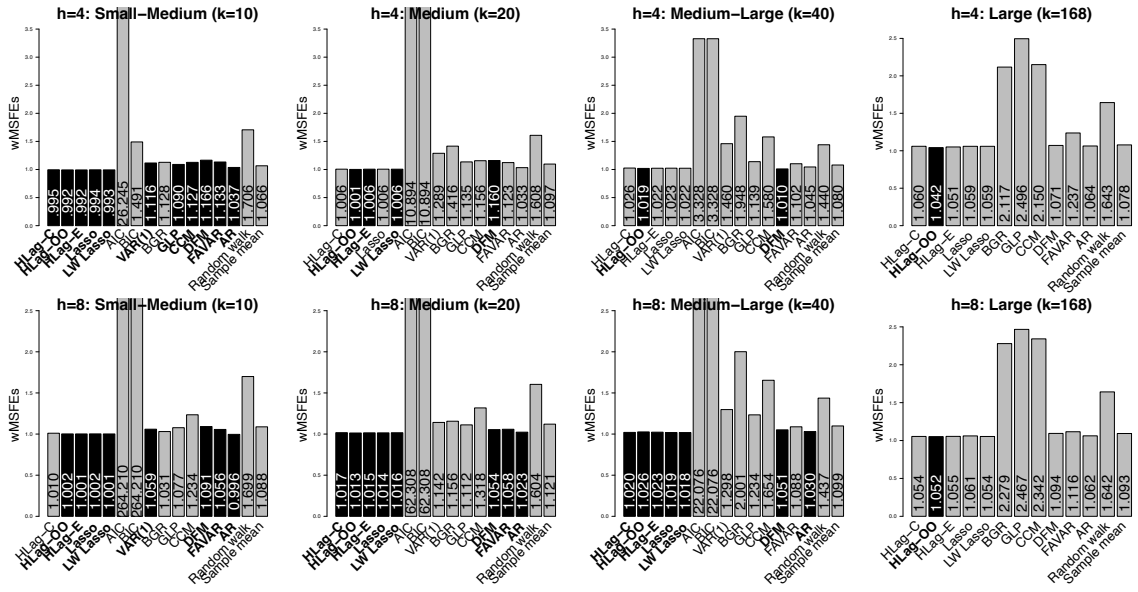


Figure 20: Rolling out-of-sample one-step-ahead $wMSFE$ for the four VAR sizes at forecast horizon $h = 4$ (top) and $h = 8$ (bottom). For each VAR size, forecast methods in the 75% MCS are in black.

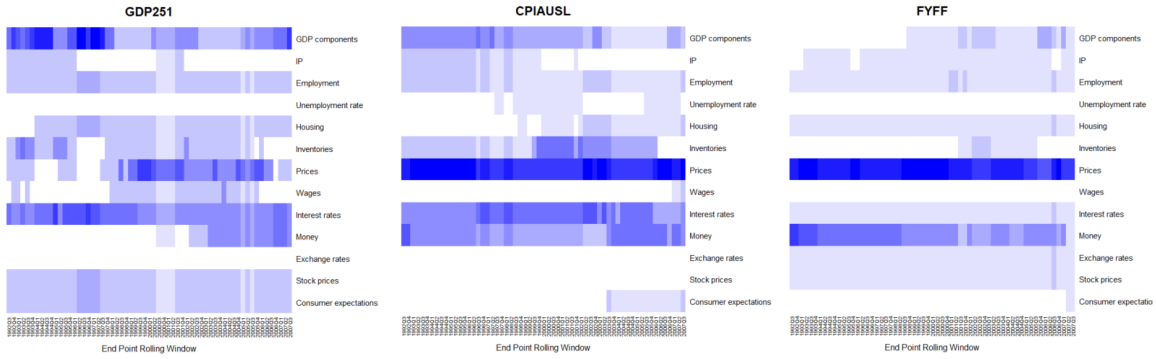


Figure 21: Fraction of non-zero coefficients in each of the 13 macro-economic categories to the total number of non-zero coefficients in the *Medium-Large* VAR estimated by elementwise HLag when forecasting *GDP251* (GDP growth, left), *CPIAUSL* (inflation, middle) and *FYFF* (Federal Funds Rate, right). The horizontal axis represents the ending date of a rolling window.

available as the FRED-QD data set, a quarterly version of the Federal Reserve Economic Data database introduced in McCracken and Ng (2016). We have performed the same empirical analysis on the FRED-QD containing $k = 210$ variables from Quarter 3, 1959 to Quarter 4, 2018 ($T = 238$). Similar findings are obtained: (i) Own-other and elementwise HLag perform comparable to the lasso methods and AR for small VAR sizes, but outperform all others for the *Large* VAR and a short forecast horizon. (ii) Own-other HLag is the preferred forecast method for several major macroeconomic indicators such as national income and product accounts and industrial production. For difficult to predict indicators, such as exchange rates, gains over the AR model are difficult to attain.

Appendix E. Financial Application

The financial data set contains information on the realized variances of $k = 16$ stock market indices listed in Table 7. All time series are log-transformed to make them stationary.

Variable	Description
AEX	Amsterdam Exchange Index
AORD	All Ordinaries Index
BFX	Belgium Bell 20 Index
BVSP	BOVESPA Index
DJI	Dow Jones Industrial Average
FCHI	Cotation Assistée en Continu Index
FTSE	Financial Times Stock Exchange Index 100
GDAXI	Deutscher Aktienindex
HSI	HANG SENG Index
IXIC	Nasdaq stock index
KS11	Korea Composite Stock Price Index
MXX	IPC Mexico
RUT	Russel 2000
SPX	Standard & Poor's 500 market index
SSMI	Swiss market index
STOXX50E	EURO STOXX 50

Table 7: Variables used in the financial application.

Appendix F. Energy Application

The energy data set contains information on $k = 26$ variables. A brief description of each variable, taken from https://archive.ics.uci.edu/ml/data sets/Appliances+energy+prediction, is provided in Table 8, along with the transformation code to make it approximately stationary. The transformation codes are: 1 = first difference of logged variables, 2 = first difference.

Variable	Description	Code
Appliances	energy use in Wh	1
Lights	energy use of light fixtures in the house in Wh	2
T1	Temperature in kitchen area, in Celsius	1
RH1	Humidity in kitchen area, in %	1
T2	Temperature in living room area, in Celsius	1
RH2	Humidity in living room area, in %	1
T3	Temperature in laundry room area	1
RH3	Humidity in laundry room area, in %	1
T4	Temperature in office room, in Celsius	1
RH4	Humidity in office room, in %	1
T5	Temperature in bathroom, in Celsius	1
RH5	Humidity in bathroom, in %	1
T6	Temperature outside the building (north side), in Celsius	2
RH6	Humidity outside the building (north side), in %	1
T7	Temperature in ironing room , in Celsius	1
RH7	Humidity in ironing room, in %	1
T8	Temperature in teenager room 2, in Celsius	1
RH8	Humidity in teenager room 2, in %	1
T9	Temperature in parents room, in Celsius	1
RH9	Humidity in parents room, in%	1
To	Temperature outside (from Chievres weather station), in Celsius	2
Pressure	From Chievres weather station, in mm Hg	1
RHout	Humidity outside (from Chievres weather station), in %	1
Wind speed	From Chievres weather station in m/s	2
Visibility	From Chievres weather station, in km	1
Tdewpoint	From Chievres weather station, C	2

Table 8: Variables used in the energy application.

References

H. Akaike. Fitting autoregressive models for prediction. *Annals of the institute of Statistical Mathematics*, 21(2):243–247, 1969.

F. Bach, R. Jenatton, J. Mairal, and G. Obozinski. Structured sparsity through convex optimization. *Statistical Science*, 27(4):450–468, 2012.

J. S. Bai and S. Ng. Determining the number of factors in approximate factor models. *Econometrica*, 70(1):191–221, 2002.

M. Banbura, D. Giannone, and L. Reichlin. Large Bayesian vector auto regressions. *Journal of Applied Econometrics*, 25(1):71–92, 2010.

S. Basu and G. Michailidis. Regularized estimation in sparse high-dimensional time series models. *The Annals of Statistics*, 43(4):1535–1567, 2015.

S. Basu, X. Q. Li, and G. Michailidis. Low rank and structured modeling of high-dimensional vector autoregressions. *IEEE Transactions on Signal Processing*, 67(5):1207–1222, 2019.

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.

B. S. Bernanke, J. Boivin, and P. Eliasz. Measuring the effects of monetary policy: A factor-augmented vector autoregressive (FAVAR) approach. *Quarterly Journal of Economics*, 120(1):387–422, 2005.

J. Bien, J. Taylor, and R. Tibshirani. A lasso for hierarchical interactions. *The Annals of Statistics*, 41(3):1111–1141, 2013.

J. Bien, F. Bunea, and L. Xiao. Convex banding of the covariance matrix. *Journal of the American Statistical Association*, 111(514):834–845, 2016.

G. N. Boshnakov and B. M. Iqelan. Generation of time series models with given spectral properties. *Journal of Time Series Analysis*, 30(3):349–368, 2009.

G. E. P. Box and G. C. Tiao. A Canonical Analysis of Multiple Time Series. *Biometrika*, 64(2):355–365, 1977.

L.M. Candanedo, V. Feldheim, and D. Deramaix. Data driven prediction models of energy use of appliances in a low-energy house. *Energy and buildings*, 140:81–97, 2017.

A. Carriero, G. Kapetanios, and M. Marcellino. Forecasting large datasets with Bayesian reduced rank multivariate models. *Journal of Applied Econometrics*, 26(5):735–761, 2011.

A. Carriero, G. Kapetanios, and M. Marcellino. Forecasting government bond yields with large Bayesian vector autoregressions. *Journal of Banking & Finance*, 36(7):2026–2047, 2012.

A. Carriero, T. E. Clark, and M. Marcellino. Large Bayesian vector autoregressions with stochastic volatility and non-conjugate priors. *Journal of Econometrics*, 212(1):137–154, 2019.

C. Chen, Z. X. Peng, and J. Z. Huang. O(1) algorithms for overlapping group sparsity. In *2014 22nd International Conference on Pattern Recognition*, pages 1645–1650. IEEE, 2014.

T. E. Clark and F. Ravazzolo. Macroeconomic forecasting performance uder alternative specifications of time-varying volatility. *Journal of Applied Econometrics*, 30(4):551–575, 2015.

R. A. Davis, P. F. Zang, and T. Zheng. Sparse vector autoregressive modeling. *Journal of Computational and Graphical Statistics*, 25(4):1077–1096, 2016.

S. Ding and S. Karlsson. Bayesian VAR models with asymmetric lags. Technical report, Orebro University, Orebro University School of Business, Orebro University, Sweden, 2014.

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.

J. Friedman, T. Hastie, and R. Tibshirani. Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, 33(1):1–22, 2010.

J. Friedman, T. Hastie, R. Tibshirani, N. Simon, B. Narasimhan, and J. Qian. *glmnet: Lasso and Elastic-Net Regularized Generalized Linear Models*, 2017. R package version 2.0-13, https://CRAN.R-project.org/package=glmnet.

D. Giannone, M. Lenza, and G. E. Primiceri. Prior selection for vector autoregressions. *Review of Economics and Statistics*, 97(2):436–451, 2015.

D. Giannone, M. Lenza, and G. E. Primiceri. Economic predictions with big data: The illustion of sparsity. *Working paper*, 2017.

P. Gilbert. Brief user's guide: Dynamic systems estimation (dse). 2005.

J. Gonzalo and J.-Y. Pitarakis. Lag length estimation in large dimensional systems. *Journal of Time Series Analysis*, 23(4):401–423, 2002.

M. Gredenhoff and S. Karlsson. Lag-length selection in VAR-models using equal and unequal lag-length procedures. *Computational Statistics*, 14(2):171–187, 1999.

F. Han, H. R. Lu, and H. Liu. A direct estimation of high dimensional stationary vector autoregressions. *Journal of Machine Learning Research*, 16:3115–3150, 2015.

P. R. Hansen, A. Lunde, and J. M. Nason. The model confidence set. *Econometrica*, 79(2): 453–497, 2011.

A. Haris, D. Witten, and N. Simon. Convex modeling of interactions with strong heredity. *Journal of Computational and Graphical Statistics*, 25(4):981–1004, 2016.

T. Hastie, R. Tibshirani, and J. Friedman. *The elements of statistical learning*. Springer, 2009.

M. S. Ho and B. E. Sorensen. Finding cointegration rank in high dimensional systems using the Johansen test: An illustration using data based Monte Carlo simulations. *Review of Economics and Statistics*, 78(4):726–732, 1996.

C. Hsiao. Autoregressive modelling and money-income causality detection. *Journal of Monetary Economics*, 7(1):85–106, 1981.

N. J. Hsu, H. L. Hung, and Y. M. Chang. Subset selection for vector autoregressive processes using lasso. *Computational Statistics & Data Analysis*, 52(7):3645–3657, 2008.

C. M. Hurvich and C. L. Tsai. Regression and time series model selection in small samples. *Biometrika*, 76(2):297–307, 1989.

A. Hyvärinen, K. Zhang, S. Shimizu, and P. O. Hoyer. Estimation of structural vector autoregression model using non-guassianity. *Journal of Machine Learning Research*, 11: 1709–1731, 2010.

L. Jacob, G. Obozinski, and J. P. Vert. Group lasso with overlap and graph lasso. *Proceedings of the 26th Annual International Conference on Machine Learning. ICML'09*, pages 433–440, 2009.

R. Jenatton, J. Mairal, G. Obozinski, and F. R. Bach. Proximal methods for sparse hierarchical dictionary learning. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 487–494, 2010.

R. Jenatton, J. Y. Audibert, and F. Bach. Structured variable selection with sparsity-inducing norms. *Journal of Machine Learning Research*, 12:2777–2824, 2011a.

R. Jenatton, J. Mairal, G. Obozinski, and F. Bach. Proximal methods for hierarchical sparse coding. *Journal of Machine Learning Research*, 12:2297–2334, 2011b.

J. W. Keating. Macroeconomic modeling with asymmetric vector autoregressions. *Journal of Macroeconomics*, 22(1):1–28, 2000.

G. Koop. Forecasting with medium and large Bayesian VARs. *Journal of Applied Econometrics*, 28(2):177–203, 2013.

G. Koop and D Korobilis. Large time-varying parameter VARs. *Journal of Econometrics*, 177(2):185–198, 2013.

G. Koop, D. Korobilis, and D. Pettenuzzo. Bayesian compressed vector autoregressions. *Journal of Econometrics*, 210(1):135–154, 2019.

J. P. Kreiss and S. Lahiri. *Bootstrap methods for time series*. In: Rao, T., Rao, S. and Rao, C. (Eds.) Handbook of Statistics 30. Time Series Analysis: Methods and Applications. North Holland, 2012.

C. L. Leng, Y. Lin, and G. Wahba. A note on the lasso and related procedures in model selection. *Statistica Sinica*, 16(4):1273–1284, 2006.

M. Lim and T. Hastie. Learning interactions via hierarchical group-lasso regularization. *Journal of Computational and Graphical Statistics*, 24(3):627–654, 2015.

J. H. Lin and G. Michailidis. Regularized estimation and testing for high-dimensional multi-block vector-autoregressive models. *Journal of Machine Learning Research*, 18:117, 2017.

R. B. Litterman. Techniques of forecasting using vector autoregressions. Working papers, Federal Reserve Bank of Minneapolis, 1979.

Y. Lou, J. Bien, R. Caruana, and J. Gehrke. Sparse partially linear additive models. *Journal of Computational and Graphical Statistics*, 25(4):1026–1040, 2016.

A. C. Lozano, N. Abe, Y. Liu, and S. Rosset. Grouped graphical Granger modeling for gene expression regulatory networks discovery. *Bioinformatics*, 25(12):i110–i118, 2009.

H. Lütkepohl. Comparison of criteria for estimating the order of a vector autoregressive process. *Journal of Time Series Analysis*, 6(1):35–52, 1985.

H. Lütkepohl. *New introduction to multiple time series analysis*. Springer, 2007.

J. Mairal, R. Jenatton, F. R. Bach, and G. Obozinski. Network flow algorithms for structured sparsity. In *Advances in Neural Information Processing Systems*, pages 1558–1566, 2010.

D. S. Matteson and R. S. Tsay. Dynamic orthogonal components for multivariate time series. *Journal of the American Statistical Association*, 106(496):1450–1463, 2011.

M. W. McCracken and S. Ng. FRED-MD: A monthly database for macroeconomic research. *Journal of Business & Economic Statistics*, 34(4):574–589, 2016.

N. Meinshausen. Relaxed lasso. *Computational Statistics & Data Analysis*, 52(1):374–393, 2007.

I. Melnyk and A. Banerjee. Estimating strucured vector autoregressive models. *Proceedings of the 33rd International Conference on Machine Learning*, 2016.

A. Neumaier and T. Schneider. Estimation of parameters and eigenmodes of multivariate autoregressive models. *ACM Transactions on Mathematical Software (TOMS)*, 27(1): 27–57, 2001.

W. B. Nicholson, D. S. Matteson, and J. Bien. VARX-L: Structured regularization for large vector autoregressions with exogenous variables. *International Journal of Forecasting*, 33 (3):627–651, 2017.

G. Nickelsburg. Small-sample properties of dimensionality statistics for fitting VAR models to aggregate economic data: A Monte Carlo study. *Journal of Econometrics*, 28(2): 183–192, 1985.

D. Percival. Theoretical properties of the overlapping groups lasso. *Electronic Journal of Statistics*, 6:269–288, 2012.

P. Radchenko and G. M. James. Variable selection using adaptive nonlinear interaction structures in high dimensions. *Journal of the American Statistical Association*, 105(492): 1541–1553, 2010.

Y. Y. She, Z. F. Wang, and H. Jiang. Group regularized estimation under structural hierarchy. *Journal of the American Statistical Association*, 113(521):445–454, 2018.

A. Shojaie and G. Michailidis. Discovering graphical Granger causality using the truncating lasso penalty. *Bioinformatics*, 26(18):i517–i523, 2010.

A. Shojaie, S. Basu, and G. Michailidis. Adaptive thresholding for reconstructing regulatory networks from time-course gene expression data. *Statistics in Biosciences*, 4(1):66–83, 2012.

C. A. Sims. Macroeconomics and reality. *Econometrica*, 48(1):1–48, 1980.

S. Song and P. Bickel. Large vector auto regressions. *arXiv preprint arXiv:1106.3915*, 2011.

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.

J. H. Stock and M. W. Watson. An empirical comparison of methods for forecasting using many predictors. *Manuscript, Princeton University*, 2005.

J. H. Stock and M. W. Watson. Generalized shrinkage methods for forecasting using many predictors. *Journal of Business & Economic Statistics*, 30(4):481–493, 2012.

G. C. Tiao and R. S. Tsay. Model specification in multivariate time series. *Journal of the Royal Statistical Society. Series B (Methodological)*, 51(2):157–213, 1989.

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

R. Tibshirani and X. T. Suo. An ordered lasso and sparse time-lagged regression. *Technometrics*, 58(4):415–423, 2016.

R. S. Tsay. *Multivariate Time Series Analysis: With R and Financial Applications*. John Wiley & Sons, 2013.

P. Tseng. On accelerated proximal gradient methods for convex-concave optimization. *submitted to SIAM Journal on Optimization*, 2008.

H. White. *Asymptotic theory for econometricians*. Academic press New York, 2001.

X. H. Yan and J. Bien. Hierarchical sparse modeling: A choice of two group lasso formulations. *Statistical Science*, 32(4):531–560, 2017.

G. Yu and J. Bien. Learning local dependence in ordered data. *Journal of Machine Learning Research*, 18:1–60, 2017.

L. Yuan, J. Liu, and J. Ye. Efficient methods for overlapping group lasso. In *Advances in neural information processing systems*, pages 352–360, 2011.

M. Yuan and Y. Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(1):49–67, 2006.

P. Zhao, G. Rocha, and B. Yu. The composite absolute penalties family for grouped and hierarchical variable selection. *The Annals of Statistics*, 37(6A):3468–3497, 2009.