Efficient Selection of Hyperparameters in Large Bayesian VARs Using Automatic Differentiation

Joshua C.C. Chan
Department of Economics
Purdue University and UTS

Liana Jacobi
Department of Economics
University of Melbourne

Dan Zhu
Department of Business Statistics and Econometrics
Monash University

June 2019

Abstract

Large Bayesian VARs with the natural conjugate prior are now routinely used for forecasting and structural analysis. It has been shown that selecting the prior hyperparameters in a data-driven manner can often substantially improve forecast performance. We propose a computationally efficient method to obtain the optimal hyperparameters based on Automatic Differentiation, which is an efficient way to compute derivatives. Using a large US dataset, we show that using the optimal hyperparameter values leads to substantially better forecast performance. Moreover, the proposed method is much faster than the conventional grid-search approach, and is applicable in high-dimensional optimization problems. The new method thus provides a practical and systematic way to develop better shrinkage priors for forecasting in a data-rich environment.

Keywords: automatic differentiation, vector autoregression, optimal hyperparameters, forecasts, marginal likelihood

JEL classifications: C11, C53, E37

1 Introduction

Since the seminal paper of Banbura, Giannone, and Reichlin (2010) showed that it is feasible to estimate large Bayesian vector autoregressions (BVARs) with over 100 variables, there has been a lot of interest in using large BVARs for forecasting and structural analysis. A few prominent examples include Carriero, Kapetanios, and Marcellino (2009), Koop (2013), Koop and Korobilis (2013) and Banbura, Giannone, Modugno, and Reichlin (2013). One key aspect of these large BVARs is the use of shrinkage priors that formally incorporate sensible non-data information, and one popular way to do so is the Minnesota-type natural conjugate prior that gives rise to a range of analytical results, including closed-form expressions of the marginal likelihood.[1] These analytical results are later used in Carriero, Clark, and Marcellino (2016) and Chan (2018) to develop efficient sampling algorithms to estimate large BVARs with flexible error covariance structures, such as stochastic volatility, serially correlated and non-Gaussian errors.

The natural conjugate prior depends on a few hyperparameters that control the degree of shrinkage and they are typically fixed at some subjectively chosen values. Alternatively, a data-based approach to select these hyperparameters might be more appealing as it reduces the number of important subjective choices required from the user. For example, Del Negro and Schorfheide (2004), Schorfheide and Song (2015) and Carriero, Clark, and Marcellino (2015) obtain the optimal hyperparameters by maximizing the marginal likelihood over a grid of possible values.[2] This grid-search approach is also incorporated in the Bayesian Estimation, Analysis and Regression toolbox (BEAR) MATLAB toolbox developed by the European Central Bank (Dieppe, Legrand, and Van Roye, 2016). However, this approach is typically time-consuming for low-dimensional problems, and it becomes computationally infeasible in higher dimensions, as the number of marginal likelihood evaluations increases exponentially in the number of hyperparameters.

We propose a computationally efficient method to obtain the optimal hyperparameters based on Automatic Differentiation (AD), which is an efficient way to compute derivatives

[1] Early seminal works of shrinkage priors were developed by Doan, Litterman, and Sims (1984) and Litterman (1986). Similar shrinkage priors for structural VARs are formulated in Leeper, Sims, and Zha (1996) and Sims and Zha (1998).

[2] Giannone, Lenza, and Primiceri (2015) show that a data-based approach of selecting the hyperparameters—compared to the convectional method of fixing them to some ad hoc values—can substantially improve the forecast performance of large BVARs.

based on the chain rule. More specifically, we apply AD to calculate the gradient of the marginal likelihood with respect to the hyperparameters, which is then used as an input in an optimization routine. By computing the gradient efficiently using AD, the proposed method is substantially faster than the conventional grid-search approach.

AD is "automatic" in the sense that for any function that maps an input vector into an output vector, there is an automatic way of evaluating its gradient without manually deriving the symbolic formula of the derivatives. More precisely, AD decomposes the function into elementary functions, and then applies the chain rule to obtain the gradient of the original function. In principle, the gradient can be computed by other commonly used methods, such as numerical finite-difference methods or symbolic differentiation. But a carefully designed AD approach is often substantially faster than these two alternatives.

While AD-based methods have been widely used in Financial Mathematics, they have only been recently introduced in Econometrics by Jacobi, Joshi, and Zhu (2018), who develop an AD approach for a comprehensive prior robustness and convergence analysis of Markov chain Monte Carlo output in the context of Bayesian estimation. Chan, Jacobi, and Zhu (2019b) extend this framework further to predictive simulation—specifically to analyze the sensitivities of point and interval forecasts from BVARs on prior hyperparameters. We continue this line of research by applying AD to obtain optimal hyperparameters and evaluate the forecast performance of the resulting BVAR.

We illustrate the new methodology with a forecasting exercise that involves 18 macroeconomic and financial variables. We first document the computational gains of using the proposed AD-based approach to obtain the optimal hyperparameters compared to the conventional grid-search approach. We show that while the proposed approach remains relatively fast for high-dimensional optimization problems, a brute-force grid-search approach would take hours or even days, and is simply impractical.

We then present forecasting results to show that the optimal hyperparameter values obtained do in fact lead to better forecast performance. Our findings therefore highlight the empirical relevance of selecting optimal hyperparameters for forecasting using large BVARs. More importantly, since the proposed method works well in high dimensions, it provides a practical and systematic way to develop better shrinkage priors for forecasting in a data-rich environment.

The rest of this paper is organized as follows. Section 2 first gives an overview of the BVAR and the natural conjugate prior, as well as a few associated analytical results. We then introduce an AD-based method to obtain the optimal hyperparameters by maximizing the marginal likelihood in Section 3. It is followed by a macroeconomic forecsating exercise to illustrate the usefulness of the proposed approach in Section 4. Lastly, Section 5 concludes and briefly discusses some future research directions.

2 The Bayesian VAR

In this section we provide background of the Bayesian VAR (BVAR), the associated natural conjugate prior, and a few useful analytical results. For textbook treatment, see, e.g., Koop and Korobilis (2010), Karlsson (2013) or Chan (2019). Let $\mathbf{y}_t = (y_{1t}, \dots, y_{nt})'$ denote the $n \times 1$ vector of dependent variables at time t. Then, a standard VAR(p) for $t = 1, \dots, T$ is given by:

$$\mathbf{y}_t = \mathbf{a} + \mathbf{A}_1 \mathbf{y}_{t-1} + \cdots + \mathbf{A}_p \mathbf{y}_{t-p} + \mathbf{u}_t,$$

where $\mathbf{a}$ is an $n \times 1$ vector of intercepts, $\mathbf{A}_1, \dots, \mathbf{A}_p$ are $n \times n$ coefficient matrices and $\mathbf{u}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma})$.

The above system can be written more compactly as follows. First, stack the dependent variables into a $T \times n$ matrix $\mathbf{Y}$ so that its t-th row is $\mathbf{y}_t'$. Let $\mathbf{Z}$ be a $T \times k$ matrix of regressors, where the t-th row is $\mathbf{z}_t' = (1, \mathbf{y}_{t-1}', \dots, \mathbf{y}_{t-p}')$ so that $k = 1 + np$. Next, let $\mathbf{A} = (\mathbf{a}, \mathbf{A}_1, \dots, \mathbf{A}_p)'$ denote the $k \times n$ matrix of VAR coefficients. Then, we can write the above VAR(p) as follows:

$$\mathbf{Y} = \mathbf{Z}\mathbf{A} + \mathbf{U}, \tag{1}$$

where $\mathbf{U}$ is a $T \times n$ matrix of innovations in which the t-th row is $\mathbf{u}_t'$. If follows that $\text{vec}(\mathbf{U}) \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma} \otimes \mathbf{I}_T)$, where $\text{vec}(\cdot)$ vectorizes a matrix into a column vector by stacking the columns and $\otimes$ denotes the Kronecker product. Finally, the likelihood function is given by

$$p(\mathbf{Y} \mid \mathbf{A}, \mathbf{\Sigma}) = (2\pi)^{-\frac{Tn}{2}} |\mathbf{\Sigma}|^{-\frac{T}{2}} \mathrm{e}^{-\frac{1}{2}\text{tr}\left(\mathbf{\Sigma}^{-1}(\mathbf{Y}-\mathbf{Z}\mathbf{A})'(\mathbf{Y}-\mathbf{Z}\mathbf{A})\right)}, \tag{2}$$

where $\text{tr}(\cdot)$ denotes the trace operator.

2.1 The Natural Conjugate Prior

The normal-inverse-Wishart prior is a joint distribution on $(\mathbf{A}, \mathbf{\Sigma})$ that is formed by combining a marginal distribution on $\mathbf{\Sigma}$ with a conditional distribution on $\mathbf{A}$ given $\mathbf{\Sigma}$. More specifically, the marginal distribution on $\mathbf{\Sigma}$ is inverse-Wishart and the conditional distribution on $\mathbf{A}$ is normal:

$$\mathbf{\Sigma} \sim \mathcal{IW}(\nu_0, \mathbf{S}_0), \quad (\text{vec}(\mathbf{A}) \,|\, \mathbf{\Sigma}) \sim \mathcal{N}(\text{vec}(\mathbf{A}_0), \mathbf{\Sigma} \otimes \mathbf{V}_\mathbf{A}),$$

and we write $(\mathbf{A}, \mathbf{\Sigma}) \sim \mathcal{NIW}(\mathbf{A}_0, \mathbf{V}_\mathbf{A}, \nu_0, \mathbf{S}_0)$. The corresponding joint density function is given by

$$p(\mathbf{A}, \mathbf{\Sigma}) = c|\mathbf{\Sigma}|^{-\frac{\nu_0+n+k+1}{2}} \text{e}^{-\frac{1}{2}\text{tr}(\mathbf{\Sigma}^{-1}\mathbf{S}_0)} \text{e}^{-\frac{1}{2}\text{tr}\left(\mathbf{\Sigma}^{-1}(\mathbf{A}-\mathbf{A}_0)'\mathbf{V}_\mathbf{A}^{-1}(\mathbf{A}-\mathbf{A}_0)\right)}, \tag{3}$$

where $c = (2\pi)^{-nk/2} 2^{-n\nu_0/2} |\mathbf{V}_\mathbf{A}|^{-n/2} \Gamma_n(\nu_0/2)^{-1} |\mathbf{S}_0|^{\nu_0/2}$, and $\Gamma_n(\cdot)$ denotes the multivariate gamma function. This normal-inverse-Wishart prior is commonly known as the natural conjugate prior and can be traced back to Zellner (1971).

The hyperparameters of this normal-inverse-Wishart prior are $\mathbf{A}_0, \mathbf{V}_\mathbf{A}, \nu_0$, and $\mathbf{S}_0$. For large systems it is important to choose these hyperparameters carefully to induce shrinkage. Below we describe one common way to elicit these hyperparameters.

First, since in our empirical application we will be working with data in growth rates, we set $\text{vec}(\mathbf{A}_0) = \mathbf{0}$ to shrink the VAR coefficients to zero. The strength of shrinkage is controlled by the prior covariance matrix $\mathbf{V}_\mathbf{A}$. Inspired by the Minnesota prior, here we assume $\mathbf{V}_\mathbf{A}$ to be diagonal with the i-th diagonal element $v_{\mathbf{A},ii}$ set as:

$$v_{\mathbf{A},ii} = \begin{cases} \frac{\kappa_1}{l^{\kappa_2} s_r^2}, & \text{for the coefficient on the } l\text{-th lag of variable } r \\ \kappa_3, & \text{for an intercept} \end{cases}$$

where s_r^2 is the sample variance of the residuals from an AR(p) model for the variable r. Hence, we simplify the task of eliciting $\mathbf{V}_\mathbf{A}$ by choosing only three key hyperparameters κ_1, κ_2 and κ_3. The hyperparameters κ_1 and κ_3 control the overall strength of shrinkage for the VAR coefficients and the intercepts, respectively. The hyperparameter κ_2 controls the level of additional shrinkage for coefficients associated to a higher lag lengh l. When

κ_2 is larger, the coefficients associated to higher lag length are shrunk more heavily to zero. A few sets of values for these three hyperparameters are commonly used in the literature. For the baseline model in the empirical application, we follow Kadiyala and Karlsson (1993, 1997) and set $\kappa_1 = 0.05$, $\kappa_2 = 1$ and $\kappa_3 = 100$.

For the marginal prior on $\mathbf{\Sigma}$, it is typically assumed to be relatively noninformative and centered around the sample covariance matrix $\text{diag}(s_1^2, \ldots, s_n^2)$. Since we will be working with large systems in the empirical application, it is of interest to investigate if shrinking the covariance matrix optimally could deliver better forecast performance. To that end, we introduce two additional hyperparameters κ_4 and κ_5, and set $k_{0,\mathbf{\Sigma}} = \kappa_4 + n + 1$ and $\mathbf{S}_{0,\mathbf{\Sigma}} = \kappa_5 \text{diag}(s_1^2, \ldots, s_n^2)$. For the baseline model we use the values $\kappa_4 = 1$ and $\kappa_5 = 1$, which are consistent with typical priors for $\mathbf{\Sigma}$ in the literature. Specifically, $\kappa_4 = 1$ is the smallest integer value such that the prior mean of the inverse-Wishart distribution exists (but the variances do not). Under these values, the prior mean of $\mathbf{\Sigma}$ is $\text{diag}(s_1^2, \ldots, s_n^2)$.

Of course, other more elaborate priors can be considered. For example, instead of using only κ_1 and κ_2 to control the level of shrinkage for all VAR coefficients, one can, for each lag length, introduce an extra hyperparameter to control the shrinkage strength. Or one can consider a non-diagonal scale matrix $\mathbf{S}_{0,\mathbf{\Sigma}}$ that contains a few hyperparameters responsible for controlling the strength of correlations between the innovations. These possibilities can be considered under the proposed framework, and we leave them for future research.

In summary, under this setup, we have altogether 5 key hyperparameters $\kappa_1, \ldots, \kappa_5$ to choose. Finding the optimal values using a grid-search approach for this high-dimensional problem is simply not practical. In Section 3 we will introduce a computationally efficient approach based on Automatic Differentiation to solve this optimization problem.

2.2 Posterior Distribution and Efficient Sampling

Given the likelihood function in (2) and the normal-inverse-Wishart prior in (3), the posterior distribution is also normal-inverse-Wishart. More specifically, it can be shown that the posterior distribution of $(\mathbf{A}, \mathbf{\Sigma})$ is given by (see, e.g., Karlsson, 2013; Chan, 2019):

$$(\mathbf{A}, \mathbf{\Sigma} \,|\, \mathbf{Y}) \sim \mathcal{NIW}(\widehat{\mathbf{A}}, \mathbf{K}_{\mathbf{A}}^{-1}, \nu_0 + T, \widehat{\mathbf{S}}),$$

where

$$\mathbf{K_A} = \mathbf{V_A^{-1}} + \mathbf{Z}'\mathbf{Z}, \quad \widehat{\mathbf{A}} = \mathbf{K_A^{-1}}(\mathbf{V_A^{-1}}\mathbf{A}_0 + \mathbf{Z}'\mathbf{Y}), \quad \widehat{\mathbf{S}} = \mathbf{S}_0 + \mathbf{A}_0'\mathbf{V_A^{-1}}\mathbf{A}_0 + \mathbf{Y}'\mathbf{Y} - \widehat{\mathbf{A}}'\mathbf{K_A}\widehat{\mathbf{A}}.$$

In particular, the posterior means of $\mathbf{A}$ and $\mathbf{\Sigma}$ are respectively $\widehat{\mathbf{A}}$ and $\widehat{\mathbf{S}}/(\nu_0 + T - n - 1)$. In addition, the marginal distribution of $\mathbf{A}$ (unconditional on $\mathbf{\Sigma}$) and the one-step-ahead predictive distribution of $\mathbf{y}_{T+1}$ are both known.

When analytical results are not available, we can estimate the quantities of interest by generating independent draws from the posterior distribution $p(\mathbf{A}, \mathbf{\Sigma} \,|\, \mathbf{Y})$. For example, the h-step-ahead predictive distribution of $\mathbf{y}_{T+h}$ for $h > 1$ is non-standard, but we can obtain posterior draws of $(\mathbf{A}, \mathbf{\Sigma})$ to construct the h-step-ahead predictive distribution via predictive simulation. We can sample $(\mathbf{A}, \mathbf{\Sigma})$ from its posterior distribution in two steps. First, we draw $\mathbf{\Sigma}$ marginally from $(\mathbf{\Sigma} \,|\, \mathbf{Y}) \sim \mathcal{IW}(\nu_0 + T, \widehat{\mathbf{S}})$. Then, given the $\mathbf{\Sigma}$ drawn, we sample $\mathbf{A}$ from the conditional distribution

$$(\mathrm{vec}(\mathbf{A}) \,|\, \mathbf{Y}, \mathbf{\Sigma}) \sim \mathcal{N}(\mathrm{vec}(\widehat{\mathbf{A}}), \mathbf{\Sigma} \otimes \mathbf{K_A^{-1}}).$$

Here the covariance matrix $\mathbf{\Sigma} \otimes \mathbf{K_A^{-1}}$ is of dimension $nk = n(np + 1)$, and sampling from this normal distribution using conventional methods—based on the Cholesky factor of the covariance matrix—would involve $\mathcal{O}(n^6)$ operations. This is especially computationally intensive when n is large. A more efficient way to sample from this conditional distribution is to exploit the Kronecker product structure in the covariance matrix and to use an efficient sampling algorithm to draw from the matrix normal distribution; see, e.g., pp. 301-302 in Bauwens, Lubrano, and Richard (1999) and Carriero, Clark, and Marcellino (2015). This more efficient approach involves only $\mathcal{O}(n^3)$ operations. In addition, we can further speed up the computations by avoiding any large matrix inversion. We refer the readers to Chan (2019) for computational details.

2.3 The Marginal Likelihood

Given the likelihood function in (2) and the normal-inverse-Wishart prior in (3), the associated marginal likelihood of the VAR(p) has the following analytical expression:

$$p(\mathbf{Y}) = \pi^{\frac{-nT}{2}} |\mathbf{V_A}|^{-\frac{n}{2}} |\mathbf{K_A}|^{-\frac{n}{2}} \frac{\Gamma_n\left(\frac{\nu_0+T}{2}\right) |\mathbf{S}_0|^{\frac{\nu_0}{2}}}{\Gamma_n\left(\frac{\nu_0}{2}\right) |\widehat{\mathbf{S}}|^{\frac{\nu_0+T}{2}}}. \tag{4}$$

The details of the derivation are given in Appendix B. Below we comment on a few computational details to improve numerical stability and computational efficiency.

First, note that to evaluate the marginal likelihood in (4), one needs not compute the inverse of the precision matrix $\mathbf{K_A}$—which can be computationally intensive when n is large—as is commonly done in the literature. Second, to prevent arithmetic underflow and overflow, we evaluate the marginal likelihood in log scale. In particular, to compute the log determinant of a square positive definite matrix, say, $\mathbf{B}$, we return $2\sum \log c_i$, where c_i is the i-th diagonal element of the Cholesky factor of $\mathbf{B}$.

3 Automatic Differentiation for Marginal Likelihood

In this section we introduce the proposed approach based on Automatic Differentiation (AD) to select optimal hyperparameters by maximizing the marginal likelihood—which is available in closed-form given in (4). In a nutshell, we apply AD to obtain the gradient of the log marginal likelihood with respect to the hyperparameters, which is then used as an input in an optimization routine, such as gradient ascent or Newton's method.[3] Since the gradient is efficiently computed using AD, this approach is substantially faster than other commonly used methods for computing the gradient, such as numerical finite-difference methods or symbolic differentiation. Below we discuss in detail how AD works.

AD is an efficient way to compute derivatives based on the chain rule. It is "automatic" in the sense that for an algorithm that maps an input vector into an output vector, there

[3]In the application we use MATLAB built-in function `fmincon` to minimize the negative log likelihood with respect to the hyperparameters. The solution, of course, coincides with the maximizer of the log marginal likelihood. The two key inputs for `fmincon` are the functions to evaluate the log marginal likelihood and its gradient, the latter of which is obtained via AD.

is an automatic way of evaluating its gradient without manually deriving the symbolic formula of the derivatives. More specifically, consider a function that maps

$$\boldsymbol{\theta}_0 \to \mathbf{G}(\boldsymbol{\theta}_0),$$

where $\boldsymbol{\theta}_0$ is the set of inputs that we are interested in computing the associated gradient. AD first decomposes the original function $\mathbf{G}$ into elementary functions, such as multiplication and exponentiation, $\mathbf{G}_1, \ldots, \mathbf{G}_k$:

$$\mathbf{G} = \mathbf{G}_k \circ \mathbf{G}_{k-1} \circ \cdots \circ \mathbf{G}_1,$$

where

$$\mathbf{G}_i : (\mathbf{x}_i, \boldsymbol{\theta}_0) \to \mathbf{x}_{i+1}$$

and $\mathbf{x}_i$ is the intermediary values at step i. Then, the derivative of $\mathbf{G}$ can be obtained via the chain rule, which is implemented automatically in the compute program:

$$\frac{\partial \mathbf{G}(\boldsymbol{\theta}_0)}{\partial \boldsymbol{\theta}_0} = \sum_{i=1}^{k} \frac{\partial}{\partial \mathbf{x}_k}\mathbf{G}_k \frac{\partial}{\partial \mathbf{x}_{k-1}}\mathbf{G}_{k-1} \cdots \frac{\partial}{\partial \mathbf{x}_{i+1}}\mathbf{G}_{i+1} \frac{\partial}{\partial \boldsymbol{\theta}_0}\mathbf{G}_i,$$

where $\frac{\partial \mathbf{G}_i}{\partial \mathbf{x}_i}, i = 1, \ldots, k$ are the intermediate gradients of the elementary functions. Depending on the software package, these intermediate gradients are often available readily as built-in functions. Otherwise they need to be provided by the user. But since they are elementary functions, their gradients are simple to compute. And once coded up, they can be added to the library to be reused in the future.

Hence, AD is different from the two other commonly used methods for computing the gradient, namely, symbolic differentiation and finite-difference methods. Due to the use of built-in derivatives of elementary functions and the application of chain rule, AD is often substantially faster than both. Similar to symbolic differentiation implemented in many widely-used softwares, AD computes the exact gradient of the original function up to floating point errors. But unlike symbolic differentiation that focuses on obtaining the exact expression of the gradient, AD evaluates the gradient *alongside* the evaluation of the original function. That is, for each $\mathbf{G}_i$ the AD program computes

$$\mathbf{G}_i^{\mathrm{AD}} : \left(\mathbf{x}_i, \frac{\partial \mathbf{x}_i}{\partial \boldsymbol{\theta}_0}, \boldsymbol{\theta}_0\right) \to \left(\mathbf{x}_{i+1}, \frac{\partial \mathbf{x}_{i+1}}{\partial \boldsymbol{\theta}_0}\right).$$

Since only floating point values are used in the above structure, a good efficiency may be obtained. Additionally, AD permits the use of control structures—e.g., loops, branches and sub-functions—common to modern computer languages but not easily amenable to symbolic differentiation.

Finite-difference methods approximate the gradient by using multiple evaluations of the original function. They are typically more computationally intensive when the number of dimensions of the domain is large. In comparison to finite-difference methods, AD requires additional model analysis and programming, but this additional effort is often justified by the improvement in the quality and speed of the calculated gradient.

Despite of its efficacy, a computationally naive implementation of AD, however, can result in prohibitively slow code and excess use of memory. Careful considerations can mitigate these effects. There are various modes and implementation subtleties associated with AD, see Griewank (1989). This paper uses the so-called forward mode implementation, i.e., we initialize the derivatives at the beginning of the algorithm and update them forward. While the backward mode can potentially increase the computational speed, as the dimension of the problem increases, the forward mode ensures the memory requirements are within a manageable constraints and allows the method to be applied to more complicated examples. We also use "operator overloading", rather than "source transformation" that typically requires the development of sophisticated compiler-type software to read in a computer program and to write a new augmented program for derivatives. The operator overloading we use here is to introduce a new class of objects that includes the differential component of the intermediary values on the expression graph. This can be easily done in modern computer languages such as C++ or MATLAB.

4 Application: Forecasting with Large BVARs

We consider a forecasting exercise with large Bayesian VARs to illustrate the usefulness of the proposed approach. After outlining the macroeconomic dataset in Section 4.1, we document the computational gains of using Automatic Differentiation (AD) to obtain the optimal hyperparameters compared to the conventional grid-search approach in Section 4.2. Then, in Section 4.3 we show that by optimizing the hyperparameters, we can substantially improve the forecast performance of standard BVARs. In particular,

we demonstrate that shrinking the error covariance matrix optimally can further improve forecast accuracy.

4.1 Data

The dataset for our forecasting exercise consists of 18 US quarterly variables and covers the quarters from 1959Q1 to 2018Q4. It is sourced from the FRED-QD database at the Federal Reserve Bank of St. Louis as described in McCracken and Ng (2016). We use a range of standard macroeconomic and financial variables, such as Real GDP, industrial production, inflation rates, labor market variables and interest rates. They are transformed to stationarity as suggested in McCracken and Ng (2016). The complete list of variables and how they are transformed is given in Appendix A.

4.2 Full Sample Results

We first find the optimal hyperparameters by maximizing the marginal likelihood given in (4) using the full sample. We consider two variants. In the first version, we optimize only κ_1, κ_2 and κ_3, while fixing κ_4 and κ_5 to their baseline values. This reflects the standard practice of the literature. In the second version, we optimize all the hyperparameters $\kappa_1, \ldots, \kappa_5$.

For each optimization exercise, we compare computational times of the proposed AD approach with that of the grid-search approach. The results are reported in Table 1. For the 3-dimensional optimization problem (i.e., optimizing the marginal likelihood with respect to only κ_1, κ_2 and κ_3), using a coarse grid with 30 grid points in each dimension takes 17.8 seconds. Doubling the grid points to 60 takes much longer, to over 2 minutes, as it takes 8 times more marginal likelihood evaluations. In contrast, using the proposed AD approach takes only 27 seconds.

For the 5-dimensional optimization problem of finding the optimal values of all 5 hyperparaemters $\kappa_1, \ldots, \kappa_5$, the grid-search approach would take hours or even days. But the proposed AD approach remains fast and takes only about about 32 seconds. Hence, the computational gains of using the AD approach can be substantial even in low dimensional problems. For high-dimensional problems, grid-search is simply impractical, whereas the

proposed approach remains feasible.

Table 1: Computation times of the proposed AD approach and the grid-search approach (in seconds). The numbers in parenthesis are the numbers of grid points in each dimension for the grid-search approach.

optimize κ_1-κ_3			optimize κ_1-κ_5		
grid (30)	grid (60)	AD	grid (30)	grid (60)	AD
17.8	138	27.2	16,020	496,800	32.0

Next, we report the optimized values of the hyperparameters in Table 2. Recall that for the baseline we set $\kappa_1 = 0.05$, $\kappa_2 = 1$, $\kappa_3 = 100$, $\kappa_4 = 1$ and $\kappa_5 = 1$ (see, e.g., Kadiyala and Karlsson, 1993, 1997). It turns out that the optimal κ_1 for both 3- and 5-dimensional optimization problems is quite close to the baseline value of 0.05. However, for other hyperparameters the optimal values can be very different from the baseline. For example, the optimal κ_2 is 3.2, which is over 3 times larger than the baseline value. These values suggest that the baseline case might be under-shrinking the VAR coefficients associated with higher lags. Overall, by using the optimal values of κ_1, κ_2 and κ_3, one can increase the log marginal likelihood by about 123.

Table 2: Baseline and optimized values of the hyperparameters.

	baseline	optimize κ_1-κ_3	optimize κ_1-κ_5
κ_1	0.05	0.051	0.041
κ_2	1	3.2	3.2
κ_3	100	28.2	24.2
κ_4	1	1	13.0
κ_5	1	1	10.3
log-ML	11,093	11,216	11,395

More interestingly, the results from the 5-dimensional optimization problem suggest that there is substantial gain in shrinking also the error covariance matrix, which is ignored in the literature so far. Recall that under our setup, the prior mean of $\mathbf{\Sigma}$ is $\kappa_5/\kappa_4 \times \text{diag}(s_1^2, \ldots, s_n^2)$, where s_r^2 is the sample variance of the residuals from an AR(p) model for the variable r. Our result suggests that the optimal prior mean of $\mathbf{\Sigma}$ is only about

77% of the sample residual variances. By shrinking the error covariance matrix optimally one can dramatically increase the log marginal likelihood by 179, compared to the case of optimizing only κ_1, κ_2 and κ_3.

In summary, our full sample results suggest that there is substantial gain in selecting multiple hyperparameters optimally. Moreover the optimal hyperparameter values obtained could be quite different from those baseline values commonly used in the literature. In the next section, we will present evidence that these "better" hyperparameter values do in fact lead to better forecast performance.

4.3 Forecasting Results

In this section we evaluate the forecast performance of BVARs with optimal hyperparameters relative to a standard benchmark where these hyperparameters are fixed at some judiciously chosen values. Our sample period is from 1959Q1 to 2018Q4 and the evaluation period starts at 1985Q1 and runs till the end of the sample. In each iteration, we use only the collection of data up to time t, denoted as $\mathbf{Y}_{1:t}$, to obtain the optimal hyperparameters by maximizing the marginal likelihood as given in (4). We consider two nested optimization problems. In the restricted version, we follow the standard practice of the literature and optimize only κ_1, κ_2 and κ_3, while fixing κ_4 and κ_5 at their baseline values. In the unrestricted version, we optimize all the hyperparameters $\kappa_1, \ldots, \kappa_5$.

We report in Figure 1 the optimal hyperparameters $\kappa_1, \kappa_2, \kappa_4$ and κ_5 over time. In the clear from the figures that there is substantial time variation in the optimal values, highlighting the empirical relevance of obtaining the optimized values, rather than setting them at some fixed values. In particular, there seems to be a structural break in the optimal values around the Great Recession of 2007-2009. For example, the optimal κ_1 increases from about 0.039 to 0.043, implying less shrinkage of the VAR coefficients is preferred by the data. Similarly, the optimal κ_4 and κ_5 drop substantially at the same time, reflecting less shrinkage of the error covariance matrix. All these results are consistent with the observation that parameter uncertainty increases at the aftermath of the Great Recession and less shrinkage of model parameters provides a better fit of the data.

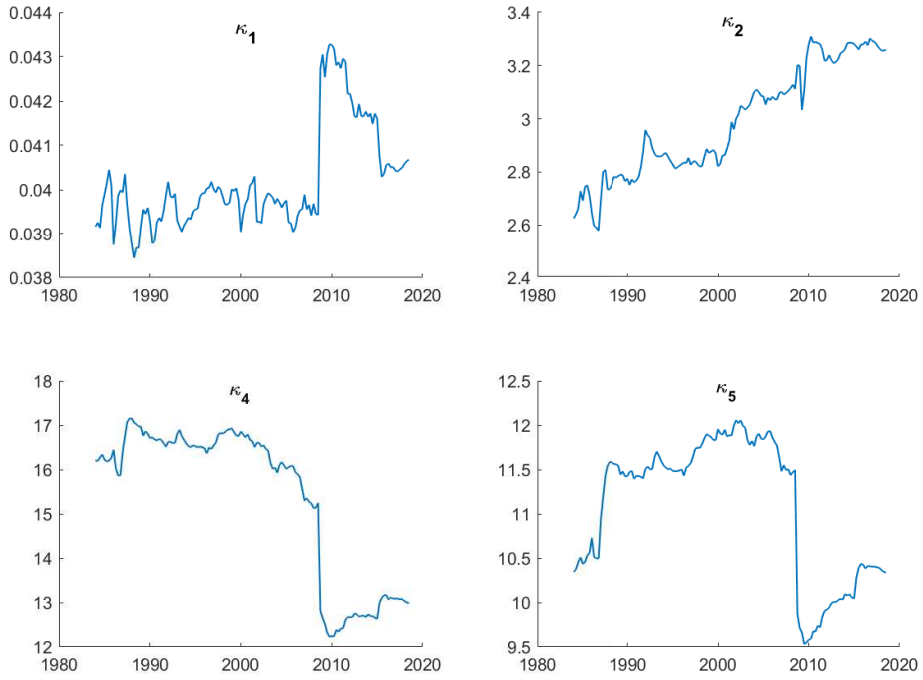

Figure 1: Optimized values of the hyperparameters $\kappa_1, \kappa_2, \kappa_4$ and κ_5 over time.

Given the substantial time variation in the optimal hyperparameters, one might expect that a BVAR where we re-optimize the hyperparameters when new data point comes in would forecast better than the BVAR in which these hyperparameters are fixed. To investigate this possibility, we turn to forecasting results next. We focus on four key macroeconomic variables, namely, Real GDP growth, PCE inflation, Fed funds rate and Unemployment rate, that are closely monitored by central banks and other policymakers.

In our forecasting exercise we evaluate both point and density forecasts. We use the conditional expectation $\mathbb{E}(y_{i,t+h} \mid \mathbf{Y}_{1:t})$ as the h-step-ahead point forecast for variable i and the predictive density $p(y_{i,t+h} \mid \mathbf{Y}_{1:t})$ as the corresponding density forecast. The metric used to evaluate the point forecasts is the root mean squared forecast error (RMSFE) defined as

$$\text{RMSFE} = \sqrt{\frac{\sum_{t=t_0}^{T-h}(y^{\text{o}}_{i,t+h} - \mathbb{E}(y_{i,t+h} \mid \mathbf{Y}_{1:t}))^2}{T-h-t_0+1}},$$

where $y^{\text{o}}_{i,t+h}$ is the actual observed value of $y_{i,t+h}$. For RMSFE, a smaller value indicates

better forecast performance.

To evaluate the density forecast, we use a measure that is based on the predictive likelihood $p(y_{i,t+h} = y^{o}_{i,t+h} \,|\, \mathbf{Y}_{1:t})$, i.e., the predictive density of $y_{i,t+h}$ evaluated at the actual value $y^{o}_{i,t+h}$. More specifically, we evaluate the density forecasts using the average of log predictive likelihoods (ALPL):

$$\text{ALPL} = \frac{1}{T-h-t_0+1} \sum_{t=t_0}^{T-h} \log p(y_{i,t+h} = y^{o}_{i,t+h} \,|\, \mathbf{Y}_{1:t}).$$

For this metric, a larger value indicates better forecast performance. For easy comparison, we report the ratios of RMSFEs of a given model to those of the standard BVAR. Hence, values smaller than unity indicate better forecast performance than the benchmark. For the average of log predictive likelihoods, we report differences from that of the standard BVAR. In this case, positive values indicate better forecast performance than the benchmark.

Table 3 reports the RMSFEs relative to the baseline BVAR where the hyperparameters are fixed. Consistent with the results in Carriero, Clark, and Marcellino (2015), using optimal hyperparameters improves point forecast performance. For example, by optimizing κ_1, κ_2 and κ_3, the RMSFEs of forecasting the unemployment rate and Federal funds rate are reduced by 7% and 11%, respectively. More interestingly, we can further improve the point forecast performance by shrinking the error covariance matrix $\mathbf{\Sigma}$ optimally. In particular, optimizing all hyperparamters $\kappa_1, \ldots, \kappa_5$ further reduces the RMSFEs associated with the unemployment rate and Federal funds rate by 2% and 3%, respectively.

Table 3: Root mean squared forecast errors relative to the baseline BVAR where the hyperparameters are fixed.

	$h = 1$		$h = 4$	
	optimize κ_1-κ_3	optimize κ_1-κ_5	optimize κ_1-κ_3	optimize κ_1-κ_5
Real GDP	0.97	0.95	0.95	0.94
PCE inflation	0.98	0.98	0.99	0.98
Fed funds rate	0.89	0.86	0.91	0.89
Unemployment rate	0.93	0.91	0.96	0.96

Next, we report in Table 4 the ALPLs relative to the baseline BVAR. Overall, these density forecast results are similar to those of the point forecasts. More specifically, for forecasting all 18 variables jointly, optimizing the hyperparameters κ_1, κ_2 and κ_3 delivers better density forecast performance relative to the baseline case. In addition, using optimal values for all the hyperparameters $\kappa_1, \ldots, \kappa_5$ further improves density forecasts.

For forecasting individual variables, optimizing κ_1, κ_2 and κ_3 never hurts density forecast performance relative to the baseline. However, for some variables optimizing also κ_4 and κ_5 reduces the forecast gains. This could be because the optimal hyperparameters are chosen by maximizing the marginal likelihood—i.e., the one-step-ahead joint density forecast performance—and they might not be optimal for some variables. Choosing the hyperparameters optimally for forecasting a subset of key variables would be an interesting research direction.

Table 4: Average of log predictive likelihoods relative to the baseline BVAR where the hyperparameters are fixed.

	$h = 1$		$h = 4$	
	optimize κ_1-κ_3	optimize κ_1-κ_5	optimize κ_1-κ_3	optimize κ_1-κ_5
All variables	0.909	1.037	0.955	1.113
Real GDP	0.012	0.009	0.022	0.025
PCE inflation	0.033	0.025	0.031	0.031
Fed funds rate	0.027	0.018	0.032	0.032
Unemployment rate	0.039	0.044	0.018	0.031

5 Concluding Remarks and Future Research

We have developed a computationally efficient method based on Automatic Differentiation to select the optimal hyperparameters for large BVARs. Using a large US dataset, we demonstrated that the computational gains of this new method compared to the convectional grid-search approach can be substantial in high-dimensional problems. In addition, we showed that by selecting the hyperparameters optimally, one can obtain notable improvement in forecast performance. Our findings therefore highlight the empirical relevance of using a data-driven approach to select hyperparameters for forecasting using

large Bayesian VARs.

For the setting considered in this paper, the marginal likelihood has an analytical expression, which can be used to speed up computations. For most complex models, however, the marginal likelihood is not available in close-formed, and its evaluation often requires Monte Carlo simulation. It would be useful to develop similar methods to choose optimal hyperparameters in those settings. Chan, Jacobi, and Zhu (2019a) take a first step in that direction by developing AD-based methods for a range of VARs and factor models. More generally, selecting optimal hyperparameters by maximizing the marginal likelihood for time-varying models, such as stochastic volatility models developed in Cogley and Sargent (2001, 2005) and Primiceri (2005), would be an important but challenging research problem.

Appendix A: Data

The dataset is sourced from the FRED-QD database at the Federal Reserve Bank of St. Louis (McCracken and Ng, 2016). It covers the quarters from 1959Q1 to 2018Q4. Table 5 lists the 18 quarterly variables and describes how they are transformed. For example, $\Delta \log$ is used to denote the first difference in the logs, i.e., $\Delta \log x = \log x_t - \log x_{t-1}$.

Table 5: Description of variables used in the forecasting exercise.

Variable	Transformation
Real Gross Domestic Product	$\Delta \log$
Personal Consumption Expenditures	$\Delta \log$
Real Disposable Personal Income	$\Delta \log$
Industrial Production Index	$\Delta \log$
Capacity Utilization: Manufacturing (SIC)	no transformation
All Employees: Total nonfarm	$\Delta \log$
Civilian Employment	$\Delta \log$
Civilian Unemployment Rate	Δ
Nonfarm Business Section: Hours of All Persons	$\Delta \log$
Housing Starts: Total	$\Delta \log$
Personal Consumption Expenditures: Chain-type Price index	$\Delta^2 \log$
Gross Domestic Product: Chain-type Price index	$\Delta^2 \log$
Consumer Price Index for All Urban Consumers: All Items	$\Delta^2 \log$
Real Average Hourly Earnings of Production and Nonsupervisory Employees: Manufacturing, deflated by Core PCE	$\Delta \log$
Effective Federal Funds Rate	Δ
Moody's Seasoned Baa Corporate Bond Yield Relative to Yield on 10-Year Treasury Constant Maturity	no transformation
Real M1 Money Stock	$\Delta \log$
S&P's Common Stock Price Index : Composite	$\Delta \log$

Appendix B: Derivation of the Marginal Likelihood

In this appendix we prove that the marginal likelihood of the VAR(p) under the normal-inverse-Wishart prior has the following expression:

$$p(\mathbf{Y}) = \pi^{\frac{-nT}{2}} |\mathbf{V_A}|^{-\frac{n}{2}} |\mathbf{K_A}|^{-\frac{n}{2}} \frac{\Gamma_n\left(\frac{\nu_0+T}{2}\right) |\mathbf{S}_0|^{\frac{\nu_0}{2}}}{\Gamma_n\left(\frac{\nu_0}{2}\right) |\widehat{\mathbf{S}}|^{\frac{\nu_0+T}{2}}}.$$

Using the likelihood in (2) and the normal-inverse-Wishart prior density in (3), the result follows from direct computation:

$$\begin{aligned}
p(\mathbf{Y}) &= \int p(\mathbf{A},\mathbf{\Sigma}) p(\mathbf{Y} \mid \mathbf{A},\mathbf{\Sigma}) \mathrm{d}(\mathbf{A},\mathbf{\Sigma}) \\
&= c(2\pi)^{-\frac{Tn}{2}} \int |\mathbf{\Sigma}|^{-\frac{\nu_0+T+n+k+1}{2}} \mathrm{e}^{-\frac{1}{2}\mathrm{tr}(\mathbf{\Sigma}^{-1}\mathbf{S}_0)} \mathrm{e}^{-\frac{1}{2}\mathrm{tr}\left[\mathbf{\Sigma}^{-1}\left((\mathbf{A}-\mathbf{A}_0)'\mathbf{V_A}^{-1}(\mathbf{A}-\mathbf{A}_0)+(\mathbf{Y}-\mathbf{ZA})'(\mathbf{Y}-\mathbf{ZA})\right)\right]} \mathrm{d}(\mathbf{A},\mathbf{\Sigma}) \\
&= c(2\pi)^{-\frac{Tn}{2}} \int |\mathbf{\Sigma}|^{-\frac{\nu_0+T+n+k+1}{2}} \mathrm{e}^{-\frac{1}{2}\mathrm{tr}(\mathbf{\Sigma}^{-1}\widehat{\mathbf{S}})} \mathrm{e}^{-\frac{1}{2}\mathrm{tr}\left(\mathbf{\Sigma}^{-1}(\mathbf{A}-\widehat{\mathbf{A}})'\mathbf{K}\mathbf{A}(\mathbf{A}-\widehat{\mathbf{A}})\right)} \mathrm{d}(\mathbf{A},\mathbf{\Sigma}) \\
&= c(2\pi)^{-\frac{Tn}{2}} \times (2\pi)^{\frac{nk}{2}} 2^{\frac{n(\nu_0+T)}{2}} |\mathbf{K_A}^{-1}|^{\frac{n}{2}} \Gamma_n\left(\frac{\nu_0+T}{2}\right) |\widehat{\mathbf{S}}|^{-\frac{\nu_0+T}{2}} \\
&= \pi^{\frac{-nT}{2}} |\mathbf{V_A}|^{-\frac{n}{2}} |\mathbf{K_A}|^{-\frac{n}{2}} \frac{\Gamma_n\left(\frac{\nu_0+T}{2}\right) |\mathbf{S}_0|^{\frac{\nu_0}{2}}}{\Gamma_n\left(\frac{\nu_0}{2}\right) |\widehat{\mathbf{S}}|^{\frac{\nu_0+T}{2}}},
\end{aligned}$$

where $c = (2\pi)^{-nk/2} 2^{-n\nu_0/2} |\mathbf{V_A}|^{-n/2} \Gamma_n(\nu_0/2)^{-1} |\mathbf{S}_0|^{\nu_0/2}$,

$$\mathbf{K_A} = \mathbf{V_A}^{-1} + \mathbf{Z}'\mathbf{Z}, \quad \widehat{\mathbf{A}} = \mathbf{K_A}^{-1}(\mathbf{V_A}^{-1}\mathbf{A}_0 + \mathbf{Z}'\mathbf{Y}), \quad \widehat{\mathbf{S}} = \mathbf{S}_0 + \mathbf{A}_0'\mathbf{V_A}^{-1}\mathbf{A}_0 + \mathbf{Y}'\mathbf{Y} - \widehat{\mathbf{A}}'\mathbf{K_A}\widehat{\mathbf{A}}.$$

In the above derivation we have used the fact that

$$\begin{aligned}
&\int |\mathbf{\Sigma}|^{-\frac{\nu_0+T+n+k+1}{2}} \mathrm{e}^{-\frac{1}{2}\mathrm{tr}(\mathbf{\Sigma}^{-1}\widehat{\mathbf{S}})} \mathrm{e}^{-\frac{1}{2}\mathrm{tr}\left(\mathbf{\Sigma}^{-1}(\mathbf{A}-\widehat{\mathbf{A}})'\mathbf{K}\mathbf{A}(\mathbf{A}-\widehat{\mathbf{A}})\right)} \mathrm{d}(\mathbf{A},\mathbf{\Sigma}) \\
&\qquad = (2\pi)^{\frac{nk}{2}} 2^{\frac{n(\nu_0+T)}{2}} |\mathbf{K_A}^{-1}|^{\frac{n}{2}} \Gamma_n\left(\frac{\nu_0+T}{2}\right) |\widehat{\mathbf{S}}|^{-\frac{\nu_0+T}{2}}.
\end{aligned}$$

Note that the right-hand side of the above equality is simply the normalizing constant of the $\mathcal{NIW}(\widehat{\mathbf{A}}, \mathbf{K_A}^{-1}, \nu_0+T, \widehat{\mathbf{S}})$ distribution.

References

Banbura, M., D. Giannone, M. Modugno, and L. Reichlin (2013): "Now-casting and the real-time data flow," in *Handbook of Economic Forecasting*, vol. 2, pp. 195–237. Elsevier.

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

Bauwens, L., M. Lubrano, and J. Richard (1999): *Bayesian Inference in Dynamic Econometric Models*. Oxford University Press, New York.

Carriero, A., T. E. Clark, and M. Marcellino (2015): "Bayesian VARs: Specification Choices and Forecast Accuracy," *Journal of Applied Econometrics*, 30(1), 46–73.

Carriero, A., T. E. Clark, and M. G. Marcellino (2016): "Common drifting volatility in large Bayesian VARs," *Journal of Business and Economic Statistics*, 34(3), 375–390.

Carriero, A., G. Kapetanios, and M. Marcellino (2009): "Forecasting exchange rates with a large Bayesian VAR," *International Journal of Forecasting*, 25(2), 400–417.

Chan, J. C. C. (2018): "Large Bayesian VARs: A Flexible Kronecker Error Covariance Structure," *Journal of Business and Economic Statistics*, Forthcoming.

——— (2019): "Large Bayesian Vector Autoregressions," *CAMA Working Paper 19/2019*.

Chan, J. C. C., L. Jacobi, and D. Zhu (2019a): "An Automated Prior Robustness Analysis in Bayesian Model Comparison," Available at: `http://joshuachan.org/papers/AD_ML.pdf`.

——— (2019b): "How Sensitive Are VAR Forecasts to Prior Hyperparameters? An Automated Sensitivity Analysis," *Advances in Econometrics*, 40A, 229–248.

Cogley, T., and T. J. Sargent (2001): "Evolving post-world war II US inflation dynamics," *NBER Macroeconomics Annual*, 16, 331–388.

——— (2005): "Drifts and volatilities: Monetary policies and outcomes in the post WWII US," *Review of Economic Dynamics*, 8(2), 262–302.

Del Negro, M., and F. Schorfheide (2004): "Priors from General Equilibrium Models for VARs," *International Economic Review*, 45, 643–673.

Dieppe, A., R. Legrand, and B. Van Roye (2016): "The BEAR toolbox," *ECB Working Paper 1934*.

Doan, T., R. Litterman, and C. Sims (1984): "Forecasting and conditional projection using realistic prior distributions," *Econometric reviews*, 3(1), 1–100.

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

Griewank, A. (1989): "On Automatic Differentiation," *Mathematical Programming: recent developments and applications*, 6(6), 83–107.

Jacobi, L., M. S. Joshi, and D. Zhu (2018): "Automated Sensitivity Analysis for Bayesian Inference via Markov Chain Monte Carlo: Applications to Gibbs Sampling," Available at SSRN: `http://dx.doi.org/10.2139/ssrn.2984054`.

Kadiyala, R. K., and S. Karlsson (1993): "Forecasting with generalized Bayesian vector auto regressions," *Journal of Forecasting*, 12(3-4), 365–378.

——— (1997): "Numerical Methods for Estimation and inference in Bayesian VAR-models," *Journal of Applied Econometrics*, 12(2), 99–132.

Karlsson, S. (2013): "Forecasting with Bayesian vector autoregressions," in *Handbook of Economic Forecasting*, ed. by G. Elliott, and A. Timmermann, vol. 2 of *Handbook of Economic Forecasting*, pp. 791–897. Elsevier.

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

Koop, G., and D. Korobilis (2010): "Bayesian Multivariate Time Series Methods for Empirical Macroeconomics," *Foundations and Trends in Econometrics*, 3(4), 267–358.

——— (2013): "Large time-varying parameter VARs," *Journal of Econometrics*, 177(2), 185–198.

Leeper, E. M., C. A. Sims, and T. Zha (1996): "What does monetary policy do?," *Brookings papers on economic activity*, 1996(2), 1–78.

Litterman, R. (1986): "Forecasting With Bayesian Vector Autoregressions — Five Years of Experience," *Journal of Business and Economic Statistics*, 4, 25–38.

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

Primiceri, G. E. (2005): "Time Varying Structural Vector Autoregressions and Monetary Policy," *Review of Economic Studies*, 72(3), 821–852.

Schorfheide, F., and D. Song (2015): "Real-Time Forecasting With a Mixed-Frequency VAR," *Journal of Business and Economic Statistics*, 33(3), 366–380.

Sims, C. A., and T. Zha (1998): "Bayesian methods for dynamic multivariate models," *International Economic Review*, 39(4), 949–968.

Zellner, A. (1971): *An Introduction to Bayesian Inference in Econometrics*. Wiley, New York.