Structured Regularization for Large Vector Autoregression

William B. Nicholson*, David S. Matteson†, and Jacob Bien‡

September 25, 2014

Abstract

The vector autoregression (VAR), has long proven to be an effective method for modeling the joint dynamics of macroeconomic time series as well as forecasting. One of the major disadvantages 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 low-frequency, high dimensional macroeconomic data. However, empirical evidence suggests that VARs which incorporate more component series tend to result in more accurate forecasts than their smaller counterparts. Existing methods which allow for the estimation of large VARs either tend to require *ad-hoc* specifications or are computationally intractable.

We adapt several prominent scalar regression regularization techniques to a vector time series context to greatly reduce the parameter space of VARs. We formulate convex optimization procedures that are amenable to efficient solutions for the time ordered high-dimensional problems we aim to solve. Through this framework, we propose a structured family of models and provide implementations which allow for both the efficient estimation and accurate forecasting of high-dimensional VARs. We demonstrate their efficacy in simulated data examples as well as an application to a large set of macroeconomic indicators.

*PhD Student, Department of Statistical Science, Cornell University, 301 Malott Hall, Ithaca, NY 14853 (E-mail: wbn8@cornell.edu; Webpage: `http://www.wbnicholson.com`)

†Assistant Professor, Department of Statistical Science and Department of Social Statistics, Cornell University, 1196 Comstock Hall, Ithaca, NY 14853, (E-mail: matteson@cornell.edu; Webpage: `https://courses.cit.cornell.edu/~dm484/`)

‡Assistant Professor, Department of Biological Statistics and Computational Biology and Department of Statistical Science, Cornell University, 1178 Comstock Hall, Ithaca, NY 14853 (E-mail: jbien@cornell.edu; Webpage: `http://faculty.bscb.cornell.edu/~bien/`)

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 expenditure, 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 ebbs and flows of contemporary macroeconomic theory. However, due to its heavy parameterization, the VAR quickly exhausts available degrees of freedom. It is ill-suited for high dimensional time series, effectively limiting applications to no more than 6 series (cf. Bernanke et al. [2005]), forcing the practictioner to specify *a priori* a reduced subset of series to include.

Almost since the VAR's inception, efforts have been made to reduce its parameterization. Early attempts, such as Litterman [1979] pursued a Bayesian approach underpinned by contemporary macroeconomic theory. In applying a Bayesian VAR with a Gaussian prior (analogous to ridge regression), 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 by a univariate random walk by shrinking model parameters 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 use of several unspecified hyperparmeters.

Modern Bayesian extensions originally proposed in Kadiyala and Karlsson [1997] and compiled by Koop [2011] allow 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] extends the Minnesota prior to a high dimensional setting with a closed-form posterior distribution. However, their approach does not perform variable selection, and their penalty parameter selection procedure appears 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. This includes the Lasso-VAR proposed by Hsu et al. [2008] and further explored in Song and Bickel [2011], Davis et al. [2012], and Medeiros and Mendes [2012]. Theoretical properties were investigated by Kock and Callot [2013] and Basu and Michailidis [2013]. The Lasso-VAR, which we introduce in Section 3, has several advantages over the Bayesian VAR as it is more computationally tractable in high dimensions, performs variable selection, and can readily compute multi-step forecasts and their associated prediction intervals.

This paper seeks to bridge the considerable gap between the regularization and macroeconomic forecasting communities. We propose numerous extensions and generalizations of the Lasso-VAR while incorporating the unique structure of the VAR model in a computationally efficient manner. Our methods: the Lasso-VAR, Lag Group Lasso-VAR, Own/Other Group Lasso-VAR, Lag Sparse Group Lasso-VAR and Own/Other Sparse Group Lasso-VAR, extend the Lasso and its structured counterparts to take into account characteristics such as a model's lag length and the delineation between a component's own lags and those of another component. These models offer great flexibility in capturing the true underlying dynamics of an economic system while imposing very mild restrictions on the parameters space. Moreover, unlike previous approaches, due to our adaptation of conventional 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 is close to or exceeds the length of the series.

In addition, unlike previous methods, our procedures can easily be applied by practitioners and avoids the use of subjective or complex hyperparameters. We also detail several extensions, including a procedure to refit a VAR based on the support estimated by our approaches (relaxed estimation), an illustration of implementing our algorithms to shrink toward a given reference (such as a vector random walk), and a framework for incorporating exogenous variables (regularized VARX). We present both a simulation study and a large macroeconomic data application to illustrate the superior forecast performance of the proposed methods over conventional VAR estimation methods.

Section 2 details the notation used throughout the paper and Section 3 introduces our structured regularization methodology. Section 4 proposes an approach for penalty parameter selection, Section 5 details relaxed estimation methods, Section 6 summarizes both a simulation study and a large macroeconomic data example, and Section 7 presents extensions which allow shrinkage to reference models and incorporating exogenous variables. The Appendix details our solution methods and algorithms.

2 Setup

Let $\{\boldsymbol{y_t} \in \mathbb{R}^k : t = 1, \ldots, T\}$ denote a k dimensional vector time series. A pth order vector autoregression $\text{VAR}_k(p)$ may be expressed as

$$\boldsymbol{Y}_t = \boldsymbol{\nu} + \sum_{\ell=1}^{p} \boldsymbol{B}_\ell \boldsymbol{Y}_{t-\ell} + \mathbf{u}_t, \tag{2.1}$$

in which $\boldsymbol{Y}_t, \boldsymbol{\nu}, \mathbf{u}_t \in \mathbb{R}^k$ for $t = 1, \ldots, T$, each $\boldsymbol{B}_\ell$ represents a coefficient matrix of dimension $k \times k$, and $\mathbf{u}_t \overset{\text{wn}}{\sim} (\mathbf{0}, \boldsymbol{\Sigma}_u)$ and p denotes the maximal lag length. The innovation covariance $\boldsymbol{\Sigma}_u$ is an unspecified finite $k \times k$ positive definite matrix. A VAR may be expressed as seemingly unrelated regressions, hence the least squares and generalized least squares estimators will coincide (cf. Zellner [1962]) in the absence of parameter restrictions. Consequently, we will not incorporate

Σ_u in the construction of our models. Basu and Michailidis [2013] details a penalized maximum likelihood approach which attempts to jointly estimate $\boldsymbol{B}_1, \ldots, \boldsymbol{B}_p$ and $\boldsymbol{\Sigma}_u^{-1}$, but doing so does not substantially improve forecasts.

It will also be convenient to express (2.1) in compact matrix notation. Define the $k \times T$ matrix $\boldsymbol{Y} = (\boldsymbol{Y}_1, \boldsymbol{Y}_2, \ldots, \boldsymbol{Y}_T)$. Let $\boldsymbol{B} = (\boldsymbol{B}_1, \boldsymbol{B}_2, \ldots, \boldsymbol{B}_p)$, which is of dimension $k \times kp$. Define a $kp \times 1$ vector $\boldsymbol{Z}_t$ as $\boldsymbol{Z}_t = (\boldsymbol{Y}_{t-1}^{'}, \ldots \boldsymbol{Y}_{t-p}^{'})^{'}$, and let $\boldsymbol{Z} = (\boldsymbol{Z}_1, \ldots, \boldsymbol{Z}_T)$, which has dimension $kp \times T$. Note that p realizations, $\boldsymbol{Y}_{-(p-1)}, \ldots, \boldsymbol{Y}_0$, are needed to initialize $\boldsymbol{Z}$. Finally, define a $k \times T$ matrix $\mathbf{U} = (\mathbf{u}_1, \ldots, \mathbf{u}_T)$, then Equation (2.1) may also be expressed as

$$\boldsymbol{Y} = \boldsymbol{\nu}\mathbf{1}^{'} + \boldsymbol{B}\boldsymbol{Z} + \mathbf{U}, \tag{2.2}$$

with $\mathbf{1}$ denoting a $T \times 1$ vector of ones.

3 Lasso-VAR

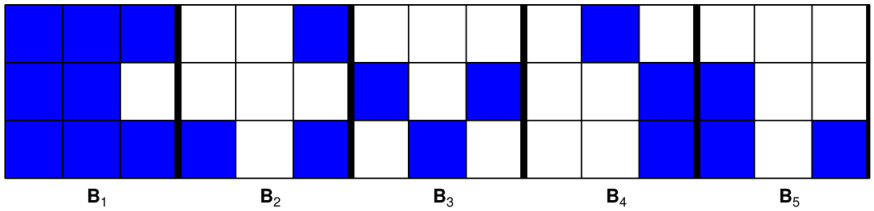
Consider the notation from Section 2, in which $\boldsymbol{Y}$ is a $k \times T$ response matrix, $\boldsymbol{Z}$ is a $kp \times T$ covariate matrix and $\boldsymbol{B}$ is a $k \times kp$ matrix of unknown coefficients.

An initial approach to reduce the dimensionality of the parameter space, extending Tibshirani [1996], is to apply an L_1 penalty to the convex least squares objective function

$$\frac{1}{2}||\boldsymbol{Y} - \boldsymbol{\nu}\mathbf{1}^{'} - \boldsymbol{B}\boldsymbol{Z}||_F^2 + \lambda||\boldsymbol{B}||_1, \tag{3.1}$$

in which $||\mathbf{X}||_F^2 = \sum_{i=1}^{m}\sum_{j=1}^{n}|x_{ij}|^2$ is the square of the Frobenius norm of $\mathbf{X}$, $||\mathbf{X}||_1 = \sum_{jk}|X_{jk}|$ is the L_1 norm, and $\lambda \geq 0$ is a penalty parameter. An L_1 penalty will induce sparsity in the coefficient matrix $\boldsymbol{B}$ by zeroing individual entries. This results in an unstructured sparsity pattern, an example of which (with p=5 and k=3) is depicted in Figure 1. If the Lasso-VAR selects $[\boldsymbol{B}_q]_{jk}$ for $q = 1, \ldots, p$, it follows that for a given $\lambda > 0$, at the value of $\widehat{\boldsymbol{B}}$ which minimizes the objective function, Equation (3.1), the linear relationship between series j and series k at lag q is zero, given the other variables selected in the model.

Figure 1: Example sparsity pattern produced by a Lasso-VAR$_3$(5)



Sparsity in the coefficient matrix is crucial when k is large because the conventional VAR is overparameterized; Sims [1980] remarked that its construction represents a "profligate parameterization." Moreover, the Lasso-VAR has an advantage over existing Bayesian methods in that it will both shrink least squares estimates toward zero as well as perform variable selection. This allows for feasible estimation for cases in which the number of regression parameters k^2p is close to or exceeds the sample size kT.

In regularization problems, the intercept is not typically shrunk. Instead, it is calculated following estimation, while $\boldsymbol{Y}$ and $\boldsymbol{Z}$ are standardized so that each row has a sample mean of zero. As a result, $\boldsymbol{\nu}$ will no longer appear in the objective function. This procedure is described in section 9.1 of the Appendix.

Since the L_1 norm is not differentiable, no closed-form solution exists for Equation (3.1), hence iterative methods are required. Our approach to solve Equation (3.1) involves the use of coordinate descent, popularized by Friedman et al. [2010]. This consists of partitioning Equation (3.1) into scalar subproblems for each $[\boldsymbol{B}]_{ij}$, solving component-wise, and then updating until convergence. This approach is computationally tractable since in the Lasso-VAR context, each subproblem has a closed-form solution. Tseng [2001] establishes that global convergence arises from solving individual subproblems in the coordinate descent framework. Our solution strategy is detailed Section 9.2.1 of the Appendix.

3.1 Structured Penalties

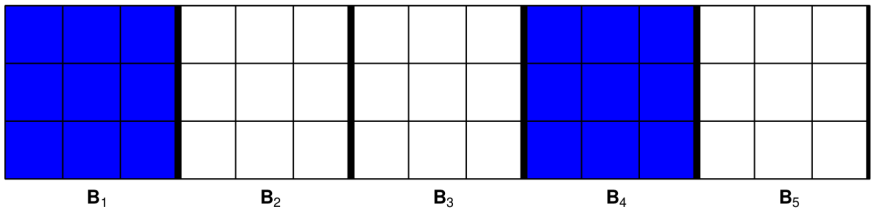
Instead of shrinking coefficients element-wise, forecasting may be substantially improved by taking advantage of the inherent structure of the VAR. For example, Song and Bickel [2011], consider two structures: assigning each row of each $\boldsymbol{B}_\ell$ to its own group, resulting in separable objective functions for each series, (*no grouping*) or partitioning the rows of each $\boldsymbol{B}_\ell$ based on natural or given data-specific partitions, such as by economy (*segmentized grouping*).

We propose generalizing this approach and partition $\boldsymbol{B}$. As a simple lag-based grouping, we examine the following objective function utilizing a *Group* Lasso penalty structure (Yuan and Lin [2006]):

$$\frac{1}{2}||\boldsymbol{Y} - \mathbf{BZ}||_F^2 + \lambda \sum_{\ell=1}^{p} ||\boldsymbol{B}_\ell||_F. \tag{3.2}$$

That is, we group the coefficient sub-matrices by their time lags, in which $\boldsymbol{B}_\ell$ is defined as in Equation (2.1). This structure is advantageous for applications in which all component series tend to exhibit comparable dynamics. It also can serve as a powerful tool for lag selection. This simple structure leads to a sparsity pattern such as that depicted in Figure 2. Unlike the Lasso-VAR,

Figure 2: Example sparsity pattern produced by a Lag Group Lasso-VAR$_3$(5)



though the subproblems in the Group-Lasso VAR are separable, they are not solvable in closed form. We instead extend the methodology of Qin et al. [2010] and transform each subproblem to a trust-region frameworkwhich can be solved efficiently as a univariate optimization problem. Details of this procedure are provided Section 9.2.2 of the Appendix.

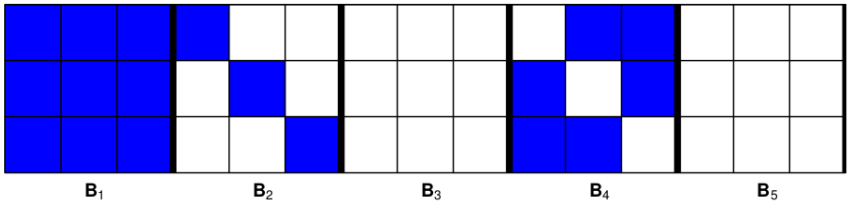
3.2 Alternative Group Structures

In many settings, it may not be appropriate to give equal consideration to every entry in a coefficient matrix $\boldsymbol{B}_\ell$. Diagonal entries, which represent a variable's own lags, are in many applications more likely to be nonzero than off-diagonal entries, which represent cross dependence with other components. We can thus partition each coefficient matrix into separate groups via the objective

$$\frac{1}{2}||\boldsymbol{Y} - \mathbf{BZ}||_F^2 + \sqrt{k}\lambda \sum_{\ell=1}^{p} ||\text{diag}(\boldsymbol{B}_\ell)||_2 + \sqrt{k(k-1)}\lambda \sum_{\ell=1}^{p} ||\boldsymbol{B}_\ell^-||_2, \tag{3.3}$$

where $\boldsymbol{B}_\ell^- = \{[\boldsymbol{B}_\ell]_{ij} : i \neq j\}$. Unlike Equation (3.2), groups differ in cardinality, which requires weighting the penalty to avoid regularization favoring larger groups. An example of this sparsity pattern is shown in Figure 3. The modifications required to implement the Own/Other-Group-Lasso-VAR are detailed Section 9.2.3 in the Appendix.

Figure 3: Example sparsity pattern produced by an Own/Other Group Lasso-VAR$_3$(5)



3.3 Sparse Group Lasso-VAR

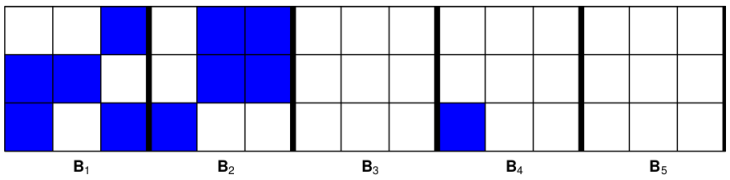
For certain applications, the Group Lasso penalty might be too restrictive. If a group is active, all coefficients in the group will be nonzero, and including a large number of groups substantially increases computation time. Moreover, it is inefficient to include an entire group if only one coefficient is nonzero. The *Sparse Group Lasso* penalty of Simon et al. [2013] allows for greater flexibility

by adding within-group sparsity to Equation (3.2) via the objective function

$$\frac{1}{2k}||\boldsymbol{Y} - \mathbf{BZ}||_F^2 + (1-\alpha)\lambda \sum_{\ell=1}^{p} ||\boldsymbol{B}_\ell||_F + \alpha\lambda||\boldsymbol{B}||_1, \tag{3.4}$$

in which $0 \leq \alpha \leq 1$ is an additional tuning parameter. Larger values of α imply strong overall sparsity, while small values of α imply strong group-wise sparsity, but minimal sparsity within-group. Note that the case where $\alpha = 0$ is equivalent to the Lag Group Lasso-VAR and $\alpha = 1$ is equivalent to the Lasso-VAR. As an alternative to estimating via cross-validation (jointly over α and λ), we relate within-group sparsity to the number of component series, and set $\alpha = \frac{1}{k+1}$. An example sparsity pattern is depicted in Figure 4. Since the inclusion of within-group sparsity

Figure 4: Example sparsity pattern produced by a Lag Sparse Group Lasso-VAR$_3$(5)



does not allow for separability, coordinate descent is no longer appropriate, therefore, following Simon et al. [2013] our solution to Equation 3.4 makes use of proximal gradient descent. This procedure can be thought of as an extension of gradient descent in which the objective function can be decomposed into a smooth and non-smooth part. The details of this approach and our implementation are provided in Section 9.2.4 of the Appendix.

Following similar methodology to the Group Lasso-VAR, the Sparse Group Lasso-VAR can also be extended to alternative groupings. Consequently, we also offer the "Own/Other" Sparse Group Lasso-VAR as an estimation procedure.

4 Penalty Parameters

4.1 Selection of Penalty Grid

Following Friedman et al. [2010], we choose the grid of potential penalty parameters to decrement in log-linear increments, starting with the smallest value in which all components of $\boldsymbol{B}$ will be zero. This value differs for each procedure and can be inferred by their respective algorithms. The starting values are summarized in Table 9 located in Section 9.3 of the Appendix. The number of gridpoints as well as the depth of the grid are left to user input.

4.2 Data-Driven Selection of λ

Due to time-dependence, our problem is not well suited to traditional n-fold cross-validation. Instead, expanding on Song and Bickel [2011] and Banbura et al. [2009], we propose choosing the optimal penalty parameter by minimizing one-step ahead mean-square forecast error (MSFE). We start by dividing the data into three periods: one for initialization, one for training, and one for forecast evaluation. Define time indices $T_1 = \lfloor \frac{T}{3} \rfloor, T_2 = \lfloor \frac{2T}{3} \rfloor$.

The period $T_1 + 1$ through T_2 is used for training and $T_2 + 1$ through T for evaluation of forecast accuracy in a rolling manner. The procedure is illustrated in Figure 5.

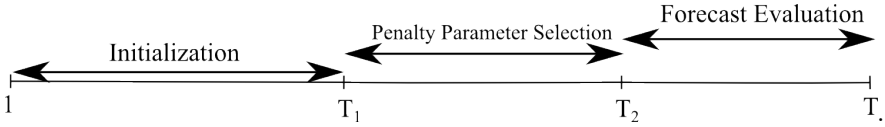


Figure 5: Illustration of Rolling Cross-Validation

Define $\hat{\mathbf{y}}_{t+1}^{\lambda}$ as the one-step ahead forecast based on all observations from $1, \ldots, t$. We consider minimization of

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

If desired, additional forecast horizons or 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 method uses the result from the previous period as an initialization or "warm start," which substantially decreases computation time. The penalty selection procedure is expressed in Algorithm 2 in the Appendix.

5 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 [2007]). An estimation procedure which 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 (Group) Lasso-VAR."

As an illustration, consider the following example which uses the results of a Lasso-VAR$_2$(2) with r nonzero coefficients to introduce linear constraints of the form

$$\text{vec}(\widehat{\boldsymbol{B}}) = \mathbf{R}\hat{\beta}, \tag{5.1}$$

in which $\mathbf{R}$ is a $k^2p \times r$ matrix with rank r and $\hat{\beta} = \text{vec}(\{\widehat{B} : [\widehat{B}]_{jk} \neq 0\})$. Within the relaxed framework, λ is held constant and the support recovered is taken as given. For example, consider the following support recovered from the aforementioned Lasso-VAR$_2$(2).

$$\hat{\boldsymbol{B}} = \begin{bmatrix} [\hat{B}_1]_{11} & [\hat{B}_1]_{12} & 0 & 0 \\ [\hat{B}_1]_{21} & [\hat{B}_1]_{22} & [\hat{B}_2]_{21} & 0 \end{bmatrix}.$$

In this case, Lasso-VAR indicates that the model contains 5 non-zero coefficients:

$$[\hat{B}_1]_{11}, [\hat{B}_1]_{12}, [\hat{B}_1]_{21}, [\hat{B}_1]_{22}, [\hat{B}_2]_{21}$$

. Then, we can express Equation (5.1) as

$$\text{vec}(\widehat{\boldsymbol{B}}) = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} [\hat{B}_1]_{11} \\ [\hat{B}_1]_{12} \\ [\hat{B}_1]_{21} \\ [\hat{B}_1]_{22} \\ [\hat{B}_2]_{21} \end{bmatrix}.$$

As previously stated, in the unrestricted VAR framework, the ordinary and generalized least squares estimators coincide. However, once restrictions are introduced, the generalized least squares (GLS) estimator is asymptotically more efficient than ordinary least squares.

Following Brüggemann [2004], we can express the GLS estimator of the Relaxed VAR as

$$\hat{\boldsymbol{B}}^{GLS} = [\mathbf{R}^{'}(\mathbf{Z}\mathbf{Z}^{'} \otimes \Sigma_u^{-1})\mathbf{R}]^{-1}\mathbf{R}^{'}(\boldsymbol{Z} \otimes \Sigma_u^{-1})\text{vec}(\boldsymbol{Y}), \tag{5.2}$$

in which $\otimes$ denotes the Kronecker product. However, since Σ_u is unknown in general, Equation (5.2) cannot be used in practice. A feasible GLS estimate may be defined by first estimating the Relaxed Least Squares (RLS) estimator

$$\hat{\boldsymbol{B}}^{\text{Relaxed}} = \mathbf{R}[\mathbf{R}'(\mathbf{Z}\mathbf{Z}' \otimes I_k)\mathbf{R}]^{-1}\mathbf{R}'(\boldsymbol{Z} \otimes I_k)\text{vec}(\boldsymbol{Y}).$$

We then use the RLS 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, $\hat{\boldsymbol{B}}^{\text{Relaxed}}$ 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

$$\bar{\Sigma}_u = \frac{1}{T - kp - 1}(\boldsymbol{Y} - \widehat{\boldsymbol{B}}^{\text{Relaxed}}\boldsymbol{Z})(\boldsymbol{Y} - \widehat{\boldsymbol{B}}^{\text{Relaxed}}\boldsymbol{Z})^{'}$$

Then, assuming $\bar{\Sigma}_u$ is non-singular, we can compute a feasible GLS as

$$\widehat{\boldsymbol{B}}^{FGLS} = \mathbf{R}[\mathbf{R}^{'}(\mathbf{Z}\mathbf{Z}^{'} \otimes \bar{\Sigma}_u^{-1})\mathbf{R}]^{-1}\mathbf{R}^{'}(\boldsymbol{Z} \otimes \bar{\Sigma}_u^{-1})\text{vec}(\boldsymbol{Y}).$$

Our applications have found $\mathbf{Z}\mathbf{Z}^{'}$ to be poorly conditioned when T is small. Moreover, as the dimension increases, conducting operations directly with the $k^2p \times k^2p$ matrix $(\mathbf{Z}\mathbf{Z}' \otimes I_k)$ exhausts memory. To ameliorate these issues of dimensionality, the refitting procedure can be conducted row-by-row if the covariance matrix is diagonal. Additionally, the conditioning of $\mathbf{Z}\mathbf{Z}^{'}$ can be improved by implementing a modification of the procedure developed by Neumaier and Schneider [2001], which adds a small regularization parameter to $\boldsymbol{Z}$ and computes $\mathbf{Z}\mathbf{Z}^{'}$ via a QR factorization. This approach is summarized in Algorithm 5 in Section 9.5 of the Appendix.

6 Empirical Results

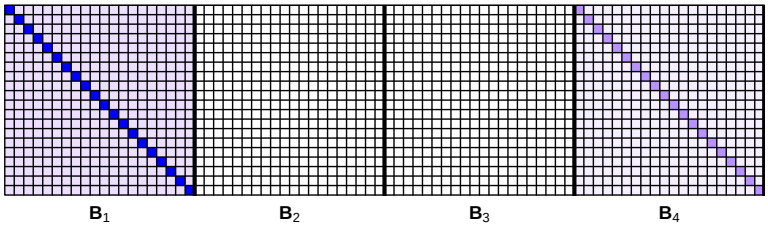
6.1 Simulation Scenarios

We evaluate our algorithms on several simulated high-dimensional VARs, conforming to different sparsity patterns; one constructed to be advantageous to each proposed structure. All simulations operate on a $\text{VAR}_{20}(4)$ of length $T = 100$ and each is repeated 100 times. The choice of 4 was selected for p because it represents one year of dependence for quarterly data, which is a common frequency of macroeconomic data. Two-thirds of the observations are used for initialization and penalty parameter selection while one-third are used for forecast evaluation.

6.1.1 Scenario 1: Sparse and Diagonally Dominant

The first coefficient matrix consists of a diagonally dominant sparsity structure, in which all diagonal elements and off-diagonal elements are identical in magnitude in matrices $\boldsymbol{B}_1$ and $\boldsymbol{B}_4$, and $\boldsymbol{B}_2$ and $\boldsymbol{B}_3$ are set identically to zero. The residual covariance matrix Σ_u is set to $(0.01)\mathbf{I}_{20}$. The sparsity pattern is depicted in Figure 6. The darker shade represents coefficients that are larger in magnitude. and the simulation results are summarized in Table 1. Under this setting, one would expect top

Figure 6: Sparsity Pattern Scenario 1: Sparse and Diagonally Dominant



performance from the "own/other" Group-Lasso VAR.

Table 1: Out of sample MSFE of one-step ahead forecasts averaged over 100 simulations: Scenario 1

Model	Average MSFE	Standard Error
Lasso	0.2400	0.0254
Lag Group Lasso	0.2321	0.0152
Lag Sparse Group Lasso	0.2326	0.0170
Own/Other Group Lasso	0.2268	0.0149
Own/Other Sparse Group Lasso	0.2271	0.0166
VAR with lag selected by AIC	0.2868	0.0198
Sample Mean	0.2688	0.0682
Random Walk	0.3336	0.0216

6.1.2 Scenario 2: Lag Sparsity

We next consider a scenario in which Lag B_1 and B_4 are dense with coefficients of the same magnitude and all other coefficients are set to zero. Under such a design, we should expect superior performance from the lag structured approaches. For this simulation, Σ_u is again set to $(0.01)\mathbf{I}_{20}$. The sparsity pattern is depicted in Figure 8, and the results are summarized in Table 2.

Figure 7: Sparsity Pattern Scenario 2: Lag Sparsity

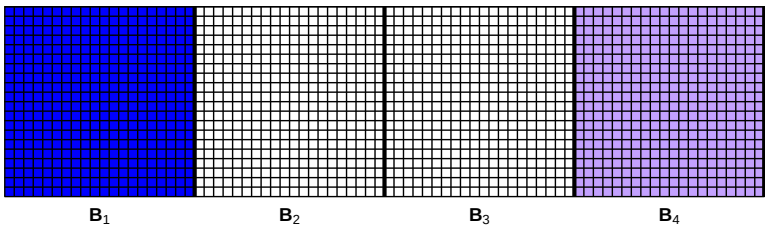


Table 2: Out of sample MSFE of one-step ahead forecasts after 100 simulations: Scenario 1

Model	Average MSFE	Standard Error
Lasso	0.3566	0.0354
Lag Group Lasso	0.3394	0.0317
Lag Sparse Group Lasso	0.3389	0.0355
Own/Other Group Lasso	0.3366	0.0315
Own/Other Sparse Group Lasso	0.3942	0.0536
VAR with lag selected by AIC	0.4303	0.0403
Sample Mean	3.04	1.440
Random Walk	7.35	4.537

6.1.3 Scenario 3: Unstructured Sparsity

We next consider a scenario in which the sparsity is completely random. Under such a design, we should expect superior performance from the unstructured Lasso-VAR. Σ_U is again set to $(0.01)\mathbf{I}_{20}$. The sparsity pattern is depicted in Figure 8. and the results are summarized in Table 3. Clearly in

Figure 8: Sparsity Pattern Scenario 3: Unstructured Sparsity

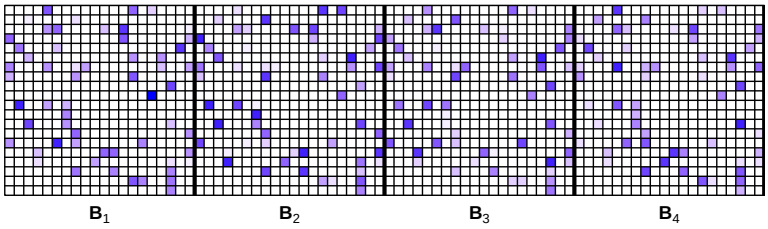


Table 3: Out of sample MSFE of one-step ahead forecasts after 100 simulations: Scenario 1

Model	Average MSFE	Standard Error
Lasso	.3431	.02590
Lag Group Lasso	.4220	.04306
Lag Sparse Group Lasso	.4509	.09460
Own/Other Group Lasso	.4767	.0941
Own/Other Sparse Group Lasso	.5633	.1297
VAR with lag selected by AIC	.7802	.0809
Sample Mean	5.059	3.969
Random Walk	7.296	5.591

the absence of any sort of structure, all of the structured approaches are substantially outperformed by the Lasso-VAR. However, it

6.1.4 Scenario 4: Structured Blockwise Sparsity, Unstructured Within-Block

Our third scenario can be thought of as a hybrid of the previous two cases. As in Scenario 2, only matrices $\boldsymbol{B}_1$ and $\boldsymbol{B}_4$ contain nonzero coefficients and, similar to Scenario 3, sparsity within each block is unstructured. Σ_U is again set to $(0.01)\mathbf{I}_{20}$. The sparsity pattern is visualized in Figure 9. In such a scenario, we should expect procedures that allow for within-group sparsity, such as the Sparse Group Lasso-VAR and Lasso-VAR to achieve the best performance. The results are summarized in Table 4.

Figure 9: Sparsity Pattern Scenario 3: Structured Blockwise, Unstructured within Block

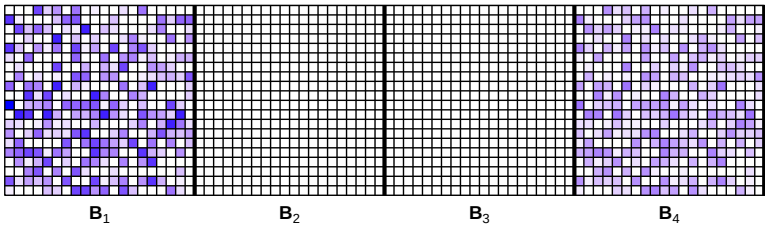


Table 4: Out of sample MSFE of one-step ahead forecasts after 100 simulations: Scenario 3

Model	Average MSFE	Standard Error
Lasso	.3531	.0213
Lag Group Lasso	.3355	.0201
Lag Sparse Group Lasso	.33941	.0205
Own/Other Group Lasso	.3334	.0200
Own/Other Sparse Group Lasso	.4007	.0243
VAR with lag selected by AIC	.4230	.0268
Sample Mean	3.1523	.2026
Random Walk	8.1067	.5952

6.1.5 Relaxed VAR Simulations

In order to examine the impact of the relaxed estimators, we ran simulations using the same sparsity pattern as Scenario 4 with several choices for $\mathbf{\Sigma}_u$. We then compared the Lasso-VAR estimator with no refitting, refitting using RLS, refitting using RFGLS, and refitting using the true $\mathbf{\Sigma}_u$ in a Relaxed GLS framework. The different covariance matrix specifications are specified in Figure 10, and the results are summarized in Table 5.

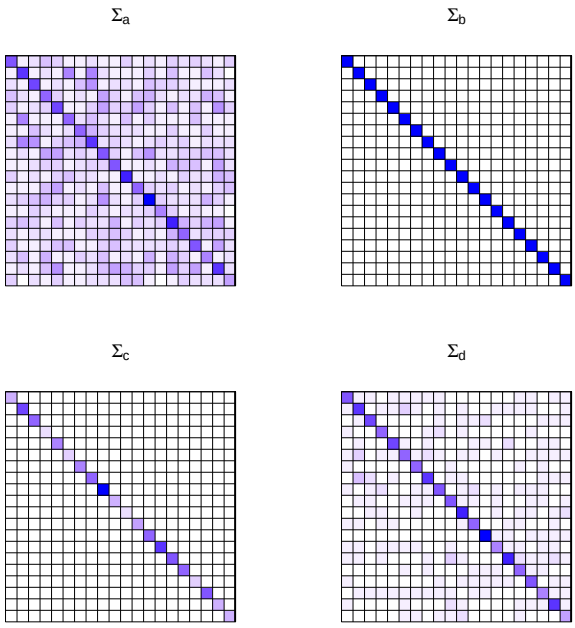


Figure 10: Sparsity Patterns used for Relaxed VAR simulation. Σ_A denotes a dense covariance matrix. Σ_B represents a diagonal matrix with equal weights, Σ_C a diagonal matrix with unequal weights, and Σ_D a sparse matrix.

Table 5: Out of sample MSFE of one-step ahead forecasts after 100 simulations: Relaxed VAR Sigma choice c

	$\mathbf{\Sigma}_u$ A		$\mathbf{\Sigma}_u$ B	
Model	Average MSFE	Standard Error	Average MSFE	Standard Error
Lasso Unrelaxed	329.87	53.03	185.42	22.27
Lasso Relaxed Least Squares	322.78	38.16	198.59	17.71
Lasso Relaxed FGLS	313.03	37.50	188.59	18.22
Lasso Relaxed GLS (True $\mathbf{\Sigma_u}$)	310.36	37.66	187.10	18.24
	$\mathbf{\Sigma}_u$ C		$\mathbf{\Sigma}_u$ D	
Lasso Unrelaxed	171.33	19.94	591.59	76.06
Lasso Relaxed Least Squares	181.00	16.39	629.60	64.66
Lasso Relaxed FGLS	172.33	16.62	596.00	65.85
Lasso Relaxed GLS (True $\mathbf{\Sigma_u}$)	170.53	16.40	590.68	62.07

6.1.6 Discussion

All of our procedures are fairly robust to sparsity patterns not conforming to their true group structure. In each scenario, every method substantially outperforms the benchmarks. Scenario 3 was the only case in which the structured approaches performed poorly relative to the Lasso-VAR. We expect that such a sparsity pattern will rarely occur in practice.

In addition, we find that our Relaxed FGLS refitting procedure performs almost as well as the Relaxed GLS approach using the true covariance matrix. However, in most scenarios, refitting does not offer a substantial improvement in forecast accuracy, though it does substantially decrease standard error.

6.2 Macroeconomic Data Application

Our methods were additionally evaluated on a large macroeconomic dataset originally compiled by Stock and Watson [2005] and augmented by Koop [2011]. It consists of 131 quarterly macroeconomic indicators, containing information about various aspects of the economy, including income, industrial production, employment, stock prices, interest rates, exchange rates, etc. Per Koop [2011], the series can be partitioned into several groups, we incorporate the following two:

- *Small*: 3 variable (Federal Funds Rate, CPI, GDP growth rate): Core group, typically used in simple DSGE models (k=3),
- *Medum* Small plus 17 additional variables, containing aggregated economic information (e.g. consumption, labor, housing, exchange rates) (k=20),

As Banbura et al. [2009] found that the greatest improvements in forecast performance occured with the *Medium* VAR, that will be our focus. The time series included are listed in Table 10 in the Appendix. Before estimation, each series is transformed to stationarity according to the transformation codes provided by Stock and Watson [2005], and standardized by subtracting the sample mean and dividing by the sample standard deviation. Quarter 3 of 1975 to Quarter 1 of 1992 is used for penalty parameter selection while Quarter 2 of 1992 to Quarter 1 of 2009 is used

for forecast evaluation. Our results are summarized in Table 6 located in Section 9.4. Note that the AIC, mean, and random walk benchmarks change slightly, as setting $p = 13$ requires additional values for initialization.

Table 6: Out of sample MSFE of one-step ahead forecasts on 20 macroeconomic indicators

	$p = 4$		$p = 13$	
Model/Penalty	MSFE	S.E. Over Evaluation Period	MSFE	S.E. Over Evaluation Period
Lasso	12.148	2.007	12.680	2.123
Lag Group Lasso	12.611	2.148	12.848	2.217
Lag Sparse Group Lasso	12.437	2.105	12.835	2.207
Own/Other Group Lasso	12.128	2.008	11.840	2.039
Own/Other Sparse Group Lasso	11.704	1.863	11.598	1.946
VAR with lag selected by AIC	14.125	2.346	14.583	2.551
Sample Mean	15.126	2.320	15.350	2.444
Random Walk	30.638	5.541	30.843	5.818

7 Extensions

7.1 Shrinking to a constant matrix C

The proposed algorithms can easily be modified to shrink toward a known constant matrix. Shrinking toward a constant matrix results in an optimization problem of the form.

$$\frac{1}{2}\|\boldsymbol{Y} - \boldsymbol{BZ}\|_F^2 + \lambda\Omega(\boldsymbol{B} - \mathbf{C}),$$

where Ω is one of the proposed penalty functions and $\mathbf{C}$ is a constant matrix that we are shrinking toward. Let $\widehat{\boldsymbol{B}}^{\lambda}(\boldsymbol{Y}, C)$ denote a solution to this problem. Now, by a change of variables $\tilde{\boldsymbol{B}} = \boldsymbol{B} - \mathbf{C}$, we get the equivalent problem

$$\|\boldsymbol{Y} - (\tilde{\boldsymbol{B}} + \mathbf{C})\boldsymbol{Z}\|_F^2 + \lambda\Omega(\tilde{\boldsymbol{B}}).$$

or

$$\|\boldsymbol{Y} - \mathbf{C}\boldsymbol{Z} - \tilde{\boldsymbol{B}}\boldsymbol{Z}\|_F^2 + \lambda\Omega(\tilde{\boldsymbol{B}}).$$

Thus, the solution to this transformed problem is given by $\widehat{\boldsymbol{B}}^{\lambda}(\boldsymbol{Y} - \mathbf{C}\boldsymbol{Z}, 0)$ and transforming back to the original variable (i.e., from $\tilde{\boldsymbol{B}}$ to $\boldsymbol{B}$), we see that

$$\hat{\boldsymbol{B}}^{\lambda}(\boldsymbol{Y}, \mathbf{C}) = \mathbf{C} + \hat{\boldsymbol{B}}^{\lambda}(\boldsymbol{Y} - \mathbf{C}\boldsymbol{Z}, 0).$$

Therefore, if we know how to solve the problem with $\mathbf{C} = 0_{k \times kp}$, then we can use this to solve the problem for general $\mathbf{C}$.

As an example, with $\mathbf{C} = (\mathbf{I}_k, \mathbf{0}_{k\times k}, \ldots, \mathbf{0}_{k\times k})$, we could implement a variant of the Minnesota Prior, in which we shrink toward a random walk. This approach could be of use in economic applications as it is widely believed that many macroeconomic time series follow a random walk (Litterman [1979]).

7.1.1 Application

In order to test this procedure, we follow the methodology of Banbura et al. [2009], who utilize the Stock and Watson dataset, but eschew stationarity transformations and work directly with the non-stationary series. We again apply our estimation procedures on the aforementioned "medium" set of series, but choose not to perform any stationarity transformations and shrink toward a random walk. Our results are summarized in Table 7.

7.2 Incorporating Exogenous Variables

In many applications, forecasts can be improved by incorporating exogenous variables, which are determined outside of the VAR. Examples of these types of variables could be leading indicators, weather related variables, or global macroeconomic indicators not related to the macroeconomic system examined, such as global oil prices. The so-called $VARX_k(p, s)$ model, which incorporates

Table 7: Out of sample MSFE of one-step ahead forecasts on 20 nonstationary macroeconomic indicators

	$p = 4$		$p = 13$	
Model/Penalty	MSFE	S.E. Over Evaluation Period	MSFE	S.E. Over Evaluation Period
Lasso	2.322	1.515	2.038	1.207
Lag Group Lasso	2.278	1.537	2.428	1.657
Lag Sparse Group Lasso	2.156	1.417	2.421	1.643
Own/Other Group Lasso	2.074	1.302	2.445	1.649
Own/Other Sparse Group Lasso	1.914	1.171	2.421	1.638
VAR with lag selected by AIC	3.709	1.881	4.111	2.299
Sample Mean	31.306	3.709	29.127	1.657
Random Walk	2.513	1.766	2.626	1.850

an m-dimensional series of exogenous variables $\boldsymbol{x_t}$, can be expressed as

$$\boldsymbol{Y}_t = \boldsymbol{\nu} + \sum_{i=1}^{p} \boldsymbol{B}_i \boldsymbol{Y}_{t-i} + \sum_{j=0}^{s} \boldsymbol{\theta}_j \mathbf{x_{t-j}} + \mathbf{u}_t, \tag{7.1}$$

In which $\boldsymbol{\theta}_j$ represents a $k \times m$ coefficient matrix of endogenous variables at lag j. Alternatively, using the compact matrix representation of Section 2, we can define

$$\begin{aligned}
\boldsymbol{\Gamma}_t &= \left[1, \quad \mathbf{y}_t^{'}, \ldots, \quad \mathbf{y}_{t-p}^{'}, \quad \mathbf{x}_{t+1}^{'}, \quad \ldots, \quad \mathbf{x}_{t-s}^{'}\right]^{'} \\
\boldsymbol{\Gamma} &= \left[\boldsymbol{\Gamma_0}, \ldots, \boldsymbol{\Gamma_{T-1}}\right] \\
\boldsymbol{\Theta} &= \left[\nu, \boldsymbol{B}_1, \ldots \boldsymbol{B}_p, \boldsymbol{\theta}_0, \ldots, \boldsymbol{\theta}_s\right]
\end{aligned}$$

then, with $\boldsymbol{Y}, \boldsymbol{\nu}$ and $\mathbf{U}$ defined as in Section 2, we have that

$$\boldsymbol{Y} = \boldsymbol{\nu}\mathbf{1}^{'} + \boldsymbol{\Theta}\boldsymbol{\Gamma} + \mathbf{U}.$$

Since each exogenous variable requires ks additional coefficients, it is natural to restrict its parameterization. We could extend the Lasso-VAR, to the Lasso-VARX, resulting in

$$\|\boldsymbol{Y} - \boldsymbol{\Theta}\boldsymbol{\Gamma}\|_F^2 + \lambda\|\boldsymbol{\Theta}\|_1.$$

However, as noted by Chiuso and Pillonetto [2010], the Group Lasso VARX approach appears more natural, as we would like to be able to differentiate between endogenous and exogenous variables. For example, we could consider the Lag Group Lasso VARX

$$\|\boldsymbol{Y} - \boldsymbol{\Theta\Gamma}\|_F^2 + \sqrt{k^2}\lambda \sum_{\ell=1}^{p} \|\boldsymbol{B}_\ell\|_F + \sqrt{k}\lambda \sum_{j=1}^{sm} \|\theta_s\|_F.$$

That is, an exogenous variable at lag s is either nonzero for all series or none at all. The Lag Sparse Group Lasso VARX and Own/Other follow similarly.

7.2.1 Application

As an application, suppose that a practioner only desire forecasts from the *small* group of macroeconomic indicators described in Section 6, but wants to use the *Medium* set of indicators as covariates. Rather than estimating the entire $400p$ variable system, a VARX version of our structured approaches could be fit with the 17 additional macroeconomic indicators treated as exogenous covariates, resulting in a more manageable $(9 + 17)p$ variable system. The results of these methods are summarized in Table 8.

Table 8: Out of sample MSFE of one-step ahead forecasts on 3 macroeconomic indicators with 17 exogenous covariates, p=13

Model/Penalty	MSFE	S.E.
Lasso	1.326	0.175
Lag Group Lasso	1.202	0.168
Own/Other Group Lasso	1.294	0.188
Lag Sparse Group Lasso	1.281	0.164
O/O Sparse Group Lasso	1.290	0.189
Sample Mean	1.490	0.236
Unrestricted VARX	2.079	0.288
Random Walk	3.388	0.671

8 Conclusion

The structured regularization framework is quite flexible in that it can accommodate a variety of potential dynamic structures. In addition, models can be refit based on a selected support, shrunk toward a known constant matrix and exogenous variables can easily be incorporated.

Each of the proposed methods consistently outperform the VAR with lag selected by AIC. Moreover, upon examining actual data, structured approaches tend to outperform their unstructured counterparts. Forecast performance for all methods appears to be robust across multiple sparsity structures. Future forecasting applications may involve higher dimensional simulation scenarios and larger macroeconomic datasets.

Our work has considerable room for extensions. In addition to the conditioning issues with the Relaxed VAR framework, our procedures require a coherent maximal lag selection mechanism. In data applications, most of our procedures tend to provide worse forecasts as the maximal lag order is increased. The currently accepted 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 would require a two-dimensional gridsearch hindering computational performance. As an additional potential issue, though our procedures perform well when shrinking toward a random walk, the nonstationarity of the data can create problems as the grid of penalty parameters is not likely to be constant over time.

An `R` package containing our algorithms and validation procedures, `BigVAR`, is forthcoming.

References

Francis Bach, Rodolphe Jenatton, Julien Mairal, and Guillaume Obozinski. Convex optimization with sparsity-inducing norms. *Optimization for Machine Learning*, pages 19–53, 2011.

Marta Banbura, Domenico Giannone, and Lucrezia Reichlin. Large bayesian vector auto regressions. *Journal of Applied Econometrics*, 25(1):71–92, 2009.

Sumanta Basu and George Michailidis. Estimation in high-dimensional vector autoregressive models. *arXiv preprint arXiv:1311.4175*, 2013.

Amir Beck and Marc Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202, 2009.

Stephen R Becker, Emmanuel J Candès, and Michael C Grant. Templates for convex cone problems with applications to sparse signal recovery. *Mathematical Programming Computation*, 3(3):165–218, 2011.

Ben S Bernanke, Jean Boivin, and Piotr Eliasz. Measuring the effects of monetary policy: a factor-augmented vector autoregressive (favar) approach. *The Quarterly Journal of Economics*, 120(1): 387–422, 2005.

Ralf Brüggemann. *Model reduction methods for vector autoregressive processes*, volume 536. Springer Verlag, 2004.

Alessandro Chiuso and Gianluigi Pillonetto. Nonparametric sparse estimators for identification of large scale linear systems. In *Decision and Control (CDC), 2010 49th IEEE Conference on*, pages 2942–2947. IEEE, 2010.

Richard A. Davis, Pengfei Zang, and Tian Zheng. Sparse vector autoregressive modeling. 2012. journal: arXiv preprint arXiv:1207.0520.

Francis X Diebold. The past, present, and future of macroeconomic forecasting. *The Journal of Economic Perspectives*, 12(2):175–192, 1998.

Jerome Friedman, Trevor Hastie, and Rob Tibshirani. Regularization paths for generalized linear models via coordinate descent. *Journal of statistical software*, 33(1):1, 2010.

Gene H Golub and Charles F Van Loan. *Matrix computations*, volume 3. JHU Press, 2012.

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

K Kadiyala and Sune Karlsson. Numerical methods for estimation and inference in bayesian var-models. *Journal of Applied Econometrics*, 12(2):99–132, 1997.

Lawrence Robert Klein and Arthur S Goldberger. An econometric model of the united states, 1929-1952, 1955.

Anders Bredahl Kock and Laurent AF Callot. Oracle inequalities for high dimensional vector autoregressions. *arXiv preprint arXiv:1311.0811*, 2013.

Gary Koop. Forecasting with medium and large bayesian vars. *Journal of Applied Econometrics*, 2011.

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

Marcelo C Medeiros and Eduardo F Mendes. Estimating high-dimensional time series models. 2012.

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

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

Jorge. Nocedal and Stephen J Wright. *Numerical optimization*, volume 2. Springer New York, 1999.

Zhiwei Qin, Katya Scheinberg, and Donald Goldfarb. Efficient block-coordinate descent algorithms for the group lasso. *Mathematical Programming Computation*, pages 1–27, 2010.

John C Robertson and Ellis William Tallman. Improving forecasts of the federal funds rate in a policy model. Technical report, Federal Reserve Bank of Atlanta, 1999.

Noah Simon, Jerome Friedman, Trevor Hastie, and Robert Tibshirani. A sparse-group lasso. *Journal of Computational and Graphical Statistics*, 22(2):231–245, 2013.

Christopher A Sims. Macroeconomics and reality. *Econometrica: Journal of the Econometric Society*, pages 1–48, 1980.

Song Song and Peter Bickel. Large vector auto regressions. 2011. journal: arXiv preprint arXiv:1106.3915.

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

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

Paul Tseng. Convergence of a block coordinate descent method for nondifferentiable minimization. *Journal of optimization theory and applications*, 109(3):475–494, 2001.

Władysław Welfe. Macroeconometric models of the united states and canada. In *Macroeconometric Models*, pages 15–46. Springer, 2013.

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

Arnold Zellner. An efficient method of estimating seemingly unrelated regressions and tests for aggregation bias. *Journal of the American statistical Association*, 57(298):348–368, 1962.

9 Appendix

9.1 Intercept Term

We can express (3.1) as:

$$\begin{aligned} f(\boldsymbol{B}, \boldsymbol{\nu}) &= \frac{1}{2}||\boldsymbol{Y} - \boldsymbol{\nu}\mathbf{1}^{'} - \boldsymbol{B}\boldsymbol{Z}||_F^2 + \lambda||\boldsymbol{B}||_1, \\ &= \frac{1}{2}\sum_{kt}(Y_{kt} - \nu_k - BZ_{kt})^2 + \lambda\sum_{kr}|B_{kr}|. \end{aligned}$$

In regularization problems, the intercept $\hat{\boldsymbol{\nu}}$ is typically not shrunk and can be derived separately from $\hat{\boldsymbol{B}} = \operatorname{argmin} f$. We can find $\hat{\boldsymbol{\nu}}$ by calculating the gradient of the unpenalized portion of

Equation (3.1) and solving with respect to the first order conditions:

$$0 = \nabla_{\boldsymbol{\nu}} f(\boldsymbol{B}, \boldsymbol{\nu}) = (\boldsymbol{Y} - \hat{\boldsymbol{\nu}}\mathbf{1}^{'} - (\widehat{\boldsymbol{B}}\boldsymbol{Z}))\mathbf{1},$$

$$\implies \hat{\boldsymbol{\nu}}_j(\lambda) = \overline{Y}_{k\cdot} - \widehat{\boldsymbol{B}}\overline{Z}_{k\cdot},$$

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

$$\frac{1}{2}||\boldsymbol{Y} - (\bar{\boldsymbol{Y}} - \boldsymbol{B}\bar{\boldsymbol{Z}})\mathbf{1}^{'} - \boldsymbol{B}\boldsymbol{Z}||_F^2 + \lambda||\boldsymbol{B}||_1 \tag{9.1}$$

$$=\frac{1}{2}||(\boldsymbol{Y} - \bar{\boldsymbol{Y}}\mathbf{1}^{'}) - \boldsymbol{B}(\boldsymbol{Z} - \bar{\boldsymbol{Z}}\mathbf{1}^{'})||_F^2 + \lambda||\boldsymbol{B}||_1. \tag{9.2}$$

Where $\bar{\mathbf{Y}}$ is a $k \times T$ matrix of row means and $\bar{\mathbf{Z}}$ is a $kp \times T$ matrix of row means. Note that since the Laplacian with respect to $\boldsymbol{\nu}$ is -T, $\hat{\boldsymbol{\nu}}$ is a maximum.

9.2 Solution Strategies

In the following sections, assume that $\boldsymbol{Y}$ and $\boldsymbol{Z}$ are centered as in Equation (9.2).

9.2.1 Lasso-VAR

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 $\boldsymbol{B}_{jr}$ as

$$\min_{\boldsymbol{B}_{jr}} \frac{1}{2}(\boldsymbol{Y}_{jt} - \sum_{\ell \neq r} \boldsymbol{B}_{j\ell}\boldsymbol{Z}_{\ell t} - \boldsymbol{B}_{jr}\boldsymbol{Z}_{jt})^2 + \lambda|\boldsymbol{B}_{jr}|. \tag{9.3}$$

Let $\boldsymbol{R}_t = \boldsymbol{Y}_{jt} - \sum_{\ell \neq r} \boldsymbol{B}_{j\ell}\boldsymbol{Z}_{\ell t}$ denote the partial residual. Then, we can rewrite Equation (9.3) as

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

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

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

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 procedure is detailed in Algorithm 1.

9.2.2 Group Lasso-VAR

Rather than vectorizing into a univariate least squares problem, we can again exploit the matrix structure for considerable compuational gains. We want to solve the subproblem

$$\frac{1}{2}||\boldsymbol{R}_{-q} - \boldsymbol{B}_q \boldsymbol{Z}_q||_F^2 + \lambda ||\boldsymbol{B}_q||_F. \tag{9.4}$$

Where $\mathbf{R}_q = \boldsymbol{Y} - \boldsymbol{B}_{-q}\boldsymbol{Z}_{-q}$ again represents the partial residual. Taking the gradient of $||\mathbf{R}_{-q} - \boldsymbol{B}_q\boldsymbol{Z}_q||_F^2$ with respect to $\boldsymbol{B}_q$ results in:

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

The subgradient with respect to $\boldsymbol{B}_q$ then is

$$\boldsymbol{B}_q\boldsymbol{Z}_q\boldsymbol{Z}_q^{'} - \boldsymbol{R}_{-q}\boldsymbol{Z}_q^{'} + \lambda\omega(\boldsymbol{B}_q)$$

Where ω is defined as:

$$\omega(\boldsymbol{B}_q) = \begin{cases} \frac{\boldsymbol{B}_q}{||\boldsymbol{B}_q||_F} & \boldsymbol{B}_q \neq 0 \\ \{U : ||U||_F \leq 1\} & \boldsymbol{B}_q = 0 \end{cases}$$

Consider the case where $\hat{\boldsymbol{B}}_q = \mathbf{0}$. Then

$$\begin{aligned}
&\frac{\hat{\boldsymbol{B}}_q\boldsymbol{Z}_q\boldsymbol{Z}_q^{'} - \boldsymbol{R}_{-q}\boldsymbol{Z}_q^{'}}{\lambda} \in \{U \text{ s.t. } ||U||_F \leq 1\},\\
&\iff ||\hat{\boldsymbol{B}}_q\boldsymbol{Z}_q\boldsymbol{Z}_q^{'} - \boldsymbol{R}_{-q}\boldsymbol{Z}_q^{'}||_F \leq \lambda,\\
&\iff ||\boldsymbol{R}_{-q}\boldsymbol{Z}_q^{'}||_F \leq \lambda,\\
&\iff \hat{\boldsymbol{B}}_q = \mathbf{0}.
\end{aligned}$$

We can conclude that $\hat{\boldsymbol{B}}_q = 0 \iff ||\boldsymbol{R}_{-q}^{'}\boldsymbol{Z}_q^{'}||_2 \leq \lambda$.

Assuming $\hat{\boldsymbol{B}}_q \neq 0$, we have that

$$\boldsymbol{B}_q\boldsymbol{Z}_q\boldsymbol{Z}_q^{'} - \boldsymbol{R}_{-q}\boldsymbol{Z}_q^{'} + \lambda(\frac{\hat{\boldsymbol{B}}_q}{||\hat{\boldsymbol{B}}_q||_F}) = 0, \tag{9.5}$$

$$\boldsymbol{B}_q\boldsymbol{Z}_q\boldsymbol{Z}_q^{'} + \lambda(\frac{\hat{\boldsymbol{B}}_q}{||\hat{\boldsymbol{B}}_q||_F}) = \boldsymbol{R}_{-q}\boldsymbol{Z}_q^{'}, \tag{9.6}$$

$$\boldsymbol{B}_q\left(\boldsymbol{Z}_q\boldsymbol{Z}_q^{'} + \frac{\lambda}{||\boldsymbol{B}_q||_F}\boldsymbol{I}_k\right) = \boldsymbol{R}_{-q}\boldsymbol{Z}_q^{\prime}. \tag{9.7}$$

Now, since $\boldsymbol{Z}_q\boldsymbol{Z}_q^{'} \succ 0$ and $\lambda > 0, \boldsymbol{Z}_q\boldsymbol{Z}_q^{'} + \frac{\lambda}{||\boldsymbol{B}_q||_F}\boldsymbol{I}_k \succ 0$, it is possible to create a trust region subproblem which coincides with Equation (9.4). However, we need to transform $\mathbf{R}_{-q}\boldsymbol{Z}_q$ into a scalar. Define

$$\begin{aligned}
\boldsymbol{p}_q &= \text{vec}(\mathbf{R}_{-q}\boldsymbol{Z}_q),\\
\mathbf{X}_q &= \boldsymbol{Z}_q\boldsymbol{Z}_q^{'} \otimes \mathbf{I}_k,\\
\boldsymbol{b}_q &= \text{vec}(\boldsymbol{B}_q).
\end{aligned}$$

Hence, we can rewrite Equation (9.7) as

$$\boldsymbol{b}_q^{'}\left(\mathbf{X}_q + \frac{\lambda}{||\boldsymbol{b}_q||_2}\boldsymbol{I}_{k^2}\right) = \boldsymbol{p}_q.$$

Resulting in the trust-region subproblem

$$\begin{aligned}
\min \frac{1}{2}\boldsymbol{b}_q^{'}\boldsymbol{X}_q\boldsymbol{X}_q^{'}\boldsymbol{b}_q + \boldsymbol{p}_q^{'}\boldsymbol{b}_q,\\
\text{s.t.}||\boldsymbol{b}_q||_2 \leq \Delta,
\end{aligned}$$

in which $\Delta > 0$ is the trust-region radius. By the Karush-Kuhn-Tucker conditions, we must have that: $\lambda(\Delta - ||\boldsymbol{b}_q^*||_2) = 0$, which implies that $||\boldsymbol{b}_q^*||_2 = \Delta$. Then, applying Theorem 4.1 of Nocedal

and Wright [1999], we can conclude that

$$\boldsymbol{b}_q^* = -\left(\boldsymbol{X}_q + \frac{\lambda}{\Delta}\boldsymbol{I}\right)^{-1}\boldsymbol{p}_q. \tag{9.8}$$

These transformations allow for the use of the methodology described in Qin et al. [2010]. Equation (9.8) can also be expressed as $\boldsymbol{b}_q^* = \Delta y_q(\Delta)$, where

$$y_q(\Delta) = -\left(\Delta\boldsymbol{X}_q + \lambda\boldsymbol{I}\right)^{-1}\boldsymbol{p}_q,$$

Note that $||y_q(\Delta)||_2 = 1$. Hence, the optimal Δ can be chosen to satisfy $||y_q(\Delta)||_2 = 1$. We can efficiently compute $||y_q(\Delta)||_2^2$ via an eigen-decomposition of $\boldsymbol{X}_q$

$$||y_q(\Delta)||_2^2 = \sum_i \frac{(\mathbf{w}_i^{'}\boldsymbol{p}_q^{'}\boldsymbol{X}_q)^2}{(\mathbf{v}_i\Delta + \lambda)^2},$$

in which $\boldsymbol{w}_i$ and $\boldsymbol{v}_i$ represent the respective eigenvectors and eigenvalues of $\boldsymbol{X}_q$. Finally, we can determine the optimal Δ by applying Newton's method to find the root of

$$\phi(\Delta) = 1 - \frac{1}{||y_j(\Delta)||_2}. \tag{9.9}$$

The full procedure is outlined in Algorithm 3. Our algorithm organizes iterations around an "active-set" as described in Friedman et al. [2010]. This approach starts by cycling through every $\boldsymbol{B}_q$ once, and then only iterating on the subset of $\boldsymbol{B}$ which 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 parameters are zero.

9.2.3 Extension of Group-Lasso VAR to Own/Other Lags

Since in this scenario the groups are not proper submatrices, Equation 3.3 must be transformed into a least squares problem. In order to do so, we define the following

$$r_{-qq} = \text{vec}(\mathbf{R}_{-qq}), \tag{9.10}$$

$$b_{qq} = \text{vec}(\boldsymbol{B}_{qq}), \tag{9.11}$$

$$\boldsymbol{M}_{qq} = Z_{qq}^{'} \otimes I_k. \tag{9.12}$$

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

$$\min_{a_q} \frac{1}{2}||\boldsymbol{M}_{qq} b_{qq} + r||_2^2 + \sqrt{\rho_{qq}}\lambda||b_{qq}||_2, \tag{9.13}$$

$$= \min_{a_q} \frac{1}{2} r^{'} r + b_{qq}^{'} \boldsymbol{M}_{qq}^{'} \boldsymbol{M}_{qq} b_{qq} + r^{'} \boldsymbol{M}_{qq} b_{qq} + \sqrt{\rho_{qq}}\lambda||b_{qq}||_2, \tag{9.14}$$

$$= \min_{b_{qq}} \frac{1}{2} b_{qq}^{'} \boldsymbol{M}_{qq}^{'} \boldsymbol{M}_{qq} b_{qq} + r^{'} \boldsymbol{M}_{qq} b_{qq} + \sqrt{\rho_{qq}}\lambda||b_{qq}||_2. \tag{9.15}$$

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

$$\frac{\partial}{\partial b_{qq}} = \boldsymbol{M}_{qq}^{'} \boldsymbol{M}_{qq} b_{qq} + \boldsymbol{M}_{qq}^{'} r + \sqrt{\rho_{qq}}\lambda\omega(b),$$

where ω is defined as

$$\omega(s) \in \begin{cases} \{\frac{s}{||s||_2}\} & s \neq 0 \\ \{u \text{ s.t. } ||u||_2 \leq 1\} & s = 0. \end{cases}$$

Thus, we can apply a slightly adapted version of Algorithm 3.

9.2.4 Sparse Group-Lasso VAR

As with the Group Lasso, we will consider the one-block subproblem:

$$\min_{\boldsymbol{B}_q} \frac{1}{2k}||\mathbf{R}_{-q} - \boldsymbol{B}_q \boldsymbol{Z}_q||_F^2 + (1-\alpha)\lambda||\boldsymbol{B}_q||_F + \alpha\lambda||\boldsymbol{B}_q||_1. \tag{9.16}$$

Since the inclusion of within-group sparsity does not allow for separability, coordinate descent is no longer appropriate, therefore, following Simon et al. [2013] our solution to Equation 3.4 will use gradient descent methods. We express Equation 9.16 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 $\boldsymbol{B}^0$ (for notational ease, let $\ell(\boldsymbol{B})$ represent the unpenalized loss function, $\boldsymbol{B} \equiv \boldsymbol{B}_q$ and $P(\boldsymbol{B})$ represent the penalty term)

$$\begin{aligned} M(\boldsymbol{B}, \boldsymbol{B}^0) &= \ell(\boldsymbol{B}^0) + \text{vec}(\boldsymbol{B} - \boldsymbol{B}^0)^{'} \text{vec}(\nabla \ell(\boldsymbol{B}^0)) + \frac{1}{2h}||\boldsymbol{B} - \boldsymbol{B}^0||_F^2 + P(\boldsymbol{B}), \\ &= \frac{1}{2k}||\boldsymbol{R}_{-q} - \boldsymbol{B}^0 \boldsymbol{Z}_q||_F^2 + \langle \boldsymbol{B} - \boldsymbol{B}^0, (\boldsymbol{B}^0 \boldsymbol{Z}_q - \boldsymbol{R}_{-q} \boldsymbol{Z}_q^{'}) \boldsymbol{Z}_q^{'} \rangle + \frac{1}{2h}||\boldsymbol{B} - \boldsymbol{B}^0||_F^2 + P(\boldsymbol{B}). \end{aligned}$$

Where h represents the step size. Our objective function is then

$$\begin{aligned} \hat{\boldsymbol{B}} &= \text{argmin}\, M(\boldsymbol{B}, \boldsymbol{B}^0), \\ &= \underset{\boldsymbol{B}}{\text{argmin}} \frac{1}{2h}||\boldsymbol{B} - \left(\boldsymbol{B}^0 - h(\boldsymbol{B}^0 \boldsymbol{Z}_q - \boldsymbol{R}_{-q}) \boldsymbol{Z}_q^{'}\right)||_F^2 + P(\boldsymbol{B}). \end{aligned}$$

Then, generalizing the arguments outlined by Simon et al. [2013], we can infer that

$$\hat{B} = \left(1 - \frac{h(1-\alpha)\lambda}{||ST(\boldsymbol{B}^0 - h(\boldsymbol{B}^0 \boldsymbol{Z}_q - \boldsymbol{R}_{-q}) \boldsymbol{Z}_q^{'}, h\alpha\lambda)||_F}\right)_+ ST(\boldsymbol{B}^0 - h(\boldsymbol{B}^0 \boldsymbol{Z}_q - \boldsymbol{R}_{-q}) \boldsymbol{Z}_{-q}^{'}, h\alpha\lambda).$$

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 it is too large, the algorithm may diverge. The conventional method for determing step size, described in Simon et al. [2013] and Beck and Teboulle [2009], is to decrease h until

$$\ell(\hat{B}, h) \leq \ell(\boldsymbol{B}) + \text{vec}(\nabla_q)^{'} \text{vec}(\Delta_{l,h}) + \frac{1}{2h}||\Delta_{l,h}||_F^2. \tag{9.17}$$

However, as noted in section 5.3 of Becker et al. [2011], Equation (9.17) has severe cancellation

errors when $\ell(\hat{\boldsymbol{B}}, h) \approx \ell(\boldsymbol{B}, h)$. They posit an alternative test, iterating until

$$\ell(\hat{\boldsymbol{B}}, h) \leq \frac{1}{2hk} ||\Delta_{l,h}||_F^2. \tag{9.18}$$

They recommend a hybrid approach: choosing Equation (9.17) when $\ell(\boldsymbol{B}, t) - \ell(\hat{\boldsymbol{B}}, t) \geq \gamma \ell(\hat{\boldsymbol{B}}, t)$, for some small $\gamma > 0$ and choosing Equation (9.18) 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 instead analytically derive 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 $\boldsymbol{B}_q$. We have that

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

The last inequality follows from the sub-multiplicity of the matrix 2-norm. Therefore, we can conclude that the Lipschitz constant is $||\boldsymbol{Z}_q \boldsymbol{Z}_q^{'}||_2 = \sqrt{\sigma_{\max}(\boldsymbol{Z}_q \boldsymbol{Z}_q^{'})}$, i.e. the largest eigenvalue of $\boldsymbol{Z}_q \boldsymbol{Z}_q^{'} \equiv \mathbf{X}$, which has dimension $k \times k$. Since $\mathbf{X}$ 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 computing the entire eigensystem. Moreover, we retain the corresponding eigenvector produced by this procedure to use as a "warm start" which substantially decreases the amount of

time required to compute the maximal eigenvalue at each time point in the cross-validation and evaluation stages.

The inner loop of of the Sparse Group-Lasso VAR procedure is detailed in Algorithm (4). An outline of the algorithm is below:

1. Iterate through all groups. Within each group:
   (a) Check if the group's coefficients are identically zero via the condition: $||(\boldsymbol{B}_q\boldsymbol{Z}_q - \boldsymbol{R}_{-q})\boldsymbol{Z}_q^{'}||_F \leq (1-\alpha)\lambda$.
   (b) If not, go to the inner loop (Algorithm 4).
   (c) Repeat until convergence.

In a manner similar to Algorithm 3, an "active-set" approach is used to minimize computation time.

9.3 Penalty Grid Selection

Table 9: Starting values of the penalty grid for each procedure.

Structure	Starting Value of Λ_{Grid}
Lasso	$\max(\boldsymbol{Z}\boldsymbol{Y}^{'})$
Block Group Lasso	$\max_q(\boldsymbol{Z}_q\boldsymbol{Y}^{'})$
Block Sparse Group Lasso	$\max_q(\boldsymbol{Z}_q\boldsymbol{Y}^{'}\alpha)$
Own/Other Group Lasso	$\max_q \frac{(\boldsymbol{Z}_q \otimes I_k)\text{vec}(\boldsymbol{Y}^{'})}{\sqrt{\rho_q}}$
Sparse Own/Other Group Lasso	$\max_q \frac{(\boldsymbol{Z}_q \otimes I_k)\text{vec}(\boldsymbol{Y}^{'})}{\sqrt{\rho_q}}\alpha$

9.4 Description of Macroeconomics Time Series

This table is a slight modification from that of Koop [2011].

Table 10: Description of the 19 macroeconomic indicators used in our analysis

Abbreviation	Series Description
GDP251	Real GDP, quantity index
CES002	Employees on nonfarm payrolls-total private
CPIAUCSL	Consumer-Price Index: All Items
PSCCOMR	Real Spot Market Price Index: All Commodities
IPS10	Industrial Production Index- Total Index
GDP252	Real Personal Consumption Expenditures: Quantity Index
LHUR	Unemployment Rate: All Workers 16 Years and Over
HSFR	Housing Starts: Non Farm(1947-1958) Total Farm and Non-Farm (1959-)
UTL11	Capacity Utilization: Manufacturing
PWFSA	Producer Price Index: Finished Goods
GDP273	Personal Consumption Expenditures: Price Index
CES275R	Real Average Hourly Earnings of Production or Non-Supervisory Workers
FYFF	Federal Funds Rate
FM2	Money Stock: M2
FM1	Money Stock: M1
FMRRA	Depository Institution Reserves: Total (Adjusted for Reserve Requirement Changes)
FSPIN	S&P's Common Stock Price Index: Industrials
FYGT10	Interest Rate: US Treasury Constant Maturity-10 Year
EXRUS	United States Effective Exchange Rate

9.5 Algorithms

Algorithm 1 LASSO-VAR(p)

Require: $\boldsymbol{Y}, \boldsymbol{Z}, \boldsymbol{B}^{\text{INI}}, M, R, \epsilon$
$\tilde{\mathbf{Y}} \leftarrow \boldsymbol{Y} - \bar{\boldsymbol{Y}}\mathbf{1}^{'}$
$\tilde{\mathbf{Z}} \leftarrow \boldsymbol{Z} - \bar{\boldsymbol{Z}}\mathbf{1}^{'}$
$\boldsymbol{\Lambda}_{\text{grid}} \leftarrow \exp(\text{sequence}(\text{start} = \log(\max(\tilde{\mathbf{Z}}\tilde{\mathbf{Y}}^{'})), \text{end} = \frac{\log(\max(\tilde{\mathbf{Z}}\tilde{\mathbf{Y}}^{'})}{R}, \text{length} = M))$
$\boldsymbol{B}^{\text{OLD}} \leftarrow \boldsymbol{B}^{\text{INI}}$

for m in $1 : M$ **do**
  $\lambda \leftarrow \Lambda(m)$
  while threshold $> \epsilon$ **do**
    for i in k, j in $kp_{\max}$ **do**
      $R_t \leftarrow Y_{jt} - \sum_{\ell \neq j} \boldsymbol{B}_{j\ell}\boldsymbol{Z}_{\ell t}$
      $B_{ij}^{\text{NEW}} \leftarrow \frac{\text{ST}(\sum_t \mathbf{R}_t \boldsymbol{Z}_{jt}, \lambda)}{\sum_t \boldsymbol{Z}_{jt}^2}$
    end for
    threshold=$\max(\frac{|\text{vec}(\boldsymbol{B}^{\text{OLD}}) - \text{vec}(\boldsymbol{B}^{\text{NEW}})|}{1+|\text{vec}(\boldsymbol{B}^{\text{OLD}})|})$
    $\boldsymbol{B}^{\text{OLD}} \leftarrow \boldsymbol{B}^{\text{NEW}}$
  end while
  $\hat{\nu} \leftarrow \bar{\boldsymbol{Y}} - \boldsymbol{B}^{\text{NEW}}\bar{\boldsymbol{Z}}$
  $\boldsymbol{B}_{\text{Array}}\{m\} \leftarrow (\hat{\nu}, \boldsymbol{B}^{\text{NEW}})$
end for
return $\boldsymbol{B}_{\text{Array}}$

Algorithm 2 LASSO-VAR(p) Cross-Validation

Require: $\boldsymbol{Y}, \boldsymbol{Z}, \boldsymbol{B}_{\text{array}}^{\text{INI}}, \Lambda_{\text{grid}}$,Relaxed
$\boldsymbol{Y} \leftarrow \bar{\boldsymbol{Y}}\mathbf{1}^{'}$
$\boldsymbol{Z} \leftarrow \bar{\boldsymbol{Z}}\mathbf{1}^{'}$
$\boldsymbol{B}_{\text{array}}^{\text{LAST}} \leftarrow \boldsymbol{B}_{\text{array}}^{\text{INI}}$

for j in $1 : \text{length}(T_2 - T_1)$ **do**
  $\boldsymbol{Y}_{\text{TRAIN}}^{(j)} \leftarrow \boldsymbol{Y}_{1:(T_1+j-p_{\max}+1)}$
  $\boldsymbol{Z}_{\text{TRAIN}}^{(j)} \leftarrow \boldsymbol{Z}_{,1:(T_1+j-p_{\max}+1)}$
  $\boldsymbol{B}^{\text{NEW}} \leftarrow \text{Lasso-VAR}(\boldsymbol{Y}_{\text{TRAIN}}^{(j)}, \boldsymbol{Z}_{\text{TRAIN}}^{(j)}, \boldsymbol{B}_{\text{array}}^{\text{LAST}}, \lambda_{\text{grid}}, \gamma)$
  if Relaxed is TRUE **then**
    $\boldsymbol{B}_{\text{array}}^{\text{NEW}} \leftarrow \text{Relaxed}(\boldsymbol{Y}_{\text{TRAIN}}^{(j)}, \boldsymbol{Z}_{\text{TRAIN}}^{(j)}, \boldsymbol{B}_{\text{array}}^{\text{NEW}})$
  end if
  for i in λ_{Grid} **do**
    $SSFE^{(i,j)} \leftarrow ||\boldsymbol{Y}_j - \mathbf{B_i}\mathbf{Z}_{,1:(j-p_{\max})}||_2^2$
    $MSSFE^{(i)} \leftarrow \frac{1}{T_2 - T_1}\sum_j SSFE\{i, j\}$
  end for
end for
$\boldsymbol{B}_{\text{array}}^{\text{LAST}} \leftarrow \boldsymbol{B}_{\text{array}}^{\text{NEW}}$
return $\lambda_{\min \text{MSSFE}}$

Algorithm 3 Group LASSO-VAR(p)

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

Define:

$$\mathbf{M}_g = \boldsymbol{Z}_g \boldsymbol{Z}_g^{'}$$
$$\mathbf{X}_g = \mathbf{M}_g \otimes \mathbf{I}_k$$

for $\lambda \in \Lambda$ **do**
  $\boldsymbol{B}_{\lambda,\mathcal{A}} \leftarrow \boldsymbol{B}_{\lambda,\text{INI}}, \mathcal{A}_\lambda \leftarrow \mathcal{A}_{\lambda,\text{INI}}$
  repeat
    $\boldsymbol{B}_{\lambda,\mathcal{A}} \leftarrow$ThresholdUpdate$(\mathcal{A}, \boldsymbol{B}_{\lambda,\mathcal{A}}, \lambda)$
    $\boldsymbol{B}_{\lambda,\mathcal{A}_{\text{FULL}}}, \mathcal{A}_\lambda \leftarrow$BlockUpdate$(\mathcal{A}_{\text{FULL}}, \boldsymbol{B}_{\lambda,\mathcal{A}}, \lambda)$
  until $\boldsymbol{B}_{\lambda,\mathcal{A}} = \boldsymbol{B}_{\lambda,\mathcal{A}_{\text{FULL}}}$
  $\hat{\nu} \leftarrow \bar{\boldsymbol{Y}} - \boldsymbol{B}_{\lambda,\mathcal{A}} \bar{\boldsymbol{Z}}$
end for
return $\hat{\nu}, B_\Lambda, A_\Lambda$
procedure BlockUpdate$(\mathcal{G}, \boldsymbol{B}_{\text{INI}}, \lambda)$
  $\boldsymbol{B} \leftarrow \boldsymbol{B}_{\text{INI}}$
  for $g \in \mathcal{G}$ **do**
    $\mathbf{R} \leftarrow \boldsymbol{B}_{-g} \boldsymbol{Z}_{-g} - \boldsymbol{Y}$
    $\mathbf{p} \leftarrow \mathbf{R} \boldsymbol{Z}_q^{'}$
    if $||p||_F \leq \lambda$ **then**
      $\boldsymbol{B}^* \leftarrow \mathbf{0}_p$
      $\mathcal{A}_g \leftarrow \emptyset$
    end if
    if $||p||_2 > \lambda$ **then**
      $\Delta \leftarrow$ the root of $\phi(\Delta)$ defined in (9.9)
      $\boldsymbol{B}_g \leftarrow -(\mathbf{X}_g + \frac{\lambda}{\Delta}\mathbf{I}_{k^2})^{-1}\mathbf{p}$
      $\mathcal{A}_g \leftarrow g$
    end if
  end for
return $\boldsymbol{B}_\lambda$, $\mathcal{A}$
end procedure
procedure ThresholdUpdate$(\mathcal{A}_\lambda, \boldsymbol{B}_{\lambda,\text{INI}}, \lambda)$
  if $\mathcal{A} = \emptyset$ **then return** $\mathbf{0}_{k \times kp}$
  end if
  if $\mathcal{A} \neq \emptyset$ **then**
    $\boldsymbol{B}_{\lambda,\text{OLD}} \leftarrow \boldsymbol{B}_{\lambda,\text{INI}}$
    repeat
      $\boldsymbol{B}_{\lambda,\text{NEW}}, \mathcal{A}_\lambda \leftarrow$ 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 4 Sparse LASSO-VAR(p) inner loop

Require: $\boldsymbol{B}^0, \boldsymbol{Z}_q, \boldsymbol{R}_{-q}$

repeat

l←1

$\boldsymbol{G}_q \leftarrow \frac{(\boldsymbol{B}^0 \boldsymbol{Z}_q - \boldsymbol{R}_{-q})\boldsymbol{Z}_q^{'}}{k}$

$\ell(U(\boldsymbol{B})) \leftarrow ||\boldsymbol{B}^0||$

$\theta^1 \leftarrow \boldsymbol{B}^0$

h← $1/\lambda_{\max}(\boldsymbol{Z}_q \boldsymbol{Z}_q^{'})$

$U(\boldsymbol{B}) \leftarrow \left(1 - \frac{h(1-\alpha)\lambda}{||ST(\boldsymbol{B}^0 - t(\boldsymbol{B}^0 \boldsymbol{Z}_q - \boldsymbol{R}_{-q})\boldsymbol{Z}_q^{'}, h\alpha\lambda)||_F}\right)_+ ST(\boldsymbol{B}^0 - h(\boldsymbol{A}^0 \boldsymbol{Z}_q - \boldsymbol{R}_{-q})\boldsymbol{Z}_{-q}^{'}, h\alpha\lambda)$

$\boldsymbol{\theta}^{l+1} \leftarrow U(\boldsymbol{B})$

$\boldsymbol{B}^{l+1} \leftarrow \boldsymbol{\theta}^{l+1} + \frac{l}{l+3}(\boldsymbol{\theta}^{l+1} - \boldsymbol{\theta}^l)$

l=l+1

until Convergence of $||\boldsymbol{B}^{l+1} - \boldsymbol{B}^l||_F$

Algorithm 5 Relaxed Feasible Generalized Least Squares

Require: $\boldsymbol{Y}, \boldsymbol{Z}, R_1, \ldots, R_k, \mathbf{R}$

$K \leftarrow [\boldsymbol{Z}^{'}, \boldsymbol{Y}]$

for i in 1:k **do**

$R1_i \leftarrow \begin{bmatrix} R_i & e_{\text{nrow}(R_i)+i} \\ \mathbf{0} & 0 \end{bmatrix}$

$K1 \leftarrow KR1_i$

$q \leftarrow$ ncol(K1)

$\delta \leftarrow (q^2 + q + 1)\sqrt{\epsilon_{\text{machine}}}$

$S \leftarrow \sqrt{\delta}\sqrt{\text{Diag}||K2_{\cdot j}||_2^2}$

$R2 \leftarrow QR.R(\begin{bmatrix} K1 \\ S \end{bmatrix})$

$R2_{11} \leftarrow R2_{1:\text{ncol}(R_i),1:\text{ncol}(R_i)}$

$R2_{12} \leftarrow R2_{1:\text{ncol}(R_i),\text{ncol}(R_i):\text{ncol}(R_i)+1}$

$\hat{\boldsymbol{B}}_i^{\text{Relaxed}} \leftarrow R_i(R2_{11}^{-1} R2_{12})$

end for

if FGLS is true **then**

$\bar{\Sigma}_u = \frac{1}{T-kp-1}(\boldsymbol{Y} - \hat{\boldsymbol{B}}^{\text{Relaxed}}\boldsymbol{Z})(\boldsymbol{Y} - \hat{\boldsymbol{B}}^{\text{Relaxed}}\boldsymbol{Z})^{'}$

$\hat{\boldsymbol{B}}^{\text{FGLS}} = [\mathbf{R}^{'}(\mathbf{Z}\mathbf{Z}^{'} \otimes \bar{\Sigma}_u^{-1})\mathbf{R}]^{-1}\mathbf{R}^{'}(\boldsymbol{Z} \otimes \bar{\Sigma}_u^{-1})\text{vec}(\boldsymbol{Y})$

end if

return $\mathbf{R}\hat{\boldsymbol{B}}^{\text{Relaxed}}$