TOOLS FOR MODELING SPARSE VECTOR AUTOREGRESSIONS

A Dissertation

Presented to the Faculty of the Graduate School

of Cornell University

in Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy

by

William B. Nicholson

August 2016

TOOLS FOR MODELING SPARSE VECTOR AUTOREGRESSIONS

William B. Nicholson, Ph.D.

Cornell University 2016

The vector autoregression (VAR) has long proven to be an effective method for modeling the joint dynamics of multivariate time series and producing joint forecasts. A major shortcoming of the VAR that has hindered its applicability is its heavy parameterization: the parameter space grows quadratically with the number of series included, quickly exhausting the available degrees of freedom. Consequently, forecasting using VARs is intractable for high-dimensional data. However, empirical evidence suggests that VARs that incorporate more component series tend to result in more accurate forecasts. Conventional methods that allow for the estimation of large VARs either tend to require ad hoc subjective specifications or are computationally infeasible.

The first two chapters discuss two frameworks that utilize structured convex penalties to reduce the parameter space of both the VAR and VAR with unmodeled exogenous variables (VARX). The first chapter details the VARX-L framework, which adapts several scalar regression regularization techniques to a vector time series setting while accounting for the intrinsic lag structure of the VARX. By extending conventional solution methods to the multivariate time series setting, we develop computationally efficient implementations of several penalized least squares VARX models.

We consider two *group lasso* extensions; the *Lag Group VARX-L* and *Own/Other*

group VARX-L, which partition the VAR and VARX least squares coefficient matrices into a collection of groups, while enforcing uniform sparsity within each group. The Lag Group VARX-L assigns each VAR lag coefficient matrix to its own group whereas the Own/Other group VARX-L divides each lag coefficient matrix into two groups: "own lags" (which represent past realizations of the marginal series of forecasting interest) and "other lags" (which represent past realizations of other series included in the VAR). Under both penalties, each column of the VARX coefficient matrix is assigned to its own group.

In certain scenarios, a group penalty can be too restrictive (e.g. if only one coefficient is truly nonzero, such a penalty forecs every coefficient in the group to be active). Consequently, we also consider imposing a *sparse group lasso* penalty to the Lag and Own/Other structure, which incorporates within-group sparsity, allowing for the ability to set coefficients within an active group to zero.

We additionally consider an unstructured penalty, the *Basic VARX-L*, which assigns each coefficient to its own group. This method does not impose any structure, but it has considerably less computational overhead than the other penalties, so it can be extended to very high-dimensional settings. Finally, we consider a nested penalty, the *Endogenous-First VARX-L*, which prioritizes endogenous coefficients (i.e. coefficients corresponding to the VAR coefficient matrix) before their exogenous counterparts.

All of our methods rely on a regularization parameter to determine the degree of sparsity to impose. In conventional regression applications, this parameter is estimated by n-fold cross validation, but such an approach does not respect time

dependence. Instead, for a given a forecast horizon h we use a rolling validation scheme, which chooses the penalty parameter as the minimizer of h-step ahead forecast error over a training period (typically around $1/3$ of the data).

We evaluate our methods in two macroeconomic data applications as well as an extensive simulation study and compare against conventional VARX dimension reduction methods, such as information criterion minimization and Bayesian ridge regression. We find that in the vast majority of scenarios, our methods produce superior forecasts at both the $1-$step ahead and $4-$step ahead forecast horizons.

The second chapter, *HVAR: High Dimensional Forecasting via Intrepretable Vector Autoregression*, considers addressing the notion of *lag order* in VAR models by directly embedding it into a convex penalty with the HVAR class of models. In addition to encouraging sparsity, this framework encourages models with low maximum lag order. Unlike conventional lag order selection methods, such as information criterion minimization, our procedures don't select a single, universal lag order and instead allow for it to vary across marginal models (i.e. the forecasting equation for each series of interest).

We present three structured HVAR penalties: a *Componentwise HVAR*, which produces a separate lag order for each marginal model, but within a series all components have the same maximum lag, *Own/Other HVAR*, which embeds an additional layer of hierarchy within a lag, prioritizing coefficients corresponding to own lags before those of other series, and the *Elementwise HVAR*, which allows for a separate lag order for each coefficient in each marginal series.

We conduct a comprehensive simulation study; examining both forecasting performance and support recovery as well as two macroeconomic data sets and find that our HVAR procedures substantially outperform both benchmark procedures as well as the conventional Lasso-VAR and a Lasso VAR with a lag weighted penalty in both forecast performance and support recovery.

The third chapter *BigVAR: Tools for Modeling Sparse Penalized Vector Autoregressions*, describes the implementation details of the VARX-L and HVAR frameworks. A major advantage of our methodology over existing high-dimensional VAR procedures is its ease of use; there are no complex or subjective hyperparameters that must be accounted for by the end user. All of the methods presented in this dissertation are reproducible in a publicly available `R` package hosted on the Comprehensive R Repository Network (cran).

In this chapter, in addition to describing implementation specifics, we also provide elaboration on our construction of information criterion minimization based approaches as well as a procedure to refit coefficients based upon the support selected by one of our models. This approach is very popular in the conventional least squares regularization framework, but it is less tractable in the multivariate time series setting as we must account for the covariance of the included series. It is very difficult to obtain a reliable covariance estimate under scenarios in which the number of potential covariates is close to or exceeds the length of the series. In order to do so, we extend a procedure that implements *feasible generalized least squares* without requiring explicit matrix inversion. We conduct a simulation study of the effectiveness of refitting via least squares and conclude that it refit-

ting is generally inadvisable as it does not lead to an improvement in forecast performance.

BIOGRAPHICAL SKETCH

William ("Will") Nicholson received dual degrees in economics and mathematical statistics at American University. While at American, he became interested in empirical policy analysis. He served as a research assistant on a project that developed a model to determine equivalent civilian wages for enlisted service positions in the US Navy. He performed additional research on the economics of education which culminated in his senior thesis *The Impact of Performance Based Pay Incentives on the Attrition of American Public School Teachers*.

After graduating from American, Will entered the economics PhD program at the University of Wisconsin where he continued to pursue his interest in empirical public policy as a research assistant in the La Follette School of Public Affairs where he worked on a project sponsored by the Social Security Administration that analyzed the spending and saving habits of retirees.

Eventually, as he became more interested in statistical methodology as opposed to policy evaluation, Will decided to leave the economics program with a Master's degree and pursue a PhD in statistics. He entered Cornell's statistics PhD program in Fall of 2011 and pursued a research agenda that combined elements of applied econometrics, time series analysis, computational statistics, and machine learning. He became very interested in financial applications and pursued this interest with an internship at a quantitative hedge fund in the summer of 2015 and will begin work at Quantbot Technolologies as a quantitative research analyst in July 2016.

ACKNOWLEDGEMENTS

I owe a tremendous debt to the many people who helped me to achieve my academic goals. First, I must thank my parents for fostering my intellectual development and encouraging me to pursue postgraduate study.

Next, my undergraduate professors who helped foster my interests in economics and statistics; in particular John Nolan and Amos Golan, who wrote reference letters for both rounds of graduate applications. At Wisconsin, I thank Steven Durlauf for all of his help in transitioning to statistics. Without their aid, I would not have had the opportunity to attend Cornell.

I thank my statistics colleagues for their friendship and support; in particular Ben Risk, Jón Steingrimsson, Lucas Mentch, Didier Chételat, Max Chen, James Li, Dan Kowal, David Sinclair, and Kerstin Frailey.

I thank my committee members David Bindel and Jacob Bien for all of their help; David for his classes on Matrix Computations and parallel computing as well as his insights on the numerical analysis issues that I encountered, and Jacob for his substantial assistance in helping me to understand statistical regularization problems, which would become the backbone of my thesis. Finally, my advisor David Matteson, not only for his aid in my research, but also for his generosity and friendship over the years.

I acknowledge financial support from Google through their "Summer of Code" in 2014, which encouraged me to develop my methodology into an R package as well as Amazon who provided me with a substantial credit that enabled me to evaluate the performance of my models in high-dimensions using their elastic

compute cloud.

Finally, I thank my girlfriend Annabelle for all of her support and encouragement.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

VARX-L: STRUCTURED REGULARIZATION FOR LARGE VECTOR AUTOREGRESSION WITH EXOGENOUS VARIABLES

1.1 Introduction

The practice of macroeconomic forecasting was spearheaded by Klein and Goldberger (1955), whose eponymous simultaneous equation system jointly forecasted the behavior of 15 annual macroeconomic indicators, including consumer expenditures, interest rates, and corporate profits. The parameterization and identification restrictions of these models were heavily influenced by Keynesian economic theory. As computing power increased, such models became larger and began to utilize higher frequency data. Forecasts and simulations from these models were commonly used to inform government policymakers as to the overall state of the economy and to influence policy decisions (Welfe, 2013). As the Klein-Goldberger model and its extensions were primarily motivated by Keynesian economic theory, the collapse of the Bretton Woods monetary system and severe oil price shocks led to widespread forecasting failure in the 1970s (Diebold, 1998). At this time, the vector autoregression (VAR), popularized by Sims (1980), emerged as an atheoretical forecasting technique underpinned by statistical methodology and not subject to the ebb and flow of contemporary macroeconomic theory.

Unfortunately, the VAR's flexibility can create modeling complications. In the

absence of prior information, the VAR assumes that every series interacts linearly with both its own past values as well as those of every other included series. Such a model is known as an *unrestricted* VAR. As most economic series are low-frequency (monthly, quarterly, or annual) there is rarely enough available data to accurately forecast using large unrestricted VARs. Such models are overparameterized, provide inaccurate forecasts, and are very sensitive to changes in economic variables. Consequently, in such applications, the VAR's parameter space must be reduced, either in a data-driven manner or based upon the modeler's knowledge of the underlying economic system. This model selection process has been described as "blending data and personal beliefs according to a subjective, undocumented procedure that others cannot duplicate" (Todd, 1990)[p. 18].

Despite their overparameterization, in many applications, large VARs can be preferable to their smaller counterparts, as small models can exclude potentially relevant variables. Ideally, if one has no prior knowledge that a variable is irrelevant, it should be included in the model. For example, as described in Lütkepohl (2014), modeling the Taylor Rule (Taylor, 1993) requires an estimate of the "output gap" between real Gross Domestic Product and potential output. The output gap is difficult to measure and can include many explanatory variables encompassing disaggregated economic measurements. Moreover, recent work by Ibarra (2012) and Hendry and Hubrich (2011) have shown that incorporating disaggregated series improves upon the forecasts of macroeconomic aggregates such as the Consumer Price Index. Hence, in these scenarios, to properly utilize all relevant economic information, a large vector autoregression with a coherent variable

selection procedure is required.

Shortly after the VAR's inception, efforts were made to develop a systematic approach to reduce its parameterization. Early attempts, such as Litterman (1979), centered upon a Bayesian approach underpinned by contemporary macroeconomic theory. In applying a Bayesian VAR with a Gaussian prior (analogous to ridge regression), specific priors were formulated based upon stylized facts regarding US macroeconomic data. For example, the popular *Minnesota prior* incorporates the prevailing belief that macroeconomic variables can be reasonably modeled by a univariate random walk via shrinking estimated models toward univariate unit root processes.

The Bayesian VAR with a Minnesota prior was shown by Robertson and Tallman (1999) to produce forecasts superior to the conventional VAR, univariate models, and traditional simultaneous equation models. However, this approach is very restrictive; in particular, it assumes that all series are contemporaneously uncorrelated, and it requires the specification of several hyperparameters. Moreover, the Minnesota prior cannot accommodate large VARs itself. As pointed out by Banbura et al. (2009), when constructing a 40 variable system, in addition to the Minnesota prior, Litterman (1986b) imposes strict economically-motivated restrictions to limit the number of variables in each equation.

Modern Bayesian VAR extensions originally proposed in Kadiyala and Karlsson (1997) and compiled by Koop (2011) show that empirical regularization methods alone allow for the accurate forecasting of large VARs. Such procedures im-

pose data-driven restrictions on the parameter space while allowing for more general covariance specifications and estimation of hyperparameters via empirical Bayes or Markov chain Monte Carlo methods. These approaches are computationally expensive, and multi-step forecasts are nonlinear and must be obtained by additional simulation. Using a conjugate Gaussian-Wishart prior, Banbura et al. (2009) extend the Minnesota prior to a high-dimensional setting with a closed-form posterior distribution. Their technique uses a single hyperparameter, which is estimated by cross-validation. However, their specification does not perform variable selection, and their penalty parameter selection procedure is more natural within a frequentist framework.

More recent attempts to reduce the parameter space of VARs have incorporated the *lasso* (Tibshirani, 1996), a least squares variable selection technique. These approaches include the *lasso-VAR* proposed by Hsu et al. (2008) and further explored by Song and Bickel (2011), Li and Chen (2014), and Davis et al. (2012). Theoretical properties were investigated by Kock and Callot (2015) and by Basu and Michailidis (2015). Gefang (2012) considers a Bayesian implementation of the elastic net, an extension of the lasso proposed by Zou and Hastie (2005) that accounts for highly correlated covariates. However, their implementation is not computationally tractable and they do not observe much of a forecasting improvement over existing methods. The lasso-VAR has several advantages over Bayesian approaches as it is more computationally efficient in high dimensions, performs variable selection, and can readily compute multi-step forecasts and their associated prediction intervals.

In many applications, a VAR's forecasts can be improved by incorporating unmodeled exogenous variables, which are determined outside of the VAR. Examples of exogenous variables are context-dependent and range from leading indicators to weather-related measurements. In many scenarios, global macroeconomic variables, such as world oil prices, may be considered exogenous. Such models are most commonly referred to as "VARX" in the econometrics literature, but they are also known as "transfer function" or "distributed lag" models.

VARX has become an especially popular approach in the modeling of small open economies, as they are generally sensitive to a wide variety of global macroeconomic variables which evolve independently of their internal indicators. For example Cushman and Zha (1997) use a structural VARX model to analyze the effect of monetary policy shocks in Canada. The VARX is also amenable under scenarios in which forecasts are desired only from a subset of the included series in a VAR, as by construction its corresponding VARX has a reduced parameterization. VARX models have received considerable attention not just in economics, but also marketing (Nijs et al., 2007), political science (Wood, 2009), and real estate (Brooks and Tsolacos, 2000).

Unfortunately, dimensionality issues have limited the utility of the VARX. As a result of the aforementioned overparameterization concerns, in the conventional unrestricted VAR context most applications are limited to no more than 6 series (Bernanke et al., 2005), forcing the practitioner to specify *a priori* a reduced subset of series to include. The VARX faces similar restrictions. As outlined in Penm

et al. (1993), lag order, the maximum number of lagged observations to include, may differ between modeled and unmodeled series. Hence, in order to select a VARX model using standard information-criterion minimization based methods, one must fit all subset models up to the predetermined maximal lag order for both the series of forecasting interest (which we refer to as *endogenous* throughout this paper) and exogenous series. Moreover, unlike the conventional VAR, zero constraints (restrictions fixing certain model parameters to zero) are generally expected.

As it is often viewed as an economic rather than statistical problem, reducing the parameter space of the VARX model has received considerably less attention. Ocampo and Rodríguez (2012) extend the aforementioned Bayesian VAR estimation methods to the VARX context. George et al. (2008) apply stochastic search variable selection to the VARX framework; it provides a data-driven method to determine zero restrictions, but their approach is not scalable to high dimensions. Chiuso and Pillonetto (2010) propose estimating a VARX model with lasso and group lasso penalties but do not elaborate on potential group structures.

This paper seeks to bridge the considerable gap between the regularization and macroeconomic forecasting communities. We develop the VARX-L framework which allows for high-dimensional penalized VARX estimation while incorporating the unique structure of the VARX model in a computationally efficient manner. In order to implement this framework, we develop substantial modifications to existing lasso and group lasso solution algorithms, which were designed

primarily for univariate regression applications with no time dependence.

We extend the lasso and its structured counterparts to impose structured sparsity on the VARX, taking into account characteristics such as lag coefficient matrices, the delineation between a component's own lags and those of another component, and a potential nested structure between endogenous and exogenous variables. Our methods offer great flexibility in capturing the underlying dynamics of an economic system while imposing minimal assumptions on the parameter space.

Moreover, unlike previous approaches, due to our adaptation of convex optimization algorithms to a multivariate time series setting, our models are well-suited for the simultaneous forecasting of high-dimensional low-frequency macroeconomic time series. In particular, our models allow for prediction under scenarios in which the number of component series and included exogenous variables is close to or exceeds the length of the series. Our procedures, which avoid the use of subjective or complex hyperparameters, are publicly available in our `R` package `BigVAR` and can easily be applied by practitioners.

Section 2.2 describes the notation used throughout the paper and introduces our structured regularization methodology. Section 1.3 provides our implementation details and presents two macroeconomic forecasting applications. Section 1.4 details the "Minnesota VARX-L," an extension that allows for the incorporation of unit root nonstationarity by shrinking toward a vector random walk, section 1.5 presents a simulation study, and section 1.6 contains our conclusion. The ap-

pendix details the solution strategies and algorithms that comprise the VARX-L class of models.

1.2 Methodology

A k-dimensional multivariate time series $\{\mathbf{y}_t\}_{t=1}^T$ and m-dimensional exogenous multivariate time series $\{\mathbf{x}_t\}_{t=1}^T$ follow a vector autoregression with exogenous variables of order (p, s), denoted $\text{VARX}_{k,m}(p, s)$, if the following linear relationship holds

$$\mathbf{y}_t = \boldsymbol{\nu} + \sum_{\ell=1}^{p} \boldsymbol{\Phi}^{(\ell)} \mathbf{y}_{t-\ell} + \sum_{j=1}^{s} \boldsymbol{\beta}^{(j)} \mathbf{x}_{t-j} + \mathbf{u}_t \text{ for } t = 1, \ldots, T, \tag{1.1}$$

in which $\boldsymbol{\nu}$ denotes a k-dimensional constant intercept vector, $\boldsymbol{\Phi}^{(\ell)}$ represents a $k \times k$ endogenous coefficient matrix at lag $\ell = 1, \ldots, p$, $\boldsymbol{\beta}^{(j)}$ represents a $k \times m$ exogenous coefficient matrix at lag $j = 1, \ldots, s$, and $\mathbf{u}_t$ denotes a k-dimensional white noise vector that is independent and identically distributed with mean zero and non-singular covariance matrix $\boldsymbol{\Sigma}_u$. A VAR, which is a special case of the VARX, can be represented by Equation (2.1) with $\boldsymbol{\beta}^{(j)} = \mathbf{0}$ for $j = 1, \ldots, s$.

In a low-dimensional setting, in which the number of included predictors is substantially smaller than the length of the series, T, the VARX model can be fit by multivariate least squares, with $\boldsymbol{\nu}, \boldsymbol{\Phi} = [\boldsymbol{\Phi}^{(1)}, \ldots, \boldsymbol{\Phi}^{(p)}]$, and $\boldsymbol{\beta} = [\boldsymbol{\beta}^{(1)}, \ldots, \boldsymbol{\beta}^{(s)}]$

estimated as

$$\underset{\boldsymbol{\nu},\boldsymbol{\Phi},\boldsymbol{\beta}}{\operatorname{argmin}} \quad \sum_{t=1}^{T} \|\mathbf{y}_t - \boldsymbol{\nu} - \sum_{\ell=1}^{p} \boldsymbol{\Phi}^{(\ell)}\mathbf{y}_{t-\ell} - \sum_{j=1}^{s} \boldsymbol{\beta}^{(j)}\mathbf{x}_{t-j}\|_F^2, \tag{1.2}$$

in which $\|A\|_F = \sqrt{\sum_{i,j} A_{ij}^2}$ denotes the Frobenius norm of a matrix A. In the absence of regularization, the VARX$_{k,m}(p, s)$ requires the estimation of $k(1 + kp + ms)$ regression parameters. In the following section, we will apply several convex penalties to Equation (2.2) which aid in reducing the parameter space of the VARX by imposing sparsity in $\boldsymbol{\Phi}$ and $\boldsymbol{\beta}$.

1.2.1 VARX-L: Structured Penalties for VARX Modeling

In this section, we introduce VARX-L, a general penalized multivariate regression framework for large VARX models. We consider structured objectives of the form

$$\min_{\boldsymbol{\nu},\boldsymbol{\Phi},\boldsymbol{\beta}} \sum_{t=1}^{T} \|\mathbf{y}_t - \boldsymbol{\nu} - \sum_{\ell=1}^{p} \boldsymbol{\Phi}^{(\ell)}\mathbf{y}_{t-\ell} - \sum_{j=1}^{s} \boldsymbol{\beta}^{(j)}\mathbf{x}_{t-j}\|_F^2 + \lambda\Big(\mathcal{P}_y(\boldsymbol{\Phi}) + \mathcal{P}_x(\boldsymbol{\beta})\Big), \tag{1.3}$$

in which $\lambda \geq 0$ is a penalty parameter, which is selected in a sequential, *rolling* manner in a procedure that is discussed in section 3.4, $\mathcal{P}_y(\boldsymbol{\Phi})$ denotes a penalty function on endogenous coefficients, and $\mathcal{P}_x(\boldsymbol{\beta})$ denotes a penalty function on exogenous coefficients. Table 3.1 details the penalty structures proposed in this paper; all but the last have this separable structure. In the following section, we will discuss each penalty structure in detail. Note that since we utilize a single penalty parameter for all model coefficients, it is required that all included series

Table 1.1: The Proposed VARX-L Penalty Functions. Note that $\mathbf{\Phi}_{\text{on}}^{(\ell)}$ and $\mathbf{\Phi}_{\text{off}}^{(\ell)}$ denote the diagonal and off-diagonal elements of coefficient matrix $\mathbf{\Phi}^{(\ell)}$, respectively.

Group Name	$\mathcal{P}_y(\mathbf{\Phi})$	$\mathcal{P}_x(\boldsymbol{\beta})$
(1.4) Lag	$\sqrt{k^2}\sum_{\ell=1}^{p}\|\mathbf{\Phi}^{(\ell)}\|_F$	$\sqrt{k}\sum_{j=1}^{s}\sum_{i=1}^{m}\|\boldsymbol{\beta}_{\cdot,i}^{(j)}\|_F$
(1.5) Own/Other	$\sqrt{k}\sum_{\ell=1}^{p}\|\mathbf{\Phi}_{\text{on}}^{(\ell)}\|_F + \sqrt{k(k-1)}\sum_{\ell=1}^{p}\|\mathbf{\Phi}_{\text{off}}^{(\ell)}\|_F$	$\sqrt{k}\sum_{j=1}^{s}\sum_{i=1}^{m}\|\boldsymbol{\beta}_{\cdot,i}^{(j)}\|_F$
(1.6) Sparse Lag	$(1-\alpha)\sqrt{k^2}\sum_{\ell=1}^{p}\|\mathbf{\Phi}^{(\ell)}\|_F + \alpha\|\mathbf{\Phi}\|_1$	$(1-\alpha)\sqrt{k}\sum_{j=1}^{s}\sum_{i=1}^{m}\|\boldsymbol{\beta}_{\cdot,i}^{(j)}\|_F + \alpha\|\boldsymbol{\beta}\|_1$
(1.7) Sparse Own/Other	$(1-\alpha)(\sqrt{k}\sum_{\ell=1}^{p}\|\mathbf{\Phi}_{\text{on}}^{(\ell)}\|_F + \sqrt{k(k-1)}\sum_{\ell=1}^{p}\|\mathbf{\Phi}_{\text{off}}^{(\ell)}\|_F) + \alpha\|\mathbf{\Phi}\|_1$	$(1-\alpha)\sqrt{k}\sum_{j=1}^{s}\sum_{i=1}^{m}\|\boldsymbol{\beta}_{\cdot,i}^{(j)}\|_F + \alpha\|\boldsymbol{\beta}\|_1$
(1.8) Basic	$\|\mathbf{\Phi}\|_1$	$\|\boldsymbol{\beta}\|_1$
(1.9) Endogenous-First	$\mathcal{P}_{y,x}(\mathbf{\Phi},\boldsymbol{\beta}) = \sum_{\ell=1}^{p}\sum_{j=1}^{k}\left(\|[\mathbf{\Phi}_{j,\cdot}^{(\ell)},\boldsymbol{\beta}_{j,\cdot}^{(\ell)}]\|_F + \|\boldsymbol{\beta}_{j,\cdot}^{(\ell)}\|_F\right)$	

are on the same scale; hence we assume that prior to estimation, the series are standardized to each have zero mean and unit variance.

Equations (3.3)-(3.4) adapt the *group lasso* penalty proposed by Yuan and Lin (2006) to the VARX setting. The group lasso partitions the parameter space into groups of related variables which are shrunk toward zero. Within a group, all variables are either identically set to zero or are all nonzero. Our choices of $\mathcal{P}_y$ and $\mathcal{P}_x$ create structured sparsity based on pre-specified groupings, which are designed to incorporate the intrinsic lagged structure of the VARX. The proposed "lag group" methods have a substantial advantage over popular Bayesian approaches in that they will both shrink least squares estimates toward zero as well as perform variable selection in a computationally efficient manner.

Sparsity in the coefficient matrix is desirable when k and m are large because the conventional VARX is grossly overparameterized. As stated in Litterman (1984), it is widely believed in macroeconomic forecasting that small bits of relevant information exist throughout the data, and economic theory is not infor-

mative with regard to the manner in which this information is scattered. The proposed VARX-L framework provides a systematic approach to filter as much information as possible, assigning each bit an appropriate weight.

The group lasso penalty function was explored in the VAR context by Song and Bickel (2011) who consider several structured groupings with a particular emphasis on creating a distinction between a series' own lags and those of another series. Theoretical properties of the use of a group lasso penalty in the VAR setting were further explored by Basu et al. (2012).

A feature of the Lag Group VARX-L is that it does not impose sparsity within a group. Song and Bickel (2011) attempt to circumvent this constraint by including several additional lasso penalties, but such an approach requires a multi-dimensional gridsearch to select penalty parameters. The penalties for the proposed Sparse Group VARX-L and Sparse Own/Other Group VARX-L, listed in Equations (3.5)-(3.6), instead implement the *sparse group lasso* (Simon et al., 2013), which extends the group lasso to allow within-group sparsity. The sparse group lasso can be viewed analogously to the elastic net (Zou and Hastie, 2005) extended to structured penalties.

The penalty for the Basic VARX-L adapts the lasso (3.7); it considers no structure, or can be viewed as a group lasso penalty that assigns each coefficient to a singleton group. In very high-dimensional scenarios, this most basic penalty has computational advantages as compared to more complex approaches. Finally, the penalty for the proposed Endogenous-First VARX-L, Equation (3.8), incorporates

a nested penalty structure such that, within a lag, endogenous coefficients are prioritized before their exogenous counterparts. Since this penalty structure is not separable in the manner of Equation (3.2), its penalty function is denoted as $\mathcal{P}_{y,x}$.

Group VARX-L

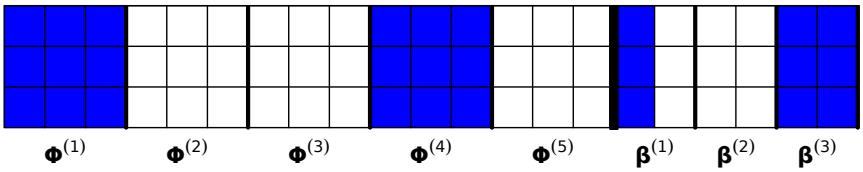
We first present the *Lag* Group VARX-L (3.3), in which the endogenous coefficients are grouped according to their lagged coefficient matrix $\mathbf{\Phi}^{(\ell)}$ for $\ell = 1, \ldots, p$, and at every lag, each exogenous component series is partitioned into its own group. This structured grouping is expressed as

$$\mathcal{P}_y(\mathbf{\Phi}) = \sqrt{k^2} \sum_{\ell=1}^{p} \|\mathbf{\Phi}^{(\ell)}\|_F, \quad \mathcal{P}_x(\boldsymbol{\beta}) = \sqrt{k} \sum_{j=1}^{s} \sum_{i=1}^{m} \|\boldsymbol{\beta}_{\cdot,i}^{(j)}\|_F.$$

Note that since the endogenous and exogenous groups differ in cardinality, it is required to weight the penalty to avoid regularization favoring larger groups. This structure implies that for each endogenous series, a coefficient matrix at lag ℓ is either entirely nonzero or entirely zero. Similarly, the relationship between an exogenous and endogenous series at lag j will either be nonzero for all endogenous series or identically zero. A potential sparsity pattern generated by this structure (with $k = 3$, $p = 5$, $m = 2$, and $s = 3$) is shown in Figure 1.1 with the active (i.e. nonzero) elements shaded.

In comparison to Bayesian regularization methods, such as stochastic search variable selection (George et al., 2008), estimating the Lag Group VARX-L is tractable even in high dimensions. We are able to extend the efficient group lasso

Figure 1.1: Example sparsity pattern (active elements shaded) produced by a Lag Group VARX-L$_{3,2}(5,3)$



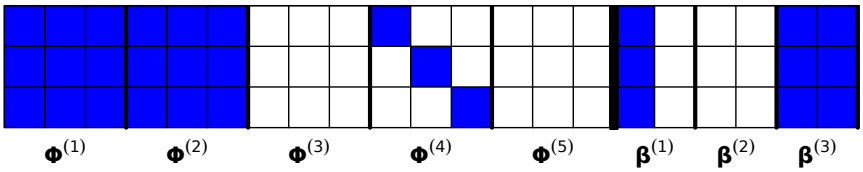
solution method proposed by Qin et al. (2010), who utilize a block coordinate descent procedure and transform each "one group" subproblem to a trust-region framework. These subproblems can then be solved efficiently via univariate optimization. Details of our algorithm are provided in section A.0.3 of the appendix.

The Lag Group structure is advantageous for applications in which all series tend to exhibit comparable dynamics, such as forecasting the disaggregate subcomponents of a composite index. It also can serve as a powerful tool for lag selection. However, in many settings, it may not be appropriate to give equal consideration to every entry in a coefficient matrix. Diagonal entries of each $\mathbf{\Phi}^{(\ell)}$, which represent regression on a series' own lags, are in many applications more likely to be nonzero than are off-diagonal entries, which represent lagged cross dependence with other components. The *Own/Other* Group VARX-L (3.4) allows for the partitioning of each lag matrix $\mathbf{\Phi}^{(\ell)}$ into separate groups by assigning the following endogenous penalty structure

$$\mathcal{P}_y(\mathbf{\Phi}) = \sqrt{k}\sum_{\ell=1}^{p}\|\mathbf{\Phi}_{\text{on}}^{(\ell)}\|_F + \sqrt{k(k-1)}\sum_{\ell=1}^{p}\|\mathbf{\Phi}_{\text{off}}^{(\ell)}\|_F,$$

in which $\mathbf{\Phi}_{\text{on}}^{(\ell)}$ denotes the diagonal elements of $\mathbf{\Phi}^{(\ell)}$ and $\mathbf{\Phi}_{\text{off}}^{(\ell)}$ denotes its off-diagonal entries.

Figure 1.2: Example sparsity pattern (active elements shaded) produced by an Own/Other Group VARX-$L_{3,2}(5, 3)$

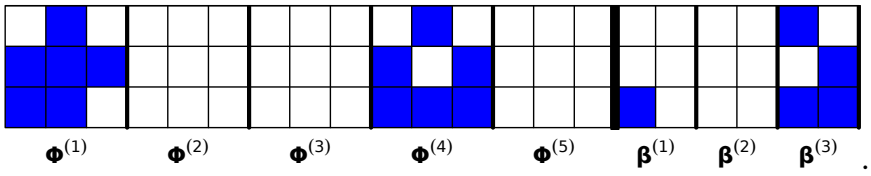


An example of this sparsity pattern is shown in Figure 1.2. The computational modifications required to utilize the Own/Other structure are detailed in section A.0.3 in the appendix. This delineation between own lags and other lags is often incorporated in macroeconomic forecasting. As detailed in Litterman (1986a), the traditional Minnesota prior operates under the assumption that a series' own past values account for most of its variation, hence they are shrunk by a smaller factor than realizations of other series. The strong forecasting performance of the VARX-L procedures that utilize the Own/Other structure in section 1.3.5 provides further justification for Litterman's beliefs.

Sparse Group VARX-L

For certain applications, a group penalty might be too restrictive. If a group is active, all coefficients in the group will be nonzero, and including a large number of groups substantially increases computation time. Moreover, it is inefficient to include an entire group if, for example, only one coefficient is truly nonzero. The *sparse group lasso*, proposed by Simon et al. (2013) allows for within-group sparsity through a convex combination of lasso and group lasso penalties. The Sparse Lag

Figure 1.3: Example sparsity pattern (active elements shaded) produced by a Sparse Lag Group VARX-L$_{3,2}(5,3)$



Group VARX-L (3.5) results in a penalty of the form

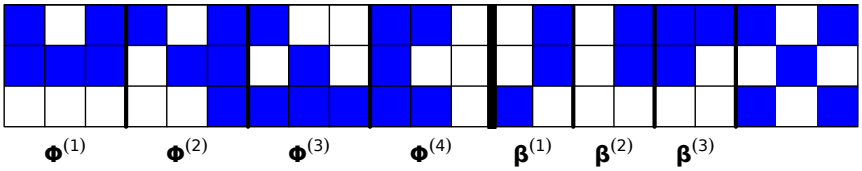
$$\mathcal{P}_y(\mathbf{\Phi}) = (1-\alpha)(\sqrt{k^2}\sum_{\ell=1}^{p}\|\mathbf{\Phi}^{(\ell)}\|_F) + \alpha\|\mathbf{\Phi}\|_1,$$

$$\mathcal{P}_x(\boldsymbol{\beta}) = (1-\alpha)(\sqrt{k}\sum_{j=1}^{s}\sum_{i=1}^{m}\|\boldsymbol{\beta}_{\cdot,i}^{(j)}\|_F) + \alpha\|\boldsymbol{\beta}\|_1,$$

in which $0 \leq \alpha \leq 1$ is an additional penalty parameter that controls within-group sparsity. Without prior knowledge of the important predictors, we weight according to relative group sizes and set $\alpha = \frac{1}{k+1}$, though α could also be estimated by cross-validation. The inclusion of the L_1 norm allows for within-group sparsity, hence even if a group is considered active individual coefficients within it can be set to zero. An example sparsity pattern is depicted in Figure 1.3.

Since the inclusion of within-group sparsity does not create a separable objective function, conventional group lasso solution methods, such as coordinate descent, are no longer applicable. Following Simon et al. (2013), our estimation algorithm for the Sparse Lag Group VARX-L makes use of proximal gradient descent. The details of this approach and our implementation are provided in section A.0.3 of the appendix. This penalty can be extended to alternative groupings. Consequently, we also consider the *Sparse Own/Other Group* VARX-L (3.6) as an

Figure 1.4: Example sparsity pattern (active elements shaded) produced by a Basic VARX-L$_{3,2}(5,3)$



estimation procedure.

Basic VARX-L

The Basic VARX-L (3.7), proposed by Chiuso and Pillonetto (2010), incorporates no structure and can be viewed as a special case of the Sparse Group VARX-L in which $\alpha = 1$, resulting in penalties of the form

$$\mathcal{P}_y(\mathbf{\Phi}) = \|\mathbf{\Phi}\|_1, \quad \mathcal{P}_x(\boldsymbol{\beta}) = \|\boldsymbol{\beta}\|_1.$$

The L_1 penalty will induce sparsity in the coefficient matrices $\mathbf{\Phi}$ and $\boldsymbol{\beta}$ by zeroing individual entries. An example sparsity pattern is depicted in Figure 1.4.

A major advantage of the Basic VARX-L over its structured counterparts is its computational tractability. Our solution approach involves the use of coordinate descent, popularized for lasso problems by Friedman et al. (2010). Coordinate descent consists of partitioning the Basic VARX-L into subproblems for each scalar element $[\mathbf{\Phi}, \boldsymbol{\beta}]_{ij}$, solving component-wise, then updating until convergence. This approach is computationally efficient since, in the Basic VARX-L context, each subproblem has a closed-form solution. Tseng (2001) establishes that global con-

vergence arises from solving individual subproblems in the coordinate descent framework. Our solution strategy and algorithm are detailed section A.0.3 of the appendix.

1.2.2 An Endogenous-First Active Set

We have previously only considered structures that assign endogenous and exogenous variables to separate groups. In this section, we consider a nested structure that can take into account the relative importance between endogenous and exogenous predictor series.

In certain scenarios, there may exist an *a priori* importance ranking among endogenous and exogenous variables. For example, the endogenous variables could represent economic indicators of interest in a small open economy, with global macroeconomic indicators serving as exogenous variables. In such a scenario, it may be desirable for exogenous variables to enter into a forecasting equation only if endogenous variables are also present at a given lag ℓ. We can consider such a structure by utilizing a *hierarchical group lasso* penalty (see, e.g. Jenatton et al. (2011)). The Endogenous-First VARX-L penalty function (3.8) takes the form

$$\mathcal{P}_{y,x}(\boldsymbol{\Phi}, \boldsymbol{\beta}) = \sum_{\ell=1}^{p} \sum_{j=1}^{k} (\|[\boldsymbol{\Phi}_{j,\cdot}^{(\ell)}, \boldsymbol{\beta}_{j,\cdot}^{(\ell)}]\|_F + \|\boldsymbol{\beta}_{j,\cdot}^{(\ell)}\|_F). \tag{1.10}$$

Under this structure, at a given lag, exogenous variables can enter the model only after the endogenous variables at the same lag. Note that this structure requires

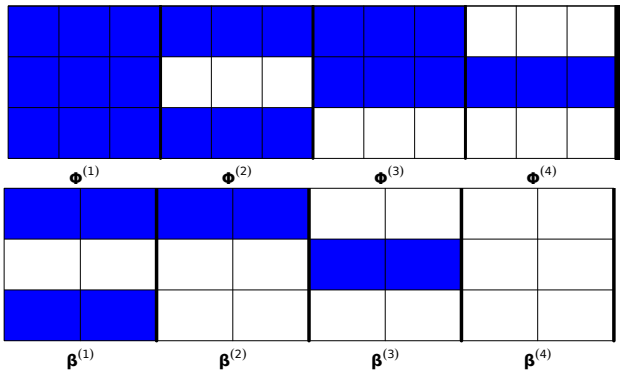


Figure 1.5: Example sparsity pattern (active elements shaded) generated by an Endogenous-First VARX-$L_{3,2}(4,4)$. Note that a row in $\boldsymbol{\beta}^{(\ell)}$ can only be nonzero if the corresponding row in $\boldsymbol{\Phi}^{(\ell)}$ is also nonzero.

that $s \leq p$. It should also be noted that (1.10) decouples across rows, allowing for separate nested structures across each endogenous series. This sparsity pattern is depicted in Figure 1.5.

Most group lasso solution methods, such as block coordinate descent, take advantage of the separability of groups to improve computational performance. Although the nested structure is not directly separable, based on the methodology of Jenatton et al. (2011), its dual can be solved in one pass of block coordinate descent. Details of the solution approach and our algorithm are provided in section A.0.3 of the appendix.

1.3 High-Dimensional Macroeconometrics

In this section, we start by evaluating our regularization procedures in two macroeconomic data applications: one high-dimensional and one low-dimensional. In our first application, we consider applying the proposed VARX-L procedures on the widely used set of US macroeconomic indicators originally constructed by Stock and Watson (2005). Our second example considers forecasting a small set of Canadian macroeconomic indicators and incorporating the previous US data as exogenous series. Section 3.4 outlines the practical implementation of our penalty parameter selection procedure, section 1.3.4 describes the benchmarks that we compare our models against, section 1.3.5 details our macroeconomic applications.

1.3.1 Practical Implementation

The regularization parameter, λ, is not known in practice and is typically chosen via cross-validation. In this section, we detail our strategy for selecting λ. Following Friedman et al. (2010), we select from a grid of potential penalty parameters that starts with the smallest value in which all components of $[\mathbf{\Phi}, \boldsymbol{\beta}]$ will be zero and decreases in log-linear increments. This value differs for each procedure and can be inferred by their respective algorithms. The starting values are summarized in Table A.1 located in section A.0.5 of the appendix. The number of grid-points, n, as well as the depth of the grid are left to user input. A deep grid and

large number of gridpoints result in increased computational costs and often do not improve forecasting performance. We have found that a grid depth $\frac{1}{25}\lambda_{\max}$ and 10 gridpoints achieve adequate forecast performance in most scenarios.

Due to time-dependence, our problem is not well-suited to traditional n-fold cross-validation. Instead, following Banbura et al. (2009), we propose choosing the optimal penalty parameter by minimizing h-step ahead mean-square forecast error (MSFE), in which h denotes the desired forecast horizon. We divide the data into three periods: one for initialization, one for training, and one for forecast evaluation. Define time indices $T_1 = \left\lfloor \frac{T}{3} \right\rfloor, T_2 = \left\lfloor \frac{2T}{3} \right\rfloor$.

We start our validation process by fitting a model using all data up to time T_1 and forecast $\hat{\mathbf{y}}_{T_1+h}^{\lambda_i}$ for $i = 1, \ldots, n$. We then sequentially add one observation at a time and repeat this process until time T_2. This procedure is illustrated in Figure 1.6.

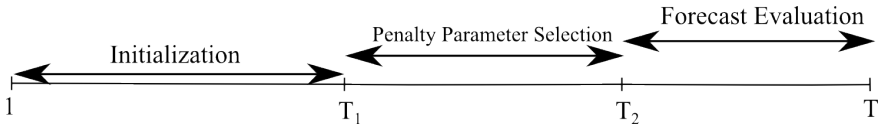


Figure 1.6: Illustration of Rolling Cross-Validation

We select $\hat{\lambda}$ as the minimizer of

$$MSFE(\lambda_i) = \frac{1}{(T_2 - T_1 - h + 1)} \sum_{t=T_1-h+1}^{T_2-h} ||\hat{\mathbf{y}}_{t+h}^{\lambda_i} - \mathbf{y}_{t+h}||_F^2. \tag{1.11}$$

Finally, from time T_2 to $T - h + 1$, we evaluate the h-step ahead forecast accuracy of $\hat{\lambda}$. If desired, additional criterion functions can be substituted. MSFE is the most

natural criterion given our use of the least squares objective function. Rather than parallelizing the cross-validation procedure, our approach uses the result from the previous period as an initialization or "warm start," which substantially decreases computation time. The penalty parameter selection procedure is presented in Algorithm 4 in the appendix.

1.3.2 Multi-step Predictions

The VARX-L framework can easily accommodate multi-step ahead forecasting. To do so, we modify our solution algorithms to calculate *direct* multi-step ahead forecasts. Essentially, computing direct h-step forecasts involves solving the standard VARX-L objective (3.2) while leaving a gap of h observations.

$$\min_{\boldsymbol{\nu},\boldsymbol{\Phi},\boldsymbol{\beta}} \sum_{t=1}^{T-h+1} \|\mathbf{y}_t - \boldsymbol{\nu} - \sum_{\ell=1}^{p} \boldsymbol{\Phi}^{(\ell)}\mathbf{y}_{t-h-\ell+1} - \sum_{j=1}^{s} \boldsymbol{\beta}^{(j)}\mathbf{x}_{t-h-j+1}\|_F^2 + \lambda\Big(\mathcal{P}_y(\boldsymbol{\Phi}) + \mathcal{P}_x(\boldsymbol{\beta})\Big).$$

Per Clark and McCracken (2013), the direct h-step ahead forecast can be calculated as

$$\hat{\mathbf{y}}_{t+h} = \hat{\boldsymbol{\nu}} + \widehat{\boldsymbol{\Phi}}^{(1)}\mathbf{y}_t + \cdots + \widehat{\boldsymbol{\Phi}}^{(p)}\mathbf{y}_{t-p+1} + \widehat{\boldsymbol{\beta}}^{(1)}\mathbf{x}_t + \cdots + \widehat{\boldsymbol{\beta}}^{(s)}\mathbf{x}_{t-s+1}.$$

The *iterative* approach, in which multi-step forecasts are computed recursively as 1−step ahead forecasts using predicted values is another popular technique to compute long-horizon forecasts. This approach is not directly extendable to the VARX setting, as we do not forecast the exogenous series.

If iterative multi-step predictions are desired, one could instead fit the full VAR_{k+m}. However, as shown in Marcellino et al. (2006), direct forecasts are more robust to model misspecification, making them more appropriate in high-dimensional settings. As our macroeconomic applications consider quarterly data, in section 1.3.5 we compute both 1−step and 4−step ahead forecasts.

1.3.3 Selecting a Structure

The VARX-L framework offers many choices for structured penalties suited toward a wide range of applications. Under scenarios in which little is known about the potential dynamic dependence of the included series, the Basic VARX-L makes no underlying structural assumptions.

The Lag and Sparse Lag Group VARX-L structures are most appropriate when the endogenous series are closely related; for example, if the series of interest comprise unemployment rates segmented by state or census region. The Own/Other and Sparse Own/Other Group VARX-L structures are most appropriate for macroeconomic applications in which a series' own lags are thought to have substantially different temporal dependence than those of "other" series. As an example, one could consider a disparate group of series traditionally examined in small-scale macroeconometric forecasting applications: the US Federal Funds Rate, the GDP Growth Rate, and the Consumer Price Index. The Endogenous-First structure is best suited toward applications in which the forecasting effec-

tiveness of the exogenous series is unknown.

One potential diagnostic tool involves fitting the Sparse Group VARX-L with both λ and α selected according to rolling cross validation. A selected value of α close to 0 indicates evidence of strong groupwise sparsity, while a value close to 1 indicates unstructured sparsity, and values in the middle provide evidence for some combination of the two.

Since the computational time required to apply all procedures is manageable, in practice, we suggest fitting several VARX-L structures and selecting the approach that achieves the best out of sample forecasting performance.

1.3.4 Methods for Comparison

A conventional VARX model selection approach in a low-dimensional setting involves fitting a $\text{VARX}_{k,m}(\ell, j)$ by least squares for $0 \leq \ell \leq p$, $0 \leq j \leq s$ and selecting lag orders for both the endogenous and exogenous series based on an information criterion, such as Akaike's Information Criterion (AIC) or Bayesian Information Criterion (BIC). Per Lütkepohl (2005), the AIC and BIC of a $\text{VARX}_{k,m}(\ell, j)$ are defined as

$$\text{AIC}(\ell, j) = \log |\widehat{\Sigma}_u^{\ell,j}| + \frac{2(k(k\ell + mj))}{T},$$
$$\text{BIC}(\ell, j) = \log |\widehat{\Sigma}_u^{\ell,j}| + \frac{\log(T)(k(k\ell + mj))}{T}.$$

in which $\widehat{\Sigma}_u^{\ell,j}$ is the residual sample covariance matrix obtained from the estimated $\text{VARX}_{k,m}(\ell, j)$, and $|\Sigma|$ represents the determinant of Σ. The selected lag orders (ℓ, j) are then chosen as the minimizer of AIC or BIC. AIC penalizes each model coefficient uniformly by a factor of two whereas BIC scales penalties relative to series length. Hence, when T is large, BIC will tend to select more parsimonious models than AIC.

We compare our methods against least squares model selection procedures that utilize AIC and BIC to select lag orders. Since we are considering high-dimensional applications, in which $\widehat{\Sigma}_u$ could be ill-conditioned, we construct our least squares estimates using a variation of the approach developed by Neumaier and Schneider (2001). This procedure constructs the least squares estimates using a QR decomposition, which obviates the need for explicit matrix inversion. In addition, following a heuristic proposed by Hansen (2013), we impose a ridge penalty: $((k \cdot \ell + m \cdot j)^2 + (k \cdot \ell + m \cdot j) + 1)\epsilon_{\text{machine}}$ scaled by the column norms of the lagged series $\mathbf{y}_{t-1}, \ldots, \mathbf{y}_{t-p}, \mathbf{x}_{t-1}, \ldots, \mathbf{x}_{t-s}$, in which $\epsilon_{\text{machine}}$ denotes machine precision. This penalty ensures that the determinant of $\widehat{\Sigma}_u$ is well-defined without noticeably impacting degree of freedom calculations.

We additionally compare our methods against two naive approaches that provide insight with regard to the level of temporal dependence in the data. We first consider the unconditional *sample mean*, which will make h-step ahead forecasts at time $t + h$ based upon the average of all observed data up to time t: $\hat{\mathbf{y}}_{t+h} = \frac{1}{t} \sum_{i=1}^{t} \mathbf{y}_i$. Scenarios in which the sample mean forecasts well relative to more sophisticated

procedures imply a weak linear relationship with lagged series. Second, we consider the vector *random walk* model, which makes h-step ahead forecasts based upon the most recent realization of the series, i.e. $\hat{\mathbf{y}}_{t+h} = \mathbf{y}_t$. Superior performance of the vector random walk indicates high persistence or a strong degree of temporal dependence, as is often observed in macroeconomic data.

Finally, we compare against the popular Bayesian VAR with a modified Minnesota Prior proposed by Banbura et al. (2009) (henceforth BGR). Their approach acts very similarly to ridge regression in that it shrinks least squares coefficients toward zero with the degree of regularization determined by a single penalty parameter. This parameter is chosen according to rolling cross validation as described in section 3.4. As in Banbura et al. (2009), instead of fitting a VARX$_{k,m}(p, s)$, we fit a VAR$_{k+m}(p)$ and select the regularization parameter as the minimizer of $h-$step ahead MSFE across the k endogenous series. This allows for BGR's method to make forecasts utilizing information from both the endogenous and exogenous series, which allows for a direct comparison with our VARX-L framework.

BGR's approach modifies the Minnesota Prior to make it computationally tractable in high dimensions, but it does not does not return sparse solutions. Superior performance of the VARX-L methods relative to BGR's approach provides evidence as to the importance of imposing sparsity in obtaining accurate forecasts. Details of our implementation of BGR's procedure are provided in section A.0.4 in the appendix.

1.3.5 Macroeconometric Applications

We evaluate our methods on the large and widely utilized macroeconomic dataset created by Stock and Watson (2005) and later amended by Koop (2011). The dataset consists of 168 quarterly US macroeconomic indicators containing information about various aspects of the economy, including income, industrial production, employment, stock prices, interest rates, exchange rates, etc. The data ranges from Quarter 2 of 1959 to Quarter 3 of 2007 ($T = 195$). Per Koop (2011), the series can be categorized into several levels; we consider the following three:

- *Small* ($k = 3$): Three variables (Federal Funds Rate, Consumer Price Index, Gross Domestic Product growth rate). Core group, typically used in simple dynamic stochastic generalized equilibrium models;
- *Medium* ($k = 20$): Small plus 17 additional variables containing aggregated economic information (e.g., consumption, labor, housing, exchange rates);
- *Medium-Large* ($k = 40$): Medium plus 20 additional aggregate variables.

For a detailed description of each set of variables, consult Koop (2011). As Banbura et al. (2009) found that the greatest improvements in forecast performance occurred with the *medium* VAR, that will be our focus. We will attempt to forecast the *medium* set of indicators ($k = 20$) while using the additional variables from the *medium-large* category as exogenous predictors ($m = 20$). Before estimation, each series is transformed to stationarity according to the specifications provided by Stock and Watson (2005) and standardized by subtracting the sample mean and

dividing by the sample standard deviation. Quarter 2 of 1976 to Quarter 3 of 1992 is used for penalty parameter selection while Quarter 4 of 1992 to Quarter 3 of 2007 is used for forecast evaluation. Our results are summarized in Table 1.2.

Table 1.2: One-step and four-step ahead MSFE of $k = 20$ macroeconomic indicators (relative to sample mean) with $m = 20$ exogenous predictors $p = 4$, $s = 4$.

Model/VARX-L Penalty Structure	One-step ahead Out of Sample Relative MSFE	Four-step ahead Out of Sample Relative MSFE
Basic	0.8064	0.9672
Lag Group	0.8747	0.9798
Own/Other Group	0.7773	0.9582
Sparse Lag Group	0.8206	0.9702
Sparse Own/Other Group	0.7823	0.9590
Endogenous-First	0.8531	0.9748
VARX with lags selected by AIC	5.0223	7.8363
VARX with lags selected by BIC	0.9455	1.1603
BGR's Bayesian VAR	0.9414	0.9765
Sample Mean	1.0000	1.0000
Random Walk	1.9909	1.8706

Most of our VARX-L procedures substantially outperform the benchmarks at both forecast horizons, with the Own/Other Group VARX-L and Sparse Own/Other Group VARX-L achieving the best performance. This provides evidence that making the distinction between a series' own lags and those of other series can improve forecasts in macroeconomic applications. The relatively poor performance of the Lag Group VARX-L suggests that a lag based grouping may be too general for such a disparate group of series and hence not appropriate for

this application.

The imposition of sparsity appears to be crucial, as BGR's Bayesian VAR performs worse than all VARX-L procedures at both horizons except for the Lag Group VARX-L at $h = 4$. It performs very similarly to the least squares VARX with lags selected by BIC at $h = 1$, but slightly better at $h = 4$.

The VARX with lags selected by AIC is substantially outperformed by the sample mean at both horizons, whereas the VARX with lags selected by BIC slightly outperforms it at $h = 1$, but is outperformed at $h = 4$. Since AIC imposes a weaker penalty for higher lag orders than AIC, it has a tendency to construct overparameterized models, whereas BIC has a tendency to underfit and misses out on potential dynamic relationships that the VARX-L procedures are able to capture. Since neither approach imposes variable selection, these models tend to result in very noisy multi-step ahead forecasts.

Canadian Macroeconomic Data Application

We next consider a low-dimensional application in which we forecast Canadian indicators using US macroeconomic series as exogenous predictors. As a small, relatively open economy Canada's macroeconomic indicators have been shown to be very sensitive to their US counterparts. In particular, Racette and Raynauld (1992) and Cushman and Zha (1997) demonstrate that the US Gross Domestic Product and Federal Funds Rate are very influential in modeling Canada's anal-

ogous monetary policy proxy variables. Taking this into consideration, we forecast $k = 4$ Canadian macroeconomic series using our previously defined *medium* dataset as exogenous predictors ($m = 20$). The endogenous series are Canadian M1 (a measure of the liquid components of money supply), Canadian Industrial Production, Canadian GDP (relative to 2000), and the Canada/US Exchange Rate.

The Canadian series range from Quarter 3 of 1960 to Quarter 3 of 2007. Quarter 3 of 1977 to Quarter 2 of 1993 is used for penalty parameter selection while Quarter 3 of 1993 to Quarter 3 of 2007 is used for forecast evaluation ($T = 191$). In addition to the standard benchmarks, we also compare against our procedures in the VAR framework, in which the exogenous predictors are ignored. Our results are summarized in Table 1.3.

Table 1.3: One-step ahead and four-step ahead MSFE (relative to sample mean) for VARX forecasts of $k = 4$ Canadian macroeconomic indicators with $m = 20$ exogenous predictors $p = 4, s = 4$ and VAR forecasts of 4 Canadian macroeconomic indicators, $p = 4$.

Model/VARX-L Penalty Structure	One-step ahead Out of Sample RMSFE	Four-step ahead Out of Sample RMSFE
Basic	0.8406	0.9187
Lag Group	0.8357	0.9285
Own/Other Group	0.8376	0.9143
Sparse Lag Group	0.8274	0.9129
Sparse Own/Other Group	0.8390	0.9181
Endogenous-First	0.8454	0.9593
VARX with lags selected by AIC	1.3680	1.7739
VARX with lags selected by BIC	0.8785	1.0941
BGR's Bayesian VAR (with exogenous series)	1.0058	0.9748
Model/VAR-L Penalty Structure	One-step ahead Out of Sample RMSFE	Four-step ahead Out of Sample RMSFE
Basic	0.8465	0.9645
Lag Group	0.8575	0.9965
Own/Other Group	0.8491	0.9604
Sparse Lag Group	0.8506	0.9623
Sparse Own/Other Group	0.8493	0.9655
VAR with lag selected by AIC	0.9190	1.1365
VAR with lag selected by BIC	0.8785	1.0941
BGR's Bayesian VAR (without exogenous series)	1.0066	0.9891
Sample Mean	1.0000	1.0000
Random Walk	1.3388	1.7180

Even at this low dimension, we find that all of our models substantially outperform the AIC and BIC benchmarks across both forecast horizons, with the Sparse Lag Group VARX-L achieving superior performance at both horizons. This low-dimensional example is better suited toward lag based groupings than our previous application. Consequently, the relative forecasting performance of the Lag Group VARX-L and Sparse Lag Group VARX-L improve substantially.

In addition, we find that our methods are able to effectively leverage relevant information from the exogenous predictors, as every VARX-L procedure achieves better out of sample performance than its corresponding VAR-L. Conversely, the information criterion based VARX approaches fail to outperform their VAR counterparts. At $h = 1$, BIC produces identical forecast error in both the VAR and VARX setting, indicating that it never selects any exogenous series.

BGR's Bayesian VAR performs poorly in this scenario, achieving similar forecast performance to the sample mean across both horizons, both with and without exogenous series, outperforming only the Lag VAR-L at $h = 4$. Its poor performance across both settings suggests that imposing sparsity is desirable even in low-dimensional applications.

1.4 Extending the VARX-L for Unit-Root Nonstationarity

In some scenarios, it may not be appropriate to shrink every coefficient toward zero. In traditional time series analysis, economic series that exhibit persistence are transformed to stationarity. However, this framework has several drawbacks. First, if no pre-established transformation guidelines are available, this process can be labor intensive and subjective. Second, as stated by Kennedy (2003), stationarity transformations destroy information about the long-run relationships of economic variables. Ideally, to effectively forecast using all available information, it would be preferable to work directly with the untransformed series. In this section, we outline a possible extension that allows for shrinking toward reference models, such as a vector random walk, that can account for mild non-stationarity, which is ubiquitous in macroeconomic data.

The "Minnesota" VARX-L

The proposed VARX-L models can easily be modified to shrink toward a known constant matrix. Shrinking toward constant matrices $\mathbf{C}_y \in \mathbf{R}^{k \times kp}, \mathbf{C}_x \in \mathbf{R}^{k \times ms}$ results in a slightly modified objective of the form

$$\min_{\boldsymbol{\nu}, \boldsymbol{\Phi}, \boldsymbol{\beta}} \sum_{t=1}^{T} \|\mathbf{y}_t - \boldsymbol{\nu} - \boldsymbol{\Phi} \boldsymbol{Y}_{t-1} - \boldsymbol{\beta} \mathbf{X}_{t-1}\|_F^2 + \lambda \Big(\mathcal{P}_y(\boldsymbol{\Phi} - \mathbf{C}_y) + \mathcal{P}_x(\boldsymbol{\beta} - \mathbf{C}_x) \Big), \tag{1.12}$$

in which $\boldsymbol{Y}_t = [\mathbf{y}_t^\top, \ldots, \mathbf{y}_{t-p}^\top]$ and $\mathbf{X}_t = [\mathbf{x}_t^\top, \ldots, \mathbf{x}_{t-s}^\top]$.

Let $[\boldsymbol{\Phi}, \boldsymbol{\beta}]^{\lambda}(\mathbf{C}_y, \mathbf{C}_x)$ denote a solution to this problem. Now, by a change of variables $\widetilde{\boldsymbol{\Phi}} = \boldsymbol{\Phi} - \mathbf{C}_y$ and $\widetilde{\boldsymbol{\beta}} = \boldsymbol{\beta} - \mathbf{C}_x$, we obtain the equivalent problem

$$\min_{\boldsymbol{\nu}, \widetilde{\boldsymbol{\Phi}}, \widetilde{\boldsymbol{\beta}}} \sum_{t=1}^{T} \|\mathbf{y}_t - \boldsymbol{\nu} - \mathbf{C}_y \boldsymbol{Y}_{t-1} - \widetilde{\boldsymbol{\Phi}} \boldsymbol{Y}_{t-1} - \mathbf{C}_x \mathbf{X}_{t-1} - \widetilde{\boldsymbol{\beta}} \mathbf{X}_{t-1}\|_F^2 + \lambda\Big(\mathcal{P}_y(\widetilde{\boldsymbol{\Phi}}) + \mathcal{P}_x(\widetilde{\boldsymbol{\beta}})\Big),$$

which can be expressed as

$$\min_{\boldsymbol{\nu}, \widetilde{\boldsymbol{\Phi}}, \widetilde{\boldsymbol{\beta}}} \sum_{t=1}^{T} \|\widetilde{\mathbf{y}}_t - \boldsymbol{\nu} - \widetilde{\boldsymbol{\Phi}} \boldsymbol{Y}_{t-1} - \widetilde{\boldsymbol{\beta}} \mathbf{X}_{t-1}\|_F^2 + \lambda\Big(\mathcal{P}_y(\widetilde{\boldsymbol{\Phi}}) + \mathcal{P}_x(\widetilde{\boldsymbol{\beta}})\Big),$$

in which $\widetilde{\mathbf{y}}_t = \mathbf{y}_t - \mathbf{C}_y \boldsymbol{Y}_{t-1} - \mathbf{C}_x \mathbf{X}_{t-1}$. We can view the solution to this transformed problem as $[\widetilde{\boldsymbol{\Phi}}, \widetilde{\boldsymbol{\beta}}]^{\lambda}(\mathbf{0}, \mathbf{0})$ operating on $\widetilde{\mathbf{y}}_t$. Hence, transforming back to the setting of Equation (1.12), we find that

$$[\boldsymbol{\Phi}, \boldsymbol{\beta}]^{\lambda}(\mathbf{C}_y, \mathbf{C}_x) = [\mathbf{C}_y, \mathbf{C}_x] + [\widetilde{\boldsymbol{\Phi}}, \widetilde{\boldsymbol{\beta}}]^{\lambda}(\mathbf{0}_{k \times kp}, \mathbf{0}_{k \times ms}).$$

As an example, consider $\mathbf{C}_y = [\mathbf{I}_k, \mathbf{0}_{k \times k}, \ldots, \mathbf{0}_{k \times k}]$, $\mathbf{C}_x = \mathbf{0}_{k \times ms}$, which implements a variant of the Minnesota prior, shrinking the VARX-L model toward a vector random walk. We refer to this extension as the "Minnesota" VARX-L. It could be very useful in economic applications as it is widely believed that many persistent macroeconomic time series can be well approximated by a random walk (Litterman, 1979).

In order to validate this alternative approach, we follow the methodology of Banbura et al. (2009), who also utilize the data from Stock and Watson (2005), but eschew stationarity transformations and work directly with the untransformed series. We again apply our VARX-L forecasting procedures by forecasting the aforementioned *medium* set of ($k = 20$) series using the remaining 20 variables

in the *medium large* set as exogenous predictors, but choose not to perform any stationarity transformations and instead shrink toward a vector random walk.

One advantage of not applying stationarity transformations is that it allows us to utilize more of our data. The data used in section 1.3.5 extends to Quarter 4 of 2008, but one series, non-borrowed depository institutional reserves (FMRNBA), becomes negative in early 2008 due in part to changes in both monetary policy and Federal Reserve accounting (Ip, 2008). The stationarity transformation guidelines provided by Stock and Watson (2005) for this series propose taking the first difference of logs, which is obviously not appropriate for negative values.

Quarter 3 of 1976 to Quarter 2 of 1993 are used for penalty parameter selection while Quarter 3 of 1993 through Quarter 4 of 2008 are used for forecast evaluation. In this application, we also shrink BGR's Bayesian VAR toward a random walk. Our results are summarized in Table 1.4.

We find the each of these Minnesota VARX-L procedures outperform the random walk at both forecast horizons with the Own/Other Group Minnesota VARX-L achieving the best out of sample performance at $h = 1$ and the Basic VARX-L performing the best at $h = 4$.

We observe that under this scenario, the choice of structure substantially affects forecasting performance. Lag based groupings, such as the Lag Group, Sparse Lag Group, and Endogenous-First perform relatively poorly at $h = 1$ but slightly improve relative to other methods at $h = 4$, however they still outper-

Table 1.4: One-step and four-step ahead MSFE (relative to a random walk) for $k = 20$ nonstationary macroeconomic indicators with m=20 exogenous predictors which shrink toward a vector random walk.

Model/Minnesota VARX-L Penalty Structure	One-step Ahead OOS RMSFE	Four-step Ahead OOS RMSFE
Basic	0.8173	0.9460
Lag Group	0.9450	0.9590
Own/Other Group	0.8155	0.9520
Sparse Lag Group	0.9858	0.9702
Sparse Own/Other Group	0.8808	0.9550
Endogenous-First	0.9746	0.9518
VARX with lag selected by AIC	1.2764	1.1896
VARX with lag selected by BIC	1.2764	1.1896
BGR's Bayesian VAR	1.3475	1.0083
Sample Mean	11.304	5.7747
Random Walk	1.0000	1.0000

form the naive methods across both horizons. Their reduced relative performance is likely due to their inability to distinguish between the diagonal random walk component and the coefficients on other lags in the coefficient matrix $\mathbf{\Phi}^{(1)}$.

AIC and BIC are not well suited toward a nonstationarity setting, hence are completely uninformative, selecting lag orders of $p = 1$ and $s = 0$ at every point in time across both horizons. BGR's procedure, despite the imposition of a random walk prior, again produces inferior forecasts to both the VARX-L procedures and the naive random walk.

1.5 Simulation Scenarios

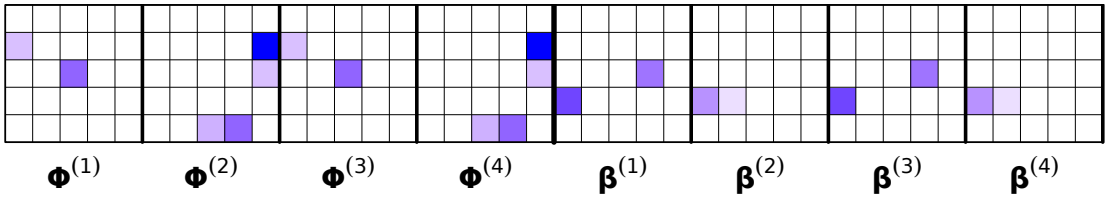
In this section, we consider evaluating the forecasting performance of our procedures on several simulated multivariate time series conforming to different sparsity patterns, with one constructed to be advantageous for each proposed structure. Our objective is to quantify relative performance under both matched and unmatched model sparsity and penalty function structures. All simulations operate on a $\text{VARX}_{5,5}(4,4)$ of length $T = 100$, and each simulation is repeated 100 times. The choice of $p = s = 4$ was selected because it represents one year of dependence for quarterly series, which is a common frequency of macroeconomic data. The middle third of the data is used for penalty parameter selection while the last third is used for forecast evaluation. Under each scenario, Σ_u is distributed according to a multivariate normal distribution with mean $\mathbf{0}_5$ and covariance $(0.01) \times \boldsymbol{I}_5$. We do not include an intercept in any simulation scenarios. The coefficient matrix from each simulation scenario was designed to ensure that a stationary process would be generated.

In order to simulate from a $\text{VARX}_{5,5}(4,4)$, we start by constructing a $\text{VAR}_{10}(4)$. Denoting the first 5 series as $\mathbf{y}_t$ and the second 5 as $\mathbf{x}_t$, we simulate according to the unidirectional relationship

$$\begin{pmatrix} \mathbf{y}_t \\ \mathbf{x}_t \end{pmatrix} = \sum_{\ell=1}^{4} \begin{pmatrix} \boldsymbol{\Phi}^{(\ell)} & \boldsymbol{\beta}^{(\ell)} \\ \mathbf{0} & \boldsymbol{\Gamma}^{(\ell)} \end{pmatrix} \begin{pmatrix} \mathbf{y}_{t-\ell} \\ \mathbf{x}_{t-\ell} \end{pmatrix} + \mathbf{u}_t,$$

in which $\boldsymbol{\Gamma}^{\ell} \in \mathbb{R}^{m \times m}$ denotes the dependence structure of the exogenous series $\mathbf{x}_t$

Figure 1.7: Sparsity Pattern Scenario 1: Unstructured Sparsity. Darker shading represents coefficients that are larger in magnitude.



(which follows the same sparsity pattern as $\mathbf{\Phi}^{\ell}$), and $\mathbf{u}_t \overset{\text{iid}}{\sim} N(\mathbf{0}, 0.01 \times \boldsymbol{I}_{10})$.

Scenario 1: Unstructured Sparsity

We first consider a scenario in which the sparsity is completely random; our sparsity pattern was generated in such a manner that each coefficient was given an equally likely probability of being active. Under such a design, we should expect superior performance from the Basic VARX-L, which assumes no group structure. We do not expect such a structure to be a common occurrence in macroeconomic applications, but it may be present in other application areas, such as internet traffic in which the included series can differ substantially and will likely not exhibit any group structure. This sparsity pattern is depicted in Figure 1.7 and the results are summarized in Table 1.5.

In this scenario, as expected, we find that the Basic VARX-L achieves the best performance. Of the structured methods, the Sparse Own/Other VARX-L performs the best, as it can partially accommodate this sparsity pattern. As expected, the other approaches, which impose a structure that is not present in the data

Table 1.5: Out of sample MSFE of one-step ahead forecasts after 100 simulations: Scenario 1. Standard errors are shown in parentheses.

Model/VARX-L Penalty Structure	MSFE	MSFE Relative to Sample Mean
Basic	0.0645 (0.0012)	0.0454
Lag Group	0.0755 (0.0010)	0.0532
Own/Other Group	0.0734 (0.0010)	0.0517
Sparse Lag Group	0.0724 (0.0009)	0.0510
Sparse Own/Other Group	0.0699 (0.0009)	0.0492
Endogenous-First	0.0779 (0.0010)	0.0549
VARX with lags selected by AIC	0.1040 (0.0017)	0.0733
VARX with lags selected by BIC	0.1183 (0.0032)	0.0833
BGR's Bayesian VAR	0.3675 (0.0124)	0.2590
Sample Mean	1.4187 (0.0681)	1.0000
Random Walk	0.8416 (0.0272)	0.5932

suffer from degraded forecasts, but all structured approaches substantially outperform the AIC and BIC benchmarks. BGR's Bayesian VAR, which cannot perform variable nor lag order selection, achieves substantially worse forecast performance than both information criterion based methods.

Scenario 2: Lag Sparsity

We next consider a scenario in which $\mathbf{\Phi}^{(4)}$ and $\boldsymbol{\beta}^{(4)}$ are dense with coefficients of the same magnitude, and all other coefficients are set to zero. Such a sparsity pattern may be present in disaggregated macroeconomic series, such as agricultural price indices which follow a purely seasonal autoregressive relationship and exhibit a

Figure 1.8: Sparsity Pattern Scenario 2: Lag Sparsity

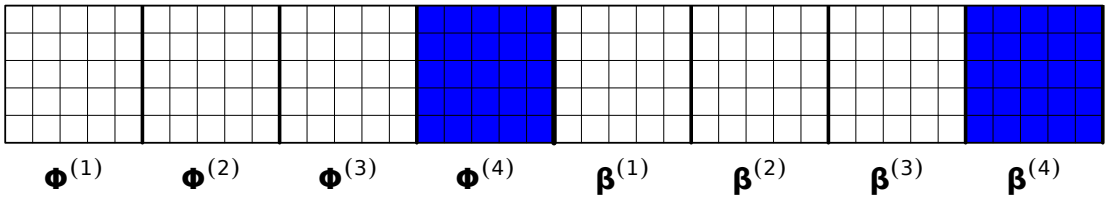


Table 1.6: Out of sample MSFE of one-step ahead forecasts after 100 simulations: Scenario 2. Standard errors are shown in parentheses.

Model/VARX-L Penalty Structure	MSFE	MSFE Relative to Sample Mean
Basic	0.0786 (0.0012)	0.1397
Lag Group	0.0709 (0.0011)	0.1260
Own/Other Group	0.0713 (0.0011)	0.1268
Sparse Lag Group	0.0739 (0.0012)	0.1314
Sparse Own/Other Group	0.0742 (0.0011)	0.1319
Endogenous-First	0.0720 (0.0011)	0.1280
VARX with lags selected by AIC	1.0084 (0.0273)	1.7933
VARX with lags selected by BIC	0.9927 (0.0282)	1.7654
BGR's Bayesian VAR	0.5769 (0.0146)	1.0259
Sample Mean	0.5623 (0.0123)	1.0000
Random Walk	1.1279 (0.0322)	2.0058

substantial degree of cross-dependence. Under such a design, we should expect superior performance from the Lag Group VARX-L, which partitions all coefficients within a lag to the same group. This sparsity pattern is depicted in Figure 1.8, and the results are summarized in Table 1.6.

As expected, we find that the Lag Group VARX-L achieves the best perfor-

mance and all structured approaches outperform the Basic VARX-L. Under this scenario, all VARX-L procedures offer a substantial improvement over the benchmarks. This is likely a result of their ability to effectively leverage the strong signal from the exogenous predictors. Note that although the AIC and BIC benchmarks utilize this exogenous information, they are restricted to select from models of sequentially increasing lag order, hence they cannot accommodate this sparsity pattern and likely overfit, resulting in comparable performance to a random walk. BGR's Bayesian VAR improves upon the information criterion based benchmarks, but since it cannot perform variable selection, it performs substantially worse than all VARX-L methods.

Scenario 3: Structured Lagwise Sparsity, Unstructured Within-Lag

Our third scenario can be thought of as a hybrid of Scenarios 1 and 2. As in Scenario 2, certain coefficient matrices are set identically to zero; only matrices $\mathbf{\Phi}^{(1)}$, $\mathbf{\Phi}^{(4)}$, $\boldsymbol{\beta}^{(1)}$, and $\boldsymbol{\beta}^{(4)}$ contain nonzero coefficients. Additionally, in a similar manner to Scenario 1, sparsity within each lag is unstructured. This scenario can be viewed as a less restrictive and more realistic version of the structure presented in Scenario 2 as it allows the degree of cross-dependence to vary across components. In such a scenario, we should expect procedures that allow for within-group sparsity, such as the Sparse Lag Group VARX-L and Basic VARX-L to achieve the best forecast performance. This sparsity pattern is depicted in Figure 1.9 and the results are summarized in Table 3.6.

Figure 1.9: Sparsity Pattern Scenario 3: Structured Lagwise, Unstructured within Lag

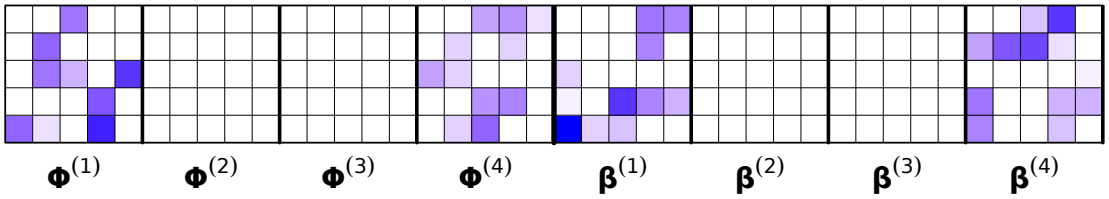
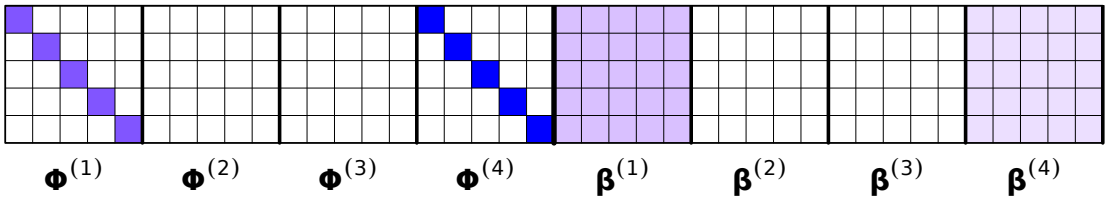


Table 1.7: Out of sample MSFE of one-step ahead forecasts after 100 simulations: Scenario 3. Standard errors are shown in parentheses.

Model/VARX-L Penalty Structure	MSFE	MSFE Relative to Sample Mean
Basic	0.0665 (0.0008)	0.1258
Lag Group	0.0696 (0.0008)	0.1317
Own/Other Group	0.0699 (0.0009)	0.1322
Sparse Lag Group	0.0677 (0.0008)	0.1281
Sparse Own/Other Group	0.0683 (0.0008)	0.1293
Endogenous-First	0.0711 (0.0009)	0.1345
VARX with lags selected by AIC	0.1300 (0.0019)	0.2458
VARX with lags selected by BIC	0.2501 (0.0061)	0.4730
BGR's Bayesian VAR	0.7568 (0.0515)	1.4314
Sample Mean	0.5287 (0.0275)	1.0000
Random Walk	1.3000 (0.0731)	2.4588

Under this scenario, the Basic VARX-L achieves the best performance, followed closely by the Sparse Lag Group VARX-L. Unlike Scenario 2, since this structure exhibits dependence in the first lag, the information-criterion based benchmarks are able to capture a portion of the true underlying structure in both the endogenous and exogenous series and thus substantially outperform the naive benchmarks. However, since they cannot account for within-lag sparsity, they are

Figure 1.10: Sparsity Pattern Scenario 4: Sparse and Diagonally Dominant



still considerably outperformed by all VARX-L methods. As in Scenario 1, BGR's Bayesian VAR performs very poorly, since it cannot perform variable or lag order selection.

Scenario 4: Sparse and Diagonally Dominant

Our final scenario consists of a diagonally-dominant sparsity structure, in which all diagonal elements in $\mathbf{\Phi}^{(1)}$ and $\mathbf{\Phi}^{(4)}$ are equal in magnitude, whereas all off-diagonal endogenous coefficients are set to zero. As in scenario 2, the coefficients in $\boldsymbol{\beta}^{(1)}$ and $\boldsymbol{\beta}^{(4)}$ are identical in magnitude. This structure incorporates the belief posited by Litterman (1986a) that macroeconomic series' own lags are more informative in forecasting applications than lags of other series. Under this setting, one would expect superior performance from the Own/Other Group VARX-L. The sparsity pattern is depicted in Figure 1.10 and the simulation results are summarized in Table 1.8.

Under Scenario 4, as expected, the Own/Other and Sparse Own/Other Group VARX-L achieve superior forecasts. Since the magnitude of coefficients within a

Table 1.8: Out of sample MSFE of one-step ahead forecasts after 100 simulations: Scenario 4. Standard errors are shown in parentheses.

Model/VARX-L Penalty Structure	MSFE	MSFE Relative to Sample Mean
Basic	0.0669 (0.0008)	0.0406
Lag Group	0.0720 (0.0008)	0.0437
Own/Other Group	0.0626 (0.0008)	0.0380
Sparse Lag Group	0.0729 (0.0011)	0.0442
Sparse Own/Other Group	0.0625 (0.0008)	0.0379
Endogenous-First	0.0725 (0.0011)	0.0440
VARX with lags selected by AIC	0.1043 (0.0015)	0.0633
VARX with lags selected by BIC	0.1044 (0.0015)	0.0634
BGR's Bayesian VAR	0.7741 (0.0394)	0.4702
Sample Mean	1.6460 (0.0902)	1.0000
Random Walk	0.7512 (0.0390)	0.4563

lag matrix varies substantially, structures that utilize lag-based groupings, such as the Lag Group and Endogenous-First VARX-L are unable to capture this discrepancy and thus perform relatively poorly. However, they still substantially outperform the benchmark procedures. We again find that VARX with lags selected by AIC and BIC perform very poorly, as they are restricted to select from sequentially increasing lag orders and cannot account for within-lag sparsity. BGR's Bayesian VAR also performs poorly for similar reasons.

Overall, all of the proposed VARX-L models are fairly robust to sparsity patterns not conforming to their true group structures. In each scenario, every method substantially outperforms all benchmark procedures. Scenario 1 is the

only case in which the structured approaches perform poorly relative to the Basic VARX-L. We expect such an unstructured sparsity pattern to occur only rarely in macroeconomic applications.

1.6 Conclusion

We have shown that the proposed VARX-L structured regularization framework is very amenable to the VARX setting in that it can simultaneously reduce its parameter space and still incorporate useful information from both endogenous and exogenous predictors. VARX-L models scale well with the dimension of the data and are quite flexible in accommodating a wide variety of potential dynamic structures. Each of the proposed methods consistently outperforms benchmark procedures both in simulations and in macroeconomic forecasting applications. Forecast performance of all models appears to be robust across multiple sparsity structures as well as forecast horizons. Moreover, upon examining actual macroeconomic data, structured VARX-L models tend to outperform the Basic VARX-L.

Our work has considerable room for extensions. This paper focuses solely on forecasting applications, but our VARX-L framework could also be extended to structural analysis and policy evaluation using an approach similar to that of Furman (2014). In addition, our current implementation requires a coherent maximal lag selection mechanism. The common procedure of choosing a lag order based on the frequency of the data is problematic in that it can lead to overfitting. One

could potentially incorporate an additional penalty parameter that grows as the lag order increases, as in Song and Bickel (2011), but this approach requires a multi-dimensional penalty parameter selection procedure and subjective specification of a functional form for the lag penalty.

An `R` package containing our algorithms and validation procedures, `BigVAR`, is available on the Comprehensive R Archive Network (cran).

Acknowledgements

The authors thank Gary Koop for providing his data transformation script, Marta Bańbura, Domenico Giannone, and Lucrezia Reichlin for sharing their BVAR code, and the attendees of the 2014 NBER/NSF Time Series Conference for their constructive comments. This research was supported by an Amazon Web Services in Education Research Grant. DSM was supported by a Xerox PARC Faculty Research Award and NSF Grant DMS-1455172. JB was supported by NSF DMS-1405746.

CHAPTER 2

HIGH DIMENSIONAL FORECASTING VIA INTERPRETABLE VECTOR AUTOREGRESSION

2.1 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 climatology, neuroscience, and signal processing. 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 of variables. Without further assumptions, VAR modeling is infeasible except in limited situations in which the number of components and the lag order are small.

Many approaches have been proposed for reducing the dimensionality of vector time series models, including canonical correlation analysis (Box and Tiao, 1977), factor models (Peña and Box, 1987), scalar component models (Tiao and Tsay, 1989), independent component analysis (Back and Weigend, 1997), principal component analysis (Stock and Watson, 2002), generalized dynamic factor models (Forni et al., 2000), and dynamic orthogonal component models (Matteson and Tsay, 2011).

Many recent approaches have instead focused on imposing sparsity in the es-

timated coefficient matrices through the use of convex regularizers such as the lasso (Tibshirani, 1996). Most of these methods are adapted from the standard regression setting and do not specifically leverage the ordered structure inherent to the lag coefficients in a VAR. We propose a new class of regularized VAR models, called hierarchical vector autoregression (HVAR), that embed lag order selection into a convex regularizer to provide more reliable forecasts and more interpretable output.

The HVAR 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*. While our motivating goal is to produce interpretable models with improved forecast performance, a convenient byproduct of the HVAR framework is a flexible and computationally efficient method for lag order selection.

Lag order selection procedures have been developed since the inception of VAR. 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 length of the series T grows while the number of components k and the maximal lag order p are held fixed (White, 2001); however, for small sample sizes, it has been observed that no criterion works well (Nickelsburg, 1985). Gonzalo and Pitarakis (2002) find that for fixed k and p, when T is relatively small, Akaike's Information Criterion (AIC) tends to overfit whereas Schwartz's Information Criterion (BIC)

tends to severely underfit. Despite their shortcomings, AIC, BIC, and corrected AIC (AICc, Hurvich and Tsai 1989) are still the preferred tools for lag order selection by most practitioners (Lütkepohl, 2005; Pfaff, 2008; Tsay, 2013).

A drawback with such approaches is 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, it unnecessarily constrains the dynamic relationship between the components, and it impedes forecast performance. 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 (which is sometimes referred to as an *asymmetric VAR*). One such procedure (Hsiao, 1981) starts from univariate autoregressive models and sequentially adds lagged components according to Akaike's "Final Prediction Error" criterion (Akaike, 1969). However, this requires an *a priori* ranking of components based on their perceived predictive power, which is inherently subjective. Keating (2001) offers a more general method which estimates all potential p^k componentwise VAR specifications and utilizes AIC or 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.

Given the difficulties with lag order selection in VAR models, 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. Early shrinkage methods, such as Litterman (1979), take a pragmatic Bayesian perspective. Many such approaches apply the *Minnesota prior,* which uses natural conjugate priors to shrink the VAR toward either an intercept-only model or toward 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 with 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, as in ridge regression, coefficients will not be estimated as exactly zero.

More recent shrinkage approaches have incorporated the lasso (Tibshirani, 1996). Hsu et al. (2008) considers 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 $2^{k^2 p}$ possible models but does not explicitly encourage lags to be small. Song and Bickel (2011) use a group lasso (Yuan and Lin, 2006) penalty structure and 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.

In Section 2.2 we introduce the HVAR framework, which attacks the traditional lag order selection problem through convex regularization. HVAR forces low lag coefficients to be selected before corresponding high lag coefficients, thereby specifically shrinking toward low lag order solutions. This is in contrast to approaches such as Song and Bickel (2011), which increase the weight of the penalty parameter with the coefficients' lag without explicitly enforcing a low-lag structure. In Section 2.2.1 we introduce three hierarchical lag structures that may be desirable when fitting VAR models to data. These structures vary in the degree to which lag order selection is common across different components. For each lag structure, a corresponding HVAR model is detailed in Section 2.2.2 for attaining that sparsity structure. The proposed methodology allows for flexible HVAR estimation in the high dimensional setting with a single tuning parameter. We develop algorithms in Section 2.3 that are computationally efficient and parallelizable across components. A simulation study in Section 2.4 and two macroeconomic applications in Section 2.5 highlight the advantages of HVAR in forecasting and lag order selection. The appendix provides additional details of our simulation methodology.

2.2 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

$$\mathbf{y}_t = \boldsymbol{\nu} + \boldsymbol{\Phi}^{(1)}\mathbf{y}_{t-1} + \cdots + \boldsymbol{\Phi}^{(p)}\mathbf{y}_{t-p} + \mathbf{u}_t, \quad \text{for } t = 1, \ldots, T, \tag{2.1}$$

conditional on initial values $\{\mathbf{y}_{-(p-1)}, \ldots, \mathbf{y}_0\}$, in which $\boldsymbol{\nu} \in \mathbb{R}^k$ denotes an intercept vector, $\{\boldsymbol{\Phi}^{(\ell)} \in \mathbb{R}^{k\times k}\}_{\ell=1}^p$ denote lag-ℓ coefficient matrices, and $\{\mathbf{u}_t \in \mathbb{R}^k\}_{t=1}^T$ denotes a mean zero white noise (serially uncorrelated) vector time series with an 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

$$\sum_{t=1}^{T} \|\mathbf{y}_t - \boldsymbol{\nu} - \sum_{\ell=1}^{p} \boldsymbol{\Phi}^{(\ell)}\mathbf{y}_{t-\ell}\|_2^2 \tag{2.2}$$

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:

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

Equation (2.1) is then simply

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

and the least squares procedure (2.2) 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 vec($\mathbf{A}$) (not 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. 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 (Song and Bickel, 2011; Davis et al., 2012; 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 (Basu and Michailidis, 2013). In what follows, we define a class of sparsity patterns, which we call hierarchical lag structures, that arises in the context of multivariate time series.

2.2.1 Hierarchical Lag Structures

In Equation (2.1), 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 hierarchical lag struc-

tures, we will use the following notational convention: for $1 \leq \ell \leq p$, let

$$\mathbf{\Phi}^{(\ell:p)} = [\mathbf{\Phi}^{(\ell)} \ \cdots \ \mathbf{\Phi}^{(p)}] \in \mathbb{R}^{k \times k(p-\ell+1)}$$

$$\mathbf{\Phi}_i^{(\ell:p)} = [\mathbf{\Phi}_i^{(\ell)} \ \cdots \ \mathbf{\Phi}_i^{(p)}] \in \mathbb{R}^{1 \times k(p-\ell+1)}$$

$$\mathbf{\Phi}_{ij}^{(\ell:p)} = [\mathbf{\Phi}_{ij}^{(\ell)} \ \cdots \ \mathbf{\Phi}_{ij}^{(p)}] \in \mathbb{R}^{1 \times (p-\ell+1)}.$$

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 hierarchical lag 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 (2.1), 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 hierarchical lag structures, in this paper we focus on the following three fundamental structures.

1. **Componentwise (C).** A componentwise hierarchical lag structure allows each of the k marginal equations from (2.1) 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 (2.1), 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 hierarchical lag structure is illustrated in Figure 2.1.

2. **Own-Other (O).** The own-other hierarchical lag structure is similar to the componentwise structure, 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,

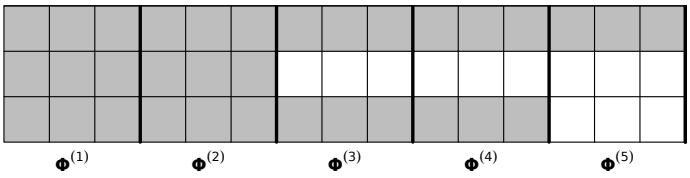
$$\mathbf{L}_{ij} = L_i^{other} \text{ for } i \neq j \text{ and } \mathbf{L}_{ii} \in \{L_i^{other}, L_i^{other} + 1\}, \text{ for } i = 1, \dots k.$$

   This hierarchical lag structure allows each component of $\mathbf{y}_t$ to have longer range lagged self-dependence than lagged cross-dependencies. This own-other hierarchical lag structure is illustrated in Figure 2.2.

3. **Elementwise (E).** Finally, we consider a completely flexible hierarchical lag structure in which the elements of $\mathbf{L}$ have no stipulated relationships. This elementwise hierarchical lag structure is illustrated in Figure 2.3.

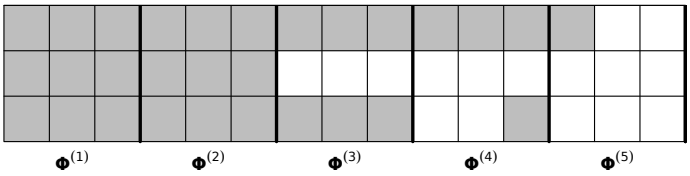
In the next section, we introduce the proposed class of HVAR estimators aimed at estimating $\text{VAR}_k(p)$ models while shrinking the elements of $\mathbf{L}$ towards zero by incorporating the three hierarchical lag structures described above.

Figure 2.1: A componentwise (C) hierarchical lag structure: $\text{HVAR}_3^C(5)$.



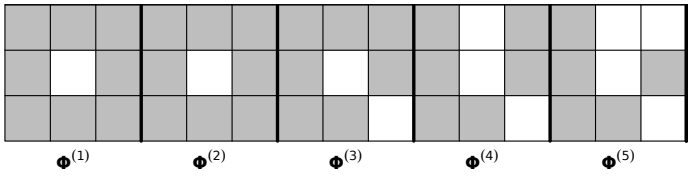
$$\mathbf{L}^C = \begin{pmatrix} 5 & 5 & 5 \\ 2 & 2 & 2 \\ 4 & 4 & 4 \end{pmatrix}$$

Figure 2.2: An own-other (O) hierarchical lag structure: $\text{HVAR}_3^O(5)$.



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

Figure 2.3: An elementwise (E) hierarchical lag structure: $\text{HVAR}_3^E(5)$.



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

2.2.2 HVAR: Hierarchical Group Lasso for Lag Structured VAR Modeling

In this section, we introduce convex penalties specifically tailored for attaining the three lag structures presented in the previous section. Our primary model-

ing tool is the hierarchical group lasso (Zhao et al., 2009), 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, 2013; Haris et al., 2014; She and Jiang, 2014), covariance estimation (Bien et al., 2014), additive modeling (Lou et al., 2014a), and time series (Suo and Tibshirani, 2014). 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.

1. The **HVAR**^{C} objective is a *componentwise* hierarchical lag structure and is defined by

$$\min_{\boldsymbol{\nu},\boldsymbol{\Phi}} \left\{ \frac{1}{2}\|\mathbf{Y} - \boldsymbol{\nu}\mathbf{1}^\top - \boldsymbol{\Phi}\mathbf{Z}\|_2^2 + \lambda \sum_{i=1}^{k} \sum_{\ell=1}^{p} \|\boldsymbol{\Phi}_i^{(\ell:p)}\|_2 \right\}, \tag{2.3}$$

   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 $\hat{\boldsymbol{\Phi}}_i^{(\ell:p)} = \mathbf{0}$ for more i, and for smaller ℓ. This componentwise hierarchical group 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. The **HVAR**^{O} objective aims for a *own-other* hierarchical lag structure and is defined by

$$\min_{\boldsymbol{\nu}, \mathbf{\Phi}} \left\{ \frac{1}{2} \|\mathbf{Y} - \boldsymbol{\nu}\mathbf{1}^\top - \mathbf{\Phi}\mathbf{Z}\|_2^2 + \lambda \sum_{i=1}^{k} \sum_{\ell=1}^{p} \left[\|\mathbf{\Phi}_i^{(\ell:p)}\|_2 + \|(\mathbf{\Phi}_{i,-i}^{(\ell)}, \mathbf{\Phi}_i^{([\ell+1]:p)})\|_2 \right] \right\}, \tag{2.4}$$

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 (2.3). 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 hierarchical lag 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. The **HVAR**^{E} objective aims for an *elementwise* hierarchical lag structure and is defined by

$$\min_{\boldsymbol{\nu}, \mathbf{\Phi}} \left\{ \frac{1}{2} \|\mathbf{Y} - \boldsymbol{\nu}\mathbf{1}^\top - \mathbf{\Phi}\mathbf{Z}\|_2^2 + \lambda \sum_{i=1}^{k} \sum_{j=1}^{k} \sum_{\ell=1}^{p} \|\mathbf{\Phi}_{ij}^{(\ell:p)}\|_2 \right\}. \tag{2.5}$$

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 (2.3).

Since all three objectives are based on hierarchical group lasso penalties, a unified computational approach to solve each is detailed in the next section.

2.3 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

$$\min_{\boldsymbol{\Phi}} \left\{ \frac{1}{2}\|\mathbf{Y} - \boldsymbol{\Phi}\mathbf{Z}\|_2^2 + \lambda \sum_{i=1}^{k} \sum_{\ell=1}^{p} \Omega_i(\boldsymbol{\Phi}_i^{(\ell:p)}) \right\}, \tag{2.6}$$

and (2.3), (2.4), and (2.5) 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 $\boldsymbol{\Phi}$:

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

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 (2.6) can be solved in parallel by solving the "one-row" subproblem

$$\min_{\boldsymbol{\Phi}_i} \left\{ \frac{1}{2}\|\mathbf{Y}_i - \boldsymbol{\Phi}_i\mathbf{Z}\|_2^2 + \lambda \sum_{\ell=1}^{p} \Omega_i(\boldsymbol{\Phi}_i^{(\ell:p)}) \right\}. \tag{2.7}$$

Jenatton et al. (2011) 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 $f_i(\mathbf{\Phi}_i) = \mathcal{L}_i(\mathbf{\Phi}_i) + \lambda\Omega_i^*(\mathbf{\Phi}_i)$, where $\mathcal{L}_i$ is differentiable with a Lipschitz continuous gradient, the proximal gradient method produces a sequence $\hat{\mathbf{\Phi}}_i[1], \hat{\mathbf{\Phi}}_i[2], \ldots$ with the guarantee that

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

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

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

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 $\mathcal{L}_i$:

$$\mathrm{Prox}_{s_m\lambda\Omega_i^*}(u) = \underset{v}{\mathrm{argmin}}\left\{\frac{1}{2}\|u - v\|_2^2 + s_m\lambda\Omega_i^*(v)\right\}. \tag{2.8}$$

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. (2011) show that for such penalties, the proximal operator has essentially a closed form solution, making it extremely efficient. It remains to note that $\mathcal{L}_i(\mathbf{\Phi}_i) = \frac{1}{2}\|\mathbf{Y}_i - \mathbf{\Phi}_i\mathbf{Z}\|_2^2$ has gradient $\nabla\mathcal{L}_i(\mathbf{\Phi}_i) = -(\mathbf{Y}_i - \mathbf{\Phi}_i\mathbf{Z})\mathbf{Z}^\top$ 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 $\mathbf{Z}$).

Beck and Teboulle (2009) develop an accelerated version of the proximal gradient method, which they call the Fast Iterative Soft-Thresholding Algorithm (FISTA). This 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,

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

and converges at rate $1/m^2$ (compared to the unaccelerated proximal gradient method's $1/m$ rate). The full procedure is detailed in Algorithm 9 and is applicable to all three HVAR estimators.

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. (2011), which shows that the proximal operator has a remarkably simple form. We write these three problems generically as

$$\hat{\mathbf{x}} = \underset{\mathbf{x}}{\mathrm{argmin}} \left\{ \frac{1}{2}\|\mathbf{x} - \tilde{\mathbf{x}}\|_2^2 + \lambda \sum_{h=1}^{H} w_h \|\mathbf{x}_{g_h}\|_2 \right\}, \tag{2.9}$$

where $g_1 \subset \cdots \subset g_H$. The key observation in Jenatton et al. (2011) is that the dual of the proximal problem (2.8) can be solved exactly in a *single pass* of blockwise coordinate descent. By strong duality, this solution to the dual provides us with

Algorithm 1: General algorithm for HVAR with penalty Ω_i^*

Require: $\mathbf{Y}, \mathbf{Z}, \hat{\mathbf{\Phi}}[0], \lambda, \epsilon$

$\hat{\mathbf{\Phi}}[1] \leftarrow \hat{\mathbf{\Phi}}[0]; \quad \hat{\mathbf{\Phi}}[2] \leftarrow \hat{\mathbf{\Phi}}[0]$

$s \leftarrow \sigma_1(Z)^{-2}$

for $i = 1, \ldots, k$ **do**

  for $m = 1, 2, \ldots$ **do**

    $\hat{\phi} \leftarrow \hat{\mathbf{\Phi}}_i[m-1] + \frac{m-2}{m+1}\left(\hat{\mathbf{\Phi}}_i[m-1] - \hat{\mathbf{\Phi}}_i[m-2]\right)$

    $\hat{\mathbf{\Phi}}_i[m] \leftarrow \text{Prox}_{s\lambda\Omega_i^*}\left(\hat{\phi} + s \cdot (\mathbf{Y}_i - \hat{\phi}\mathbf{Z})\mathbf{Z}^\top\right)$

    if $\|\hat{\phi} - \hat{\mathbf{\Phi}}_i[m]\|_\infty \leq \epsilon$ **then**

      break

    end if

  end for

end for

return $\hat{\mathbf{\Phi}}[m]$

a solution to problem (2.8). Furthermore, the updates of each block are extremely simple, corresponding to a groupwise-soft-thresholding operation. Algorithm 4 shows the solution to (2.9), which includes all three of our penalties as special cases.

Algorithm 2: Solving Problem (2.9)

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 $\hat{\mathbf{x}}$.

2.4 Simulation Study

In this section we compare the proposed HVAR methods with competing VAR modeling approaches. After detailing these comparison methods below, we describe three simulation scenarios and discuss the forecast and lag order selection performance of each estimator. Finally, we examine the performance of the proposed HVAR methods in a low-dimensional simulation setting while allowing the maximal lag order p to vary.

2.4.1 Comparison Methods

A standard method in lower dimensional settings is to fit a $VAR_k(\ell)$ with least squares for $0 \leq \ell \leq p$ and then to select a universal lag order ℓ using AIC or BIC.

Per Lütkepohl (2005), the AIC and BIC of a $VAR_k(\ell)$ are defined as

$$\text{AIC}(\ell) = \log\det(\hat{\boldsymbol{\Sigma}}_u^\ell) + \frac{2k^2\ell}{T}, \tag{2.10}$$

$$\text{BIC}(\ell) = \log\det(\hat{\boldsymbol{\Sigma}}_u^\ell) + \frac{\log(T)k^2\ell}{T}, \tag{2.11}$$

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 the first set of simulations that follow, we cannot use least squares for $\ell > 1$, thus for a simple benchmark we instead estimate a $VAR_k(1)$ by least squares:

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

where $\boldsymbol{Z}^{(1)} = [\mathbf{y_0} \ \cdots \ \mathbf{y_{T-1}}]$. We also include two well-known regularization approaches. The *lasso* estimates the VAR using an L_1-penalty:

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

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:

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

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 HVAR penalties in that it does not necessarily produce sparsity patterns in which all coefficients beyond a certain lag order are zero.

Finally, we compare against two naive approaches to serve as simple baselines: the unconditional *sample mean* corresponds to the intercept-only model,

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

which makes one-step-ahead forecasts of the form $\hat{\mathbf{y}}_{t+1} = \frac{1}{t}\sum_{\ell=1}^{t}\mathbf{y}_\ell$; and 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 $\hat{\mathbf{y}}_{t+1} = \mathbf{y}_t$.

2.4.2 Simulation Settings

In order to demonstrate the efficacy of the HVAR methods in applications with various lag structures, we evaluate forecasts produced by the proposed methods under several simulation scenarios. In these scenarios, we have $k = 60$ components, a maximal lag order of $p = 12$, and a series length of $T = 100$; the error covariance is taken to be $\boldsymbol{\Sigma}_u = 0.01 \cdot \mathbf{I}_{60}$. All simulations are generated from stationary coefficient matrices. The steps taken to ensure the stationarity of the simulation structures are described in detail in Section B.0.7 of the appendix. Simulation results are based on 100 iterations.

The penalty parameters were selected using the rolling cross-validation approach utilized by Song and Bickel (2011) and Banbura et al. (2009), with the middle third of the data used for penalty parameter selection and the last third for forecast evaluation. In the case of the lag-weighted lasso, λ and α were jointly selected. Given an evaluation period $(T_1, T_2]$, we use *mean-squared one-step-ahead forecast error* (MSFE) as a measure of forecast performance:

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

where $\hat{\mathbf{y}}_{i,t+1}$ denotes the forecast of a method for time $t+1$ and component i based on observing the series up to time t.

We evaluate the methods under three lag structures.

Simulation Scenario 1: Componentwise Lag Structure. In this scenario, we simulate according to an $\text{HVAR}^C_{60}(5)$ structure. In particular, we choose the maxlag matrix

$$\mathbf{L} = [1, 2, 3, 4, 5]^\top \otimes (\mathbf{1}_{12}\mathbf{1}_{60}^\top).$$

This 60×60 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 12, each group taking on a distinct maxlag in $\{1, 2, 3, 4, 5\}$. A coefficient matrix $\mathbf{\Phi}$ with maxlag matrix $\mathbf{L}$ is used in Scenario 1's simulations and its magnitudes are depicted in Figure 2.8. The prediction performance of the methods under study is shown in Table 2.1. The componentwise and own-other HVAR methods perform best, which is to be expected since both methods are geared explicitly toward a

Figure 2.4: Sparsity pattern (and magnitudes) of the $\text{HVAR}^C_{60}(5)$ structure used in simulation Scenario 1.

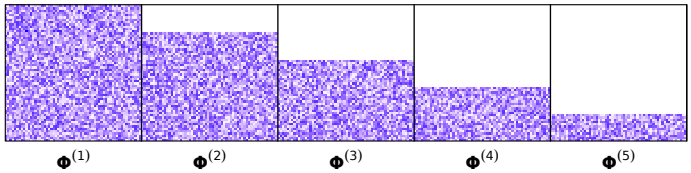
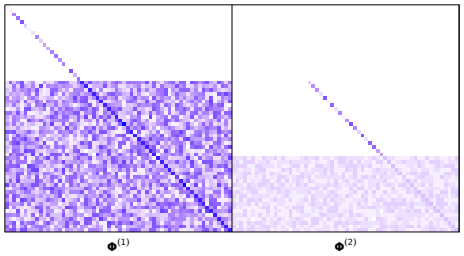


Table 2.1: Out-of-sample mean-squared one-step-ahead forecast error (standard errors are in parentheses) for Scenario 1 based on 100 simulations.

Class	**Method**	**MSFE**
HVAR	Componentwise	0.0464 (0.0005)
	Own-other	0.0470 (0.0005)
	Elementwise	0.0546 (0.0005)
VAR	Lasso	0.0598 (0.0006)
	Lag-weighted lasso	0.0547 (0.0006)
	Least squares	0.1599 (0.0020)
Other	Sample mean	0.1295 (0.0034)
	Random walk	0.3054 (0.0107)

lag structure as in Scenario 1. The lag-weighted lasso and elementwise HVAR perform similarly, and both of them do better than the regular lasso. With a total of $pk^2 = 12(60)^2 = 43,200$ coefficient parameters to estimate, the methods that assume an ordering are greatly advantaged over a method like the lasso that does not exploit this knowledge. One exception is the $\text{VAR}_{60}(1)$ model that is fit using least squares: Despite this method's explicit orientation toward modeling recent behavior, it suffers both because it misses important longer range lag coefficients

Figure 2.5: Sparsity pattern (and magnitudes) of the $\text{HVAR}^{O}_{60}(2)$ structure used in simulation Scenario 2.



and because it is an unregularized estimator of $\mathbf{\Phi}^{(1)}$ and therefore has high variance.

Simulation Scenario 2: Own-Other Lag Structure. In this scenario, we create the matrix $\mathbf{\Phi}$ in such a manner that it differentiates between *own* and *other* coefficients. The coefficients of a series' "own lags" (i.e., $\mathbf{\Phi}^{(\ell)}_{ii}$) are larger in magnitude than those of "other lags" (i.e., $\mathbf{\Phi}^{(\ell)}_{ij}$ with $i \neq j$). The magnitude of coefficients decreases as the lag order increases. The $\text{HVAR}^{O}_{60}(2)$ model we simulate is depicted in Figure 2.5. The first 20 rows can be viewed as univariate autoregressive models in which only the *own* term is nonzero; in the next 20 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 20 rows have nonzeros throughout the first $2k$ coefficients, with *own* coefficients dominating *other* coefficients in magnitude. The results from this scenario are shown in Table 2.2. Here, all three HVAR methods lead the competing methods. As one would expect, the own-other HVAR procedure achieves the best forecasting performance, with the componentwise HVAR performing only slightly worse. We find that the

Table 2.2: Out-of-sample mean-squared one-step-ahead forecast error (standard errors are in parentheses) for Scenario 2 based on 100 simulations.

Class	**Method**	**MSFE**
HVAR	Componentwise	0.0193 (0.0001)
	Own-other	0.0183 (0.0001)
	Elementwise	0.0210 (0.0002)
VAR	Lasso	0.0270 (0.0003)
	Lag-weighted lasso	0.0228 (0.0003)
	Least squares	0.0544 (0.0007)
Other	Sample mean	0.2948 (0.0178)
	Random walk	0.1381 (0.0086)

lag-weighted lasso performs worse than the elementwise HVAR, but much better than the lasso without weights. As with the previous scenario, the least-squares approach is not competitive.

Simulation Scenario 3: Elementwise Lag Structure. In this final scenario, we simulate under an $\text{HVAR}^E_{60}(4)$ model, meaning that the maxlag is allowed to vary not just across rows but also *within* rows. The maxlag matrix is given by

$$\mathbf{L} = \begin{pmatrix} 4 & 3 & 2 & 1 \\ 3 & 2 & 1 & 0 \\ 2 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix} \otimes (\mathbf{1}_{15}\mathbf{1}_{15}^\top).$$

A coefficient matrix corresponding to this lag structure is depicted in Figure 2.6,

Figure 2.6: Sparsity pattern (and magnitudes) of the $\text{HVAR}^{E}_{60}(4)$ structure used in simulation Scenario 3.

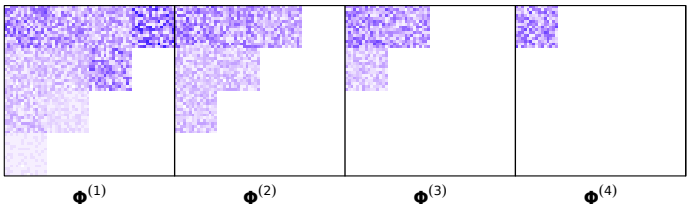


Table 2.3: Out-of-sample mean-squared one-step-ahead forecast error (standard errors are in parentheses) for Scenario 3 based on 100 simulations.

Class	**Method**	**MSFE**
HVAR	Componentwise	0.0323 (0.0007)
	Own-other	0.0335 (0.0007)
	Elementwise	0.0295 (0.0004)
VAR	Lasso	0.0379 (0.0007)
	Lag-weighted lasso	0.0380 (0.0007)
	Least squares	0.1071 (0.0016)
Other	Sample mean	0.0915 (0.0038)
	Random walk	0.1253 (0.0039)

and the results are shown in Table 2.3.

As expected, the elementwise HVAR method outperforms all others. The chosen $\mathbf{L}$ violates the own-other lag structure in 45 of the 60 rows (and it violates the componentwise lag structure in every row). Even so, these two HVAR methods outperform the lasso and weighted lasso methods.

2.4.3 Lag Order Selection

While our primary intent in introducing the HVAR framework is better forecast performance and improved interpretability, one can also view HVAR 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 2.2.1. Based on an estimate $\hat{\mathbf{\Phi}}$ of the autoregressive 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 rolling 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. This refitting procedure is described in detail in Section B.2 in 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" heuristic 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).

Table 2.4: Lag selection performance (standard errors in parentheses) for Scenario 1 based on 100 simulations.

Class	**Method**	$\|\|\hat{\mathbf{L}} - \mathbf{L}\|\|_1 / \|\|\mathbf{L}\|\|_1$
HVAR	Componentwise	0.5898 (0.0495)
	Own-other	0.5792 (0.0447)
	Elementwise	0.9684 (0.0034)
VAR	Lasso	0.9968 (0.0013)
	Lag-weighted lasso	0.9936 (0.0023)
Other	Sample mean	1.0000 (0.0000)

We measure a procedure's lag order selection performance based on the sum of absolute differences between $\mathbf{L}$ and $\hat{\mathbf{L}}$:

$$||\hat{\mathbf{L}} - \mathbf{L}||_1 = \sum_{ij} |\hat{\mathbf{L}}_{ij} - \mathbf{L}_{ij}|.$$

In particular, Tables 2.4, 2.5, and 2.6 report this value for various methods relative to that of the sample mean (which chooses $\hat{\mathbf{L}}_{ij} = 0$ for all i and j).

In Scenario 1, the own-other and componentwise HVARs achieve the best performance; every other approach scarcely outperforms the benchmark. The fact that the own-other and componentwise HVARs perform best is no surprise given that they both are geared toward a lag structure as in Scenario 1.

In Scenario 2, the own-other HVAR achieves the best performance followed by the componentwise HVAR; the elementwise HVAR performs much worse than these but still better than all other methods.

Table 2.5: Lag selection performance (standard errors in parentheses) for Scenario 2 based on 100 simulations.

Class	**Method**	$\|\|\hat{\mathbf{L}} - \mathbf{L}\|\|_1 / \|\|\mathbf{L}\|\|_1$
HVAR	Componentwise	0.5376 (0.0060)
	Own-other	0.4329 (0.0061)
	Elementwise	0.9561 (0.0009)
VAR	Lasso	1.0395 (0.0023)
	Lag-weighted lasso	1.0551 (0.0031)
Other	Sample mean	1.0000 (0.0000)

Table 2.6: Lag selection performance (standard errors in parentheses) for Scenario 3 based on 100 simulations.

Class	**Method**	$\|\|\hat{\mathbf{L}} - \mathbf{L}\|\|_1 / \|\|\mathbf{L}\|\|_1$
HVAR	Componentwise	1.3260 (0.0382)
	Own-other	1.2573 (0.0365)
	Elementwise	0.8782 (0.0035)
VAR	Lasso	1.0121 (0.0020)
	Lag-weighted lasso	1.0144 (0.0043)
Other	Sample mean	1.0000 (0.0000)

Interestingly, in Scenario 3, the elementwise HVAR approach is the only method to perform better than the sample-mean baseline. We see that the HVAR methods that incorrectly assume that maxlag should be constant (or near constant) within a row pay a price in lag order selection, making them even worse than the lasso methods.

2.4.4 Simulation Scenario 4: Robustness of HVAR as p increases

We additionally examine the impact of the upper bound for maximal lag order p on HVAR's performance. Ideally, provided that p is large enough to capture the dynamics of the system, its choice should have little impact on forecast performance. However, we should expect regularizers that treat each coefficient democratically, such as the lasso, to experience degraded forecast performance as p increases.

As an experiment, we simulate from an $\text{HVAR}^{C}_{10}(5)$ while increasing the upper bound on the maximal lag order to substantially exceed the true $\mathbf{L}$. 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 $\mathbf{L} = 0$. Figure 2.7 depicts the coefficient matrix $\mathbf{\Phi}$ and its magnitudes.

We consider varying $p \in (1, 5, 12, 25, 50)$. As p increases, we should expect the performance of the HVAR procedures to remain relatively constant whereas the lasso and information-criterion based methods should return worse forecasts.

Figure 2.7: Sparsity pattern (and magnitudes) of the $\text{HVAR}^{C}_{10}(5)$ structure used in simulation Scenario 4.

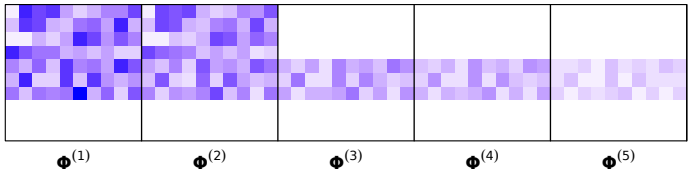


Figure 2.8: Simulation Results: Scenario 4 (AIC omitted due to extremely poor performance)

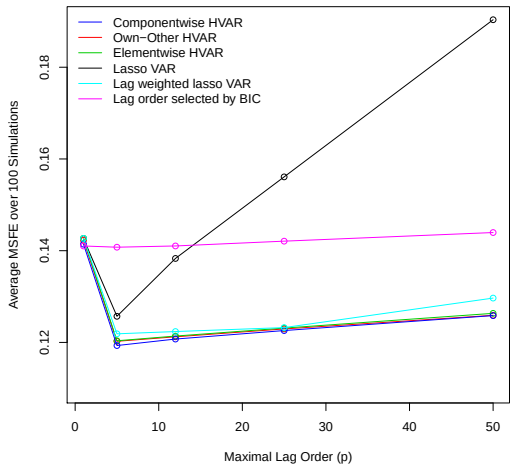


Table 2.7: Out-of-sample mean-squared one-step-ahead forecast error (standard errors in parentheses) for Scenario 4 based on 100 simulations (T=200).

Class	**Method**	**MSFE (p=1)**	**MSFE (p=5)**	**MSFE (p=12)**	**MSFE (p=25)**	**MSFE (p=50)**
HVAR	Componentwise	0.0141 (0.0010)	0.0119 (0.0077)	0.0120 (0.0008)	0.0122 (0.0009)	0.0125 (0.0010)
	Own-other	0.0142 (0.0010)	0.0120 (0.0008)	0.0121 (0.0008)	0.0123 (0.0008)	0.0126 (0.0010)
	Elementwise	0.0142 (0.0011)	0.0120 (0.0008)	0.0121 (0.0008)	0.0123 (0.0008)	0.0126 (0.0009)
VAR	Lasso	0.0142 (0.0011)	0.0125 (0.0008)	0.0138 (0.0010)	0.0156 (0.0013)	0.0190 (0.0016)
	Lag-weighted lasso	0.0143 (0.0011)	0.0121 (0.0008)	0.0122 (0.0008)	0.0123 (0.0015)	0.0129 (0.0012)
	AIC	0.0141 (0.0010)	0.0117 (0.0080)	0.0535 (0.0070)	0.0781 (0.0121)	0.0855 (0.0130)
	BIC	0.0141 (0.0010)	0.0140 (0.0011)	0.0141 (0.0011)	0.0142 (0.0012)	0.0144 (0.0013)

At $p = 1$ all models are misspecified. Since no method is capable of capturing the true dynamics of series 1-7 in Figure 2.8, all perform poorly. As expected, after ignoring $p = 1$, the componentwise HVAR achieves the best performance across all other choices for p, although own-other and elementwise HVAR's performances are within one standard error. Among the information-criterion based methods, AIC performs substantially worse than BIC as p 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 is somewhat more robust to the lag order, but still achieves worse forecasts than every HVAR procedure under all scenarios except for $p = 25$ where it performs comparably.

2.5 Data Analysis

2.5.1 Macroeconomic Application

We now apply the proposed HVAR estimation methods to a collection of US macroeconomic time series compiled by Stock and Watson (2005) and augmented by Koop (2011). The full dataset contains 168 quarterly macroeconomic indicators over 45 years, representing information about many aspects of the US economy, including income, employment, stock prices, exchange rates, etc. The full list of variables considered is available in Koop (2011), who defines various nested

groups of components:

- *Small ($k = 3$)*: Federal Funds Rate, CPI, and GDP growth rate; this core group is commonly considered in basic Dynamic Stochastic Generalized Equilibrium modeling;
- *Medium ($k = 20$)*: Small group plus 17 additional variables, including indices for consumption, labor, and housing, as well as exchange rates;
- *Medium-Large ($k = 40$)*: Medium group plus 20 additional aggregate variables;
- *Large ($k = 168$)*: Medium-Large group plus 128 additional variables, consisting primarily of the components that make up the aggregated variables (e.g. subsets of Gross Domestic Product, Bond Interest Rates, Industrial Production, etc).

Forecast Comparisons. We initially focus on forecasting the *Medium-Large* ($k = 40$) and *Large* ($k = 168$) groups. We apply the transformation codes provided by Stock and Watson (2005) to make the data approximately stationary, then we standardize each series to have sample mean zero and variance one. Quarter 3 of 1977 to Quarter 3 of 1992 is used for penalty parameter selection while Quarter 4 of 1992 to Quarter 4 of 2007 is used for forecast performance comparisons. Following the convention from Banbura et al. (2009), we set the maximal lag order p to 13. In the *Large* group, VAR by AIC and BIC are overparameterized and not included.

Table 2.8: Rolling out of sample one-step ahead MSFE for the Medium-Large ($k = 40$) and Large ($k = 168$) groups of macroeconomic indicators.

		Medium-Large	**Large**
Class	**Method**	$k = 40$	$k = 168$
HVAR	Componentwise	0.5342	0.6188
	Own-other	0.5102	0.5749
	Elementwise	0.5138	0.5752
VAR	Lasso	0.5385	0.5916
	Lag-weighted lasso	0.5262	0.5876
	AIC	2.9750	N/A
	BIC	0.7502	N/A
Other	Sample mean	0.6861	0.7486
	Random walk	1.1775	1.3306

The rolling out of sample one-step-ahead mean square forecast error (MSFE) for the *Medium-Large* and *Large* groups are summarized in Table 2.9. The proposed HVAR methods outperformed the lasso, least squares, and both information-criterion based models for the *Medium-Large* group over this evaluation period. Among the HVAR methods, the more flexible own-other and elementwise structures performed similarly, and better than the componentwise structure. The sample mean and random walk forecast results are provided for comparison.

As the number of component series k increases, the componentwise hierarchical lag order 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 Table 2.9 we see that the component-wise HVAR performs more similarly to the lasso and least squares methods for the *Large* group over this evaluation period. The own-other and elementwise HVAR methods again had the best forecasting performance. 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.

Lag Order Selection. The *Small* group includes Real Gross Domestic Product (GDP251), a measure of economic activity, Consumer Price Index (CPIAUSL), a measure of inflation, and the Federal Funds Rate (FYFF), a measure of monetary policy. This core subset is generally of primary interest to forecasters and policy-makers. We now examine the estimated lag order of these three component series from a fitted HVAR^{E}_{40} model of the *Medium-Large* group over the entire observation period. The estimated lag order is shown in Figure 2.11.

Most of the lag orders chosen by the elementwise HVAR have an underlying economic interpretation. The Federal Funds Rate (FYFF), has been shown in Bernanke and Blinder (1992) to be an important predictor of several measures of economic activity, including the components of Gross Domestic Product. Additionally, the "Taylor Rule" (Taylor, 1993) suggests that the Federal Funds Rate is set to control inflation, hence we should expect changes in the previous quarter to aid in forecasting inflation.

The non-core component series which have high maximal lag orders are also economically rational. *FYGT10*, the interest rate on 10 year maturity Treasury Bonds, has a maxlag of 6 in the Consumer Price Index regression and 3 in the Federal Funds Rate regression. 10 year bond yields historically serve as a proxy for the Federal Reserve's monetary policy, and here we see that it aids in predicting both inflation and the federal funds rate. *UTL11*, industrial capacity utilization, is an important indicator of economic activity, and similar to inflation, it is widely believed that the Federal Reserve sets its Federal Funds Rate in order to achieve a target level of capacity utilization (McElhattan, 1978). *GDP263*, which denotes real exports has a maxlag of 8 in the GDP growth rate regression; Marin (1992) showed that there exists a Granger causal relationship between the US growth rate and net exports. *GDP273* is a price index constructed by the Bureau of Economic Analysis which has objectives similar to the Consumer Price Index, hence, they appear to exhibit a high degree of lagged dependence.

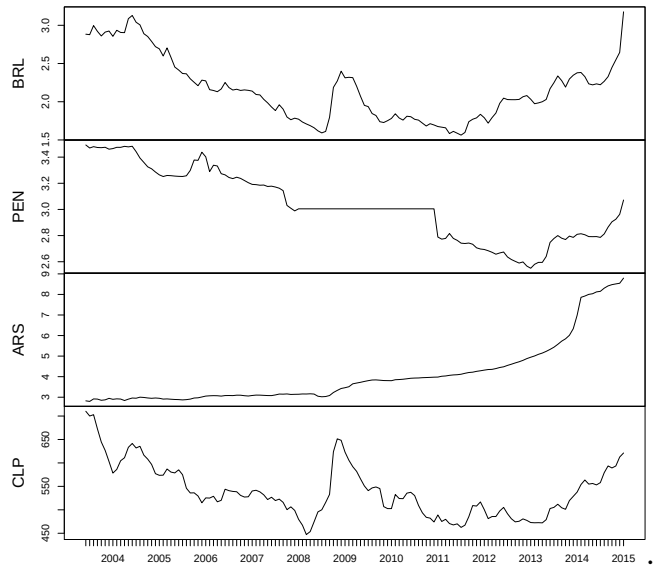
2.5.2 Exchange Rate Application

We additionally consider utilizing the proposed HVAR procedures to forecast monthly exchange rates. Exchange rates are notoriously difficult to forecast; most economically-motivated models are substantially outperformed by a simple random walk (Meese and Rogoff, 1983). However, recent developments by Carriero et al. (2009) have demonstrated that jointly forecasting a panel of exchange rates using a Bayesian VAR can lead to substantial forecast improvements over a ran-

dom walk.

Our application forecasts the monthly exchange rates of the currencies of four South American nations in relation to the US dollar. These currencies include the Argentinian Peso (ARS), Brazilian Real (BRL), Chilean Peso (CLP), and the Peruvian Nuevo Sol (PEN). We chose a cross-section of emerging currencies rather than their more established counterparts as emerging currencies are subjected to a wide range of monetary policies but are still closely related, both economically and geographically. Our data were acquired from the proprietary CUR database hosted on `Quandl` and ranges from June of 2003 to January 2015 ($T = 140$). The series are plotted in Figure 2.9.

Figure 2.9: Plots of the monthly exchange rate vis-a-vis the US dollar for the Brazilian Real (BRL, first), the Peruvian Nuevo Sol (PEN, second), Argentinian Peso (ARS, third), and the Chilean Peso (CLP, fourth).



Note that the Argentinian Peso and Peruvian Nuevo Sol both follow strikingly

different trends than the other two currencies. This is likely due to their respective nations' aggressive inflation-targeting monetary policies which require intense intervention to promote stability as compared to the relatively free-floating policies of the other two countries, in which rates tend to vary according to market demand (Frenkel and Rapetti, 2010).

Since all of these series exhibit evidence of nonstationarity, instead of shrinking every coefficient to zero, following Carriero et al. (2009), we modify our procedures to instead shrink toward a vector random walk. The training period ranges from November 2006 to June 2010 and the forecast evaluation ranges from July 2010 to January 2015. Table 2.9 displays the rolling out of sample one-step-ahead MSFE (relative to a random walk).

We find that every HVAR and lasso VAR model improves upon the random walk's forecasting performance, with the own-other HVAR and componentwise HVAR achieving the greatest improvement. These results suggest that even at a low dimension, imposing the HVAR framework leads to a substantial improvement in forecasting performance.

As in the previous application, we examine the estimated lag orders selected by the elementwise HVAR for each of the exchange rates, which are plotted in Figure 2.10. Though the true underlying exchange rate dynamics are likely beyond the scope of this model, the maxlags selected offer interesting economic insights. There is a strong bilateral relationship between Argentina and Peru (maxlags of 6 and 5, respectively). This could be the result of a strong degree of interdependence

Table 2.9: Rolling out of sample one-step ahead MSFE for $k = 4$ monthly exchange rate forecasts (relative to a random walk), $p = 12$

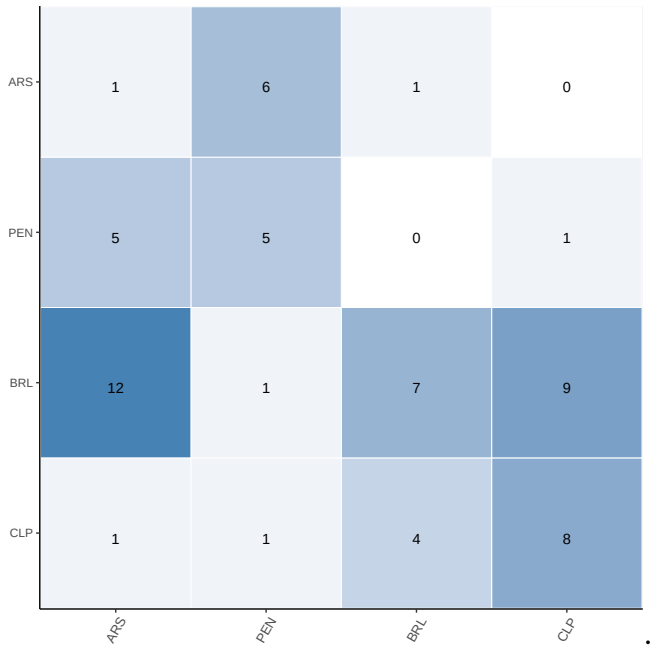
Class	**Method**	Relative MSFE
HVAR	Componentwise	0.885
	Own-other	0.878
	Elementwise	0.914
VAR	Lasso	0.935
	Lag-weighted lasso	0.943
	AIC	1.064
	BIC	1.093
Other	Sample mean	44.00
	Random walk	1.000

between their economies, as both economies are centered around exports of raw materials. Argentina's exchange rate appears to be very important in forecasting Brazil's (maxlag of 12), but the reverse is not true. This unilateral relationship could be due to Brazil's considerable imports from Argentina.

Additionally, there appears to be a strong bilateral relationship between Brazil and Chile (maxlags of 9 and 4, respectively). Though Brazil officially practices a "managed floating" exchange rate, in practice its government intervenes heavily to control the exchange rate in an effort to protect its international competitiveness (Leahy, 2012). It is possible that the importance of Chile's exchange rates is the result of Brazil's interventionist response to global macroeconomic conditions that

affect both economies.

Figure 2.10: Plot of $\hat{L}^E$, denoting the estimated elementwise maxlag for each exchange rate series.

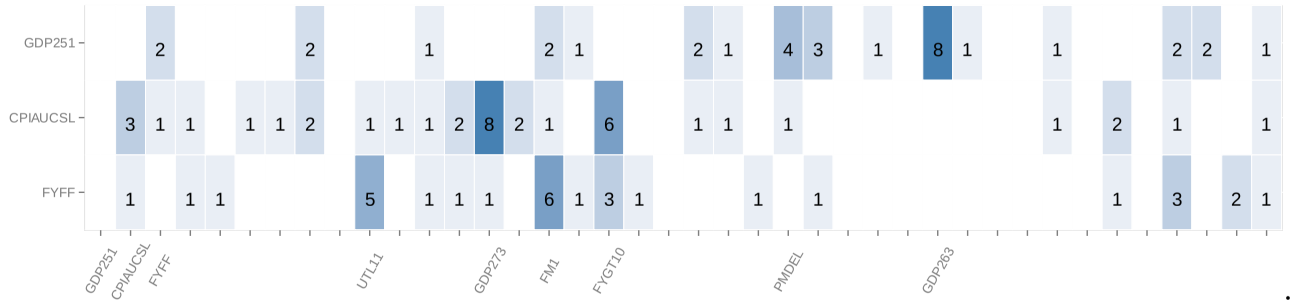


2.6 Discussion

By incorporating the property that more recent lags convey more information than distant lags, the hierarchical VAR approaches offer 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. Our methods scale well and are computationally feasible in high dimensions. Implementations of our methods are available in the R package `BigVAR`, which is hosted on the Comprehensive R Archive Network (cran).

Figure 2.11: The first three rows of $\hat{L}^E$, denoting the estimated elementwise maxlag for each series in the *Medium-Large* group using the $HVAR^E$ method. Components with maxlag of zero are left empty. The first component, Federal Funds Rate (FYFF), has been shown in Bernanke and Blinder (1992) to be an important predictor of several measures of economic activity, including the components of Gross Domestic Product. Additionally, the "Taylor Rule" (Taylor, 1993) suggests that the Federal Funds Rate is set to control inflation, hence we should expect changes in the previous quarter to aid in forecasting inflation.



CHAPTER 3

BIGVAR: TOOLS FOR MODELING SPARSE PENALIZED VECTOR AUTOREGRESSIONS

3.1 Introduction

For decades, the vector autoregression (VAR) and vector autoregression with unmodeled exogenous variables (VARX) have served as essential tools in forecasting multivariate time series. However, in the absence of regularization, the VAR and VARX are heavily overparameterized, often forcing practitioners to arbitrarily specify a reduced subset of series to model.

Recent years have witnessed tremendous developments toward the incorporation of regularization methods in the forecasting of high-dimensional multivariate time series with a particular interest in the lasso (Tibshirani, 1996) and its structured variants (the group lasso, Yuan and Lin (2006) and sparse group lasso, Simon et al. (2013)). All of these methods can be expressed as penalized least squares optimization problems which can be solved efficiently with iterative nonsmooth convex optimization algorithms, such as coordinate descent (Friedman et al., 2010) and generalized gradient descent (Beck and Teboulle, 2009).

Despite growing interest in the area, there has been relatively little progress in the development of software that allows for the modeling of sparse high-dimensional VARs and VARXs. Many authors, including Davis et al. (2012) and

Song and Bickel (2011) implement their penalized VAR models as modifications of the existing implementation `glmnet` (Friedman et al., 2009), a package that is not designed for time-dependent problems and offers limited multivariate and structured support.

Moreover, we have found a dearth of `R` packages that even allow for the estimation of a high-dimensional VAR or VARX by least squares. The `ar.ols` function in base `R` employs explicit matrix inversion, hence it is not tractable in high-dimensional settings and does not have VARX support. The `VAR` function in the package `vars` fits the VAR equation-by-equation via least squares using `lm`, which can cause complications under scenarios in which the number of covariates is close to or exceeds the length of the series, as such an implementation ignores degrees of freedom and can potentially can lead to numerically unstable results.

`BigVAR` adapts the aforementioned penalized regression solution algorithms from the regularization literature to a multivariate time series setting, allowing for the simultaneous forecasting of many potentially interrelated time series. If forecasts are only desired from a subset of included series, `BigVAR` utilizes the VARX-L framework (Nicholson et al., 2016a) to effectively leverage the information from unmodeled *exogenous* series to improve the forecasts of modeled *endogenous* series.

We additionally offer a class of Hierarchical Vector Autoregression (HVAR) procedures (Nicholson et al., 2016b) that address the notion of lag order by imposing a nested group lasso penalty in the VAR context. Finally, for comparison

purposes, we offer very fast and numerically stable implementations of information criterion based models which fit VAR and VARX models by least squares as the minimizer of either AIC or BIC.

Section 3.2 details our notation and provides an overview of the VARX-L and HVAR frameworks and Section 3.3 details the practical implementation of `BigVAR` with a macroeconomic data example. Section 3.4 provides an overview of several post-estimation refitting procedures as well as a simulation study, and Section 3.5 contains our conclusion. Our appendix elaborates upon our solution methods and algorithms.

3.2 Notation and Overview of `BigVAR` Procedures

Let $\{\mathbf{y_t}\}_{t=1}^T$ denote a k dimensional vector time series and $\{\mathbf{x}_t\}_{t=1}^T$ denote an m-dimensional unmodeled *exogenous* series. A vector autoregression with exogenous variables of order (p,s) , VARX$_{k,m}(p, s)$, can be expressed as

$$\mathbf{y}_t = \boldsymbol{\nu} + \sum_{\ell=1}^{p} \boldsymbol{\Phi}^{(\ell)} \mathbf{y}_{t-\ell} + \sum_{j=1}^{s} \boldsymbol{\beta}^{(j)} \mathbf{x}_{t-j} + \mathbf{u}_t \text{ for } t = 1, \ldots, T, \tag{3.1}$$

in which $\boldsymbol{\nu}$ denotes a $k \times 1$ intercept vector, each $\boldsymbol{\Phi}^{(\ell)}$ represents a $k \times k$ endogenous (modeled) coefficient matrix, each $\boldsymbol{\beta}^{(j)}$ represents a $k \times m$ exogenous (unmodeled) coefficient matrix, and $\mathbf{u}_t \overset{\text{wn}}{\sim} (\mathbf{0}, \boldsymbol{\Sigma}_u)$. Note the the VAR is a special case of Equation (3.1) in which the second summation ($\sum_{j=1}^{s} \boldsymbol{\beta}^{(j)} \mathbf{x}_{t-j}$) is not included.

3.2.1 The VARX-L Framework

To reduce the parameter space of the VARX, the VARX-L framework applies structured convex penalties to the least squares VARX problem, resulting in the objective

$$\min_{\boldsymbol{\nu},\boldsymbol{\Phi},\boldsymbol{\beta}} \sum_{t=1}^{T} \|\mathbf{y}_t - \boldsymbol{\nu} - \sum_{\ell=1}^{p} \boldsymbol{\Phi}^{(\ell)}\mathbf{y}_{t-\ell} - \sum_{j=1}^{s} \boldsymbol{\beta}^{(j)}\mathbf{x}_{t-j}\|_F^2 + \lambda\Big(\mathcal{P}_y(\boldsymbol{\Phi}) + \mathcal{P}_x(\boldsymbol{\beta})\Big), \tag{3.2}$$

in which $\|A\|_F$ denotes the Frobenius norm of matrix A (i.e. the elementwise 2-norm), $\boldsymbol{\Phi} = [\boldsymbol{\Phi}^{(1)}, \ldots, \boldsymbol{\Phi}^{(p)}]$, $\boldsymbol{\beta} = [\boldsymbol{\beta}^{(1)}, \ldots, \boldsymbol{\beta}^{(s)}]$, $\lambda \geq 0$ is a penalty parameter estimated by sequential cross-validation, $\mathcal{P}_y(\boldsymbol{\Phi})$ represents the group penalty structure on endogenous coefficients, and $\mathcal{P}_x(\boldsymbol{\beta})$ represents the group penalty structure on exogenous coefficients.

These penalties impose structured sparsity based upon a partition of the parameter space that takes into account the intrinsic structure of the VARX. All VARX-L penalty structures are detailed in Table 3.1. Observe that groups are weighted by their cardinality to prevent regularization favoring larger groups. Plots of example sparsity patterns (with nonzero, or *active* coefficients shaded) are depicted in Figure 3.1. In the following sections, we will describe each penalty structure in more detail.

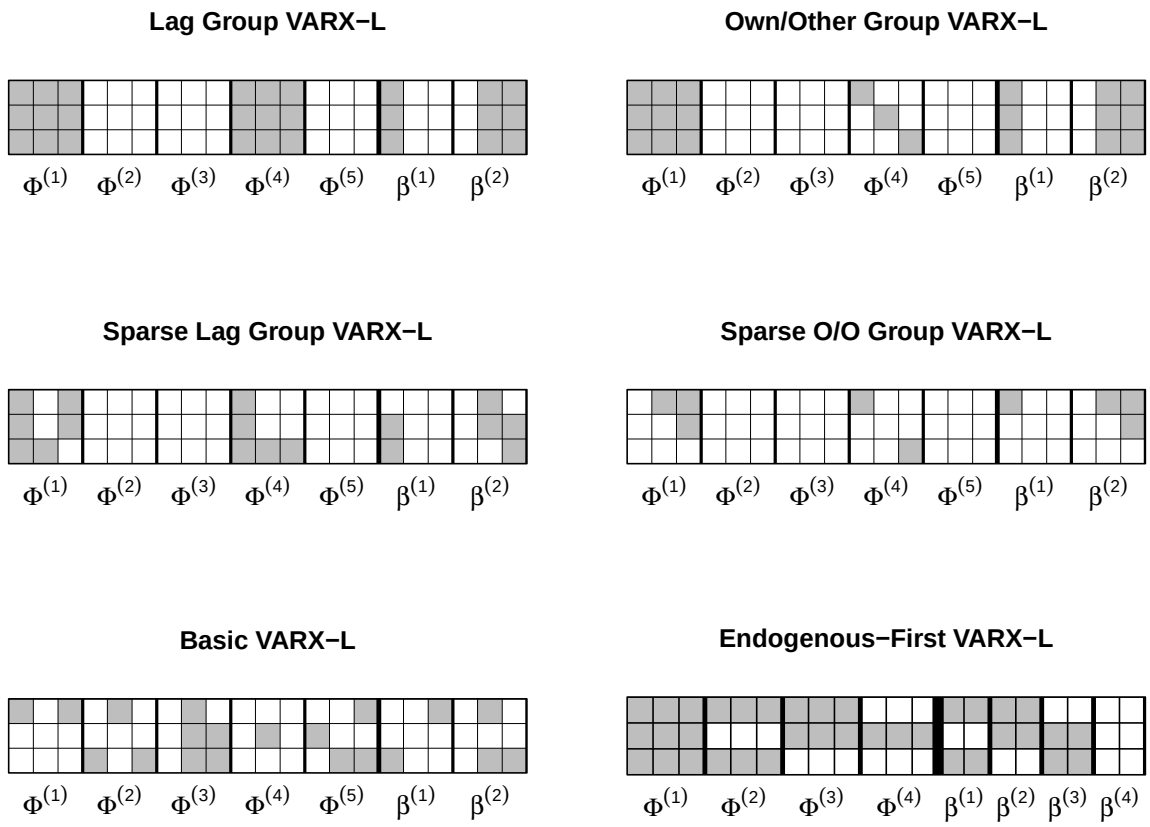


Figure 3.1: Examples of VARX-L Sparsity Patterns (k=3, p=5; m=2, s=3). The gray shading denotes nonzero 'active' coefficients whereas white denotes coefficients that have been set to zero.

Group Lasso Penalties

The group lasso (Yuan and Lin, 2006) has emerged as a popular penalized regression procedure that partitions all model coefficients into a collection of disjoint groups that can take into account the inherent structure of a multivariate time series. Within a group, all coefficients will either be set to zero or the group will be *active* and all coefficients will be nonzero. We consider two group structures

Table 3.1: VARX-L Penalty Functions (Reproduced from (Nicholson et al., 2016a)). Note that $\mathbf{\Phi}_{\text{on}}^{(\ell)}$ and $\mathbf{\Phi}_{\text{off}}^{(\ell)}$ denote the diagonal and off-diagonal elements of coefficient matrix $\mathbf{\Phi}^{(\ell)}$, respectively.

Group Name	$\mathcal{P}_y(\mathbf{\Phi})$	$\mathcal{P}_x(\boldsymbol{\beta})$
(3.3) Lag	$\sqrt{k^2}\sum_{\ell=1}^{p}\|\|\mathbf{\Phi}^{(\ell)}\|\|_F$	$\sqrt{k}\sum_{j=1}^{s}\sum_{i=1}^{m}\|\|\boldsymbol{\beta}_{\cdot,i}^{(j)}\|\|_F$
(3.4) Own/Other	$\sqrt{k}\sum_{\ell=1}^{p}\|\|\mathbf{\Phi}_{\text{on}}^{(\ell)}\|\|_F + \sqrt{k(k-1)}\sum_{\ell=1}^{p}\|\|\mathbf{\Phi}_{\text{off}}^{(\ell)}\|\|_F$	$\sqrt{k}\sum_{j=1}^{s}\sum_{i=1}^{m}\|\|\boldsymbol{\beta}_{\cdot,i}^{(j)}\|\|_F$
(3.5) Sparse Lag	$(1-\alpha)\sqrt{k^2}\sum_{\ell=1}^{p}\|\|\mathbf{\Phi}^{(\ell)}\|\|_F + \alpha\|\|\mathbf{\Phi}\|\|_1$	$(1-\alpha)\sqrt{k}\sum_{j=1}^{s}\sum_{i=1}^{m}\|\|\boldsymbol{\beta}_{\cdot,i}^{(j)}\|\|_F + \alpha\|\|\boldsymbol{\beta}\|\|_1$
(3.6) Sparse Own/Other	$(1-\alpha)(\sqrt{k}\sum_{\ell=1}^{p}\|\|\mathbf{\Phi}_{\text{on}}^{(\ell)}\|\|_F + \sqrt{k(k-1)}\sum_{\ell=1}^{p}\|\|\mathbf{\Phi}_{\text{off}}^{(\ell)}\|\|_F) + \alpha\|\|\mathbf{\Phi}\|\|_1$	$(1-\alpha)\sqrt{k}\sum_{j=1}^{s}\sum_{i=1}^{m}\|\|\boldsymbol{\beta}_{\cdot,i}^{(j)}\|\|_F + \alpha\|\|\boldsymbol{\beta}\|\|_1$
(3.7) Basic	$\|\|\mathbf{\Phi}\|\|_1$	$\|\|\boldsymbol{\beta}\|\|_1$
(3.8) Endogenous-First	$\mathcal{P}_{y,x}(\mathbf{\Phi},\boldsymbol{\beta}) = \sum_{\ell=1}^{p}\sum_{j=1}^{k}\left(\|\|[\mathbf{\Phi}_{j,\cdot}^{(\ell)}, \boldsymbol{\beta}_{j,\cdot}^{(\ell)}]\|\|_F + \|\|\boldsymbol{\beta}_{j,\cdot}^{(\ell)}\|\|_F\right)$	

for the endogenous covariates: a lag based grouping (*Lag* Group VARX-L, expression 3.3 in Table 3.1) and a grouping that distinguishes between a series' *own* lags (diagonal entries of $\mathbf{\Phi}^{(\ell)}$) and those of other series (off diagonal entries of $\mathbf{\Phi}^{(\ell)}$) (*Own/Other* Group VARX-L, expression 3.4). The Own/Other grouping incorporates the widely held stylized fact in macroeconometrics that a series' own lags have different dynamic dependence than those from other series (Litterman, 1979).

Though both penalties employ the same solution algorithm, since the partitioning under the Lag Group VARX-L forms proper submatrices, it is possible to directly solve the matrix optimization problem as opposed to performing a least squares transformation, resulting in substantially less computational overhead than the Own/Other scenario.

Both the Own/Other and Lag Group VARX-L partition exogenous coefficients

by column. Our experiences have found that assigning each exogenous covariate to its own group substantially increases computation time without an improvement in forecast performance and an exogenous lag-based grouping is too general. Hence, the column-based grouping serves as a compromise; allowing for a degree of flexibility while still resulting in a computationally efficient optimization problem.

Sparse Group Lasso Penalties

In certain scenarios, a group penalty can be too restrictive. If a group is active, all of its coefficients are potentially nonzero. On the other hand, specifying a large number of groups will substantially increase computation time and, in our experience, generally does not lead to improvements in forecasting performance.

As a compromise, we consider applying *sparse group lasso* penalties (expressions 3.5 and 3.6 in Table 3.1) proposed by Simon et al. (2013), which allow for "within-group" sparsity via a convex combination of L_1(unstructured sparsity) and L_2 (structured sparsity) penalties. `BigVAR` offers the Sparse VARX-L for both the Lag and Own/Other structured groupings.

By default α, the parameter that sets the weights of the two penalties and is constrained to be between 0 and 1, is chosen to according to a heuristic ($\frac{1}{k+1}$) to control within-group sparsity. `BigVAR` also permits for the joint cross validation of λ and α. Performing joint cross validation allows for the Sparse Group VARX-

L to function as a powerful diagnostic tool to determine the applicability of a structured grouping. A selected value of close to zero provides strong evidence of structured sparsity whereas a value close to one points to a lack of structure.

Basic Penalty

The Basic VARX-L (expression 3.7 in Table 3.1) is the most general grouping and can be viewed as partitioning each variable into its own group or as applying an unstructured lasso penalty to the entire VARX coefficient matrix. It does not incorporate any of the structure of the VARX, but it results in a comparably simpler optimization problem, allowing it to scale to much larger problems than structured penalties.

Nested Penalty Structures

The previous penalty structures are disjoint groupings that partition $[\boldsymbol{\Phi}, \boldsymbol{\beta}]$. In certain scenarios, one might wish to assign a preference to endogenous versus exogenous variables. The *Endogenous-First* VARX-L (expression 3.8 in Table 3.1) utilizes a nested penalty to prioritize endogenous series. At a given lag, an exogenous series can enter the model only if their endogenous counterpart is nonzero. Note that by construction this penalty decouples across series, allowing for endogenous/exogenous dependence to vary. It is additionally required that $p \geq s$, otherwise such a nested penalty structure would not be appropriate.

Solution Methods

In order to solve the optimization problems in the form of Equation 3.2, we employ computationally tractable algorithms designed for non-smooth convex functions. Our solution methods do not make calls to external packages or commercial convex solvers and are optimized for time dependent problems. All of our solution algorithms are coded in `C++` and linked to `R` via `Rcpp` (Eddelbuettel and François, 2011), `RcppArmadillo` (Eddelbuettel and Sanderson, 2014), and `RcppEigen` (Bates et al., 2012). The specific algorithms that we utilize for each procedure are displayed in Table C.2 in Section C.0.17 of the appendix. Implementation details are provided in the appendix of Nicholson et al. (2016a).

3.2.2 Hierarchical Vector Autoregression (HVAR)

The VARX-L procedures remain agnostic with regard to lag order selection. Hence, as the maximum lag order increases forecast performance may start to degrade, as each group is treated democratically despite more distant lags generally tending to be less useful in forecasting. Within the VAR context, we utilize the HVAR class of models (Nicholson et al., 2016b) which alleviate this issue by embedding lag order into *hierarchical group lasso* penalties.

In addition to returning sparse solutions, our HVAR$_k(p)$ procedures induce regularization toward models with low maximum lag order. To allow for greater

flexibility, instead of imposing a single, universal lag order (as information criterion minimization based approaches tend to do), we allow it to vary across marginal models (i.e. the rows of the coefficient matrix $\mathbf{\Phi} = [\mathbf{\Phi}^{(1)}, \ldots, \mathbf{\Phi}^{(p)}]$). `BigVAR` includes three HVAR models as well as the "Lag-weighted Lasso," which incorporates a lasso penalty that increases geometrically as the lag order increases. These procedures are presented in Table 3.2 and example sparsity patterns of the HVAR procedures and the Lag-weighted Lasso are depicted in Figure 3.2.

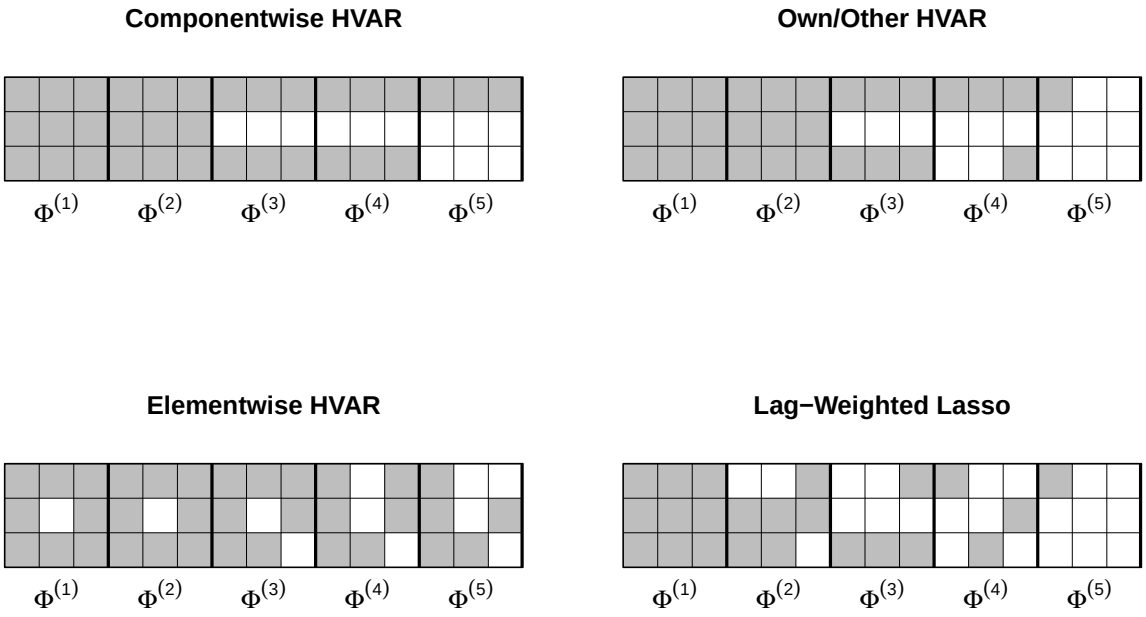


Figure 3.2: Examples of Sparsity Patterns for the HVAR procedures and the Lag-Weighted Lasso (k=3,p=5)

Table 3.2: HVAR Penalty Functions .

Group Name	$\mathcal{P}_y(\mathbf{\Phi})$
(3.9) Componentwise	$\sum_{i=1}^{k}\sum_{\ell=1}^{p}\|\|\mathbf{\Phi}_i^{(\ell:p)}\|\|_2$.
(3.10) Own/Other	$\sum_{i=1}^{k}\sum_{\ell=1}^{p}\left[\|\|\mathbf{\Phi}_i^{(\ell:p)}\|\|_2 + \|\|(\mathbf{\Phi}_{i,-i}^{(\ell)}, \mathbf{\Phi}_i^{([\ell+1]:p)})\|\|_2\right]$
(3.11) Elementwise	$\sum_{i=1}^{k}\sum_{j=1}^{k}\sum_{\ell=1}^{p}\|\|\mathbf{\Phi}_{ij}^{(\ell:p)}\|\|_2$
(3.12) Lag-weighted Lasso	$\sum_{\ell=1}^{p}\ell^{\gamma}\|\|\mathbf{\Phi}^{(\ell)}\|\|_1$

Componentwise HVAR

The Componentwise HVAR (defined in expression (3.9) in Table 3.2), allows for the maximum lag order to vary across marginal models, but within a series all components have the same maximum lag. This structure allows for k potentially different lag orders,

Own/Other HVAR

The Own/Other HVAR (defined in expression (3.10) in Table 3.2), is similar to the Componentwise HVAR, but imposes an additional layer of hierarchy within a lag: prioritizing coefficients of lagged values of the series of forecasting interest (i.e. "own" lags) over those of other series. This penalty incorporates a common specification in the Bayesian VAR with a Minnesota Prior (Litterman, 1979) that "own" lags are more informative for forecasting purposes than "other" lags,

Elementwise HVAR

The Elementwise HVAR (defined in expression (3.11) in Table 3.2) is the most general structure; in each marginal model, each series may have its own maximum lag. Under this framework, there are k^2 possible lag orders

Lag-weighted Lasso

In addition, for comparison purposes we provide a *Lag-weighted Lasso* (expression (3.12) in Table 3.2), which consists of a lasso penalty that increases geometrically with lag; $\gamma \in [0, 1]$ is an additional penalty parameter that is jointly estimated with λ according to sequential cross validation. This is similar to the approach proposed by Song and Bickel (2011). Though it encourages greater regularization at more distant lags, it does not explicitly force sparsity and requires the specification of an arbitrary functional form as well as an additional penalty parameter.

3.2.3 Penalty Parameter Selection

In order to account for time dependence, selection of the penalty parameter λ is conducted in a rolling manner. The penalty parameter, $\hat{\lambda}$, is selected from a grid of values $\lambda_1, \ldots, \lambda_n$. We perform sequential cross validation between times $T_1 - h + 1$ and $T_2 - h + 1$, in which h denotes forecast horizon. At $T_1 - h + 1$, we forecast $\hat{\mathbf{y}}_{T_1+h}^{\lambda_i}$ for $i = 1, \ldots, n$, and sequentially add observations until time $T_2 - h + 1$. $T_2 - h + 2$

through $T - h + 1$ is used for out of sample forecast evaluation.

Unless otherwise specified, `BigVAR` sets $T_1 = \left\lfloor \frac{T}{3} \right\rfloor, T_2 = \left\lfloor \frac{2T}{3} \right\rfloor$. We choose $\hat{\lambda}$ as the minimizer of h-step ahead MSFE:

$$MSFE(\lambda_i) = \frac{1}{(T_2 - T_1 - h + 1)} \sum_{t=T_1-h+1}^{T_2-h} \|\hat{\mathbf{y}}_{t+h|t}^{\lambda_i} - \mathbf{y}_{t+h}\|_2^2,$$

In the VAR context, there are two possible methods to obtain multi-step ahead forecasts: iterated one-step ahead predictions or directly forecasting the longer horizon. Per Clark and McCracken (2013), in the VAR context, iterated h-step ahead forecasts have the form:

$$\hat{\mathbf{y}}_{t+h|t} = \hat{\nu} + \sum_{\ell=1}^{p} \widehat{\mathbf{\Phi}}^{(\ell)} \hat{\mathbf{y}}_{t+h-i|t},$$

whereas the alternative involves directly forecasting h-step ahead forecasts

$$\hat{\mathbf{y}}_{t+h} = \hat{\nu} + \sum_{\ell=1}^{p} \widehat{\mathbf{\Phi}}^{(\ell)} \mathbf{y}_{t+1-i}.$$

Both approaches have advantages; as noted by Marcellino et al. (2006), the direct approach could provide more accurate forecasts if the VAR is misspecified, however, if the model is correctly specified, the iterated approach is theoretically more efficient. In the VAR setting, `BigVAR` allows for the choice of either iterated or direct forecasts when optimizing over forecasts horizons greater than one. In the VARX setting only direct forecasts are available, since we do not return forecasts of exogenous series.

If the user wishes to employ their own penalty parameter selection routine, they can do so by calling `BigVAR.est` within their code. This procedure will be discussed in Section 3.3.3.

3.3 Forecasting VAR(X) models with `BigVAR`

In this section, we demonstrate how to utilize `BigVAR` to forecast a set of quarterly macroeconomic indicators procured from the St. Louis Federal Reserve Economic Database (FRED) via Quandl. We consider forecasting four US macroeconomic series:

(i) Consumer Price Index (CPI),

(ii) Federal Funds Rate (FFR),

(iii) Gross Domestic Product (GDP),

(iv) M1 (a measure of the liquid components of the money supply).

We first download the data using the API provided in the `Quandl` package and then transform each series to stationarity by taking the log difference of CPI, M1, and GDP and the log of FFR (since it is already expressed as a rate).

The GDP and CPI series start in Quarter 1 of 1947, but since the Federal Funds Rate was not officially published until 1954 and M1 was not recorded until 1959, we discard all realizations of GDP and CPI before Quarter 3 of 1959. The data

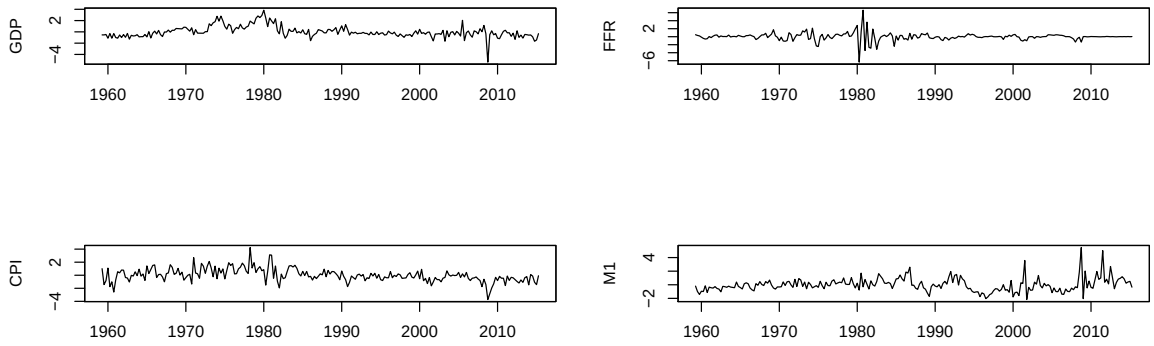


Figure 3.3: Plots of Standardized Quarterly GDP, Federal Funds Rate, CPI, and M1

ranges through Quarter 2 of 2015, resulting in $T = 224$. As is standard in the regularization framework, before estimation we standardize each series to have zero mean and unit variance.

3.3.1 Constructing an object of class `BigVAR`

In an effort to streamline functionality, `BigVAR` incorporates `R`'s *s*4 object class system. In order to fit a model, the user constructs an object of class `BigVAR` that contains the data as well as model specifications. A `BigVAR` object can be created with the wrapper function `constructModel`, which encompasses both the HVAR and VARX-L frameworks.

In examining Figure 3.3, we observe considerable fluctuations in the CPI and FFR series in the early 1980s, owing to the period's rapid inflation and the resulting contractionary monetary policy. Here, we choose to minimize the influence of

this period when selecting our penalty parameters. Below, we construct an Elementwise $\text{HVAR}_4(4)$ and use data from Quarter 1 of 1985 to Quarter 1 of 2005 for penalty parameter selection.

```
library(BigVAR)


T1 <- which(index(Y)=="1985 Q1")
T2 <- which(index(Y)=="2005 Q1")


Model1=constructModel(as.matrix(Y),p=4,
    struct="HVARELEM",gran=c(25,10),verbose=FALSE,
VARX=list(),T1=T1,T2=T2)
```

The required arguments for `constructModel` are:

- `Y`: a $T \times k$ multivariate time series (in matrix form),
- `p`: predetermined maximum lag order,
- `gran`: two arguments that characterize the grid of penalty parameters: the first denotes the depth of the grid and the second the number of candidate penalty parameters.

The choices for the argument `struct` are presented in Table C.1 in Section C.0.17 in the appendix. In the `BigVAR` framework, `gran` denotes the only "hyperparameter" that must be set by the end user. Following Friedman et al. (2010),

the grid of penalty values starts with the smallest value in which all coefficients will be zero, then decrements in log linear increments. The grid ends at a fraction of this maximum value (as dictated by the first argument in `gran`). These bounds are detailed in the appendix of Nicholson et al. (2016a).

In practice these bounds can be coarse. Consequently, to avoid scenarios in which several candidate penalty parameters return coefficient matrices of identically zero, `BigVAR` utilizes an empirical procedure to determine tighter bounds. In order to do so, we expand upon the approach presented in Algorithm 3 of Lou et al. (2014b). Starting with the theoretically determined bound, we employ a bisection routine in order to find a tighter data-driven bound. Our implementation of this procedure is detailed in Algorithm 9 in Section C.0.17 in the appendix. In practice, we find that the best choices for grid depth tend to be between 10 and 50, depending on the number of series included and the forecast horizon.

The number of penalty parameters is also left to user input. The package `glmnet` calls for 100 penalty parameters by default. However, in our applications we have found no substantial forecasting improvement in considering any more than 10. If the user wishes to provide their own penalty parameters, they can do so through `gran`, but they must also set the optional argument `ownlambdas` to `TRUE`. The additional optional arguments to `constructModel` and their default values are:

- `RVAR`: Relaxed VAR(X) indicator to refit based upon the coefficients recovered from a VARX-L or HVAR procedure according to least squares (default:

FALSE). This method will be discussed in greater detail in Section 3.4.

- MN: option for the *Minnesota VAR(X)*, which shrinks parameter estimates toward a vector random walk (default FALSE).
- h: forecast horizon (default 1).
- verbose: indicator for progress bar (default TRUE).
- IC: indicator to return AIC and BIC benchmarks (default TRUE).
- VARX: list of VARX specifications (default list()).
- T1: start of cross validation period (default $\lfloor \frac{T}{3} \rfloor$).
- T2: start of forecast evaluation period (default $\lfloor \frac{2T}{3} \rfloor$).
- ONESE: indicator for *One Standard Error* heuristic described in Hastie et al. (2009) which selects the largest penalty parameter within one standard error of the minimizer of MSFE (default FALSE).
- recursive: indicator determining if recursive multi-step predictions are desired as opposed to direct (default FALSE, applicable only for VAR models with $h > 1$).
- alpha: vector of candidate values for α if dual cross validation is desired for the Sparse Lag or Sparse Own/Other structured penalties (all entries must be between 0 and 1, the default value is $\frac{1}{k+1}$).
- C: vector denoting series to be shrunk toward a random walk instead of toward zero (used in situations in which some series exhibit signs of non-stationarity, while others don't). This scenario will be discussed in greater detail in Section 3.3.3, (default $\mathbf{0}_k$, applicable only if MN is TRUE).

3.3.2 Implementation

In order to fit a model with `BigVAR` using rolling cross validation, we simply need to execute the method `cv.BigVAR` on an object of class `BigVAR` as detailed below. To fit an Elementwise $\text{HVAR}_4(4)$, we simply run the command

```
Model1Results = cv.BigVAR(Model1)
```

An object of class `BigVAR.Results` is returned. By default, the output displays model characteristics, such as the penalty structure, maximum lag order, the value of λ selected by rolling cross validation, and both in-sample and out-of-sample MSFE. For comparison purposes, the out-of-sample MSFE from several benchmarks, including the sample mean, random walk, and the least squares VAR or VARX with lags selected by AIC and BIC are also returned.

```
Model1Results


## *** BigVAR MODEL Results ***
## Structure
## [1] "HVARELEM"
## Forecast Horizon
## [1] 1
## Minnesota VAR
## [1] FALSE
```

```
## Maximum Lag Order
## [1] 4
## Optimal Lambda
## [1] 6.8498
## Grid Depth
## [1] 25
## Index of Optimal Lambda
## [1] 9
## In-Sample MSFE
## [1] 1.843
## BigVAR Out of Sample MSFE
## [1] 4.771
## *** Benchmark Results ***
## Conditional Mean Out of Sample MSFE
## [1] 5.372
## AIC Out of Sample MSFE
## [1] 5.123
## BIC Out of Sample MSFE
## [1] 5.428
## RW Out of Sample MSFE
## [1] 6.868
```

`Model1Results` also includes

```
# Coefficient matrix at end of evaluation period
Model1Results@betaPred
# Residuals at end of evaluation period
Model1Results@resids
# Lagged Values at end of evaluation period
Model1Results@Zvals
```

3.3.3 Diagnostics and Additional Features

This section details the features of BigVAR that both tailor to the specific forecasting scenarios of the end-user and ensure that the most accurate possible forecasts are delivered.

The "Minnesota" Lasso

As opposed to shrinking every coefficient toward zero, all of the procedures in `BigVAR` can be modified to instead shrink toward a vector random walk (i.e. $\mathbf{\Phi}^{(1)} = I_k$, all other coefficient matrices are still shrunk toward zero). Such a modification is akin to the Bayesian VAR with Minnesota Prior of Litterman (1979). This approach can be useful in scenarios exhibiting evidence of unit-root nonstationarity, which is commonplace in macroeconomic data. For more details about this approach, see Section 4 of Nicholson et al. (2016a).

`BigVAR` also allows for the option of shrinking some series toward zero while shrinking others toward a random walk. This can be of use in applications, such as that presented in Banbura et al. (2009), in which a large cross section of series are examined; most are roughly stationary, but a few exhibit a substantial degree of persistence.

In examining Figure 3.3, we observe substantial persistence in the M1 series while the other series appear stationary. We could attempt to shrink M1 toward a random walk while shrinking the others toward zero. However, as can be observed below, doing so degrades forecast performance.

```
Model1MN <- constructModel(as.matrix(Y), 4, "HVARELEM",
c(25, 10),
T1 = T1, T2 = T2, verbose = FALSE, MN = TRUE,
C = c(0, 0, 0, 1))
Model1MNresults <- cv.BigVAR(Model1MN)
mean(Model1MNresults@OOSMSFE)


## [1] 4.806
```

Evaluating a Choice of Structure

If the practitioner is unsure as to the choice of a VARX-L structure, one potential selection approach involves fitting a Sparse Lag or Sparse Own/Other VARX-L

with both λ and α selected by sequential cross validation. The selected choice of α should provide some insight as to the importance of structure in the data.

```
# Construct grid of candidate alphas between zero and 1
alpha <- seq(0, 1, length = 10)
Model2 = constructModel(as.matrix(Y), p = 4,
struct = "SparseLag", gran = c(25,
    10), verbose = FALSE,
VARX = list(), alpha = alpha, T1 = T1, T2 = T2)
SparseLagDiag <- cv.BigVAR(Model2)
# Selected value of alpha
SparseLagDiag@alpha
```

```
## [1] 0.2222222
```

```
# Resulting out of sample MSFE
mean(SparseLagDiag@OOSMSFE)
```

```
## [1] 4.713579
```

```
Model3 = constructModel(as.matrix(Y), p = 4,
struct = "SparseOO", gran = c(25,
    10), verbose = FALSE,
VARX = list(), alpha = alpha, T1 = T1, T2 = T2)
```

```
SparseOODiag <- cv.BigVAR(Model3)
# Selected value for alpha
SparseOODiag@alpha


## [1] 0.3333333


# Resulting out of sample MSFE
mean(SparseOODiag@OOSMSFE)


## [1] 4.674584
```

We observe that in the Sparse Lag setting, the selected value of $\alpha \approx 0.22$ is very close to our heuristic ($\frac{1}{k+1} = 0.2$), indicating that the level of within-group sparsity determined by our heuristic is appropriate for this application. In the Sparse Own/Other setting, the selected value is ≈ 0.33, indicating that a slightly greater degree of within-group sparsity than that imposed by the heuristic may be appropriate.

Penalty Grid Position

The `plot` method of a `BigVAR.results` object visualizes the position of $\hat{\lambda}$ over the grid of candidate values. Figure 3.4 plots the in-sample MSFE for each value of λ over the training period with the minimum value highlighted. It is desirable

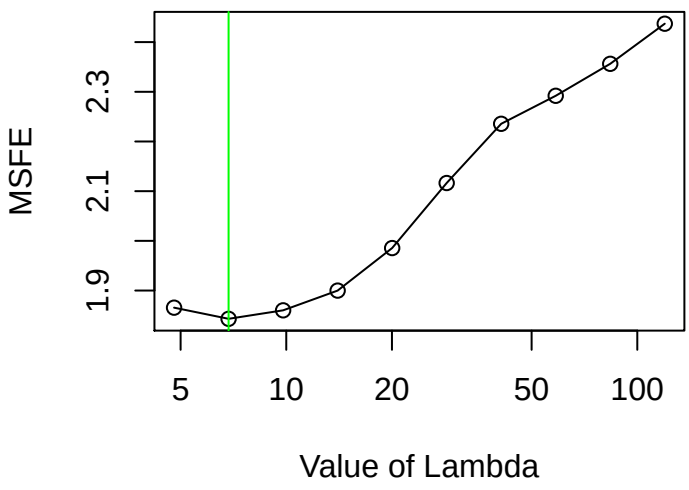


Figure 3.4: In-sample MSFE for each candidate penalty parameter.

for $\hat{\lambda}$ to be near the middle of the grid; if it is at the lower boundary, increasing the depth of the grid may lead to improved forecasting performance. In examining Figure 3.4, we see that $\hat{\lambda}$ is not at the lower boundary of the penalty grid. If it were, we could simply construct a deeper penalty grid by increasing the first parameter of the "gran" argument in `ConstructModel`.

```
plot(Model1Results)
```

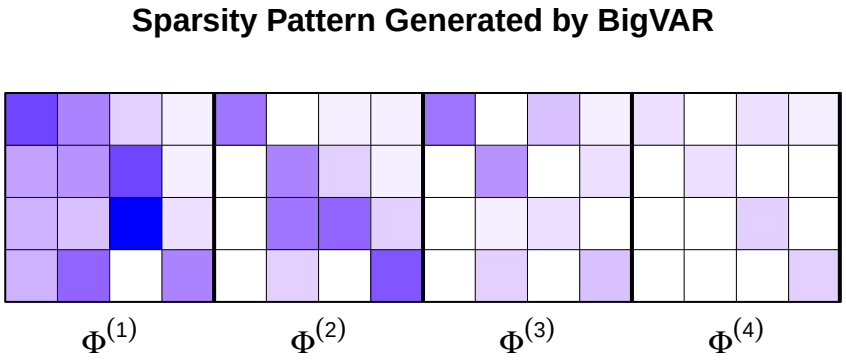


Figure 3.5: Sparsity plot generated by the Elementwise HVAR with active elements shaded. Darker coefficients are larger in magnitude.

Visualizing Sparsity Patterns

The method `SparsityPlot.BigVAR.results` allows for the ability to view the sparsity pattern of the final estimated coefficient matrix $[\mathbf{\Phi}, \boldsymbol{\beta}]$ in the out of sample forecast evaluation period, which fits a model with the selected penalty parameter using all available data. Figure 3.5 depicts this sparsity pattern for the Elementwise $\text{HVAR}_4(4)$ example. Darker shading indicates coefficients that are larger in magnitude. We observe that coefficients on the diagonal are larger in magnitude, indicating that "own" lags are relatively more important in forecasting than those of "other" series even though we did not explicitly consider an Own/Other structured grouping.

```
SparsityPlot.BigVAR.results(Model1Results)
```

VARX-L Estimation

If the user wishes to fit a VARX-L model, first the series should be arranged in a $T \times (k+m)$ matrix or such that the first k columns are endogenous (modeled) series and the remaining m are exogenous (unmodeled) series.

After doing so, a list of VARX specifications needs to be passed to `constructModel`. The list must contain two elements: k denotes the number of endogenous series and s the maximum lag order for the exogenous series. For example, if we want to forecast GDP and the Federal Funds Rate using CPI and M1 as exogenous series (with $s = 4$), we simply need to specify:

```
VARX = list()
VARX$k = 2   # 2 endogenous series
VARX$s = 4   # maximum lag order of 4 for exogenous series
Model2 <- constructModel(as.matrix(Y),
4, "SparseLag", gran = c(50, 10), VARX = VARX, verbose = FALSE)
Model2Results = cv.BigVAR(Model2)
```

N-step Ahead Out-of-sample Predictions

Out of sample predictions can be obtained via the `predict` method. Multi-step ahead VAR forecasts are computed recursively using standard methods described in chapter 2 of Lütkepohl (2005). While n-step ahead predictions are available for

VAR models, we currently only allow 1-step ahead predictions for VARX models unless new data is provided.

```
# One-step ahead VAR forecasts
predict(Model1Results, 1)


##              [,1]
## [1,] -0.77288097
## [2,] -0.06281180
## [3,] -0.34507176
## [4,]  0.06941941


# Multi-step VARX prediction with new data
VARXExample <- constructModel(as.matrix(Y), 4, "Basic",
gran = c(50, 10), VARX = list(k = 2,
    s = 4), verbose = FALSE)
result <- cv.BigVAR(VARXExample)
# Holdout data
holdout


##                CPI        FFR        GDP            M1
## 2015 Q3 -1.092013 0.01706719 -0.7863598  0.009156325
## 2015 Q4 -0.842867 0.09126610 -1.0337862 -0.257181315
## 2016 Q1 -1.187883 0.10775474 -1.2655224  0.370885724
```

```
predict(result, n.ahead = 3
, newxreg = matrix(holdout[, 3:4], ncol = 2))


##             [,1]          [,2]
## [1,] -0.2123561 -0.003871842
```

Estimation with Fixed λ

A user may wish to initially estimate $\hat{\lambda}$ by rolling cross-validation and continue to use that value as new data becomes available or potentially apply their own penalty parameter selection technique. In such a scenario, it would not be desirable to fit a model with `cv.BigVAR`.

We provide an alternative function `BigVAR.est`, which requires an object of class `BigVAR` as input and fits a VARX-L or HVAR model using all available data for either a fixed grid of λ values or a grid determined by the data (as is done in `cv.BigVAR`). For example, suppose that wish to re-estimate our Elementwise $\text{HVAR}_4(4)$ model with newly available data using the $\hat{\lambda}$ that was selected in Section 3.3.2.

```
# new data
holdout
# augment data in original BigVAR object
```

```
Model1@Data <- as.matrix(rbind(Y, holdout))
# Extract the optimal lambda from our BigVAR results object
lambda <- Model1Results@OptimalLambda
# Set ownlambas indicator TRUE in BigVAR object
Model1@ownlambdas = TRUE
# Replace granularity specs with choice of lambda
Model1@Granularity <- lambda
BigVAR.est(Model1)
# returns a list containing:
a k x (kp+ms+1) x n array,
in which n denotes
# the number of penalty parameters
and a vector of penalty parameters
# corresponding to each slice of the array
```

Simulating multivariate time series

When developing new methods, it is often good practice to evaluate their performance on simulated data. `BigVAR` offers the ability to simulate realizations from user-provided VAR coefficient and covariance matrices via the function `MultVARSim`. In order to simulate from a $\text{VAR}_k(p)$, we convert its coefficient ma-

trix to block companion form (following equation 2.1.8 in Lütkepohl (2005)):

$$\mathbf{A} = \begin{bmatrix} \mathbf{\Phi}^{(1)} & \mathbf{\Phi}^{(2)} & \dots & \mathbf{\Phi}^{(p-1)} & \mathbf{\Phi}^{(p)} \\ \boldsymbol{I}_k & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{I}_k & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \boldsymbol{I}_k & \mathbf{0} \end{bmatrix} \tag{3.13}$$

An example is shown below.

```
# included VAR_3(3) coefficient matrix
in BigVAR in block companion form
data(Generator)
k <- 3
A[1:k,]


##        [,1]  [,2] [,3]  [,4]  [,5]  [,6]  [,7]  [,8] [,9]
## [1,] -0.29  0.00  0.0 -0.62  0.00  0.00 -0.49  0.00 0.00
## [2,] -0.26 -0.20  0.0 -0.77 -0.36  0.00 -1.24 -0.07 0.00
## [3,] -0.66  0.75  1.3  0.30 -0.40 -0.44  0.36  0.05 0.03


SigmaU <- .01*diag(k) #Scaled identity covariance
YSim <- MultVARSim(k,A,3,SigmaU,T=100)
```

When constructing a coefficient matrix, one needs to be judicious in ensuring

stationarity. Stationarity requires that the all eigenvalues of $\mathbf{A}$ have modulus less than 1. As stated in Roy et al. (2014), there is generally no link between the magnitude of elements in a coefficient matrix and stationarity. For example, consider the case where $k = 2$ and $p = 1$. The $VAR_2(1)$ coefficient matrix

$$\mathbf{\Phi} = \begin{bmatrix} 0 & 0 \\ \epsilon & 0 \end{bmatrix} \tag{3.14}$$

is stationary for any value of ϵ.

Recent developments by Boshnakov and Iqelan (2009) provide a framework to guarantee stationary VAR coefficient matrices, but their method cannot impose structured sparsity, which limits its utility in evaluating the performance of the VARX-L and HVAR class of models.

3.3.4 Structural Macroeconomic Analysis

Though `BigVAR` is primarily designed to forecast high-dimensional time series, it can also be of use in analyzing the joint dynamics of a group of interrelated time series. In order to conduct policy analysis, many macroeconomists make use of VARs to examine the impact of shocks to certain variables on the entire system (holding all other variables fixed). This is know as impulse response analysis. It has the potential to be very important in a high-dimensional setting as omitting variables from a system can lead to major distortions (Lin, 2006).

For example, a macroeconomist may wish to analyze the impact of a 100 basis

point increase in the Federal Funds Rate on all included series over the next 8 quarters. To do so, we can utilize the function `generateIRF`, which converts the last estimated coefficient matrix to fundamental form (for details, see Section C.0.12 in the appendix). The impulse responses generated from this "shock" are depicted in Figure 3.6.

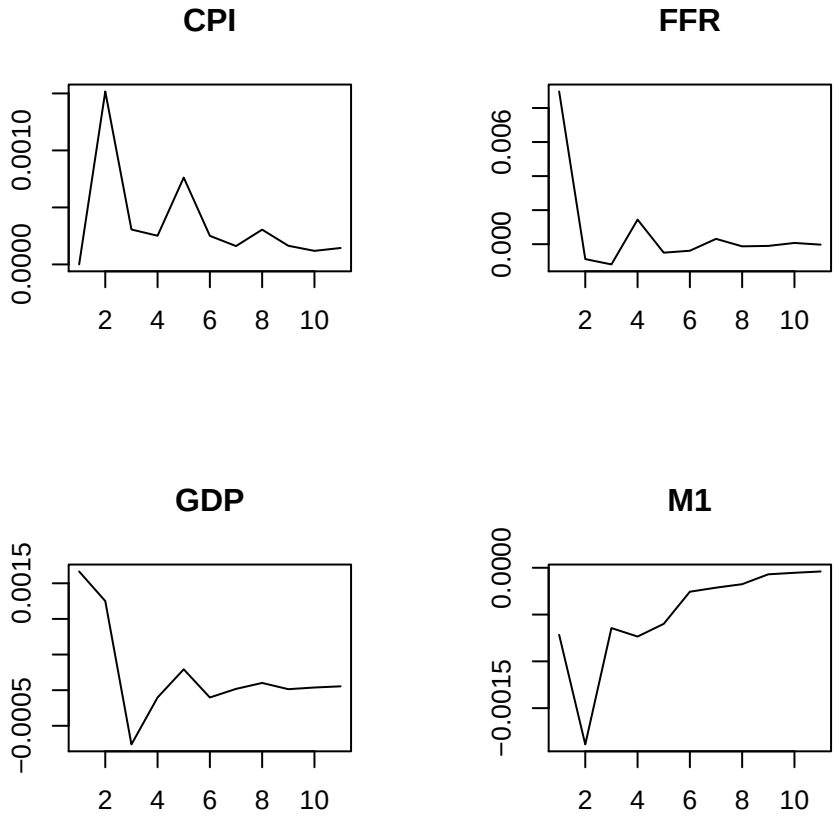


Figure 3.6: Impulse responses generated as the result of a 100 basis point increase to the Federal Funds Rate

3.3.5 Information Criterion Benchmarks

By default, we compare our methods to the conventional approach of selecting from a universal, sequentially increasing lag order as chosen by AIC or BIC and fitting the resulting VAR or VARX by least squares. Due to poor numerical stability as well as substantial computational overhead, we do not recommend including this benchmark when working in high dimensions (i.e. $kp \approx T$), hence we offer the option to disable it (by setting IC to `FALSE` in `constructModel`).

We implement the numerically stable and computationally efficient technique proposed in Neumaier and Schneider (2001), which calculates the least squares VARX using a QR decomposition that does not require explicit matrix inversion. `BigVAR` contains two functions that fit least squares VAR and VARX according to information criterion minimization. `VARXForecastEval`, which evaluates the h-step ahead forecasting performance of a VAR or VARX with lags selected by AIC or BIC over an evaluation period, and `VARXFit`, which fits a least squares VAR or VARX with the lag order selected by AIC or BIC. `VARXForecastEval` is called automatically by `cv.BigVAR` if `IC` is set to `TRUE` in `constructModel`. Implementation details are provided in Section C.0.11 of the appendix.

```
# Least Squares AIC VARX
LSAIC <- VARXFit(Y, 12, "AIC", NULL)
# VARX Forecast Eval with BIC
Pass in matrix of zeros for exogenous series
```

```r
# This matrix is not used in the VAR setting
X <- matrix(0, nrow = nrow(Y), ncol = 1)
# Shift by p quarters to account
for initialization in order to match
# cv.BigVAR output
BICEval <- VARXForecastEval(as.matrix(Y)[(p + 1):nrow(Y), ]
, X, p, 0, T2 - p, nrow(Y) - p, "BIC", 1)
mean(BICEval)


## [1] 5.443206
```

3.4 Refitting with least squares

Within the regularization framework, it is often of interest to use the lasso and its structured variants for variable selection, while refitting the support selected according to least squares. Belloni and Chernozhukov (2009) prove that a post-selection least squares refitting procedure has smaller bias than the conventional lasso in the univariate regression setting. In this section, we extend the refitting framework to the VAR context and perform a detailed simulation study to explore the forecasting performance of several potential refitting procedures.

Post-selection estimation has been considered in time-dependent problems by

Song and Bickel (2011), who refit by least squares based upon the support chosen by their structured VAR penalties. However, such an approach does not take into account Σ_u, the VAR innovation covariance matrix. A seminal result from Zellner (1962) shows that, in the absence of parameter restrictions, ordinary least squares and generalized least squares coincide in the VAR framework. However, once parameter restrictions are introduced, generalized least squares is more efficient.

A feasible generalized least squares VAR that incorporates parameter restrictions (such as setting coefficients to zero) is introduced in Brüggemann (2004) and is utilized by Davis et al. (2012) in the context of constrained maximum likelihood VAR estimation. Details of this approach are provided by Equations (C.6) and (C.7) in Section C.0.13 of the appendix.

We have found this formulation to be unsuited for our framework. First, in the early stages of rolling cross validation for short series, we often face scenarios in which the number of potential least squares parameters is close to exceeds the length of the series. Hence, taking the inverse of a poorly conditioned covariance matrix in these situations results in substantial estimation error. As an alternative, we propose extending the iterated feasible generalized least squares approach developed by Foschi et al. (2004), which formulates the feasible generalized least squares problem in a framework that avoids explicit matrix inversion. Details of our implementation are provided in Section C.0.15 in the appendix.

3.4.1 Simulation Study

In this section, we conduct a detailed simulation study to evaluate the forecasting performance of several refitting procedures. First, we consider the conventional *relaxed least squares* approach which simply refits the support selected according to restricted least squares (as defined by Equation (C.5) in the appendix). Second, we consider a *weighted relaxed least squares approach* which refits according to feasible generalized least squares using a covariance matrix with the diagonal entries set to the unconditional variance of each marginal series, and all other elements set to zero. Next, we consider the iterated feasible GLS approach, which iteratively refines the covariance matrix utilizing the procedure outlined in Algorithm 11 the appendix. Finally, we compare against the "oracle" procedure in which we perform generalized least squares using the covariance matrix from which the data was generated.

In this section, we operate exclusively in the Basic VAR-L setting. We do not believe that it is appropriate to refit when imposing structure; the groupings impose a ridge-like regularization effect which is not preserved after a least squares transformation.

We consider simulating from a $\text{VAR}_8(4)$ with an unstructured sparsity pattern as depicted in Figure 3.7 and we consider four covariance matrices that are discussed in the following sections. For each covariance matrix, we simulate a VAR of length 200 and use the middle third of the data for penalty parameter selection

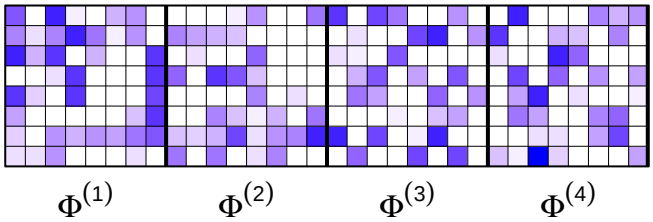


Figure 3.7: Sparsity Pattern of the $VAR_8(4)$ Coefficient Matrix Used in all Simulation Scenarios

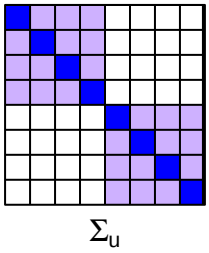


Figure 3.8: Covariance Matrix Used in Simulation Scenario 1

and the final third for forecast evaluation. We record the average 1-step ahead MSFE over the evaluation period for each simulation and repeat this process 100 times.

Simulation Scenario 1: Sparse Hub Structure

The first covariance matrix we consider is sparse with two *cliques*. Each series within the clique has identical covariance and it is set to zero outside of the clique. The variance is identical across all observations. Note that since we do not impose sparsity in our covariance estimation, our IFGLS procedure will not be able to capture this structure.

Table 3.3: Out of sample MSFE of one-step ahead forecasts after 100 simulations: Scenario 1

Model	Average MSFE	Standard Error
Basic	2.4527	0.0188
Basic Relaxed Least Squares	2.4756	0.0201
Basic Weighted Least Squares	2.4991	0.0204
Basic IFGLS	2.4526	0.0199
Basic Oracle	2.4483	0.0197
Sample Mean	3.6568	0.0370
Random Walk	7.0373	0.0808
Least Squares AIC VAR	2.9113	0.0236
Least Squares BIC VAR	3.6212	0.0357

We observe that the Oracle GLS achieves the best forecasting performance, though both the IFGLS procedure and the Basic VAR-L are well within one standard error. The relaxed least squares outperforms weighted least squares which subsequently outperforms all naive methods. It should be noted that all Basic VAR-L methods achieve very similar forecast performance, suggesting that in this scenario, there is little to be gained in terms of forecasting improvements by refitting.

Simulation Scenario 2: Poorly Conditioned

We next consider simulating using a covariance matrix with a high condition number. We constructed the covariance matrix to have a condition number of

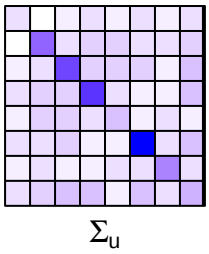


Figure 3.9: Covariance Matrix Used in Simulation Scenario 2

50,214,428. In such a scenario, the conventional feasible GLS estimator is inadvisable as computing Σ_u^{-1} will result in substantial estimation error. This is a scenario that we have encountered in the early stages of sequential cross validation, in which the length of the time series is relatively small compared to the number of potential model coefficients. Under this scenario, we should expect the Oracle GLS estimator to perform very poorly as a result of this imprecision. Since our IFGLS procedure does not require explicit matrix inversion, it should be relatively robust to a poorly conditioned covariance matrix.

Under this scenario, we find that any form of refitting only serves to degrade forecast performance; the Basic VAR-L achieves the best performance. The IFGLS performs relatively well, better than any other refitting method and within one standard error of the Basic VAR-L. The Oracle GLS, as it is trying to incorporate a nearly singular covariance, achieves relatively poor performance, on par with weighted least squares. Notice that the AIC and BIC VARs, both of which incorporate the covariance in lag order selection, achieve the exact same forecasting performance.

Table 3.4: Out of sample MSFE of one-step ahead forecasts after 100 simulations: Scenario 2

Model	Average MSFE	Standard Error
Basic	6.4045	0.0519
Basic Relaxed Least Squares	6.5135	0.0539
Basic Weighted Least Squares	6.6003	0.0588
Basic IFGLS	6.4464	0.0544
Basic Oracle	6.5979	0.0561
Sample Mean	9.7013	0.1106
Random Walk	18.5767	0.2505
Least Squares AIC VAR	7.5102	0.0679
Least Squares BIC VAR	7.5102	0.0679

Scenario 3: Scaled Identity

We next consider the case in which the covariance matrix is set to $0.1 \times I_k$. This scenario examines the robustness of the IFGLS framework in cases where an estimate of the covariance should provide no aid in forecasting. The results from this scenario are detailed in Table 3.5.

In this setting, we again find that the Basic VAR-L achieves the best forecasting performance, substantially outperforming all refitting procedures, which are all within one standard error of each other. This suggests that in settings in which there is no contemporaneous dependence, any type of refitting will only serve to degrade forecast performance.

Table 3.5: Out of sample MSFE of one-step ahead forecasts after 100 simulations: Scenario 3

Model	Average MSFE	Standard Error
Basic	0.8700	0.0055
Basic Relaxed Least Squares	0.8779	0.0057
Basic Weighted Least Squares	0.8775	0.0060
Basic IFGLS	0.8782	0.0060
Basic Oracle	0.8773	0.0060
Sample Mean	1.3218	0.0122
Random Walk	2.6096	0.0285
Least Squares AIC VAR	1.0273	0.0073
Least Squares BIC VAR	1.3219	0.0122

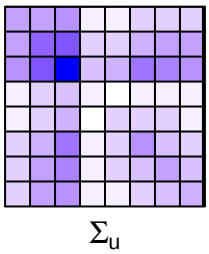


Figure 3.10: Covariance Matrix Used in Simulation Scenario 4

Scenario 4: Dense matrix

Our final scenario considers a well-conditioned dense covariance matrix as shown in Figure 3.10. In this setting, we should expect the IFGLS estimator and Oracle to achieve the best performance, as they are best able to capture the true covariance structure.

Table 3.6: Out of sample MSFE of one-step ahead forecasts after 100 simulations: Scenario 4

Model	Average MSFE	Standard Error
Basic Unrelaxed	3.0012	0.0446
Basic Relaxed Least Squares	3.0355	0.0447
Basic Weighted Least Squares	3.0598	0.0448
Basic IFGLS	2.9641	0.0406
Basic Oracle	3.1561	0.0510
Sample Mean	4.4773	0.0719
Random Walk	7.9507	0.1340
Least Squares AIC VAR	3.5527	0.0550
Least Squares BIC VAR	3.5527	0.0550

We find that the IFGLS procedure achieves the best forecasting performance, followed by the unrelaxed Basic VAR-L. All other refitting procedures perform substantially worse. Surprisingly, the Oracle GLS performs the worst of any regularization procedure. This demonstrates yet again that even if we can obtain a reliable estimate for the covariance matrix, it provides no guarantee of forecasting improvement.

Empirical Example

We additionally consider examining the performance of these models on the macroeconomic data examined in Section 3.3.2.

In applying refitting procedures to actual data, we find that none of the pro-

Table 3.7: Out of sample MSFE of one-step ahead forecasts of 4 US macroeconomic series

Model	Out-of-sample MSFE
Basic Unrelaxed	4.7353
Basic Relaxed Least Squares	4.7995
Basic Weighted Least Squares	4.8196
Basic IFGLS	4.8440
Sample Mean	5.3722
Random Walk	6.8677
Least Squares AIC VAR	5.3722
Least Squares BIC VAR	5.4281

posed methods lead to forecasting improvements over the Basic VAR-L.

3.4.2 Summary

We observe that in most simulation scenarios as well as our empirical application, refitting does not lead to substantial improvements in forecasting performance. This lack of improvement is likely due to several factors. First, it is possible that a different penalty parameter selection procedure is more appropriate when refitting is involved. In our experience, the penalty parameter selected by sequential cross validation tends to "over-select" model coefficients, choosing many coefficients that are technically active, but extremely small in magnitude. It would appear that very small magnitude model coefficients should not be refit, but defining a cutoff magnitude is challenging.

Belloni et al. (2011) develop a hypothesis testing procedure that can be used to determine which coefficients to refit, but it does not extend to a multivariate time-dependent setting. In addition, as pointed out by Belloni and Chernozhukov (2009), when refitting, the optimal penalty parameter should larger than in the unrelaxed setting. This suggests that proper incorporation of refitting requires the development of an alternative penalty parameter selection procedure that encourages more sparse solutions.

Despite its relatively lackluster forecasting performance, the IFGLS framework could be potentially useful in applications other than forecasting, such as generating impulse response functions (as discussed in Section 3.3.4), in which a reliable estimate of the innovation covariance matrix is crucial for an accurate depiction of the joint dynamics of the included series.

3.5 Conclusion

`BigVAR` offers a convenient framework for the forecasting of high-dimensional multivariate time series with structured convex penalties. Our methodology is transparent and can easily be understood and applied by practitioners and academics alike. Our package is currently available on the Comprehensive R Archive Network.

APPENDIX A

APPENDIX TO CHAPTER 1

A.0.1 Compact Matrix Notation

In deriving the solution methods for our algorithms, we find it convenient to express the VARX using compact matrix notation

$$\begin{array}{llll} \boldsymbol{Y} = [\mathbf{y}_1, \ldots, \mathbf{y}_T] & (k \times T); & \mathbf{1} = [1, \ldots, 1]^\top & (T \times 1). \\ \boldsymbol{Z}_t = [\mathbf{y}_t^\top, \ldots, \mathbf{y}_{t-p}^\top, \mathbf{x}_t^\top, \ldots, \mathbf{x}_{t-s}^\top]^\top & [(kp + ms) \times 1]; & \boldsymbol{Z} = [\boldsymbol{Z}_2, \ldots, \boldsymbol{Z}_{T-1}] & [(kp + ms) \times T]; \\ \boldsymbol{\Phi} = [\boldsymbol{\Phi}^{(1)}, \boldsymbol{\Phi}^{(2)}, \ldots, \boldsymbol{\Phi}^{(p)}] & (k \times kp); & \boldsymbol{\beta} = [\boldsymbol{\beta}^{(1)}, \ldots, \boldsymbol{\beta}^{(s)}] & [k \times ms]; \\ \boldsymbol{B} = [\boldsymbol{\Phi}, \boldsymbol{\beta}] & [k \times (kp + ms)]; & \mathbf{U} = [\mathbf{u}_1, \ldots, \mathbf{u}_T] & (k \times T) \end{array}$$

Equation (2.1) then becomes

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

and the least squares procedure (2.2) can be expressed as minimizing $\frac{1}{2}\|\boldsymbol{Y} - \boldsymbol{\nu}\mathbf{1}^\top - \boldsymbol{B}\boldsymbol{Z}\|_F^2$ over $\boldsymbol{\nu}$ and $\boldsymbol{B}$.

A.0.2 Intercept Term

In regularization problems, the intercept $\hat{\boldsymbol{\nu}}$ is not typically regularized and can instead be derived separately. Using compact matrix notation, we can express the unpenalized portion of (3.2) as

$$f(\boldsymbol{B}, \boldsymbol{\nu}) = \frac{1}{2}\|\boldsymbol{Y} - \boldsymbol{\nu}\mathbf{1}^\top - \boldsymbol{B}\boldsymbol{Z}\|_F^2, \tag{A.1}$$

We can find $\hat{\boldsymbol{\nu}}$ by calculating the gradient of (A.1) with respect to $\boldsymbol{\nu}$

$$0 = \nabla_{\boldsymbol{\nu}} f(\boldsymbol{B}, \boldsymbol{\nu}) = (\boldsymbol{Y} - \hat{\boldsymbol{\nu}}\mathbf{1}^\top - \widehat{\boldsymbol{B}}\boldsymbol{Z})\mathbf{1},$$
$$\implies \hat{\boldsymbol{\nu}}_j(\lambda) = \overline{Y}_{k\cdot} - \widehat{\boldsymbol{B}}\overline{Z}_{k\cdot},$$

in which $\overline{Y}_{k\cdot} = \frac{1}{T}\sum_t \boldsymbol{Y}_{kt}$, and $\overline{Z}_{k\cdot} = \frac{1}{T}\sum_t \boldsymbol{Z}_{kt}$. This provides some insight into the scaling, as we can rewrite (A.1) as

$$\begin{aligned} &\min_{\boldsymbol{B}} \frac{1}{2}\|\boldsymbol{Y} - (\overline{Y} - \boldsymbol{B}\overline{Z})\mathbf{1}^\top - \boldsymbol{B}\boldsymbol{Z}\|_F^2, \\ &= \min_{\boldsymbol{B}} \frac{1}{2}\|(\boldsymbol{Y} - \overline{Y}\mathbf{1}^\top) - \boldsymbol{B}(\boldsymbol{Z} - \overline{Z}\mathbf{1}^\top)\|_F^2, \end{aligned} \tag{A.2}$$

in which $\overline{Y}$ is a $k \times 1$ vector of row means and $\overline{Z}$ is a $(kp + ms) \times 1$ vector of row means.

A.0.3 Solution Strategies

In the following sections, assume that $\boldsymbol{Y}$ and $\boldsymbol{Z}$ are centered as in Equation (A.2).

Basic VARX-L

Utilizing the coordinate descent framework, we can find $\widehat{\boldsymbol{B}}$ via scalar updates. To generalize to a multivariate context, we can express the one-variable update for the (j, r) entry of $\boldsymbol{B}$, $\boldsymbol{B}_{jr}$ as

$$\min_{\boldsymbol{B}_{jr}} \frac{1}{2}\sum_t (\boldsymbol{Y}_{jt} - \sum_{\ell \neq r} \boldsymbol{B}_{j\ell}\boldsymbol{Z}_{\ell t} - \boldsymbol{B}_{jr}\boldsymbol{Z}_{jt})^2 + \lambda|\boldsymbol{B}_{jr}|. \tag{A.3}$$

Let $\mathbf{R}_t = \boldsymbol{Y}_{jt} - \sum_{\ell \neq r} \boldsymbol{B}_{j\ell} \boldsymbol{Z}_{\ell t}$ denote the partial residual. Then, we can rewrite Equation (A.3) as

$$g_{jr}(\boldsymbol{B}) = \min_{\boldsymbol{B}_{jr}} \frac{1}{2} \sum_t (\mathbf{R}_t - \boldsymbol{B}_{jr} \boldsymbol{Z}_{jt})^2 + \lambda |\boldsymbol{B}_{jr}|$$
$$= \min_{\boldsymbol{B}_{jr}} \frac{1}{2} \sum_t (\mathbf{R}_t^2 - \boldsymbol{B}_{jr}^2 \boldsymbol{Z}_{jt}^2 - 2\mathbf{R}_t \boldsymbol{Z}_{jt} \boldsymbol{B}_{jr}) + \lambda |\boldsymbol{B}_{jr}|.$$

Now, differentiating with respect to $\boldsymbol{B}_{jr}$ gives the subgradient as

$$\partial g_{jr}(\boldsymbol{B}) \ni \boldsymbol{B}_{jr} \sum_t \boldsymbol{Z}_{jt}^2 - \sum_t \mathbf{R}_t \boldsymbol{Z}_{jt} + \lambda \psi(\boldsymbol{B}_{jr}),$$

where we define $\psi(\boldsymbol{B}_{jr})$ as

$$\psi \in \begin{cases} \{\text{sgn}(\boldsymbol{B}_{jr})\} & \boldsymbol{B}_{jr} \neq 0 \\ [-1, 1] & \boldsymbol{B}_{jr} = 0. \end{cases}$$

For $\hat{\boldsymbol{B}}_{jr}$ to be a global minimum, $0 \in \partial g(\hat{\boldsymbol{B}}_{jr})$. After some algebra, the optimal update can be expressed as

$$\hat{\boldsymbol{B}}_{jr} \leftarrow \frac{\mathcal{ST}(\sum_t \mathbf{R}_t \boldsymbol{Z}_{jt}, \lambda)}{\sum_t \boldsymbol{Z}_{jt}^2}.$$

Where $\mathcal{ST}$ represents the soft-threshold operator

$$\mathcal{ST}(x, \phi) = \text{sgn}(x)(|x| - \phi)_+,$$

sgn denotes the signum function, and $(|x| - \phi)_+ = \max(|x| - \phi, 0)$. The Basic VARX-L procedure is detailed in Algorithm 9.

Lag Group VARX-L

Rather than vectorizing the Lag Group VARX-L and solving the corresponding univariate least squares problem, if the groups are proper submatrices we can

exploit the matrix structure for considerable computational gains. Without loss of generality, we will consider the "one lag" problem for $\mathbf{\Phi}^{(q)}$ (the problem for $\boldsymbol{\beta}^{(q)}$ is analogous).

$$\min_{\mathbf{\Phi}^{(q)}} \frac{1}{2}||\mathbf{R}_{-q} - \mathbf{\Phi}^{(q)}\mathbf{Z}_q||_F^2 + \lambda||\mathbf{\Phi}^{(q)}||_F, \tag{A.4}$$

in which, for notational ease, we directly incorporate the weighting into the penalty parameter by defining $\lambda = k\lambda$, $\mathbf{R}_q = \mathbf{\Phi}^{(-q)}\mathbf{Z}_{-q} - \boldsymbol{Y} \in \mathbb{R}^{k\times T}$ again represents the partial residual. Taking the gradient of $||\mathbf{R}_{-q} - \mathbf{\Phi}^{(q)}\mathbf{Z}_q||_F^2$ with respect to $\mathbf{\Phi}^{(q)}$ results in

$$\begin{aligned}
&\nabla_{\mathbf{\Phi}^{(q)}} \frac{1}{2}||\mathbf{R}_{-q} - \mathbf{\Phi}^{(q)}\mathbf{Z}_q||_F^2 = \nabla_{\mathbf{\Phi}^{(q)}} \operatorname{Tr}\left((\mathbf{R}_{-q} - \mathbf{\Phi}^{(q)}\mathbf{Z}_q)(\mathbf{R}_{-q} - \mathbf{\Phi}^{(q)}\mathbf{Z}_q)^\top\right), \\
=&\mathbf{\Phi}^{(q)}\mathbf{Z}_q\mathbf{Z}_q^\top - \mathbf{R}_{-q}\mathbf{Z}_q^\top.
\end{aligned}$$

In which Tr denotes the trace operator. The subgradient of (A.4) with respect to $\mathbf{\Phi}^{(q)}$ is then

$$\mathbf{\Phi}^{(q)}\mathbf{Z}_q\mathbf{Z}_q^\top - \mathbf{R}_{-q}\mathbf{Z}_q^\top + \lambda\omega(\mathbf{\Phi}^{(q)}),$$

where ω is defined as

$$\omega(\mathbf{\Phi}^{(q)}) = \begin{cases} \dfrac{\mathbf{\Phi}^{(q)}}{\|\mathbf{\Phi}^{(q)}\|_F} & \mathbf{\Phi}^{(q)} \neq 0 \\ \{U : \|U\|_F \leq 1\} & \mathbf{\Phi}^{(q)} = 0. \end{cases}$$

Consider the case where $\hat{\mathbf{\Phi}}^{(q)} = \mathbf{0}$. Then

$$\begin{aligned}
&\frac{\hat{\mathbf{\Phi}}^{(q)}\mathbf{Z}_q\mathbf{Z}_q^\top - \mathbf{R}_{-q}\mathbf{Z}_q^\top}{\lambda} \in \{U : \|U\|_F \leq 1\}, \\
&\iff ||\hat{\mathbf{\Phi}}^{(q)}\mathbf{Z}_q\mathbf{Z}_q^\top - \mathbf{R}_{-q}\mathbf{Z}_q^\top||_F \leq \lambda, \\
&\iff ||\mathbf{R}_{-q}\mathbf{Z}_q^\top||_F \leq \lambda, \\
&\iff \hat{\mathbf{\Phi}}^{(q)} = \mathbf{0}.
\end{aligned}$$

We can conclude that $\hat{\mathbf{\Phi}}^{(q)} = 0 \iff \|\mathbf{R}_{-q}\mathbf{Z}_q^\top\|_F \leq \lambda$. Now, assuming $\hat{\mathbf{\Phi}}^{(q)} \neq 0$, we have that

$$\begin{aligned}
&\mathbf{\Phi}^{(q)}\mathbf{Z}_q\mathbf{Z}_q^\top - \mathbf{R}_{-q}\mathbf{Z}_q^\top + \lambda(\frac{\mathbf{\Phi}^{(q)}}{\|\mathbf{\Phi}^{(q)}\|_F}) = 0,\\
&\mathbf{\Phi}^{(q)}\mathbf{Z}_q\mathbf{Z}_q^\top + \lambda(\frac{\mathbf{\Phi}^{(q)}}{\|\mathbf{\Phi}^{(q)}\|_F}) = \mathbf{R}_{-q}\mathbf{Z}_q^\top,\\
&\mathbf{\Phi}^{(q)}\left(\boldsymbol{Z}_q\boldsymbol{Z}_q^\top + \frac{\lambda}{\|\mathbf{\Phi}^{(q)}\|_F}\boldsymbol{I}_k\right) = \mathbf{R}_{-q}\mathbf{Z}_q^\top. \qquad \text{(A.5)}
\end{aligned}$$

Now, since $\boldsymbol{Z}_q\boldsymbol{Z}_q^\top$ is positive definite and $\lambda > 0$, we can infer that $\boldsymbol{Z}_q\boldsymbol{Z}_q^\top + \frac{\lambda}{\|\mathbf{\Phi}^{(q)}\|_F}\boldsymbol{I}_k$ is positive definite, hence it is possible to create a trust region subproblem that coincides with Equation (A.4). However, we need to transform $\mathbf{R}_{-q}\mathbf{Z}_q^\top \in \mathbb{R}^{k\times k}$ into a scalar. Define

$$\begin{aligned}
\boldsymbol{r}_q &= \text{vec}(\mathbf{R}_{-q}\mathbf{Z}_q^\top),\\
\mathbf{G}_q &= \boldsymbol{Z}_q\boldsymbol{Z}_q^\top \otimes \mathbf{I}_k,\\
\boldsymbol{\phi}_q &= \text{vec}(\mathbf{\Phi}^{(q)}),
\end{aligned}$$

in which $\otimes$ denotes the Kronecker product. Hence, we can rewrite Equation (A.5) as

$$\boldsymbol{\phi}_q^\top\left(\mathbf{G}_q + \frac{\lambda}{\|\boldsymbol{\phi}_q\|_F}\boldsymbol{I}_{k^2}\right) = \boldsymbol{r}_q.$$

Applying the same transformation to the original subproblem, we can express Equation A.4 as the trust region subproblem

$$\begin{aligned}
&\min \frac{1}{2}\boldsymbol{\phi}_q^\top\boldsymbol{G}_q\boldsymbol{G}_q^\top\boldsymbol{\phi}_q + \boldsymbol{r}_q^\top\boldsymbol{\phi}_q,\\
&\text{s.t.}\|\boldsymbol{\phi}_q\|_F \leq \Delta,
\end{aligned}$$

in which $\Delta > 0$ denotes to the trust-region radius which corresponds to the optimal solution of Equation A.4. These modifications allow for the use of the block coordinate descent algorithm described in Qin et al. (2010). Expanding upon their arguments, by the Karush-Kuhn-Tucker (KKT) conditions, we must have that: $\lambda(\Delta - \|\phi_q^*\|_F) = 0$, which implies that $\|\phi_q^*\|_F = \Delta$. Then, applying Theorem 4.1 of Nocedal and Wright (1999), we can conclude that

$$\phi_q^* = -\left(\boldsymbol{G}_q + \frac{\lambda}{\Delta}\boldsymbol{I}_{k^2}\right)^{-1} \boldsymbol{r}_q. \tag{A.6}$$

Qin et al. (2010) remarks that Equation (A.6) can also be expressed as $\phi_q^* = \Delta y_q(\Delta)$, where

$$y_q(\Delta) = -\left(\Delta\boldsymbol{G}_q + \lambda\boldsymbol{I}_{k^2}\right)^{-1} \boldsymbol{r}_q, \tag{A.7}$$

Note that, based on the KKT conditions, $\|y_q(\Delta)\|_F = 1$. Hence, the optimal Δ can be chosen to satisfy $\|y_q(\Delta)\|_F = 1$. We can efficiently compute $\|y_q(\Delta)\|_F^2$ via an eigen-decomposition of $\boldsymbol{G}_q$. We start by rewriting Equation (A.7) as

$$\begin{aligned} y_q(\Delta) &= -\left(\Delta\boldsymbol{W}V\boldsymbol{W} + \lambda\boldsymbol{I}_{k^2}\right)^{-1} \boldsymbol{r}_q, \\ &= -\boldsymbol{W}(\Delta V + \lambda\boldsymbol{I}_{k^2})^{-1}\boldsymbol{W}\boldsymbol{r}_q, \end{aligned}$$

in which the first line follows from the spectral decomposition of a symmetric positive definite matrix. Now, letting $\Psi = (\Delta V + \lambda\boldsymbol{I}_{k^2})^{-1}$, expanding on the arguments

of Qin et al. (2010), we find

$$\begin{aligned}
\|y_q(\Delta)\|_F^2 &= \mathrm{Tr}\left((\boldsymbol{W}\Psi\boldsymbol{W}^{-1}\boldsymbol{r}_q)(\boldsymbol{W}\Psi\boldsymbol{W}^{-1}\boldsymbol{r}_q)^\top\right), \\
&= \mathrm{Tr}\left(\boldsymbol{W}\Psi\boldsymbol{W}^{-1}\boldsymbol{r}_q\boldsymbol{r}_q^\top\boldsymbol{W}^{-1\top}\Psi^\top\boldsymbol{W}^\top\right), \\
&= \mathrm{Tr}\left(\boldsymbol{r}_q^\top\boldsymbol{W}^{-1\top}\Psi^\top\boldsymbol{W}^\top\boldsymbol{W}\Psi\boldsymbol{W}^{-1}\boldsymbol{r}_q\right), \\
&= \mathrm{Tr}\left(\boldsymbol{r}_q^\top\boldsymbol{W}^{-1\top}\Psi^\top\Psi\boldsymbol{W}^{-1}\boldsymbol{r}_q\right), \\
&= \mathrm{Tr}\left(\boldsymbol{W}^\top\boldsymbol{r}_q\boldsymbol{r}_q^\top\boldsymbol{W}\Psi^\top\Psi\right).
\end{aligned}$$

Finally, we can express $\|y_q(\Delta)\|_F^2$ as

$$\|y_q(\Delta)\|_F^2 = \sum_i \frac{(\mathbf{w_i}^\top\boldsymbol{r}_q)^2}{(\mathbf{v}_i\Delta + \lambda)^2},$$

in which $\boldsymbol{w}_i$ denotes the columns of $\boldsymbol{W}$ and $\boldsymbol{v}_i$ the diagonal elements of $\boldsymbol{V}$. Qin et al. (2010) notes that we can determine the optimal Δ by applying Newton's method to find the root of

$$\Omega(\Delta) = 1 - \frac{1}{\|y_q(\Delta)\|_F}. \tag{A.8}$$

The full Lag Group VARX-L procedure is detailed in Algorithm 5. Our algorithm organizes iterations around an "active-set" as described in Friedman et al. (2010). This approach starts by cycling through every group and then only iterating on the subset of $\boldsymbol{B}$ that are nonzero (the "active-set") until convergence. If a full pass through all $\boldsymbol{B}$ does not change the active set, the algorithm has converged, otherwise the process is repeated. This approach considerably reduces computation time, especially for large values of λ in which most model coefficients are zero.

Own/Other Group VARX-L

In the Own/Other setting since the groups are not proper submatrices, in order to properly partition each $\mathbf{\Phi}^{(\ell)}$ into separate groups for own and other lags, Equation (3.2) must be transformed into a least squares problem. To perform a least squares transformation, we define the following

$$
\begin{aligned}
r_{-qq} &= \text{vec}(\mathbf{R}_{-qq}), \\
\phi_{qq} &= \text{vec}(\mathbf{\Phi}^{(q)}_{\text{on}}), \\
\boldsymbol{M}_{qq} &= (\boldsymbol{Z}^\top \otimes I_k)_{qq}.
\end{aligned}
$$

Then, the one block subproblem for own lags (group qq) can be expressed as

$$
\begin{aligned}
&\min_{\phi_{qq}} \frac{1}{2}||\boldsymbol{M}_{qq}\phi_{qq} + r_{-qq}||_F^2 + \lambda||\phi_{qq}||_F, \\
=&\min_{\phi_{qq}} \frac{1}{2} r_{-qq}^\top r + \phi_{qq}^\top \boldsymbol{M}_{qq}^\top \boldsymbol{M}_{qq}\phi_{qq} + r_{-qq}^\top \boldsymbol{M}_{qq}\phi_{qq} + \lambda||\phi_{qq}||_F, \\
=&\min_{\phi_{qq}} \frac{1}{2} \phi_{qq}^\top \boldsymbol{M}_{qq}^\top \boldsymbol{M}_{qq}\phi_{qq} + r_{-qq}^\top \boldsymbol{M}_{qq}\phi_{qq} + \lambda||\phi_{qq}||_F.
\end{aligned}
$$

At $\hat{\phi}_{qq}$, we must have that $0 \in \partial f(\hat{\phi}_{qq})$. The subgradient can be expressed as

$$
\frac{\partial}{\partial \phi_{qq}} = \boldsymbol{M}_{qq}^\top \boldsymbol{M}_{qq}\phi_{qq} + \boldsymbol{M}_{qq}^\top r + \lambda\omega(\phi_{qq}),
$$

where ω is defined as

$$
\omega(s) \in \begin{cases} \{\frac{s}{\|s\|_F}\} & s \neq 0 \\ \{u : \|u\|_F \leq 1\} & s = 0. \end{cases}
$$

Thus, after applying these transformations, we can apply a slightly adapted version of Algorithm 5.

Sparse Lag Group VARX-L

As with the Lag Group VARX-L, we will consider the one-block subproblem for lag $\mathbf{\Phi}^{(q)}$

$$\min_{\mathbf{\Phi}^{(q)}} \frac{1}{2k}||\mathbf{R}_{-q} - \mathbf{\Phi}^{(q)}\mathbf{Z}_q||_F^2 + (1-\alpha)\lambda||\mathbf{\Phi}^{(q)}||_F + \alpha\lambda||\mathbf{\Phi}^{(q)}||_1. \tag{A.9}$$

Since the inclusion of within-group sparsity does not allow for separability, coordinate descent based procedures are no longer appropriate, therefore, following Simon et al. (2013) our solution to the Sparse Lag Group VARX-L utilizes gradient descent based methods. We express Equation (A.9) as the sum of a generic differentiable function with a Lipschitz gradient and a non-differentiable function.

We start by linearizing the quadratic approximation of the unpenalized loss function that only makes use of first-order information around its current estimate $\mathbf{\Phi}_0$ (borrowing from Simon et al. (2013), for notational ease, let $\mathbf{\Phi} \equiv \mathbf{\Phi}^{(q)}$, $\ell(\mathbf{\Phi})$ represent the unpenalized loss function, and $\mathcal{P}(\mathbf{\Phi})$ represent the penalty term). Then, we can express the linearization as

$$\begin{aligned} M(\mathbf{\Phi}, \mathbf{\Phi}_0) &= \ell(\mathbf{\Phi}_0) + \text{vec}(\mathbf{\Phi} - \mathbf{\Phi}_0)^\top \text{vec}(\nabla\ell(\mathbf{\Phi}_0)) + \frac{1}{2d}||\mathbf{\Phi} - \mathbf{\Phi}_0||_F^2 + \mathcal{P}(\mathbf{\Phi}), \\ &= \frac{1}{2k}||\mathbf{R}_{-q} - \mathbf{\Phi}_0\mathbf{Z}_q||_F^2 + \langle \mathbf{\Phi} - \mathbf{\Phi}_0, (\mathbf{\Phi}_0\mathbf{Z}_q - \mathbf{R}_{-q})\mathbf{Z}_q^\top \rangle + \frac{1}{2d}||\mathbf{\Phi} - \mathbf{\Phi}_0||_F^2 + \mathcal{P}(\mathbf{\Phi}), \end{aligned}$$

in which d represents the step size. Removing terms independent of $\mathbf{\Phi}$, our objective function becomes

$$\begin{aligned} &\underset{\mathbf{\Phi}}{\text{argmin}}\, M(\mathbf{\Phi}, \mathbf{\Phi}_0), \\ &= \underset{\mathbf{\Phi}}{\text{argmin}} \frac{1}{2d}||\mathbf{\Phi} - \left(\mathbf{\Phi}_0 - d(\mathbf{\Phi}_0\mathbf{Z}_q - \mathbf{R}_{-q})\mathbf{Z}_q^\top\right)||_F^2 + \mathcal{P}(\mathbf{\Phi}). \end{aligned}$$

Then, generalizing the arguments outlined by Simon et al. (2013), we can infer that the optimal update $U(\mathbf{\Phi})$ can be expressed as

$$U(\mathbf{\Phi}) = \left(1 - \frac{d(1-\alpha)\lambda}{\|ST(\mathbf{\Phi}_0 - d(\mathbf{\Phi}_0\mathbf{Z}_q - \mathbf{R}_{-q})\mathbf{Z}_q^\top, d\alpha\lambda)\|_F}\right)_+ ST(\mathbf{\Phi}_0 - d(\mathbf{\Phi}_0\mathbf{Z}_q - \mathbf{R}_{-q})\mathbf{Z}_{-q}^\top, d\alpha\lambda).$$

As in Simon et al. (2013), we apply a Nesterov accelerated update. At step j, we update according to

$$\hat{\mathbf{\Phi}}[j] \leftarrow \hat{\mathbf{\Phi}}[j-1] + \frac{j}{j+3}(U(\mathbf{\Phi}) - \hat{\mathbf{\Phi}}[j-1]), \tag{A.10}$$

which, per Beck and Teboulle (2009) converges at rate $1/j^2$ as opposed to the $1/j$ rate of the standard proximal gradient descent.

The calculation of the step size h can be problematic. Ideally, the step size should be as large as possible, as it leads to faster convergence, but if the step size is too large, the algorithm may diverge. The conventional method for determining step size, described in Simon et al. (2013) and Beck and Teboulle (2009), is to decrease h until

$$\ell(\widehat{\mathbf{\Phi}}, h) \leq \ell(U(\mathbf{\Phi})) + \langle \mathrm{vec}(\widehat{\mathbf{\Phi}} - U(\mathbf{\Phi})), \mathrm{vec}(\nabla\ell(U(\mathbf{\Phi}), h))\rangle + \frac{1}{2hk}\|\widehat{\mathbf{\Phi}} - U(\mathbf{\Phi})\|_F^2. \tag{A.11}$$

However, as noted in section 5.3 of Becker et al. (2011), Equation (A.11) has severe cancellation errors when $\ell(\widehat{\mathbf{\Phi}}, h) \approx \ell(U(\mathbf{\Phi}), h)$. They posit a more conservative approach, iterating until

$$\ell(\widehat{\mathbf{\Phi}}, h) \leq \frac{1}{2hk}\|\widehat{\mathbf{\Phi}} - U(\mathbf{\Phi})\|_F^2. \tag{A.12}$$

They recommend a hybrid approach: choosing Equation (A.11) when $\ell(\widehat{\mathbf{\Phi}}, h) - \ell(U(\mathbf{\Phi}), h) \geq \gamma\ell(\widehat{\mathbf{\Phi}}, h)$, for some small $\gamma > 0$ and choosing Equation (A.12) otherwise.

Unfortunately, we have found even this hybrid approach to be unstable. This could be due to the use of a Nesterov-style accelerated update which, per Bach et al. (2011), can result in the algorithm not decreasing at each step, causing the above specifications to diverge. We use a constant step size according to the Lipschitz constant, H, which must satisfy

$$\|\nabla_X \ell(X) - \nabla_Y \ell(Y)\| \leq H\|X - Y\|.$$

Consider two submatrices $\mathbf{A}^{(q)}$ and $\mathbf{C}^{(q)}$. We have that

$$\begin{aligned}
\nabla_{\mathbf{A}^{(q)}} \ell(\mathbf{A}^{(q)}) &= \mathbf{A}^{(q)} \mathbf{Z}_q \mathbf{Z}_q^\top - \mathbf{R}_{-q} \mathbf{Z}_q^\top, \\
\nabla_{\mathbf{C}^{(q)}} \ell(\mathbf{C}^{(q)}) &= \mathbf{C}^{(q)} \mathbf{Z}_q \mathbf{Z}_q^\top - \mathbf{R}_{-q} \mathbf{Z}_q^\top, \\
\implies \nabla_{\mathbf{A}^{(q)}} \ell(\mathbf{A}^{(q)}) - \nabla_{\mathbf{C}^{(q)}} \ell(\mathbf{C}^{(q)}) &= (\mathbf{A}^{(q)} - \mathbf{C}^{(q)}) \mathbf{Z}_q \mathbf{Z}_q^\top, \\
\implies \|(\mathbf{A}^{(q)} - \mathbf{C}^{(q)}) \mathbf{Z}_q \mathbf{Z}_q^\top\|_2 &\leq \|\mathbf{A}^{(q)} - \mathbf{C}^{(q)}\|_2 \|\mathbf{Z}_q \mathbf{Z}_q^\top\|_2.
\end{aligned}$$

The last inequality follows from the sub-multiplicity of the matrix 2-norm. Therefore, we can conclude that the Lipschitz constant is $\|\mathbf{Z}_q \mathbf{Z}_q^\top\|_2 = \sqrt{\sigma_1(\mathbf{Z}_q)}$, i.e. the square root of the largest singular value of $\mathbf{Z}_q$, which has dimension $k \times k$ for $\mathbf{\Phi}^{(1)}, \ldots, \mathbf{\Phi}^{(p)}$ and is a scalar for exogenous groups. Since $\mathbf{Z}_q \mathbf{Z}_q^\top$ is symmetric and positive definite, it is diagonalizable, and the maximum eigenvalue can be efficiently computed using the power method, described in Golub and Van Loan (2012).

As only the maximum eigenvalue is required, the power method is much more computationally efficient than a computation of the entire eigensystem. Moreover, we retain the corresponding eigenvector produced by this procedure to use as a

"warm start" that substantially decreases the amount of time required to compute the maximal eigenvalue at each point in time during the cross-validation and forecast evaluation procedures.

In a manner similar to Algorithm 5, an "active-set" approach is used to minimize computation time. The inner loop of of the Sparse Group VARX-L procedure is detailed in Algorithm (6). An outline of the algorithm is below:

1. Iterate through all groups. For each group:
   (a) Check if the group is active via the condition: $\|(\boldsymbol{\Phi}^{(q)}\boldsymbol{Z}_q - \mathbf{R}_{-q})\boldsymbol{Z}_q^\top\|_F \leq (1-\alpha)\lambda$.
   (b) If active, go to the inner loop (Algorithm 6), if not active, set group identically to zero.
   (c) Repeat until convergence.

Upon performing the least squares transformations as in the Own/Other Group VARX-L (section A.0.3), the Sparse Own/Other Group VARX-L follows almost the exact same procedure as its lag counterpart.

Endogenous-First VARX-L

The Endogenous-First VARX-L is of the form

$$\min_{\boldsymbol{\Phi},\boldsymbol{\beta}} \frac{1}{2}\|\boldsymbol{Y} - \boldsymbol{B}\boldsymbol{Z}\|_F^2 + \lambda \sum_{\ell=1}^{p}\sum_{j=1}^{k}\left(\|[\boldsymbol{\Phi}_j^{(\ell)}, \boldsymbol{\beta}_{j,\cdot}^{(\ell)}]\|_F + \|\boldsymbol{\beta}_{j,\cdot}^{(\ell)}\|_F\right).$$

Since the optimization problem decouples across rows, we will consider solving the *one row* subproblem (for row i)

$$\min_{\boldsymbol{\Phi}_i, \boldsymbol{\beta}_i} \frac{1}{2}\|\boldsymbol{Y}_i - \boldsymbol{B}_i\boldsymbol{Z}\|_F^2 + \lambda \sum_{\ell=1}^{p} \left(\|[\boldsymbol{\Phi}_i^{(\ell)}, \boldsymbol{\beta}_{i,\cdot}^{(\ell)}]\|_F + \|\boldsymbol{\beta}_{i,\cdot}^{(\ell)}\|_F \right). \tag{A.13}$$

In a manner similar to the Sparse Group VARX-L, the Endogenous-First VARX-L is solved via proximal gradient descent. For ease of notation, let $\mathcal{P}(\boldsymbol{\Phi}, \boldsymbol{\beta})$ represent the nested penalty. The update step for the Endogenous-First VARX-L (at step j) can be expressed as

$$\boldsymbol{B}_i[j] = \text{Prox}_{d\lambda, \mathcal{P}(\boldsymbol{\Phi}, \boldsymbol{\beta})}(\boldsymbol{B}_i[j-1] - d\nabla\ell(\boldsymbol{B}_i)), \tag{A.14}$$

in which d denotes step size and $\ell(\boldsymbol{B}_i)$ denotes the unpenalized loss function. Note that $\nabla\ell(\boldsymbol{B}_i) = -(\boldsymbol{Y}_i - \boldsymbol{B}_i\boldsymbol{Z})\boldsymbol{Z}^\top$. Similar to the Sparse Group VARX-L setting, a fixed step size is used; $d = \frac{1}{\sigma_1(\boldsymbol{Z})}$. To speed convergence, as in the Sparse Group VARX-L update step (A.10), we apply a similar Nesterov-style accelerated update:

$$\hat{\boldsymbol{B}} \leftarrow \hat{\boldsymbol{B}}[j-1] + \frac{j-2}{j+1}(\hat{\boldsymbol{B}}[j-1] - \hat{\boldsymbol{B}}[j-2]),$$

Thus, (A.14) becomes

$$\boldsymbol{B}_i[j] = \text{Prox}_{d\lambda, \mathcal{P}(\boldsymbol{\Phi}, \boldsymbol{\beta})}(\hat{\boldsymbol{B}} - d\nabla\ell(\boldsymbol{B}_i)), \tag{A.15}$$

Definition A.0.1 (Jenatton et al. (2011))**.** *The proximal operator associated with the Endogenous-First VARX can be expressed as*

$$Prox_{h_j\lambda, \mathcal{P}(\boldsymbol{\Phi}, \boldsymbol{\beta})} \arg\min_{v \in \mathbf{R}^{kp+ms}} \{\frac{1}{2}\|u - v\|_F^2 + h\lambda\mathcal{P}(v)\} \tag{A.16}$$

in other words, the proximal operator will map a vector $u \in \mathbf{R}^{kp+ms}$ to the unique solution of (A.16)*.*

Jenatton et al. (2011) observed that the dual of (A.15) can be solved with one pass of block coordinate descent. Moreover, the block updates are extremely simple and available in closed-form. Algorithm 7 details the prox function for the Endogenous-First VARX-L. Note that it consists of p separate nested structures for each series. Thus, solving (A.15) essentially amounts to calling the same proximal function p times at each update step.

A.0.4 Banbura et al. (2009) Implementation

The Bayesian VAR proposed by Banbura et al. (2009) utilizes a normal inverted Wishart Prior. Defining $\phi = \text{vec}(\mathbf{\Phi})$, the prior has the form

$$\phi|\Omega \sim N(\phi_0, \Omega \otimes \Omega_0)$$
$$\Omega \sim iW(S_0, \alpha_0),$$

in which iW denotes the inverse Wishart distribution. This prior is implemented by adding the following dummy observations to $\boldsymbol{Y}$ and $\boldsymbol{Z}^\top$ (which we define as

$\boldsymbol{X}$):

$$\boldsymbol{Y}_{d_1} = \begin{pmatrix} \text{diag}(\delta\sigma_1, \ldots, \delta\sigma_k)/\lambda \\ 0_{k\times(p-1)\times k} \\ \text{diag}(\sigma_1, \ldots, \sigma_k) \\ 0_{1\times k} \end{pmatrix}$$

$$\boldsymbol{X}_{d_1} = \begin{pmatrix} 0_{kp\times 1} & J_p \otimes \text{diag}(\sigma_1, \ldots, \sigma_k) \\ 0_{k\times 1} & 0_{k\times kp} \\ \epsilon & 0_{1\times kp} \end{pmatrix}.$$

The scale parameters for the prior variances of each series, $\sigma_1, \ldots, \sigma_k$, are estimated by univariate autoregressive models. $J_p = \text{diag}(1, 2, \ldots, p)$, and ϵ denotes the prior on the intercept and is set to a very small number (e.g. 1e-5). δ serves as an indicator for the prior belief that the series have high persistence. We set $\delta = 0$ in all of our forecasting applications except for the Minnesota VARX-L application in section 1.4.

In addition to the above construction, following Doan et al. (1984), BGR adds a prior that imposes a bound on the sum of coefficients by shrinking $\Pi = (I_k - \boldsymbol{\Phi}_1 - \cdots - \boldsymbol{\Phi}_p)$ toward zero. This prior is implemented by adding the additional dummy observations

$$\boldsymbol{Y}_{d_2} = \text{diag}(\delta\mu_1, \ldots, \delta\mu_k)/\tau,$$
$$\boldsymbol{X}_{d_2} = \begin{pmatrix} \mathbf{0}_{k\times 1} & 1_{1\times p} \otimes \text{diag}(\delta\mu_1, \ldots, \delta\mu_n)/\tau \end{pmatrix},$$

in which $\mu_1, \ldots, \mu_k$ are meant to capture the average level of each of the series, set according to their unconditional means and τ denotes a loose prior which is set

to 10λ. After appending the dummy observations to $\boldsymbol{Y}$ and $\boldsymbol{X}$ and creating the augmented matrices $\boldsymbol{Y}_*$ and $\boldsymbol{X}_*$, the posterior mean can be calculated in closed form as:

$$\tilde{\boldsymbol{\Phi}}^\top = (\boldsymbol{X}_*^\top \boldsymbol{X}_*)^{-1} \boldsymbol{X}_*^\top \boldsymbol{Y}_*$$

A.0.5 Penalty Grid Selection

Table A.1: Starting values of the penalty grid for each procedure; ρ_q denotes the number of variables in group q.

Structure	Starting Value of Λ_{Grid}
Basic	$\|\boldsymbol{Z}\boldsymbol{Y}^\top\|_\infty$
Lag Group	$\max_q(\|\boldsymbol{Z}_q\boldsymbol{Y}^\top\|_F)$
Sparse Lag	$\max_q(\|\boldsymbol{Z}_q\boldsymbol{Y}^\top\|_F)\alpha$
Own/Other Group	$\max_q(\|(\boldsymbol{Z}^\top \otimes \boldsymbol{I}_k)_q \text{vec}(\boldsymbol{Y})\|_F / \sqrt{\rho_q})$
Sparse Own/Other Group	$\max_q(\|(\boldsymbol{Z}^\top \otimes \boldsymbol{I}_k)_q \text{vec}(\boldsymbol{Y})\|_F / \sqrt{\rho_q})\alpha$
Endogenous-First	$\max_i(\|\boldsymbol{Z}\boldsymbol{Y}_i^\top\|_F)$,

A.0.6 Algorithms

Algorithm 3: Basic-VARX-L$_{k,m}(p, s)$

Require: $\boldsymbol{Y}, \boldsymbol{Z}, \boldsymbol{B}^{\mathrm{INI}}, \lambda$

$\boldsymbol{B}^{\mathrm{OLD}} \leftarrow \boldsymbol{B}^{\mathrm{INI}}$

repeat

for i in k, j in $kp + ms$ **do**

$\boldsymbol{R} \leftarrow \boldsymbol{Y}_{i\cdot} - \sum_{\ell \neq j} \boldsymbol{B}_{i\ell} \boldsymbol{Z}_{\ell\cdot}$

$\boldsymbol{B}_{ij}^{\mathrm{NEW}} \leftarrow \frac{\mathrm{ST}(\sum_t \boldsymbol{R}\boldsymbol{Z}_{j\cdot,\lambda})}{\sum_t \boldsymbol{Z}_{jt}^2}$

end for

$\boldsymbol{B}^{\mathrm{OLD}} \leftarrow \boldsymbol{B}^{\mathrm{NEW}}$

until Desired threshold is reached

$\hat{\boldsymbol{\nu}} \leftarrow \bar{\boldsymbol{Y}} - \boldsymbol{B}^{\mathrm{NEW}} \bar{\boldsymbol{Z}}$

return $\hat{\boldsymbol{\nu}}, \boldsymbol{B}^{\mathrm{NEW}}$

Algorithm 4: Basic VARX-L(p,s) Cross-Validation

Require: $\boldsymbol{Y}, \boldsymbol{Z}, \boldsymbol{B}^{\text{INI}}, \Lambda_{\text{grid}}, h$

$\boldsymbol{B}^{\text{LAST}} \leftarrow \boldsymbol{B}^{\text{INI}}$

for j in $[T_1, T_2 - h]$ **do**

$\boldsymbol{Y}^{(j)}_{\text{TRAIN}} \leftarrow \boldsymbol{Y}_{,h:(j-1)}$

$\boldsymbol{Z}^{(j)}_{\text{TRAIN}} \leftarrow \boldsymbol{Z}_{,1:(j-h+1)}$

for i in Λ_{Grid} **do**

$\boldsymbol{\nu}_i, \boldsymbol{B}^{\text{NEW}}_i \leftarrow$ Basic-VARX-L($\boldsymbol{Y}^{(j)}_{\text{TRAIN}}, \boldsymbol{Z}^{(j)}_{\text{TRAIN}}, \boldsymbol{B}^{\text{LAST}}_i, \lambda_i, \epsilon$)

$\boldsymbol{Z}^{(j)}_{\text{TEST}} \leftarrow \boldsymbol{Z}_{,(j+1)}$

$SSFE^{(j,i)} \leftarrow \|\boldsymbol{Y}_{j+h} - [\nu_i, \boldsymbol{B}^{\text{NEW}}_i] * [\mathbf{1}, \boldsymbol{Z}^{(j)}_{TEST}]\|^2_F$

$\boldsymbol{B}^{\text{LAST}}_i \leftarrow \boldsymbol{B}^{\text{NEW}}_i$

end for

for i in Λ_{Grid} **do**

$MSFE^{(i)} \leftarrow \frac{1}{T_2 - T_1 - h + 1} \sum_j SSFE^{(j,i)}$

end for

end for

return $\lambda_{\hat{i}}$, where $\hat{i} = \text{argmin}_i \, MSFE^{(i)}$

Algorithm 5: Lag Group VARX-L$_{k,m}(p, s)$ with active-set strategy

Require: $\boldsymbol{B}_{\text{INI}}, \mathcal{G}, \boldsymbol{Y}, \boldsymbol{Z}, \mathcal{A}_{\text{INI}}, \Lambda$

Define:

$$\text{for } g = 1, \ldots, p + ms:$$
$$\mathbf{G}_g = \mathbf{M}_g \otimes \mathbf{I}_k.$$

for $\lambda \in \Lambda_{\text{Grid}}$ **do**

$\boldsymbol{B}_{\lambda,\mathcal{A}_\lambda} \leftarrow \boldsymbol{B}_{\lambda,\text{INI}}$,

$\mathcal{A}_\lambda \leftarrow \mathcal{A}_{\lambda,\text{INI}}$

repeat

$\boldsymbol{B}_{\lambda,\mathcal{A}_\lambda} \leftarrow$ThresholdUpdate$(\mathcal{A}_\lambda, \boldsymbol{B}_{\lambda,\mathcal{A}_\lambda}, \lambda)$

$\boldsymbol{B}_{\lambda,\mathcal{A}_{\text{FULL}}}, \mathcal{A}_\lambda \leftarrow$BlockUpdate$(\mathcal{A}_{\text{FULL}}, \boldsymbol{B}_{\lambda,\mathcal{A}_\lambda}, \lambda)$

until $\boldsymbol{B}_{\lambda,\mathcal{A}_\lambda} = \boldsymbol{B}_{\lambda,\mathcal{A}_{\text{FULL}}}$

$\hat{\nu} \leftarrow \bar{\boldsymbol{Y}} - \boldsymbol{B}_{\lambda,\mathcal{A}} \bar{\boldsymbol{Z}}$

end for

return $\hat{\nu}, \boldsymbol{B}_\Lambda, A_\Lambda$

procedure BLOCKUPDATE$(\mathcal{G}, \boldsymbol{B}_{\text{INI}}, \lambda)$ ▷ Makes one full pass through all groups

$\boldsymbol{B} \leftarrow \boldsymbol{B}_{\text{INI}}$

for $g \in \mathcal{G}$ **do**

$\mathbf{R} \leftarrow \boldsymbol{B}_{-g} \boldsymbol{Z}_{-g} - \boldsymbol{Y}$

$\mathbf{r} \leftarrow \mathbf{R} \boldsymbol{Z}_q^\top$

if $\|\mathbf{r}\|_F \leq \lambda$ **then**

$\boldsymbol{B}_g^* \leftarrow \mathbf{0}_{|g|}$

$\mathcal{A}_g \leftarrow \emptyset$

end if

if $\|\mathbf{r}\|_F > \lambda$ **then**
$\Delta \leftarrow$ the root of $\Omega(\Delta)$ defined in (A.8)
$\text{vec}(\boldsymbol{B}_g) \leftarrow -(\mathbf{G}_g + \frac{\lambda}{\Delta}\mathbf{I})^{-1}\mathbf{r}$
$\mathcal{A}_g \leftarrow g$
end if
end for
return $\boldsymbol{B}_\lambda$, $\mathcal{A}$
end procedure
procedure THRESHOLDUPDATE($\mathcal{A}_\lambda, \boldsymbol{B}_{\lambda,\text{INI}}, \lambda$) ▷ Iterates through active set until convergence
if $\mathcal{A} = \emptyset$ **then return** $\mathbf{0}_{k\times kp+ms}$
end if
if $\mathcal{A} \neq \emptyset$ **then**
$\boldsymbol{B}_{\lambda,\text{OLD}} \leftarrow \boldsymbol{B}_{\lambda,\text{INI}}$
repeat
$\boldsymbol{B}_{\lambda,\text{NEW}}, \mathcal{A}_\lambda \leftarrow \text{BlockUpdate}(\mathcal{A}_\lambda, \boldsymbol{B}_{\lambda,\text{OLD}}, \lambda)$
$\boldsymbol{B}_{\lambda,\text{OLD}} \leftarrow \boldsymbol{B}_{\lambda,\text{NEW}}$
until Desired threshold is reached
end if
return $\boldsymbol{B}_{\lambda,\text{NEW}}, \mathcal{A}$
end procedure

Algorithm 6: Sparse Group VARX-L inner loop

Require: $\mathbf{\Phi}_0, \mathbf{Z}_q, \mathbf{R}_{-q}$

$h \leftarrow \frac{1}{\sigma_1(\mathbf{Z}_q)}$

$\mathbf{\Phi}_0 \leftarrow \mathbf{\Phi}^1$

repeat

$j \leftarrow 1$

$\boldsymbol{F}_q \leftarrow \frac{(\mathbf{\Phi}^j \mathbf{Z}_q - \mathbf{R}_{-q})\mathbf{Z}_q^\top}{k}$

$\gamma^j \leftarrow \mathbf{\Phi}^j$

$$\mathrm{vec}(\gamma^{(j+1)}) \leftarrow \left(1 - \frac{h(1-\alpha)\lambda}{\|ST(\mathbf{\Phi}^j - h(\mathbf{\Phi}^j \mathbf{Z}_q - \mathbf{R}_{-q})\mathbf{Z}_q^\top, h\alpha\lambda)\|_F}\right)_+ ST(\mathrm{vec}(\mathbf{\Phi}^j) - h\mathrm{vec}(\boldsymbol{F}_q), h\alpha\lambda)$$

$\mathbf{\Phi}^{j+1} \leftarrow \gamma^{j+1} + \frac{j}{j+3}(\gamma^{j+1} - \gamma^j)$

$j \leftarrow j + 1$

until Desired threshold is reached

Algorithm 7: Endogenous-First VARX-L Proximal Problem

Require: $\tilde{v}, \lambda, k, p, m, s$

for i=1,. . . ,p **do**

$g_1 \leftarrow [((i-1)\cdot k+1) : ((i-1)\cdot k+k)]$

$g_2 \leftarrow [((i-1)\cdot m+p\cdot k) : ((i-1)\cdot m+p\cdot k+m)]$

$v_{\{g_1,g_2\}} \leftarrow \text{Prox}(v_{\{g_1,g_2\}}, \lambda, k)$

end for

return v.

procedure PROX(v,λ,k)

$h_2 \leftarrow (k+1) : (k+m)$

$h_1 \leftarrow 1 : (k+m)$

for $j = 1, 2$ **do**

$v_{h_j} \leftarrow (1 - \lambda/\|v_{h_j}\|_F)_+ v_{h_j}$

end for

return v.

end procedure

APPENDIX B

APPENDIX TO CHAPTER 2

B.0.7 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 (2005)

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

For $\mathbf{A}$ to be stationary, its maximum eigenvalue must be less than 1. In general, it is very difficult to generate stationary coefficient matrices. Boshnakov and Iqelan (2009) offers a potentially viable procedure that utilizes the unique structure of B.1, 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.

B.0.8 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, 2007a). 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 $\widehat{\boldsymbol{B}}$ denote the coefficient matrix recovered from one of our algorithms 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:

$$\text{vec}(\widehat{\mathbf{\Phi}}_{\text{Relaxed}}) = \mathbf{V}[\mathbf{V}^\top(\boldsymbol{Z}\boldsymbol{Z}^\top \otimes \boldsymbol{I}_k)\mathbf{V}]^{-1}\mathbf{V}^\top(\boldsymbol{Z} \otimes \boldsymbol{I}_k)\text{vec}(\boldsymbol{Y}), \tag{B.2}$$

in which $\otimes$ denotes the Kronecker operator. In general, it is ill-advised to directly form B.2. 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 B.2 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.

B.0.9 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

$$\widehat{\mathbf{\Phi}}_{\text{Relaxed}_i} = (V_i(V_i^\top \boldsymbol{Z}\boldsymbol{Z}^\top V_i)^{-1} V_i^\top \boldsymbol{Z}\boldsymbol{Y}_i)^\top. \tag{B.3}$$

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

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

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 $(r_i^2 + r_i + 1)\epsilon_{\text{machine}}$. The full refitting algorithm is detailed in Algorithm 8.

Algorithm 8: Relaxed Least Squares

Require: $\boldsymbol{Z}, \boldsymbol{Y}, V_1, \ldots, V_k$

for $i = 1, 2, \ldots, k$ **do**

$\mathbf{K}_i \leftarrow [(V_i \boldsymbol{Z})^\top, Y_i]$

$D \leftarrow (r_i^2 + r_i + 1)\epsilon_{\text{machine}} \text{diag}(\|\mathbf{K}_{i\cdot}\|_2)$

$R, Q \leftarrow QR(\begin{bmatrix} \mathbf{K}_i \\ D \end{bmatrix})$

$\widehat{\mathbf{\Phi}}_{\text{Relaxed}_i} \leftarrow (V_i (R_{11}^{-1} R_{12})^\top)^\top$

end for

return $\widehat{\mathbf{\Phi}}_{\text{Relaxed}}$.

APPENDIX C

APPENDIX TO CHAPTER 3

C.0.10 Notation

When detailing our algorithms, we find it convenient to express the VARX in compact matrix notation.

$$\begin{aligned} \boldsymbol{Y} &= [\mathbf{y}_1, \ldots, \mathbf{y}_T]; & \mathbf{X} &= [\mathbf{x}_1, \ldots, \mathbf{x}_T] \\ \boldsymbol{Z}_t &= [1, \mathbf{y}_t^\top, \ldots, \mathbf{y}_{t-p}^\top, \mathbf{x}_t^\top, \ldots, \mathbf{x}_{t-s}^\top] & \boldsymbol{Z} &= [\boldsymbol{Z}_2; \ldots; \boldsymbol{Z}_{T-1}] \\ \boldsymbol{\Phi} &= [\boldsymbol{\Phi}^{(1)}, \boldsymbol{\Phi}^{(2)}, \ldots, \boldsymbol{\Phi}^{(p)}] & \boldsymbol{\beta} &= [\boldsymbol{\beta}^{(1)}, \ldots, \boldsymbol{\beta}^{(s)}] \\ \boldsymbol{B} &= [\nu, \boldsymbol{\Phi}, \boldsymbol{\beta}] & \mathbf{U} &= [\mathbf{u}_1, \ldots, \mathbf{u}_T] \end{aligned} .$$

We can then express the VARX as

$$\boldsymbol{Y} = \boldsymbol{B}\boldsymbol{Z} + \mathbf{U},$$

in which $\mathbf{U} \overset{\text{iid}}{\sim} (0, I_T \otimes \Sigma_u)$.

C.0.11 Computing Information Criterion Based Benchmarks

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

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

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

$$R = \begin{array}{c} \begin{matrix} kp + ms + 1 & k \end{matrix} \\ \begin{bmatrix} R_{11} & R_{12} \\ 0 & R_{22} \end{bmatrix} \begin{matrix} kp + ms + 1 \\ k \end{matrix} \end{array}$$

Then, we can compute the least squares estimate $\widehat{\boldsymbol{B}}$ as

$$\begin{aligned} \widehat{\boldsymbol{B}} &= (R_{12}^\top R_{11} (R_{11}^\top R_{11})^{-1})^\top, \\ &= (R_{12}^\top R_{11} R_{11}^{-1} (R_{11}^\top)^{-1})^\top, \\ &= (R_{12}^\top (R_{11}^\top)^{-1})^\top, \\ &= (R_{11}^{-1} R_{12})^\top, \end{aligned}$$

which can be evaluated with a triangular solver, hence does not require explicit matrix inversion. We can then obtain the residual covariance $\widehat{\Sigma}_u$ as:

$$\frac{R_{22}^\top R_{22}}{T}$$

Our implementation of this procedure in the context of VAR and VARX lag order selection is described in Algorithm 11 in Section C.0.17.

C.0.12 Generating Impulse Response Functions

In order to perform impulse response analysis, the system needs to be identified, hence we need to convert the VAR to a moving average representation. Following

Lütkepohl (2005) and Lin (2006), we can convert a VAR(p) process to MA form as follows. First convert the VAR(p) to VAR(1), as in Equation (B.1).

$$\boldsymbol{Y}_t = \boldsymbol{\nu} + \mathbf{A}\boldsymbol{Y}_{t-1} + \mathbf{U}_t.$$

Then, (assuming the coefficient matrix generates a stationary process), it can be represented as

$$\boldsymbol{Y}_t = \boldsymbol{\mu} + \sum_{i=0}^{\infty} \mathbf{A}^i U_{t-i}, \tag{C.1}$$

We then obtain the MA representation by pre-multiplying both sides of Equation (C.1) by the $k \times kp$ matrix J = $[I_k, \mathbf{0}, \mathbf{0}]$, resulting in

$$J\boldsymbol{Y}_t = J\boldsymbol{\nu} + \sum_{i=0}^{\infty} J\mathbf{A}^i J^\top J U_{t-i},$$

$$\mathbf{y}_t = \boldsymbol{\nu} + \sum_{i=0}^{\infty} \Gamma_i \mathbf{u}_{t-i},$$

in which Γ_i is the MA coefficient matrix (constructed by $J\mathbf{A}^i J^\top$) measuring the impulse response. Since, the covariance of $\mathbf{U}_t$ is not diagonal, we cannot perform impulse response without a factorization. Traditionally, the Cholesky Decomposition is used to factor $\Sigma_u = CC^\top$, where C is a lower triangular matrix. After this factorization, the MA coefficients can be expressed as:

$$\boldsymbol{Y}_t = \sum_{i=0}^{\infty} \Theta_i w_{t-i}, \tag{C.2}$$

in which $\Theta_i = \Gamma_i C$, $w_t = C^{-1}U_t$, $E(w_t w_t^\top) = I_k$. Then, with D representing the diagonal of the Cholesky factor P and defining $W = CD^{-1}, \Sigma_w = DD^\top$, it is possible to use Equation (C.2) to model a response function to trace the effect of a shock over n periods by examining Θ_i for $i = 1, \ldots, n$.

C.0.13 Relaxed (Group) Lasso-VAR

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, 2007b), (Belloni and Chernozhukov, 2009)). This approach has been extended to the VAR setting by Song and Bickel (2011), who briefly remark that they use their group lasso models for variable selection and refit based on least squares.

Refitting with least squares in the penalized VAR setting is inefficient and completely ignores the VAR's structure. As demonstrated in Zellner (1962), in the absence of parameter restrictions, the ordinary and generalized least squares estimators (GLS) coincide in the VAR framework. However, once restrictions are introduced, the generalized least squares estimator is asymptotically more efficient than ordinary least squares.

An estimation procedure that can take into account linear restrictions (such as fixing some parameters at zero) is referred to in the time series literature as a "Restricted VAR," and was explored in the context of constrained likelihood Lasso-VAR estimation by Davis et al. (2012). As we use this method to re-estimate nonzero coefficients, to avoid confusion we will refer to this two-step estimation procedure as a "Relaxed Basic VAR-L."

Notation

Let $\widehat{\mathbf{\Phi}}$ denote the coefficient matrix obtained from a structured regularization procedure (e.g. a Basic VAR-L), that returned r nonzero coefficients. The selected coefficients can be expressed as linear constraints of the form

$$\text{vec}(\widehat{\mathbf{\Phi}}) = \mathbf{R}\hat{\phi}, \tag{C.3}$$

in which $\mathbf{R}$ is a $(k^2p + k) \times r$ selection matrix of rank r consisting of columns from an identity matrix of dimension $k^2p + k$, and $\hat{\phi} = \text{vec}(\{\widehat{\mathbf{\Phi}} : \widehat{\mathbf{\Phi}}_{jk} \neq 0\})$. Within the relaxed framework, λ is held constant and the support recovered is taken as given. Following Brüggemann (2004), we can express the GLS estimator of the Relaxed VAR as

$$\text{vec}(\widehat{\mathbf{\Phi}}^{GLS}) = \mathbf{R}[\mathbf{R}^\top(\boldsymbol{Z}\boldsymbol{Z}^\top \otimes \Sigma_u^{-1})\mathbf{R}]^{-1}\mathbf{R}^\top(\boldsymbol{Z} \otimes \Sigma_u^{-1})\text{vec}(\boldsymbol{Y}), \tag{C.4}$$

in which $\otimes$ denotes the Kronecker product. However, since Σ_u is unknown in general, Equation (C.4) cannot be used in practice. The two step procedure to construct a "feasible" GLS estimator starts by calculating the Relaxed Least Squares (RLS) estimator

$$\text{vec}(\widehat{\mathbf{\Phi}}^{\text{Rlx}}) = \mathbf{R}[\mathbf{R}^\top(\boldsymbol{Z}\boldsymbol{Z}^\top \otimes I_k)\mathbf{R}]^{-1}\mathbf{R}^\top(\boldsymbol{Z} \otimes I_k)\text{vec}(\boldsymbol{Y}). \tag{C.5}$$

The RLS estimator is then used to estimate Σ_u. If estimating Σ_u is not tractable, which can occur when the series length T is small relative to the number of component series k, $\hat{\mathbf{\Phi}}^{\text{Rlx}}$ can be used to return "unshrunk" parameter estimates under

the assumption that Σ_u is the identity matrix. Otherwise Σ_u can be estimated by

$$\widehat{\Sigma}_u = \frac{1}{T - (p \times k)}(\boldsymbol{Y} - \widehat{\boldsymbol{\Phi}}^{\text{Rlx}}\boldsymbol{Z})(\boldsymbol{Y} - \widehat{\boldsymbol{\Phi}}^{\text{Rlx}}\boldsymbol{Z})^\top \tag{C.6}$$

Then, assuming $\widehat{\Sigma}_u$ is non-singular, the feasible GLS estimator can be expressed as

$$\text{vec}(\widehat{\boldsymbol{\Phi}}^{FGLS}) = \mathbf{R}[\mathbf{R}^\top(\boldsymbol{Z}\boldsymbol{Z}^\top \otimes \widehat{\Sigma}_u^{-1})\mathbf{R}]^{-1}\mathbf{R}^\top(\boldsymbol{Z} \otimes \widehat{\Sigma}_u^{-1})\text{vec}(\boldsymbol{Y}). \tag{C.7}$$

Due to the poor numerical properties detailed by Foschi and Kontoghiorghes (2003), it is inadvisible to form (C.6) or (C.7) directly. In addition, our applications have found $\boldsymbol{Z}\boldsymbol{Z}^\top$ to be poorly conditioned when T is small. Moreover, as the dimension increases, conducting operations directly with the $(k^2p + k) \times (k^2p + k)$ matrix $(\boldsymbol{Z}\boldsymbol{Z}^\top \otimes I_k)$ exhausts memory.

To ameliorate these issues of dimensionality, the refitting procedure can be conducted in parallel across rows of $\boldsymbol{B}$ if the covariance matrix is assumed to be the identity. Additionally, the conditioning of $\boldsymbol{Z}\boldsymbol{Z}^\top$ can be improved by implementing a modification of the procedure developed by Neumaier and Schneider (2001) (discussed in Section C.0.11), which adds a small (on the order of $\epsilon_{\text{machine}}$) regularization penalty to $\boldsymbol{Z}$ and $\boldsymbol{Y}$ and computes (C.5) via a QR factorization that does not require explicit matrix inversion (for details, see the Appendix of Nicholson et al. (2016b)). However, this approach cannot be extended to incorporate a non-identity covariance. The following sections detail an iterative procedure that constructs the feasible GLS estimator (C.7) without explicit matrix inversion.

C.0.14 Generalized Least Squares

Consider the conventional least squares problem with design matrix $X \in \mathbb{R}^{m \times n}$ ($n < m$) of full rank, response vector $y \in \mathbb{R}^m$, and vector of unknown coefficients $\beta \in \mathbb{R}^n$.

$$y = X\beta + \epsilon$$
$$\epsilon \overset{\text{iid}}{\sim} N(0, \Sigma)$$

Generalized Least Squares (see e.g. Björck (1996)) incorporates an $m \times m$ semidefinite symmetric matrix Σ, leading to the optimization problem

$$\min_{\beta} \|C^{-1}(X\beta - y)\|_2,$$

in which $\Sigma = CC^\top$. The normal equations for the GLS problem take the form

$$X^\top \Sigma^{-1} X\beta = X^\top \Sigma^{-1} y, \tag{C.8}$$

Equation (C.8) can be unstable if X or Σ are ill-conditioned. Instead of solving (C.8) directly, Foschi et al. (2004) recommend formulating the system as the generalized linear least squares problem (GLLSP)

$$\underset{v,\beta}{\operatorname{argmin}}\, v^\top v \text{ s.t. } y = X\beta + Cv, \tag{C.9}$$

in which $v \in \mathbb{R}^m, Cv = \epsilon$. This problem can be solved with a generalized QR factorization (Paige, 1979) which does not require explicit matrix inversion. The

generalized QR factorization involves the QR factorization of X

$$Q^\top X = \begin{bmatrix} R \\ 0 \end{bmatrix} \begin{matrix} n \\ m-n, \end{matrix} \tag{C.10}$$

$$\tag{C.11}$$

in which $Q \in \mathbb{R}^{m \times m}$, is an orthogonal matrix and $R \in \mathbb{R}^{n \times n}$ is upper triangular, and the product RQ[1] decomposition of $Q^\top C$ The product RQ decomposition takes the form

$$(Q^\top C)P = U,$$

$$U = \begin{matrix} n & m-n \end{matrix} \\ \begin{bmatrix} U_{11} & U_{12} \\ 0 & U_{22} \end{bmatrix} \begin{matrix} n \\ m-n \end{matrix},$$

in which $P \in \mathbb{R}^{m \times m}$ is an orthogonal matrix and $U \in \mathbb{R}^{m \times m}$ is upper triangular. Since P is orthogonal, $\|v\|_2 = \|P^\top v\|_2$, so Equation (C.9) can be reformulated as:

$$\min_{v,\beta} \|P^\top v\|_2^2 \text{ s.t. } Q^\top y = Q^\top X\beta + Q^\top CPP^\top v,$$

$$\iff \min_{v,\beta} \|P^\top v\|_2^2 \text{ s.t. } Q^\top y = Q^\top QR\beta + UP^\top v$$

[1]The RQ decomposition of X can be computed from the QR decomposition of $X^\top$ with the rows reversed. The resulting R from the QR decomposition then needs to be transposed with its rows and columns reversed

next, define $P^\top v = (v_1^\top, v_2^\top)$, in which v_1 has length n and v_2 has length m-n. Then, we can express the GLLSP as

$$\underset{v_1, v_2, \beta}{\operatorname{argmin}} \|v_1\|_2^2 + \|v_2\|_2^2 \text{ subject to} \tag{C.12}$$

$$y_1 = R\beta + U_{11}v_1 + U_{12}v_2, \tag{C.13}$$

$$y_2 = U_{22}v_2, \tag{C.14}$$

in which

$$Q^\top y = \begin{bmatrix} y_1 \\ \\ y_2 \end{bmatrix} \begin{matrix} n \\ \\ m-n \end{matrix} \tag{C.15}$$

Paige (1979) notes that since R is of full rank, Equation (C.13) can always be solved for β once v_1 and v_2 are given, hence Equation (C.14) gives the constraints on v, reducing the problem to

$$\underset{v}{\operatorname{argmin}} \|v_1\|_2^2 + \|v_2\|_2^2$$

$$\text{subject to}$$

$$y_2 = U_{22}v_2.$$

Now, since v_1 no longer appears in the constraints, we set it to zero and calculate $\hat{\beta}_{\text{FGLS}}$ by solving the triangular systems

$$R\beta = y_1 - U_{12}v_2,$$

$$U_{22}v_2 = y_2,$$

for v_2 and β. We can then estimate $\widehat{\Sigma}$ as $v_2^\top v_2/(m-n)$.

C.0.15 Application to Relaxed Feasible Generalized Least Squares

The previous approach can be modified to take into account the structure of the VAR. Foschi et al. (2002), develop an extension of this procedure in the context of seemingly unrelated regressions with common regressors and later extend it to the VAR in Foschi and Kontoghiorghes (2003). Their framework can easily be extended to our context of Relaxed VAR estimation. We start by formulating the Generalized Linear Least Squares Problem

$$\begin{aligned} &\underset{V}{\operatorname{argmin}} \|V\|_F \\ &\text{subject to } \operatorname{vec}(\boldsymbol{Y}^\top) = ((I_k \otimes \boldsymbol{Z}^\top)\mathbf{R})\phi_{\text{FGLS}} + \operatorname{vec}(VC^\top), \end{aligned}$$

in which $\mathbf{R}$ is the $(k^2p+k)\times r$ restriction matrix of $\mathbf{\Phi}^\top$ and $\operatorname{vec}(\widehat{\mathbf{\Phi}}^\top_{\text{FGLS}}) = \mathbf{R}\phi_{\text{FGLS}}$. Note that this application uses $\boldsymbol{Z}^\top$ because its corresponding Kronecker product produces a block diagonal structure. For notational ease, we will define $\mathbf{X} = (I_k \otimes \boldsymbol{Z}^\top)\mathbf{R}$. Here, C is a lower triangular Cholesky factor such that $\Sigma_u = CC^\top$, and V is defined as a random matrix satisfying the relationship $(C \otimes I_T)\operatorname{vec}(V) = \operatorname{vec}(U)$.

First, note that we do not need to directly compute the QR factorization of $\boldsymbol{X}$. We can instead compute the QR factorization of $\boldsymbol{Z}^T R_i$ for each $i = 1, \ldots, k$, in which R_i is the $(kp + 1) \times r_i$ restriction matrix for series i (denoted by a row in $\mathbf{\Phi}$). The Q

corresponding to $\mathbf{X}$ can then be formed as:

$$\mathbf{Q} = \begin{pmatrix} \oplus_i \widetilde{Q}_i & \oplus_i \widehat{Q}_i \end{pmatrix} = \begin{pmatrix} \widetilde{Q}_1 & & \widehat{Q}_1 & \\ & \ddots & & \ddots \\ & & \widetilde{Q}_k & & \widehat{Q}_k, \end{pmatrix} \tag{C.16}$$

in which $\widetilde{Q}_i$ represents the "economy" Q for series i (of dimension $T \times r_i$) and $\widehat{Q}_i$ represents the orthogonal completion for series i (of dimension $T \times (kp + 1 - r_i)$). The upper triangular matrix can be constructed similarly (to avoid confusion, we denote it as $\mathcal{R}$).

Next, we construct

$$\begin{aligned} \text{vec}\,(\widetilde{\boldsymbol{Y}}) &= \widetilde{\mathbf{Q}}^\top \text{vec}\,(\boldsymbol{Y}) \\ \text{vec}\,(\widehat{\boldsymbol{Y}}) &= \widehat{\mathbf{Q}}^\top \text{vec}\,(\boldsymbol{Y}). \end{aligned}$$

Now, consider the generalized QR decomposition:

$$\mathbf{Q}^\top (C \otimes I_T)\mathbf{P} = \begin{array}{c} \begin{matrix} r & kT - r \end{matrix} \\ \begin{bmatrix} W_{11} & W_{12} \\ 0 & W_{22} \end{bmatrix} \begin{matrix} r \\ kT-r \end{matrix} \end{array}$$

Using the above decomposition, the GLLSP becomes

$$\underset{\widetilde{v},\widehat{v}}{\text{argmin}}\, \|\widetilde{v}\|_2 + \|\widehat{v}\|_2 \tag{C.17}$$

$$\text{subject to} \begin{pmatrix} \text{vec}(\widetilde{\boldsymbol{Y}}) \\ \text{vec}(\widehat{\boldsymbol{Y}}) \end{pmatrix} = \begin{pmatrix} \mathcal{R} \\ 0 \end{pmatrix} \phi_{\text{FGLS}} + \begin{pmatrix} W_{11} & W_{12} \\ 0 & W_{22} \end{pmatrix} \begin{pmatrix} \text{vec}(\widetilde{v}) \\ \text{vec}(\widehat{v}) \end{pmatrix} \tag{C.18}$$

in which

$$\mathbf{P}^\top \operatorname{vec}(V) = \begin{bmatrix} \operatorname{vec}(\tilde{v}) \\ \operatorname{vec}(\hat{v}) \end{bmatrix} \begin{matrix} r \\ kT-r \end{matrix}.$$

Since, as in the univariate generalized least squares scenario, the constraints for $\tilde{v}$ are always satisfied, we set $\tilde{v} = 0$ and solve the triangular system

$$W_{22} \operatorname{vec} \hat{v} = \operatorname{vec}(\hat{y}).$$

After doing so, we compute

$$\operatorname{vec}(v^*) = W_{12} \operatorname{vec}(\hat{v}).$$

Then, we solve the final triangular system

$$\mathcal{R}\phi_{\text{FGLS}} = \operatorname{vec}(\widetilde{\boldsymbol{Y}}) - \operatorname{vec}(v^*)$$

The RQ decomposition of $\mathbf{Q}^\top(C \otimes I_T)$ is the most computationally expensive component, requiring $O(k^3T^3)$ floating point operations. Foschi et al. (2002) conduct the RQ decomposition in two stages, first calculating

$$\mathbf{Q}^\top(C \otimes I_T)\mathbf{Q} = \begin{matrix} \\ r \\ kT-r \end{matrix} \begin{matrix} \begin{matrix} r & kT-r \end{matrix} \\ \begin{bmatrix} \widetilde{W}_{11} & \widetilde{W}_{12} \\ \widetilde{W}_{21} & \widetilde{W}_{22} \end{bmatrix} \end{matrix}. \tag{C.19}$$

in which each $\widetilde{W}_{ij}$ is block upper triangular. In the second stage, the RQ decompositions is computed

$$\begin{pmatrix} \widetilde{W}_{21} & \widetilde{W}_{22} \end{pmatrix} \widetilde{P} = \begin{pmatrix} 0 & W_{22} \end{pmatrix},$$

and W_{12} is constructed by using the previously calculated $\widetilde{P}$

$$\begin{pmatrix} \widetilde{W}_{11} & \widetilde{W}_{12} \end{pmatrix} \widetilde{\mathbf{P}} = \begin{pmatrix} W_{11} & W_{12} \end{pmatrix}.$$

Therefore, we can conclude that $\mathbf{P} = \mathbf{Q}\widetilde{\mathbf{P}}$. Foschi et al. (2002) details efficient algorithms which take advantage of the Kronecker structure when computing these factorizations. One can also avoid explicitly forming $\mathbf{Q}^\top (C \otimes I_T)\mathbf{Q}$ by separately constructing each submatrix in Equation C.19 (derivations are in Section C.0.16). This procedure is summarized in Algorithm 11 in Section C.0.17.

Updating $\widehat{\Sigma}_U$

As Σ_U is typically unknown, this approach is usually an iterative procedure. Initially, $\widehat{\Sigma}_U$ is set to I_k, and is updated based on the model residuals $\widetilde{U}$, which are calculated as follows

$$\begin{aligned} \text{vec}\,(\widetilde{U}) &= \widetilde{\boldsymbol{Y}} - (I_k \otimes R)\widehat{\phi}_{\text{FGLS}} \\ \widehat{\Sigma}_u^{(j+1)} &= \frac{\widetilde{U}^\top \widetilde{U} + \widehat{\boldsymbol{Y}}^\top \widehat{\boldsymbol{Y}}}{T - (p \times k)} \end{aligned}$$

One can update $\widehat{\Sigma}_u$ and $\widehat{\phi}_{\text{FGLS}}$ until some convergence criterion is satisfied; we currently use the matrix 2-norm (i.e. the maximum singular value of $\widehat{\Sigma}_u^{(j+1)} - \widehat{\Sigma}_u^{(j)}$). Note that $\widehat{\boldsymbol{Y}}$ does not change across iterations, so the only component that needs to be updated is $\widetilde{U}$.

A potential complication arises when $\widehat{\Sigma}_u^{j+1}$ is not positive definite. In this scenario, since it is not possible to take its Cholesky decomposition, the algorithm

will break down. Foschi et al. (2002) propose computing the Cholesky factor C directly from the QL decomposition of

$$\begin{bmatrix} \widetilde{U} \\ \widetilde{\boldsymbol{Y}} \end{bmatrix}.$$

However, this decomposition does not guarantee that the diagonal of C will be positive. As an alternative, in the rare instances when $\widehat{\Sigma}_u$ is not positive definite, we propose factoring $\widehat{\Sigma}_u$ using the singular value decomposition

$$\widehat{\Sigma}_u = UDU^\top$$
$$Q, C = \text{QR}(D^{1/2}U^\top),$$

i.e., C is recovered from the R in the QR decomposition of $D^{-1/2}U^\top$.

C.0.16 Additional Details

Construction of Submatrices in Equation C.19

Each of the 4 submatrices in Equation C.19 can be expressed in closed form; as combinations of the "economy" QR decompositions for each series ($\widetilde{Q}_i$) as well as its orthogonal completion ($\widehat{Q}_i$).

$$W_{11} \in \mathbb{R}^{r \times r} = \begin{bmatrix} C_{1,1} I_{r_1} & C_{2,1} \widetilde{Q}_2^\top \widetilde{Q}_1 & \dots & \dots & C_{k,1} \widetilde{Q}_k^\top \widetilde{Q}_1 \\ \mathbf{0} & C_{2,2} I_{r_2} & C_{3,2} \widetilde{Q}_3^\top \widetilde{Q}_2 & \dots & C_{k,2} \widetilde{Q}_k^\top \widetilde{Q}_2 \\ \vdots & \ddots & \ddots & \ddots & \ddots \\ \mathbf{0} & \mathbf{0} & \dots & C_{kk} I_{r_k} & \mathbf{0} \end{bmatrix}$$

$$W_{22} \in \mathbb{R}^{kT-r \times kT-r} = \begin{bmatrix} C_{1,1} I_{T-r_1} & C_{2,1} \widehat{Q}_2^\top \widehat{Q}_1 & \dots & \dots & C_{1,k} \widehat{Q}_k^\top \widehat{Q}_1 \\ \mathbf{0} & C_{2,2} I_{T-r_2} & C_{2,3} \widehat{Q}_3^\top \widehat{Q}_2 & \dots & C_{2,k} \widehat{Q}_k^\top \widehat{Q}_2 \\ \vdots & \ddots & \ddots & \ddots & \ddots \\ \mathbf{0} & \mathbf{0} & \dots & C_{kk} I_{T-r_k} & \mathbf{0} \end{bmatrix}$$

$$W_{12} \in \mathbb{R}^{r \times kT-r} = \begin{bmatrix} \mathbf{0}_{r_1} & C_{2,1} \widetilde{Q}_1^\top \widehat{Q}_2 & \dots & \dots & C_{1,k} \widetilde{Q}_1^\top \widehat{Q}_1 \\ \mathbf{0} & \mathbf{0}_{r_2} & C_{2,3} \widetilde{Q}_2^\top \widehat{Q}_3 & \dots & C_{2,k} \widetilde{Q}_2^\top \widehat{Q}_k \\ \vdots & \ddots & \ddots & \ddots & \ddots \\ \mathbf{0} & \mathbf{0} & \dots & \dots & \mathbf{0} \end{bmatrix}$$

$$W_{21} \in \mathbb{R}^{kT-r \times r} = \begin{bmatrix} \mathbf{0}_{r_1} & C_{2,1} \widetilde{Q}_2^\top \widehat{Q}_1 & \dots & \dots & C_{1,k} \widetilde{Q}_k^\top \widehat{Q}_1 \\ \mathbf{0} & \mathbf{0}_{r_2} & C_{2,3} \widetilde{Q}_3^\top \widehat{Q}_2 & \dots & C_{2,k} \widetilde{Q}_k^\top \widehat{Q}_2 \\ \vdots & \ddots & \ddots & \ddots & \ddots \\ \mathbf{0} & \mathbf{0} & \dots & \dots & \mathbf{0} \end{bmatrix}$$

in which r_i denotes the number of active coefficients for series i.

$$\mathbf{Q}^\top (C \otimes I_T) \mathbf{Q} = \begin{array}{c} \begin{array}{cc} r & kT-r \end{array} \\ \left[\begin{array}{cc} \widetilde{W}_{11} & \widetilde{W}_{12} \\ \widetilde{W}_{21} & \widetilde{W}_{22} \end{array} \right] \begin{array}{c} r \\ kT-r \end{array} \end{array}.$$

C.0.17 Tables and Algorithms

Table C.1: Arguments for `struct` in `constructModel`. X denotes "True" while . denotes "False."

Struct Argument	Penalty	VAR	VARX	Univariate
"Lag"	Lag Group	X	X	X
"OwnOther"	Own/Other Group	X	X	.
"SparseLag"	Lag Sparse Group	X	X	.
"SparseOO"	O/O Sparse Group	X	X	.
"Basic"	Basic	X	X	X
"EF"	Endogenous-First	.	X	.
"HVARC"	Componentwise Hierarchical	X	.	X
"HVAROO"	Own/Other Hierarchical	X	.	.
"HVARELEM"	Elementwise Hierarchical	X	.	.
"Tapered"	Lag Weighted Lasso	X	.	.

Table C.2: Solution Algorithms employed for each structured penalty

Algorithm	Solution Procedure	Reference
Lag	Block Coordinate Descent	Qin et al. (2010)
Own/Other	Block Coordinate Descent	.
Lag Sparse	Proximal Gradient Descent	Beck and Teboulle (2009)
Own/Other Sparse	Proximal Gradient Descent	.
Basic	Coordinate Descent	Friedman et al. (2010)
Endogenous-First	Fast Iterative Soft Thresholding	Jenatton et al. (2011)

Algorithm 9: Iterative procedure to determine $\lambda_{\max}$

Require: $\boldsymbol{Y}, \boldsymbol{Z}, \boldsymbol{B}, \lambda_{\max \text{ coarse}}, \epsilon$

$\lambda_{\text{HIGH}} \leftarrow \lambda_{\max \text{ coarse}}$

$\lambda_{\text{LOW}} \leftarrow 0$

while $\lambda_{\text{HIGH}} - \lambda_{\text{LOW}} > \epsilon$ **do**

$\lambda \leftarrow \frac{\lambda_{\text{HIGH}} + \lambda_{\text{LOW}}}{2}$

$\boldsymbol{B} \leftarrow$ BigVAR Model$(\boldsymbol{Y}, \boldsymbol{Z}, \boldsymbol{B}, \lambda)$

if $\|B\|_{\infty} = 0$ **then**

$\lambda_{\text{HIGH}} \leftarrow \lambda$

else

$\lambda_{\text{LOW}} \leftarrow \lambda$

end if

end whilereturn λ

Algorithm 10: Fit a VARX according to information criterion minimization

Require: $\boldsymbol{Y}, \boldsymbol{Z}, \boldsymbol{B}, p, s,$ criterion

for $i = 0, \ldots, p$ **do**

for $j = 0, \ldots, s$ **do**

if $i > 0$ & $j > 0$ **then**

$K = [\boldsymbol{Z}^{\top}_{:,\left(1:(ki+1),(kp+2):((kp+2):js)\right)}, \boldsymbol{Y}^{\top}]$

else if $i = 0$ & $j > 0$ **then**

$K = [\boldsymbol{Z}^{\top}_{:,(kp+2):(kp+2+js)}, \boldsymbol{Y}^{\top}]$

else if $i > 0$ & $j = 0$ **then**

$K = [\boldsymbol{Z}^{\top}_{:,(1:(ki+1))}, \boldsymbol{Y}^{\top}]$

else if $i = 0$ & $j = 0$ **then**

$K = [\boldsymbol{1}]$

end if

$\widehat{\Sigma_u} \leftarrow$ VARXFit(K)
if criterion="AIC" **then**
$IC[i, j] \leftarrow |\widehat{\Sigma_u}| + \frac{2(k(ki+mj+1))}{T-\max(i,j)}$
else if criterion="BIC" **then**
$IC[i, j] \leftarrow |\widehat{\Sigma_u}| + \frac{\log(T-\max(i,j))(k(ki+mj+1)}{T-\max(i,j)}$
end if
end for
end for
return $\hat{p}, \hat{s}$ as the minimum entry of $IC[i, j]$
procedure VARXFIT(K)
$Q, R \leftarrow QR(K)$
$R_{11} = R_{1:(kp+ms+1),1:(kp+ms+1)}$
$R_{12} = R_{1:(kp+ms+1),(kp+ms+1):(kp+ms+k+1)}$
$R_{22} = R_{(kp+ms+1):(kp+ms+k+1),(kp+ms+1):(kp+ms+k+1)}$
$\widehat{B} \leftarrow (R_{11}^{-1} R_{12})^\top$
$\widehat{\Sigma_u} \leftarrow \frac{R_{22}^\top R_{22}}{\text{nrow}(K)}$
return $\widehat{B}$,$\widehat{\Sigma_u}$
end procedure

Algorithm 11: Iterated Feasible GLS (adapted from Algorithm 21.1 of Foschi et al. (2002))

Require: $\boldsymbol{Y}, \boldsymbol{X}, \boldsymbol{B}, p, k, \Sigma_u^{(0)}, \epsilon_1, \epsilon_2$

$\mathbf{R} \leftarrow \text{RestrictionMatrix}(\boldsymbol{B}^\top, \epsilon_1)$ ▷ selects coefficients greater in magnitude than ϵ_1.

$\mathbf{Q} \leftarrow$ determined according to Equation(C.16)

$$\mathcal{R} \leftarrow \begin{pmatrix} \mathcal{R}_1 & & \\ & \ddots & \\ & & \mathcal{R}_k, \end{pmatrix}$$

$\text{vec}(\widetilde{\boldsymbol{Y}}) = \widetilde{\mathbf{Q}}^\top \boldsymbol{Y}$

5: $\text{vec}(\widehat{\boldsymbol{Y}}) = \widehat{\mathbf{Q}}^\top \boldsymbol{Y}$

$C \leftarrow \text{Cholesky}(\Sigma_u^{(0)})$

$j \leftarrow 1$

while threshold> ϵ_2 **do**

10: $$\mathbf{Q}^\top (C \otimes I_T)\mathbf{Q} = \begin{array}{c} \\ r \\ kT-r \end{array} \begin{array}{c} \begin{array}{cc} r & kT-r \end{array} \\ \begin{bmatrix} \widetilde{W}_{11} & \widetilde{W}_{12} \\ \widetilde{W}_{21} & \widetilde{W}_{22} \end{bmatrix} \end{array}.$$

$$\begin{pmatrix} \widetilde{W}_{21} & \widetilde{W}_{22} \end{pmatrix} \widetilde{\mathbf{P}} = \begin{pmatrix} 0 & W_{22} \end{pmatrix}$$
$$\begin{pmatrix} \widetilde{W}_{11} & \widetilde{W}_{12} \end{pmatrix} \widetilde{\mathbf{P}} = \begin{pmatrix} W_{11} & W_{12} \end{pmatrix}.$$ ▷ RQ Decomposition

Triangular Solve (for v) $\quad W_{22}\text{vec}(v) = \text{vec}(\widehat{\boldsymbol{Y}})$

$v^* = W_{12}\text{vec}(v)$

Triangular Solve $\quad \mathcal{R}\text{vec}(\widehat{\phi}^{IFGLS}) = \text{vec}(\widetilde{\boldsymbol{Y}} - v)$

$\text{vec}(\widetilde{U}) \leftarrow \widetilde{\boldsymbol{Y}} - (I_k \otimes \mathcal{R}_1)\text{vec}(\widehat{\phi}^{IFGLS})$

$\Sigma_u^{(j)} \leftarrow \frac{\widetilde{U}^\top \widetilde{U} + \widehat{\boldsymbol{Y}}^\top \widehat{\boldsymbol{Y}}}{T-(p\times k)}$

$\epsilon_2 \leftarrow \|\Sigma_u^{(j)} - \Sigma_u^{(j-1)}\|_2$ ▷ Operator Norm

$C \leftarrow \text{Cholesky}(\Sigma_u^{(j)})$

end while

$\widehat{\boldsymbol{\Phi}}^{(IFGLS)} \leftarrow \mathbf{R}\text{vec}(\widehat{\phi}^{IFGLS})$

return $\widehat{\boldsymbol{\Phi}}^{(IFGLS)}, \Sigma_u^{(j)}$

BIBLIOGRAPHY

Akaike, H. (1969). Fitting autoregressive models for prediction. *Annals of the institute of Statistical Mathematics*, 21(1):243–247.

Bach, F., Jenatton, R., Mairal, J., and Obozinski, G. (2011). Convex optimization with sparsity-inducing norms. *Optimization for Machine Learning*, pages 19–53.

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

Back, A. D. and Weigend, A. S. (1997). A First Application of Independent Component Analysis to Extracting Structure from Stock Returns. *International Journal of Neural Systems*, 8:473–484.

Banbura, M., Giannone, D., and Reichlin, L. (2009). Large bayesian vector auto regressions. *Journal of Applied Econometrics*, 25(1):71–92.

Basu, S. and Michailidis, G. (2013). Estimation in high-dimensional vector autoregressive models. *arXiv preprint arXiv:1311.4175*.

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

Basu, S., Shojaie, A., and Michailidis, G. (2012). Network granger causality with inherent grouping structure. *arXiv preprint arXiv:1210.3711*.

Bates, D., Francois, R., and Eddelbuettel, D. (2012). Rcppeigen: Rcpp integration for the eigen templated linear algebra library. *R package version 0.3*, 1.

Beck, A. and Teboulle, M. (2009). A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202.

Becker, S. R., Candès, E. J., and Grant, M. C. (2011). Templates for convex cone problems with applications to sparse signal recovery. *Mathematical Programming Computation*, 3(3):165–218.

Belloni, A. and Chernozhukov, V. (2009). Least squares after model selection in high-dimensional sparse models.

Belloni, A., Chernozhukov, V., and Hansen, C. (2011). Inference for high-dimensional sparse econometric models. *arXiv preprint arXiv:1201.0220*.

Bernanke, B. S. and Blinder, A. S. (1992). The federal funds rate and the channels of monetary transmission. *The American Economic Review*, pages 901–921.

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

Bien, J., Bunea, F., and Xiao, L. (2014). Convex banding of the covariance matrix. *arXiv preprint arXiv:1405.6210*.

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

Björck, A. (1996). *Numerical methods for least squares problems*. Siam.

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

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

Brooks, C. and Tsolacos, S. (2000). Forecasting models of retail rents. *Environment and Planning A*, 32(10):1825–1840.

Brüggemann, R. (2004). *Model reduction methods for vector autoregressive processes*, volume 536. Springer Verlag.

Carriero, A., Kapetanios, G., and Marcellino, M. (2009). Forecasting exchange rates with a large bayesian var. *International Journal of Forecasting*, 25(2):400–417.

Chiuso, A. and Pillonetto, G. (2010). Nonparametric sparse estimators for identification of large scale linear systems. In *Decision and Control (CDC), 2010 49th IEEE Conference on*, pages 2942–2947. IEEE.

Clark, T. E. and McCracken, M. W. (2013). Evaluating the accuracy of forecasts from vector autoregressions. *VAR Models in Macroeconomics–New Developments and Applications: Essays in Honor of Christopher A. Sims (Advances in Econometrics, Volume 32) Emerald Group Publishing Limited*, 32:117–168.

Cushman, D. O. and Zha, T. (1997). Identifying monetary policy in a small open economy under flexible exchange rates. *Journal of Monetary economics*, 39(3):433–448.

Davis, R. A., Zang, P., and Zheng, T. (2012). Sparse vector autoregressive modeling. journal: arXiv preprint arXiv:1207.0520.

Diebold, F. X. (1998). The past, present, and future of macroeconomic forecasting. *The Journal of Economic Perspectives*, 12(2):175–192.

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

Doan, T., Litterman, R., and Sims, C. (1984). Forecasting and conditional projection using realistic prior distributions. *Econometric reviews*, 3(1):1–100.

Eddelbuettel, D. and François, R. (2011). Rcpp: Seamless R and C++ integration. *Journal of Statistical Software*, 40(8):1–18.

Eddelbuettel, D. and Sanderson, C. (2014). Rcpparmadillo: Accelerating r with high-performance c++ linear algebra. *Computational Statistics and Data Analysis*, 71:1054–1063.

Forni, M., Hallin, M., Lippi, M., and Reichlin, L. (2000). The Generalized Dynamic-Factor Model: Identification and Estimation. *Review of Economics and Statistics*, 82(4):540–554.

Foschi, P., Garin, L., and Kontoghiorghes, E. J. (2002). Numerical and computational strategies for solving seemingly unrelated regression models. In *Computational Methods in Decision-Making, Economics and Finance*, pages 405–427. Springer.

Foschi, P. and Kontoghiorghes, E. J. (2003). Estimation of var models computational aspects. *Computational Economics*, 21(1-2):3–22.

Foschi, P., Kontoghiorghes, E. J., and Nägeli, H.-H. (2004). Numerical methods for estimating linear econometric models.

Frenkel, R. and Rapetti, M. (2010). A concise history of exchange rate regimes in latin america.

Friedman, J., Hastie, T., and Tibshirani, R. (2009). glmnet: Lasso and elastic-net regularized generalized linear models. *R package version*, 1.

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

Furman, Y. (2014). Var estimation with the adaptive elastic net. *Available at SSRN 2456510*.

Gefang, D. (2012). Bayesian doubly adaptive elastic-net lasso for var shrinkage. journal: International Journal of Forecasting.

George, E. I., Sun, D., and Ni, S. (2008). Bayesian stochastic search for var model restrictions. *Journal of Econometrics*, 142(1):553–580.

Gilbert, P. (2005). Brief users guide: Dynamic systems estimation (dse). *Available in the file doc/dse-guide. pdf distributed together with the R bundle dse, to be downloaded from http://cran. r-project. org*.

Golub, G. H. and Van Loan, C. F. (2012). *Matrix computations*, volume 3. JHU Press.

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

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

Hansen, B. (2013). *Econometrics*. Manuscript.

Haris, A., Witten, D., and Simon, N. (2014). Convex modeling of interactions with strong heredity. *arXiv preprint arXiv:1410.3517*.

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

Hendry, D. F. and Hubrich, K. (2011). Combining disaggregate forecasts or combining disaggregate information to forecast an aggregate. *Journal of Business & Economic Statistics*, 29(2).

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

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

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

Ibarra, R. (2012). Do disaggregated cpi data improve the accuracy of inflation forecasts? *Economic Modelling*, 29(4):1305–1313.

Ip, G. (2008). Non-borrowed reserves: False alarm. *The Wall Street Journal*.

Jenatton, R., Mairal, J., Obozinski, G., and Bach, F. (2010). Proximal methods for sparse hierarchical dictionary learning. In *Proceedings of the International Conference on Machine Learning (ICML)*.

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

Kadiyala, K. and Karlsson, S. (1997). Numerical methods for estimation and inference in bayesian var-models. *Journal of Applied Econometrics*, 12(2):99–132.

Keating, J. W. (2001). Macroeconomic modeling with asymmetric vector autoregressions. *Journal of Macroeconomics*, 22(1):1–28.

Kennedy, P. (2003). *A guide to econometrics*. MIT press.

Klein, L. R. and Goldberger, A. S. (1955). An econometric model of the united states, 1929-1952.

Kock, A. B. and Callot, L. (2015). Oracle inequalities for high dimensional vector autoregressions. *Journal of Econometrics*, 186(2):325–344.

Koop, G. (2011). Forecasting with medium and large bayesian vars. *Journal of Applied Econometrics*.

Leahy, J. (2012). Brazil admits tight hold over exchange rate. *Financial Times*.

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

Li, J. and Chen, W. (2014). Forecasting macroeconomic time series: Lasso-based approaches and their forecast combinations with dynamic factor models. *International Journal of Forecasting*, 30(4):996–1015.

Lim, M. and Hastie, T. (2013). Learning interactions through hierarchical group-lasso regularization. *arXiv preprint arXiv:1308.2719*.

Lin, J.-L. (2006). Teaching notes on impulse response function and structural var. *Institute of Economics, Academia Sinica, Department of Economics, National Chengchi University*, pages 1–9.

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

Litterman, R. B. (1984). Forecasting and policy analysis with bayesian vector autoregression models. *Quarterly Review*, (Fall).

Litterman, R. B. (1986a). Forecasting with bayesian vector autoregressionsfive years of experience. *Journal of Business & Economic Statistics*, 4(1):25–38.

Litterman, R. B. (1986b). A statistical approach to economic forecasting. *Journal of Business & Economic Statistics*, 4(1):1–4.

Lou, Y., Bien, J., Caruana, R., and Gehrke, J. (2014a). Sparse partially linear additive models. *arXiv preprint arXiv:1407.4729*.

Lou, Y., Bien, J., Caruana, R., and Gehrke, J. (2014b). Sparse partially linear additive models. *arXiv preprint arXiv:1407.4729*.

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

Lütkepohl, H. (2005). New introduction to multiple time series analysis.

Lütkepohl, H. (2014). Structural vector autoregressive analysis in a data rich environment: A survey.

Marcellino, M., Stock, J. H., and Watson, M. W. (2006). A comparison of direct and iterated multistep ar methods for forecasting macroeconomic time series. *Journal of econometrics*, 135(1):499–526.

Marin, D. (1992). Is the export-led growth hypothesis valid for industrialized countries? *The Review of Economics and Statistics*, pages 678–688.

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

McElhattan, R. (1978). Estimating a stable-inflation capacity-utilization rate. *Economic Review*, Fall:20–30.

Meese, R. A. and Rogoff, K. (1983). Empirical exchange rate models of the seventies: Do they fit out of sample? *Journal of international economics*, 14(1):3–24.

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

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

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

Nicholson, W., Matteson, D., and Bien, J. (2016a). VARX-L: Structured Regularization for Large Vector Autoregression with Exogenous Variables. arXiv preprint arXiv:1508.07497.

Nicholson, W. B., Bien, J., and Matteson, D. S. (2016b). High dimensional forecasting via interpretable vector autoregression. *arXiv preprint arXiv:1412.5250*.

Nickelsburg, G. (1985). 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.

Nijs, V. R., Srinivasan, S., and Pauwels, K. (2007). Retail-price drivers and retailer profits. *Marketing Science*, 26(4):473–487.

Nocedal, J. and Wright, S. J. (1999). *Numerical optimization*, volume 2. Springer New York.

Ocampo, S. and Rodríguez, N. (2012). An introductory review of a structural var-x estimation and applications. *Revista Colombiana de Estadística*, 35(3):479–508.

Paige, C. (1979). Computer solution and perturbation analysis of generalized linear least squares problems. *Mathematics of Computation*, 33:171–183.

Peña, D. and Box, G. E. P. (1987). Identifying a Simplifying Structure in Time Series. *Journal of the American Statistical Association*, 82(399):836–843.

Penm, J. H., Penm, J. H., and Terrell, R. (1993). The recursive fitting of subset varx models. *Journal of Time Series Analysis*, 14(6):603–619.

Pfaff, B. (2008). *Analysis of integrated and cointegrated time series with R*. Springer.

Qin, Z., Scheinberg, K., and Goldfarb, D. (2010). Efficient block-coordinate descent algorithms for the group lasso. *Mathematical Programming Computation*, pages 1–27.

Racette, D. and Raynauld, J. (1992). Canadian monetary policy: will the checklist approach ever get us to price stability? *Canadian Journal of Economics*, pages 819–838.

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

Robertson, J. C. and Tallman, E. W. (1999). Improving forecasts of the federal funds rate in a policy model. Technical report, Federal Reserve Bank of Atlanta.

Roy, A., McElroy, T. S., and Linton, P. (2014). Estimation of causal invertible varma models. *arXiv preprint arXiv:1406.4584*.

She, Y. and Jiang, H. (2014). Group Regularized Estimation under Structural Hierarchy. *ArXiv e-prints*.

Simon, N., Friedman, J., Hastie, T., and Tibshirani, R. (2013). A sparse-group lasso. *Journal of Computational and Graphical Statistics*, 22(2):231–245.

Sims, C. A. (1980). Macroeconomics and reality. *Econometrica: Journal of the Econometric Society*, pages 1–48.

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

Stock, J. H. and Watson, M. W. (2002). Forecasting Using Principal Components from a Large Number of Predictors. *Journal of the American Statistical Association*, 97(460):1167–1179.

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

Suo, X. and Tibshirani, R. (2014). An ordered lasso and sparse time-lagged regression. *arXiv preprint arXiv:1405.6447*.

Taylor, J. B. (1993). Discretion versus policy rules in practice. In *Carnegie-Rochester conference series on public policy*, volume 39, pages 195–214. Elsevier.

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

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

Todd, R. M. (1990). Improving economic forecasting with bayesian vector autoregression. *Modelling economic series*, pages 214–34.

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

Tseng, P. (2001). Convergence of a block coordinate descent method for nondifferentiable minimization. *Journal of optimization theory and applications*, 109(3):475–494.

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

Welfe, W. (2013). Macroeconometric models of the united states and canada. In *Macroeconometric Models*, pages 15–46. Springer.

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

Wood, B. D. (2009). Presidential saber rattling and the economy. *American Journal of Political Science*, 53(3):695–709.

Yuan, M. and Lin, Y. (2006). Model selection and estimation in regression with

grouped variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(1):49–67.

Zellner, A. (1962). An efficient method of estimating seemingly unrelated regressions and tests for aggregation bias. *Journal of the American statistical Association*, 57(298):348–368.

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

Zou, H. and Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2):301–320.