Mixed-Frequency Bayesian VAR Models in R: the mfbvar package

Sebastian Ankargren
Uppsala University

Yukai Yang
Uppsala University,
Stockholm School of Economics

Abstract

Time series are often sampled at different frequencies, which leads to mixed-frequency data. Mixed frequencies are often neglected in applications as high-frequency series are aggregated to lower frequencies. In the **mfbvar** package, we introduce the possibility to estimate Bayesian vector autoregressive (VAR) models when the set of included time series consists of monthly and quarterly variables. The package implements several common prior distributions as well as stochastic volatility methods. The mixed-frequency nature of the data is handled by assuming that quarterly variables are weighted averages of unobserved monthly observations. We provide a user-friendly interface for model estimation and forecasting. The capabilities of the package are illustrated in an application.

Keywords: vector autoregression, steady-state prior, stochastic volatility, time series, R.

1. Introduction

Vector autoregressive (VAR) models constitute an important tool for multivariate time series analysis. They are, in their original form, easy to fit and to use and have hence been used for various types of policy analyses as well as for forecasting purposes. A major obstacle in applied VAR modeling is the curse of dimensionality: the number of parameters grows quadratically in the number of variables, and having several hundred or even thousands of parameters is not uncommon. Thus, VAR models estimated by maximum likelihood are usually associated with bad precision. As a remedy, Bayesian estimation has become widely popular following Litterman (1986) and the so-called Minnesota prior, which regularizes the estimation such that the parameters are shrunk towards a stylized view of macroeconomic time series. In the traditional Minnesota prior, the prior belief is that the time series are independent random walks. The prior puts prior densities more tightly around zero for higher-order lags, thus implying that recent lags should be relatively more important than more distant lags. The Minnesota prior, and variations thereof, has been successful in forecasting; for examples, the reader is referred to Bańbura, Giannone, and Reichlin (2010) and Karlsson (2013) and the references therein. For an accessible introduction to VAR modeling in macroeconomics, see Stock and Watson (2001).

Another prior that has shown promising results with respect to forecasting is the steady-state prior proposed by Villani (2009). In the Minnesota prior, common practice is to put a loose prior on the intercept in the VAR model. The steady-state prior employs an alternative parametrization of the model in which the unconditional mean (the steady state) is present. Thus, one can put a prior distribution on the steady states, for which there are often beliefs, rather than on the constant term. Numerous applications of Bayesian VARs with this prior exist in the literature, see for instance Jarociński and Smets (2008); Österholm (2010); Clark (2011); Ankargren, Bjellerup, and Shahnazarian (2017).

In most applications, researchers use single-frequency aggregated data in order to effortlessly be able to estimate the models. A common situation in macroeconomics is to include both the rate of inflation and GDP growth in a model. The inflation rate is typically published monthly, whereas GDP growth is published quarterly. Thus, a necessary first step in order to use traditional approaches is to aggregate the monthly inflation rate to the quarterly frequency.

By using newer techniques, there is no need to aggregate to the lowest common frequency. Mixed data sampling (MIDAS) methods allow for various frequencies of the data to co-exist in the model (Ghysels, Sinko, and Valkanov 2007). Moreover, bridge methods can be used to tackle the mixed-frequency problem (Baffigi, Golinelli, and Parigi 2004). Foroni and Marcellino (2013) provided a survey of mixed-frequency methods.

The approach that the **mfbvar** package implements is a state-space based approach in which it is assumed that low-frequency variables are observed linear combinations of underlying high-frequency processes. This implies that there is a latent monthly process for GDP growth that is unobserved, and what is observed is a weighted average of said latent process. By assuming such a structure, the model can be estimated by extending the single-frequency Bayesian VAR model estimation techniques with an auxiliary step that draws from the posterior distribution of the latent process.

The state-space-based mixed-frequency Bayesian VAR was proposed by Schorfheide and Song (2015) using a Minnesota-style normal inverse Wishart prior. In Ankargren, Unosson, and Yang (2019), a similar model was presented but with a steady-state prior. The **mfbvar** package implements the mixed-frequency VAR with Minnesota and steady-state priors and stochastic volatility in a user-friendly way.

The implementation of Bayesian VARs in R is not new; the **BMR** package (O'Hara 2017) presents the possibility to estimate single-frequency BVARs with either the Minnesota or steady-state prior. The BEAR toolbox (Dieppe, Legrand, and van Roye 2016), developed at the European Central Bank, provides the same functionality and more for MATLAB users. Moreover, various Bayesian VARs can also be estimated in EViews. However, none of these alternatives provide mixed-frequency estimation. The existing implementations of mixed-frequency estimation closest to ours is the MATLAB code accompanying the paper by Schorfheide and Song (2015) and the **midasr** package (Ghysels, Kvedaras, and Zemlys 2016) implementing MIDAS regression in R.

2. Mixed-Frequency Bayesian VAR models

Suppose that the system evolves at the monthly frequency. Let x_t be an $n \times 1$ monthly process. Decompose $x_t = (x'_{m,t}, x'_{q,t})'$ into n_m monthly variables, and a n_q-dimensional latent process for the quarterly observations. By letting $y_t = (y'_{m,t}, y'_{q,t})'$ denote observations, it is implied that $y_{m,t} = x_{m,t}$ as the monthly part is always observed. For the remaining quarterly variables, we instead observe a weighted average of x_q. There are two common aggregations used in the literature: intra-quarterly averaging and triangular aggregation. The former assumes the relation between observed and latent variables to be

$$y_{q,t} = \begin{cases} \frac{1}{3}(x_{q,t} + x_{q,t-1} + x_{q,t-2}), & t \in \{\text{Mar}, \text{Jun}, \text{Sep}, \text{Dec}\} \\ \varnothing, & \text{otherwise,} \end{cases}$$

and was used by e.g. Schorfheide and Song (2015) for modeling data in log-levels. The second alternative is the triangular weighting scheme employed by Mariano and Murasawa (2003), where

$$y_{q,t} = \begin{cases} \frac{1}{9}(x_{q,t} + 2x_{q,t-1} + 3x_{q,t-2} + 2x_{q,t-3} + x_{q,t-4}), & t \in \begin{Bmatrix} \text{Mar}, & \text{Jun} \\ \text{Sep}, & \text{Dec} \end{Bmatrix} \\ \varnothing, & \text{otherwise.} \end{cases}$$

Intra-quarterly averaging is recommended when the dataset consists of series in log-levels, whereas the triangular weighting scheme is appropriate when the data enter the model as growth rates. In practice, the difference is often negligible.

As the system is assumed to evolve at the monthly frequency, we specify a VAR(p) model for x_t:

$$x_t = \phi + \Phi_1 x_{t-1} + \cdots + \Phi_p x_{t-p} + \epsilon_t, \quad \epsilon_t \sim \text{N}(0, \Sigma). \tag{1}$$

The VAR(p) model can be written in companion form, where we let $z_t = (x'_t, x'_{t-1}, \ldots, x'_{t-p+1})'$. Thus, we obtain

$$z_t = \pi + \Pi z_{t-1} + u_t, \quad u_t \sim \text{N}(0, \Omega), \tag{2}$$

where π, Π and Ω are the corresponding companion form matrices constructed from $(\phi, \Phi_1, \ldots, \Phi_p, \Sigma)$; see Hamilton (1994).

It is now possible to specify the observation equation as

$$y_t = M_t \Lambda z_t, \tag{3}$$

where M_t is a deterministic selection matrix and Λ an aggregation matrix based on the weighting scheme employed. The M_t matrix in Equation (3) yields a time-varying observation vector by selecting rows corresponding to variables which are observed, whereas Λ aggregates the underlying latent process. For more details, see Schorfheide and Song (2015) and Ankargren *et al.* (2019).

The posterior distribution of interest is $p(X, \Theta|Y)$, where $X = (x_1, \ldots, x_T)'$, $Y = (y_1, \ldots, y_T)'$ and Θ collects the parameters of the model. This posterior distribution is intractable, but a Gibbs sampler can be employed in order to numerically approximate the posterior. Thus, estimation can be carried out by Markov Chain Monte Carlo (MCMC) and Gibbs sampling. We alternate between drawing from the conditional posterior of X given the parameters and from the conditional posterior of the parameters given X. That is, we alternate between drawing from the conditional distributions

$$p(X|\Theta, Y) \quad \text{and} \quad p(\Theta|X).$$

The two equations (2) and (3) constitute the transition and measurement equations of a state-space model. By conditioning on the parameters and the data, one can make a draw from the conditional posterior $p(X|\Theta, Y)$ by use of a simulation smoother (Durbin and Koopman 2002). Given X, the parameters are conditionally independent of Y and a draw from $p(\Theta|X)$ can be made as in the familiar single-frequency case with the data being X.

The preceding description of the VAR model assumes a constant error covariance matrix. In recent years, there has been a growing interest in relaxing this assumption and modeling heteroskedasticity by use of stochastic volatility models. Seminal work include Primiceri (2005) and Cogley and Sargent (2005), who used VARs with time-varying parameters and stochastic volatilities and have had large influence ever since. For forecasting, allowing for stochastic volatility often improves the predictive ability, in particular when the forecasting performance is evaluated with respect to density forecasts. Important work in this regard include Clark (2011); D'Agostino, Gambetti, and Giannone (2013); Clark and Ravazzolo (2015), whose results demonstrate the usefulness of stochastic volatilities. In the **mfbvar** package, a time-varying error covariance matrix can be modeled using either the common stochastic volatility model by Carriero, Clark, and Marcellino (2016) or using the factor stochastic volatility model based on the work by Kastner, Frühwirth-Schnatter, and Freitas Lopes (2017).

In the remainder of this section, we describe the prior distributions available in **mfbvar** and discuss some aspects of the implementations for sampling from the posterior distribution.

2.1. Priors for Regression Parameters

Minnesota-style priors

The model can be written on matrix form as

$$X = W\Gamma + E,$$

where $W = (W_1, \dots, W_T)'$ with $W_t = (x'_{t-1}, \dots, x'_{t-p}, 1)'$, $E = (\epsilon_1, \dots, \epsilon_T)'$, and $\Gamma = (\Phi', \phi)'$. The Minnesota prior for Γ takes one of two forms depending on the specification of the error covariance component. We will use the term "Minnesota prior" to refer to the prior for the model with intercept in order to more easily contrast it with an alternative specification discussed in a later subsection. It does not, however, refer to the original Minnesota prior with a fixed diagonal error covariance matrix as used by Litterman (1986); rather, it should be interpreted as a normal prior for the regression parameters—including intercept—based on the Minnesota prior beliefs.

Conditional normal prior A common prior for VAR models is a joint normal inverse Wishart prior for (Γ, Σ) in which the prior Γ is constructed conditionally on the error covariance Σ. The conditional prior distribution for Γ is in this case a multivariate normal distribution of the form

$$\mathrm{vec}(\Gamma)|\Sigma \sim \mathrm{N}(\mathrm{vec}(\underline{\Gamma}), \Sigma \otimes \underline{\Xi}), \tag{4}$$

where $\underline{\Gamma}, \underline{\Xi}$ are prior parameters specified by the researcher. The **mfbvar** package follows common practice and specifies the structure of the moments of the prior distribution along the lines of the Minnesota prior beliefs, yielding

$$\underline{\Gamma}(\underline{\gamma}) = \begin{pmatrix}\mathrm{diag}(\underline{\gamma}) & 0_{n\times[(p-1)+1]}\end{pmatrix}' \tag{5}$$

$$\xi_i = \begin{cases} \frac{\lambda_1^2}{(l^{\lambda_3} s_r)^2}, & \text{lag } l \text{ of variable } r, i = (l-1)n + r \\ \lambda_4^2, & i = np + 1 \end{cases}, \tag{6}$$

where ξ_i are the diagonal elements of $\underline{\Xi}$, and s_j^2 for $j = 1, \dots, p$ are obtained as the residual variances from AR(4) regressions. As is customary, the prior means of all regression parameters are set to zero, except the AR(1) parameters $(\underline{\gamma})$.

Independent normal prior The conditional normal prior imposes a symmetry in the prior for Γ through the Kronecker structure. An alternative specification is to assume Γ to be independent of the error covariance a priori. The prior is

$$\mathrm{vec}(\Gamma) \sim \mathrm{N}(\mathrm{vec}(\underline{\Gamma}), \underline{\Omega}) \tag{7}$$

where $\underline{\Omega}$ is the $n(np+1) \times n(np+1)$ diagonal matrix containing the prior variances of $\Phi_l^{(i,j)}$, i.e. element (i, j) of Φ_l that relates lag l of variable j to variable i, and the vector of intercepts ϕ.

The prior variances are given by

$$\mathsf{VAR}(\Phi_l^{(i,j)}) = \begin{cases} \frac{\lambda_1^2}{(l^{\lambda_3})^2}, & \text{if } i = j \\ \frac{\lambda_1^2 \lambda_2^2}{(l^{\lambda_3})^2} \frac{s_i^2}{s_j^2}, & \text{otherwise.} \end{cases} \tag{8}$$

The main difference between (6) and (8) is that the conditional normal prior in (6) enforces the restriction of symmetrical shrinkage of parameters in all equations. This restriction is not imposed in (8). The implication is that in (6) λ_2 is implicitly set to $\lambda_2 = 1$ (i.e. no penalization of terms corresponding to lags of other variables in a given equation), whereas (8) allows cross-variable shrinkage to be enforced. The prior variance of the intercept is as before $\mathsf{VAR}(\phi) = 10^4 I_n$.

Steady-state prior

The steady-state prior proposed by Villani (2009) reformulates (1) to be on the mean-adjusted form

$$\Phi(L)(x_t - \Psi d_t) = \epsilon_t, \tag{9}$$

where $\Phi(L) = (I_n - \Phi_1 L - \cdots - \Phi_p L^p)$ is an invertible lag polynomial. The intercept ϕ in (1) can be replaced by the more general deterministic term $\Phi_0 d_t$, where Φ_0 is $n \times m$ and d_t is $m \times 1$. The steady-state parameters Ψ in (9) relate to Φ_0 through $\Psi = [\Phi(L)]^{-1}\Phi_0$. By the reformulation, we obtain parameters Ψ that immediately yield the unconditional mean of x_t—the steady state. The rationale is that while it is potentially difficult to express prior beliefs about Φ_0, eliciting prior beliefs about Ψ is often easier.

Original steady-state prior In the original steady-state prior proposed by Villani (2009), the prior for $\psi = \text{vec}(\Psi)$ is given by $\psi \sim \text{N}(\underline{\psi}, \underline{\Omega}_\psi)$. The prior distribution for Φ in **mfbvar** is either of the conditional or independent normal priors in (4) and (7), respectively. The only modification is that the constant column of Γ is excluded.

Hierarchical steady-state prior Louzis (2019) suggested an extension of the steady-state prior that utilizes a hierarchical specification based on the hierarchical normal-gamma shrinkage prior proposed by Griffin and Brown (2010); see also Huber and Feldkircher (2019) for an application of the normal-gamma prior to VAR models. There are two main reasons that justify the hierarchical steady-state prior. First, the normal-gamma prior induces a heavy-tailed unconditional prior for the steady-state parameters, which means that the prior generally pulls the posterior distribution towards the prior means, but with the possibility of larger deviations due to the excess kurtosis of the prior. Second, Louzis (2019) relied on default values for the hyperpriors following the suggestion by Huber and Feldkircher (2019). The consequence is that only prior means must be specified. It is oftentimes easier to have a reasonable prior belief for the means of the steady-state parameters than for the variances, and the issue of specifying the prior is therefore simplified. The structure of the prior is

$$\begin{aligned}
\psi_j | \omega_{\psi,j} &\sim \text{N}(\underline{\psi}_j, \omega_{\psi,j}) \\
\omega_{\psi,j} | \phi_\psi, \lambda_\psi &\sim \text{G}(\phi_\psi, 0.5\phi_\psi\lambda_\psi) \\
\phi_\psi &\sim \text{Exp}(1) \\
\lambda_\psi &\sim \text{G}(c_0, c_1) \\
j &= 1, \ldots, nm
\end{aligned} \tag{10}$$

where $\text{G}(a, b)$ denotes the gamma distribution with shape-rate parametrization, and $\text{Exp}(c)$ denotes the exponential distribution. Ankargren *et al.* (2019) employed both types of steady-state priors in mixed-frequency BVARs and found that the hierarchical prior performed well.

2.2. Error Covariance Priors

The **mfbvar** package includes both homoskedastic and heteroskedastic specifications, which are described next.

Constant volatility

Two of the more common priors for (Γ, Σ) that are used for homoskedastic VAR models are the normal inverse Wishart and normal-diffuse priors. The former combines the conditional normal prior for Γ with an inverse Wishart prior for Σ, and the latter uses the independent normal prior for Γ in conjunction with a diffuse Jeffreys' prior for Σ. For this reason, the two alternatives for a homoskedastic Σ in **mfbvar** are the inverse Wishart and diffuse priors.

Inverse Wishart prior The inverse Wishart prior is specified as

$$\begin{aligned}
&\Sigma \sim \text{iW}(\underline{S}, \underline{\nu}) \\
&\underline{S} = (\underline{\nu} - n - 1)\,\text{diag}(s_1^2, \ldots, s_n^2) \\
&\underline{\nu} = n + 2
\end{aligned} \tag{11}$$

where s_i^2 are n residual variances from auxiliary AR(4) regressions. The degrees of freedom is fixed to $n+2$ to ensure that the prior variance exists. The inverse Wishart prior for Σ is in **mfbvar** always used with a conditional normal prior for the regression parameters—with or without a separate steady-state prior—to yield a standard normal inverse Wishart prior.

Diffuse prior The diffuse prior for Σ is the Jeffreys' prior given by

$$p(\Sigma) \propto |\Sigma|^{-(n+1)/2}.$$

The **mfbvar** package uses the diffuse prior only in combination with the independent normal prior for the regression parameters—possibly with an additional prior for the steady-state parameters—so that the standard normal-diffuse prior is obtained.

Stochastic volatility

The **mfbvar** package also includes two methods that relax the usual assumption of the error covariance matrix being constant over time. These two methods are common stochastic volatility, proposed by Carriero *et al.* (2016), and factor stochastic volatility using the methodology developed by Kastner *et al.* (2017).

Common stochastic volatility The common stochastic volatility specification presented by Carriero *et al.* (2016) assumes that the covariance structure in the model is constant over time, but adds a factor that enables time-dependent scaling of the error covariance matrix. More specifically, it is assumed that

$$\mathsf{VAR}(\epsilon_t | f_t, \Sigma) = f_t \Sigma,$$

where f_t is a scalar, Σ is inverse Wishart as in (11), and

$$\begin{aligned} \log f_t &= \rho \log f_{t-1} + v_t \\ v_t &\sim \mathrm{N}(0, \sigma^2) \\ \rho &\sim \mathrm{N}(\underline{\mu}_\rho, \underline{\Omega}_\rho; |\rho| < 1) \\ \sigma^2 &\sim \mathrm{IG}(\underline{d} \cdot \underline{\sigma}^2,\ \underline{d}), \end{aligned} \tag{12}$$

where $\mathrm{N}(a, b; |x| < c)$ denotes the truncated normal distribution with support $(-c, c)$, and $\mathrm{IG}(a, b)$ is the inverse gamma distribution with parameters (a, b).

The common stochastic volatility model for the error covariance matrix further requires the conditional normal prior in (4) for Γ. The computational benefits associated with the normal inverse Wishart prior, which are due to its symmetry and Kronecker factorization, still apply under the common stochastic volatility framework, and the method therefore offers a simple and parsimonious way of coping with changing volatility with relatively little effort. This form of volatility was used by Götz and Hauzenberger (2018); Ankargren *et al.* (2019) who documented improved forecasts compared to the standard prior with constant volatility.

Factor stochastic volatility The **mfbvar** package also allows for relaxing the assumption of a constant error covariance matrix by modeling the time-varying error covariance matrix using a factor stochastic volatility model. This line of modeling VAR models has previously been pursued by Kastner and Huber (2018) for single-frequency VAR models, and Ankargren and Jonéus (2019b) have discussed its advantages for estimating large-scale mixed-frequency VARs. The factor stochastic volatility model in **mfbvar** is based on the approach taken by Kastner *et al.* (2017), see also Aguilar and West (2000), which decomposes the error term in (1) as

$$\begin{aligned}\epsilon_t &= \Lambda^f f_t + \nu_t \\ f_t &\sim \mathrm{N}(0, \Omega_t^f) \\ \nu_t &\sim \mathrm{N}(0, \Omega_t^\nu)\end{aligned}$$

where Λ^f $(n \times r)$ is a matrix of factor loadings, f_t $(r \times 1)$ a vector of latent factors and ν_t $(n \times 1)$ is a vector of idiosyncratic error terms. Both Ω_t^f and Ω_t^ν are diagonal with elements $\omega_{t,i}^f$ and $\omega_{t,j}^\nu$ for $i = 1, \dots, r$ and $j = 1, \dots, n$. The diagonal elements, in turn, evolve as geometric AR(1) processes given by

$$\begin{aligned}\log \omega_{t,i}^f &= \phi_i^f \log \omega_{t-1,i}^f + \sigma_i^f \epsilon_{t,i}^f, \quad \epsilon_{t,i}^f \sim \mathrm{N}(0,1) \\ \log \omega_{t,j}^\nu &= \mu_j^\nu + \phi_i^\nu (\log \omega_{t-1,j}^\nu - \mu_j^\nu) + \sigma_j^\nu \epsilon_{t,j}^\nu, \quad \epsilon_{t,j}^\nu \sim \mathrm{N}(0,1)\end{aligned}$$

where the factor-related volatility processes, $\log \omega_{t,i}^f$, is assumed to have zero mean for identification purposes (see Kastner *et al.* 2017 for more information).

The time-varying error covariance matrix is under the factor stochastic volatility model

$$\Sigma_t = \mathsf{VAR}(\epsilon_t | \Lambda^f, \Omega_t^\nu, \Omega_t^f) = \Lambda^f \Omega_t^f \Lambda^{f'} + \Omega_t^\nu.$$

The non-diagonal terms are therefore governed by the factor component, whereas the variances are driven by both the common component and the idiosyncratic term.

The prior distributions for the parameters relating to the factor stochastic volatility model, as well as the procedure for sampling from the posterior, follow Kastner and Frühwirth-Schnatter (2014); Kastner *et al.* (2017). The factor loadings are given independent normal priors $\Lambda_{ij}^f \sim \mathrm{N}(0, B_\Lambda)$. The unconditional mean for the idiosyncratic volatility processes are equipped with normal priors $\mu_j^\nu \sim \mathrm{N}(b_\mu, B_\mu)$. The autoregressive parameters are given the prior distribution $(\phi_i^f + 1)/2 \sim \mathrm{Beta}(a_0, b_0)$ and $(\phi_j^\nu + 1)/2 \sim \mathrm{Beta}(a_0, b_0)$, which ensures that the log volatility processes are stationary. Finally, the prior distribution for the variance parameters is $(\sigma_i^f)^2, (\sigma_j^\nu)^2 \sim \mathrm{G}(0.5, 0.5 B_\sigma^{-1})$. While the common stochastic volatility approach hinges on the use of the symmetric normal inverse Wishart prior, a VAR with a factor stochastic volatility model instead uses the less restrictive independent normal prior in (7).

2.3. Implementation of Posterior Sampling

Gibbs sampling is employed to obtain draws from the posterior distribution of the latent processes and the parameters. Estimation of mixed-frequency Bayesian VARs can be time-consuming and computationally intensive. For this reason, all of the MCMC algorithms employed are fully implemented in C++ using the **Armadillo** library (Sanderson and Curtin 2016) through the use of **Rcpp** and **RcppArmadillo** (Eddelbuettel and Francois 2011; Eddelbuettel and Sanderson 2014). We describe the main blocks of the MCMC algorithms used in some more detail in what follows. Because the outstanding feature of mixed-frequency VARs, as opposed to standard single-frequency VARs, is the simulation smoothing step conducted to sample from the posterior of the high-frequency latent processes, we describe this step more carefully.

Simulation smoother

The state-space model in (2)–(3) is simple to implement in existing software as it fits into the standard framework for state-space models with no correlation between equations. Tusell (2011) provides an overview of available packages in R for Kalman filtering and smoothing, which could be used for this purpose. However, this naive implementation does not constitute a viable option as dimensions grow—it will involve a large number of operations on matrices of size $np \times np$ which will be infeasibly time consuming even for moderate n and p.

The simulation smoother implemented in **mfbvar** is based on the suggestions by Schorfheide and Song (2015) and Ankargren and Jonéus (2019a) that employ a compact form to reduce the computational burden. To see how this can alleviate the problem, first decompose

$$\phi = \begin{pmatrix} \phi_m, \\ \phi_q \end{pmatrix}, \quad \Phi_i = \begin{pmatrix} \Phi_{i,mm} & \Phi_{i,mq} \\ \Phi_{i,qm} & \Phi_{i,qq} \end{pmatrix}$$

and let $\Phi_j = (\Phi_{1,j}, \dots, \Phi_{p,j})$ where $j \in \{mm, mq, qm, qq\}$. The idea is now to exploit the fact that for some $T_0 \leq T$, all monthly variables are observed for all $t = 1, \dots, T_0$. This fact suggests that the monthly variables can be omitted from the state equation and instead enter the system through the exogenous terms in a state-space model. The reformulation, however, implies that the state and measurement errors will be correlated, which thus requires the use of an algorithm based on a state-space model with between-equation correlation.

Let now $\epsilon_t = Le_t$ with $e_t \sim \mathrm{N}(0, I_n)$ and L being the lower triangular Cholesky factor of Σ, and let the exogenous terms be given by $c_{m,t} = \phi_m + \sum_{i=1} \Phi_{i,mm} y_{m,t-i}$ and $d_{q,t} = \phi_q + \sum_{i=1} \Phi_{i,qm} y_{m,t-i}$. The compact state-space model is then specified as

$$\begin{aligned} \begin{pmatrix} y_{m,t} \\ y_{q,t} \end{pmatrix} &= \begin{pmatrix} c_{m,t} \\ M_{q,t} 0_{n_q} \end{pmatrix} + \begin{pmatrix} 0_{n_m} & \Phi_{mq} \\ M_{q,t}\Lambda & 0_{n_q} \end{pmatrix} \begin{pmatrix} x_{q,t} \\ z_{q,t-1} \end{pmatrix} + \begin{pmatrix} I_{n_m} & 0_{n_m \times n_q} \\ \multicolumn{2}{c}{M_{q,t} 0_{n_q \times n}} \end{pmatrix} Le_t \\ \begin{pmatrix} x_{q,t} \\ z_{q,t-1} \end{pmatrix} &= \begin{pmatrix} d_{q,t} \\ 0_{pn_q} \end{pmatrix} + \begin{pmatrix} \Phi_{qq} & 0_{n_q} \\ I_{pn_q} & 0_{n_q} \end{pmatrix} \begin{pmatrix} x_{q,t-1} \\ z_{q,t-2} \end{pmatrix} + \begin{pmatrix} 0_{n_q \times n_m} & I_{n_q} \\ \multicolumn{2}{c}{0_{pn_q \times n}} \end{pmatrix} Le_t, \end{aligned} \tag{13}$$

where $M_{q,t}$ refers to the quarterly rows of M_t. From (13), one can deduce the system matrices for a state-space model with between-equation correlation. See Schorfheide and Song (2015) for more information.

The dimension of the state vector in (13) is $n_q(p+1)$ compared to np in (2) and thus offers a substantial reduction when n_q is relatively small. The drawback of the compact representation is that it is only applicable for $t = 1, \dots, T_0$; for $t = T_0, \dots, T$, when only a subset of the monthly variables is observed, it is no longer an alternative. The typical pattern in many macroeconomic datasets constructed in real time is that some variables, usually interest rates and stock market variables, are available immediately when a month has ended. Macroeconomic variables are, on the other hand, often available with a lag of one or two months. The speed of the simulation smoother can be notably improved by exploiting that not all of the monthly variables are missing at the end of the sample. To handle the final part of the sample, the **mfbvar** package uses the adaptive algorithm proposed by Ankargren and Jonéus (2019a). The algorithm is adaptive in the sense that it augments the state vector in (13) with variables that are missing at the end; this feature is in contrast to the procedure used by Schorfheide and Song (2015), which instead moves to the full companion form defined in (2).

Equation (13) can be generalized into

$$\begin{aligned}
\begin{pmatrix} y_{m,t} \\ y_{q,t} \end{pmatrix} &= Z_t \begin{pmatrix} x_{\mathcal{U}_t,t} \\ x_{q,t} \\ \vdots \\ x_{\mathcal{U}_t,t-p} \\ x_{q,t-p} \end{pmatrix} + C_t \begin{pmatrix} y_{\mathcal{O}_{t-1},t-1:t-p} \\ 1 \end{pmatrix} + G_t e_t \\
\begin{pmatrix} x_{\mathcal{U}_t,t} \\ x_{q,t} \\ \vdots \\ x_{\mathcal{U}_t,t-p} \\ x_{q,t-p} \end{pmatrix} &= T_t \begin{pmatrix} x_{\mathcal{U}_{t-1},t-1} \\ x_{q,t} \\ \vdots \\ x_{\mathcal{U}_{t-1},t-p-1} \\ x_{q,t-p-1} \end{pmatrix} + D_t \begin{pmatrix} y_{\mathcal{O}_{t-1},t-1:t-p} \\ 1 \end{pmatrix} + H_t e_t.
\end{aligned} \tag{14}$$

where $\mathcal{U}_t$ denotes the indexes relating to the set of monthly variables that are unobserved at time t, $\mathcal{O}_t$ denotes the set of indexes for the monthly variables that are observed at time t, and

$$y_{\mathcal{O}_{t-1},t-1:t-p} = (y'_{\mathcal{O}_{t-1},t-1}, \dots, y'_{\mathcal{O}_{t-1},t-p})'$$

with $y_{\mathcal{O}_{t-1},t-j}$ denoting the value at time $t-j$ of the set of monthly variables observed at time $t-1$. The above formulation generalizes (13) in that it allows for any number of monthly variables to be missing at time t, but only those that are missing data are included in the state vector. As demonstrated by Ankargren and Jonéus (2019a), this approach can lead to large gains when the simulation smoother is used for real-time data featuring an unbalanced dataset. For more details and explicit descriptions of the system matrices in (14), see Ankargren and Jonéus (2019a).

To generate a draw from the posterior distribution $p(X|\Theta, Y)$, we employ Algorithm 2 in Durbin and Koopman (2002):

1. Generate an artifical sample $\{x_t^+, y_t^+\}_{t=1}^T$ using the recursions in (1) and (3)
2. Create $y_t^* = y_t - y_t^+$ and smooth to obtain $\hat{x}_t^* = \mathsf{E}(x_t^*|\Theta, Y^*)$
3. Take $\{x_t^+ + \hat{x}_t^*\}_{t=1}^T$ as the draw from $p(X|\Theta, Y)$

To initialize, we condition on the first p observations indexed by $t = -p+1, \dots, 0$. For the quarterly observations, missing observations in the pre-sample are replaced by the preceding observation. In addition, as is discussed by Jarociński (2015), when smoothing based on y_t^* we set constant terms to zero.

Constant error covariance matrix

If the error covariance matrix is assumed to be constant, then by standard results the error covariance matrix can easily be sampled from an inverse Wishart posterior distribution, see Karlsson (2013) for these results described in detail.

Common stochastic volatility

Under the common stochastic volatility specification, the VAR can be transformed into a homoskedastic model given the volatility factor. The matrix Σ can therefore be sampled from its conditional posterior distribution using standard results with parameters based on the transformed model, see Carriero *et al.* (2016); Ankargren *et al.* (2019).

The posterior distributions of the volatility parameters (ρ, σ^2) are truncated normal and inverse gamma, respectively. Sampling of the latent volatility is conducted using the auxiliary mixture approach developed by Kim, Shephard, and Chib (1998), where we use the sampler presented by McCausland, Miller, and Pelletier (2011) based on an extension of the implementation used in the **stochvol** package (Kastner 2016).

Factor stochastic volatility

If, instead, the factor stochastic volatility model is used to capture time-varying variances in the model, then we use an implementation adapted from the **factorstochvol** package (Kastner 2019) for obtaining a draw from the posterior of the volatilities, factors and the related parameters. The interested reader is referred to Kastner *et al.* (2017) for a thorough description of the methodology used by the **factorstochvol** package.

Regression parameters

The conditional posterior distribution of the regression parameters is multivariate normal. Under the normal inverse Wishart prior, the covariance matrix of the vector of regression parameters features a Kronecker structure. A procedure that is more efficient than computing the posterior covariance and then sampling from the multivariate normal distribution is to use the matrix-normal form of the matrix of regression parameters directly. See the discussion and samplers in Karlsson (2013); Carriero *et al.* (2016).

When factor stochastic volatility is used in the model, the computational effort required for sampling from the conditional posterior of the regression parameters is reduced. Conditional on $\Lambda^f f_t$, the equations in the VAR are independent. The independence enables independent sampling of the parameters in each equation. There is, therefore, no need to compute the Cholesky decomposition of the covariance matrix of all parameters, which would otherwise be prohibitive even for models that are relatively moderate in size. Note that under the normal inverse Wishart prior, Cholesky decomposing the full covariance matrix is also avoided, albeit for a different reason.

The actual sampling of regression parameters when the factor stochastic volatility model is assumed is conducted using one of two algorithms. Both algorithms exploit that the posterior is on the form $\mathrm{N}(A^{-1}b, A^{-1})$, where there is an intimate connection between the first and second moments. The first algorithm, described in Rue (2001), has complexity $O(n^3p^3)$ for sampling each equation's parameters, and is used when the number of parameters in each equation is less than the sample size T. When the sample size exceeds the number of parameters in each equation, the algorithm proposed by Bhattacharya, Chakraborty, and Mallick (2016) is used. The latter algorithm has complexity $O(T^2np)$ and is the faster alternative when $T < np$. Further speed improvements can be obtained by sampling the parameters in each equation in parallel. The package allows for doing so when the factor stochastic volatility model is employed using the **RcppParallel** package (Allaire, Francois, Ushey, Vandenbrouck, Geelnard, and Intel 2019).

Steady-state parameters

The conditional posterior distribution of the steady-state parameters is normal and low-dimensional and therefore easily sampled from. The conditional posterior of the hierarchical prior's global shrinkage parameter λ_ψ is a gamma distribution, whereas the kurtosis parameter ϕ_ψ has no known conditional posterior distribution. The **mfbvar** package implements a Metropolis-Hastings sampler with a random walk proposal on the log-scale, where the scale of the proposal distribution is either fixed or updated adaptively as in Roberts and Rosenthal (2009). The conditional posterior distributions of the local shrinkage parameters $\omega_{\psi,j}$ are generalized inverse Gaussian, which we sample from using the **GIGrvg** package implementing the methods developed by Hörmann and Leydold (2014).

2.4. Marginal Data Density Estimators

The hyperparameters of the prior distributions are crucial for good perfomance of the models. For the Minnesota and steady-state priors that use the normal inverse Wishart specification, the priors are parametrized by the hyperparameters λ_1 and λ_3. In the literature, a standard choice has become $\lambda_1 = 0.2$ and $\lambda_3 = 1$, but in many applications it is plausible that this choice is subpar. One method of selecting values of the hyperparameters is to compute the marginal data density over a range of values for λ_1 and λ_3 and selecting the maximizing pair. The **mfbvar** package allows the marginal data density to be estimated when the normal inverse Wishart prior is used employing the estimators suggested by Schorfheide and Song (2015) and Ankargren, Unosson, and Yang (2018).[1]

Minnesota prior

In the case of a Minnesota prior, marginal data density estimation follows Schorfheide and Song (2015). Consider a single quarterly observation under the intra-quarterly average aggregation scheme, $y_{q,3} = \frac{1}{3}(x_{q,3} + x_{q,2} + x_{q,1})$, and its relation to $x_{q,3}$. Given $y_{q,3}$, $x_{q,2}$ and $x_{q,1}$, $x_{q,3}$ is known. Hence, we can write

$$\begin{pmatrix} x_{q,3} \\ x_{q,2} \\ x_{q,1} \end{pmatrix} = \begin{pmatrix} 3 & -1 & -1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} y_{q,3} \\ x_{q,2} \\ x_{q,1} \end{pmatrix}$$

For this reason, we partition X into (Y, Z) where Z collects the *inherently* missing observations, i.e. the first two missing monthly observations per quarter, and Y all observed data. [In the example, $Y_{1:3} = y_{q,3}$ and $Z_{1:3} = (x_{q,1}, x_{q,2})$]. The marginal data density estimator derived by Schorfheide and Song (2015) is

$$\hat{p}(Y; \lambda_1, \lambda_3) = c \left[\frac{1}{R} \sum_{r=1}^{R} \frac{f(Z^{(r)}; \hat{\mu}_Z, \hat{\Sigma}_Z, \kappa)}{p(X^{(r)})} \right]^{-1}, \tag{15}$$

where c is a constant accounting for the change of variables from $p(Y, Z)$ to $p(X)$. If we by $T_{q,i}$ denote the number of observations of the ith quarterly variable, then $\log c = \sum_{i=1}^{n_q} T_{q,i} \log(3)$.

[1] The marginal data density is also known as the marginal likelihood. We use the former term so that our terminology is in line with Schorfheide and Song (2015); Ankargren *et al.* (2018).

Furthermore, the density f is a truncated normal distribution given as

$$
\begin{aligned}
&f(x;\mu_x,\Sigma_x,\kappa)\\
&\qquad=\frac{1}{\kappa}\frac{|\Sigma_x|^{-1/2}}{2\pi^{n_x/2}}\exp\left\{-\frac{1}{2}(x-\hat{\mu}_x)'\Sigma_x^{-1}(x-\hat{\mu}_x)\right\}\\
&\qquad\qquad\times I\left\{(x-\hat{\mu}_x)'\Sigma_x^{-1}(x-\hat{\mu}_x)\leq\chi^2_{n_x}(\kappa)\right\},
\end{aligned}
$$

for x a vector with n_x elements and $\kappa\in(0,1]$. The moments in the numerator term in (15) are

$$
\begin{aligned}
\hat{\mu}_Z&=\frac{1}{R}\sum_{r=1}^{R}Z^{(r)}\\
\hat{\Sigma}_Z&=\frac{1}{R}\sum_{r=1}^{R}Z^{(r)}Z^{(r)\prime}-\hat{\mu}_Z\hat{\mu}_Z'.
\end{aligned}
$$

The denominator term in (15) is the density of a matrix t distribution with parameters $\mathrm{Mt}(W^{(r)}\underline{\Gamma},(I_T+W^{(r)}\underline{\Xi}W^{(r)\prime})^{-1},\underline{S},\underline{\nu})$ (see Karlsson 2013 for the density and more details).

Note that the marginal data density in (15) is a function of (λ_1,λ_3) through the draws of $\Phi^{(r)}$, which in turn affect all other draws.

Steady-state prior

If the steady-state prior is used in conjunction with the normal inverse Wishart prior, the marginal density computation follows Ankargren *et al.* (2018) and is

$$
\hat{p}(Y;\lambda_1,\lambda_3)=\frac{p(Y|\bar{\Phi},\bar{\Sigma},\bar{\psi})p(\bar{\Phi},\bar{\Sigma})}{\hat{p}(\bar{\Phi},\bar{\Sigma}|\bar{\psi},Y)}\frac{p(\bar{\psi})}{\hat{p}(\bar{\psi}|Y)},
$$

where bars represent posterior means. As in the Minnesota case, (λ_1,λ_3) have a direct effect on $\Phi^{(r)}$ and, consequently, an indirect effect on the other draws.

The likelihood $p(Y|\bar{\Phi},\bar{\Sigma},\bar{\psi})$ and the priors $p(\bar{\Phi},\bar{\Sigma})$ and $p(\bar{\psi})$ are known and can be evaluated exactly, whereas $\hat{p}(\bar{\Phi},\bar{\Sigma}|\bar{\psi},Y)$ is a numerical approximation from an auxiliary Gibbs sampler and $\hat{p}(\bar{\psi}|Y)$ is a 'Rao-Blackwellized' estimate (Gelfand, Smith, and Lee 1992). Specifically, we compute

$$
\hat{p}(\bar{\Phi},\bar{\Sigma}|\bar{\psi},Y)=\frac{1}{R}\sum_{r=1}^{R}p(\bar{\Phi},\bar{\Sigma}|\bar{\psi},X^{(r)},Y),
$$

where $\{\Phi^{(r)},\Sigma^{(r)},X^{(r)}\}_{r=1}^{R}$ is obtained from a reduced Gibbs step (Chib 1995; Fuentes-Albero and Melosi 2013) where we approximate $p(\Phi,\Sigma,X|\bar{\psi},Y)$ by drawing from the conditional posteriors

$$
p(\Phi,\Sigma|\bar{\psi},X)\quad\text{and}\quad p(X|\Phi,\Sigma,\bar{\psi},Y).
$$

The Rao-Blackwellized estimate of $p(\bar{\psi}|Y)$ is

$$
\hat{p}(\bar{\psi}|Y)=\frac{1}{R}\sum_{r=1}^{R}p(\bar{\psi}|\Phi^{(r)},\Sigma^{(r)},X^{(r)},Y)
$$

using draws $\{\Phi^{(r)}, \Sigma^{(r)}, X^{(r)}\}_{r=1}^{R}$ from the full Gibbs sampler.

3. The R Package **mfbvar**

The workflow promoted by the **mfbvar** package consists of three main steps. First, an object containing data, hyperparameters and settings is constructed. Second, the prior object is used as input to the main estimation function. Third, the results are processed. The purpose for disentangling the first and second steps is to separate specification from estimation, thereby making the calls in R easier to decipher.

3.1. Specification

The main function for performing the first step is `set_prior`. Its arguments are listed in Table 1. The most important arguments are described below.

- `Y`: a matrix, data frame or tibble with the data. Quarterly variables should be in the final columns of `Y` and store quarterly observations in the final month of each quarter (and `NA` in the remaining cells). For full compatibility with plotting and post-processing functions, the column names should be meaningful variable names. Row names should be monthly dates formatted as `YYYY-MM-DD`. If `Y` is a data frame, a single non-numeric column is allowed to be included, which is then assumed to include the dates. The **mfbvar** package provides some support for single-frequency models in which no observations are missing; in this case, `freq` must be specified accordingly.

- `aggregation`: the aggregation scheme, either `"average"` for the intra-quarterly average, or `"triangular"` for the triangular weighting used by Mariano and Murasawa (2003). The latter is typically used for modeling growth rates, and the former for log-levels; our experience, however, indicates that results tend to be relatively indifferent to the choice.

- `freq`: a character vector indicating the sampling frequency. Either `"m"` for monthly variables or `"q"` for quarterly variables.

- `prior_Pi_AR1`: a numeric vector providing the prior mean for the AR(1) parameters ($\underline{\gamma}$).

- `lambda1, lambda2, lambda3, lambda4`: one-dimensional numeric vectors providing the hyperparameters for overall tightness, cross-variable tightness, lag decay and intercept variance. The defaults are 0.2, 0.5, 1, and 10^4, respectively. Note that `lambda2` is only used in the independent normal specification for Γ, which is only employed if the diffuse prior or a factor stochastic volatility model is used for Σ. The prior variance for the intercept, `lambda4`, is only used if the steady-state prior is not used.

- `prior_psi_mean, prior_psi_Omega`: numeric vector and matrix giving the prior mean and covariance of the steady-state parameters. If the deterministic component of the model includes more terms than a constant, the prior moments are for $\psi = \text{vec}(\Psi)$ and this order should be respected. The package includes a helper function, `interval_to_moments`, for simplifying the specification of priors for the steady states. The function `interval_to_moments` takes a matrix of prior $100 * (1 - \alpha)$ % intervals and returns the mean vector and covariance matrix needed for `set_prior`.

In order to encourage and enable a pipe-like process of specification, the function `update_prior(prior_obj, ...)` can be used to add further specifications to `prior_obj`, where `prior_obj` is the object returned from `set_prior`. The class of the return from `set_prior` and `update_prior` is `mfbvar_prior` for which methods for the generic functions `summary`, `print` and `plot` are implemented (see Section 4).

3.2. Estimation

With an object containing the specifications in place, the second step of the **mfbvar** workflow is to estimate the model. The function for doing so is `estimate_mfbvar`, with arguments:

- `mfbvar_prior`: an object of class `mfbvar_prior` obtained from `set_prior`/`update_prior`
- `prior`: a string equal to `"minn"`, `"ss"` or `"ssng"` for estimating the model using the Minnesota prior, the steady-state prior, or the hierarchical steady-state prior
- `variance`: a string equal to `"iw"`, `"diffuse"`, `"csv"` or `"fsv"` for selecting the inverse Wishart prior, the diffuse prior, the common stochastic volatility model, or the factor stochastic volatility model for the error covariance matrix in the model
- `...`: additional arguments that are passed on to `update_prior` for temporarily overriding settings in `mfbvar_prior`

3.3. Processing

For processing the results, **mfbvar** provides three functions to simplify this step:

- `predict`: a method for the generic `predict` function is implemented. The function returns a data frame according to the concept of tidy data (Wickham 2014) with the forecasts for the variables. The forecasts of the quarterly variables are returned either as monthly or quarterly forecasts. The data frame includes all post-burn-in draws, or quantiles of the posterior predictive distribution.
- `plot`: the generic `plot` function can be used on the estimated object. The plot displays the forecasts and, if applicable, the posterior steady-state intervals.
- `volplot`: a plotting function for displaying the posterior standard deviation of the errors over time. Only applicable if stochastic volatility is used.

As discussed in Section 2, the package also includes estimators for the marginal data density when the normal inverse Wishart prior is used. Given an object `x` obtained from `estimate_mfbvar` with `variance = "iw"`, the marginal data density is estimated by calling `mdd(x)`. If `prior = "minn"`, the only argument that can be provided is `p_trunc` giving the degree of truncation of the truncated normal distribution.

				Regression prior (`prior`)			Variance (`var`)			
Category	Argument	Description	Default	`"minn"`	`"ss"`	`"ssng"`	`"iw"`	`"diffuse"`	`"csv"`	`"fsv"`
General	`Y`	data		✓	✓	✓	✓	✓	✓	✓
	`aggregation`	aggregation scheme	`"average"`	✓	✓	✓	✓	✓	✓	✓
	`freq`	frequencies of variables		✓	✓	✓	✓	✓	✓	✓
	`prior_Pi_AR1`	$\underline{\gamma}$ in (5)	0	✓	✓	✓	✓	✓	✓	✓
	`lambda1`	overall tightness; λ_1 in (6)	0.2	✓	✓	✓	✓	✓	✓	✓
	`lambda2`	cross-var shrinkage; λ_2 in (6)	0.5	✓	✓	✓		✓		✓
	`lambda3`	lag decay; λ_3 in (6)	1	✓	✓	✓	✓	✓	✓	✓
	`lambda4`	intercept variance; λ_4 in (6)	10,000	✓			✓	✓	✓	✓
	`block_exo`	block exogeneity		✓	✓	✓		✓		✓
	`n_lags`	number of lags (p)		✓	✓	✓	✓	✓	✓	✓
	`n_fcst`	number of forecasts	0	✓	✓	✓	✓	✓	✓	✓
	`n_thin`	thinning frequency	1	✓	✓	✓	✓	✓	✓	✓
	`n_burnin`	burn-in draws		✓	✓	✓	✓	✓	✓	✓
	`n_reps`	main draws		✓	✓	✓	✓	✓	✓	✓
	`verbose`	progress bar in console	`FALSE`	✓	✓	✓	✓	✓	✓	✓
SS	`d`	deterministc term			✓	✓	✓	✓	✓	✓
prior	`d_fcst`	`d` for forecasting period			✓	✓	✓	✓	✓	✓
	`prior_psi_mean`	prior mean, $\underline{\psi}$			✓	✓	✓	✓	✓	✓
	`prior_psi_Omega`	prior covariance, $\underline{\Omega}_{\psi}$			✓	✓	✓	✓	✓	✓
	`check_roots`	keep only stable $\Phi(L)$	`FALSE`		✓	✓	✓	✓	✓	✓
Hier.	`s`	scale of proposal for ϕ_{ψ}	-1000 (adaptive)			✓	✓	✓	✓	✓
SS	`prior_ng`	(c_0, c_1) for λ_{ψ} in (10)	0.01			✓	✓	✓	✓	✓
Common	`prior_phi`	mean and variance of ϕ (12)	(0.9, 0.1)	✓	✓	✓			✓	
SV	`prior_sigma2`	df and mean of σ^2 (12)	(0.01, 4)	✓	✓	✓			✓	
Factor	`n_fac`	number of factors		✓	✓	✓				✓
SV	`n_cores`	number of cores	1	✓	✓	✓				✓
	`...`	additional fsv args		✓	✓	✓				✓

Table 1: Overview of elements in the prior object. Checkmarks indicate that the argument applies for the prior or variance specification that is given at the top of the respective columns.

3.4. Use by Other Packages

The computational burden can be large for mixed-frequency VARs and special care has been paid to the implementation of, in particular, the simulation smoother and the sampling of regression parameters. Because of the modularity of MCMC, these implementations can be leveraged by other packages extending the mixed-frequency VAR further. The following functions, implemented in C++ via **RcppArmadillo**, can therefore easily be imported by other packages:

- `simsm_adaptive`, `simsm_adaptive_univariate`: the adaptive simulation smoother presented by Ankargren and Jonéus (2019a), and its extension with univariate filtering suggested by Ankargren and Jonéus (2019b)

- `mvn_rue`, `mvn_rue_int`, `mvn_bcm`: procedures for sampling from multivariate normal posterior distributions using the algorithm by Rue (2001), the Rue (2001) algorithm including non-zero means for the AR(1) parameters, and the Bhattacharya *et al.* (2016) algorithm

- `update_csv`, `update_fsv`, `update_ssng`: functions for the sampling steps required for common stochastic volatility, factor stochastic volatility, and the hierarchical steady-state prior

The functions are available as header files in `mfbvar/inst/include` and can therefore easily be imported by other packages, see e.g. Wickham (2015, chap. 10).

4. Illustration

To illustrate the basic funcionality of the **mfbvar** package we here estimate mixed-frequency Bayesian VAR models on US data. The data can be retrieved from the ALFRED database provided by the Federal Reserve Bank of St. Louis through the **alfred** package (Kleen 2018). The model we will use includes inflation and unemployment, which are published monthly, and GDP growth, which is published quarterly.

```
R> library("purrr")
R> library("dplyr")
R> library("ggplot2")
R> library("alfred")
R>
R> variables <- c("CPIAUCSL", "UNRATE", "GDPC1")
R> out <- map(variables, get_alfred_series,
+              observation_start = "1980-01-01",
+              observation_end = "2018-11-01",
+              realtime_start = "2018-12-10",
+              realtime_end = "2018-12-10")
```

The data start in January 1980 and we retrieve the vintage available on December 10, 2018. The **mfbvar** package expects data to be provided in a data frame or a matrix. Each row represents a month, and quarterly variables should store observations on the final month of each quarter and NA elsewhere. The data obtained from FRED needs two adjustments before being used by **mfbvar**. First, quarterly GDP is stored on the first month of each quarter and needs to be moved to the end of the quarter for compatibility with **mfbvar**. The observed date of GDP is therefore shifted by two months.

```
R> out[[3]]$date <- out[[3]]$date + months(2)
```

Second, consumer price index and real GDP need to be transformed into annualized growth rates.

```
R> log_diff <- function(x, lag = 1) {
+    c(rep(NA, lag), 1200/lag * diff(log(x), lag = lag))
+ }
R>
R> mf_df <- reduce(out, full_join, by = c("date", "realtime_period")) %>%
+    mutate(CPIAUCSL = log_diff(CPIAUCSL),
+           GDPC1 = log_diff(GDPC1, lag = 3)) %>%
+    filter(date >= "1980-04-01") %>%
+    select(-realtime_period)
R>
R> tail(mf_df)
```

```
          date  CPIAUCSL UNRATE    GDPC1
459 2018-06-01 1.5413092    4.0 4.074644
460 2018-07-01 2.0504124    3.9       NA
```

Inflation	Unemployment	GDP
$(1, 3)$	$(4, 8)$	$(1, 3)$

Table 2: 95 % prior probability intervals for steady states (unconditional means)

```
461 2018-08-01 2.6712683    3.9       NA
462 2018-09-01 0.7049857    3.7 3.444026
463 2018-10-01 3.9602192    3.7       NA
464 2018-11-01        NA    3.7       NA
```

With the data in place, the workflow of the package next requires the user to first specify an object containing all the prior information, and then calling the main function to estimate the model.

4.1. Setting the Prior

To create an initial, minimal prior we call `set_prior()` with the following arguments:

```
R> library("mfbvar")
R> prior <- set_prior(Y = mf_df, freq = c("m", "m", "q"),
+                     n_lags = 4, n_burnin = 1000, n_reps = 1000)
```

The `print` method for the prior displays what model specifications can be used with the provided information.

```
R> prior

The following elements of the prior object have not been set:
 block_exo d d_fcst prior_psi_mean prior_psi_Omega n_fac

Checking if the steady-state prior can be used... FALSE
 Missing elements: d prior_psi_mean
Checking if a Minnesota-style prior can be used... TRUE

Checking if common stochastic volatility can be used... TRUE

Checking if factor stochastic volatility can be used... FALSE
 Missing element: n_fac
```

As is indicated, we need to provide the deterministic term d_t as well as the prior mean for the parameters in ψ to also enable the possibility of estimating the model using the (hierarchical) steady-state prior. In many applications, $d_t = 1$ and so the only deterministic term is the unconditional mean. The prior is commonly specified as independent 95 % prior probability intervals. The prior intervals used here are given in Table 2 and mirror those used by Louzis (2019); Ankargren *et al.* (2019).

A helper function `interval_to_moments()` is included in **mfbvar** to convert intervals to prior moments $\underline{\psi}$ and $\underline{\Omega}_\psi$. Having obtainted the moments, the prior is updated to include also specifications that enable estimation of the model using the steady-state prior.

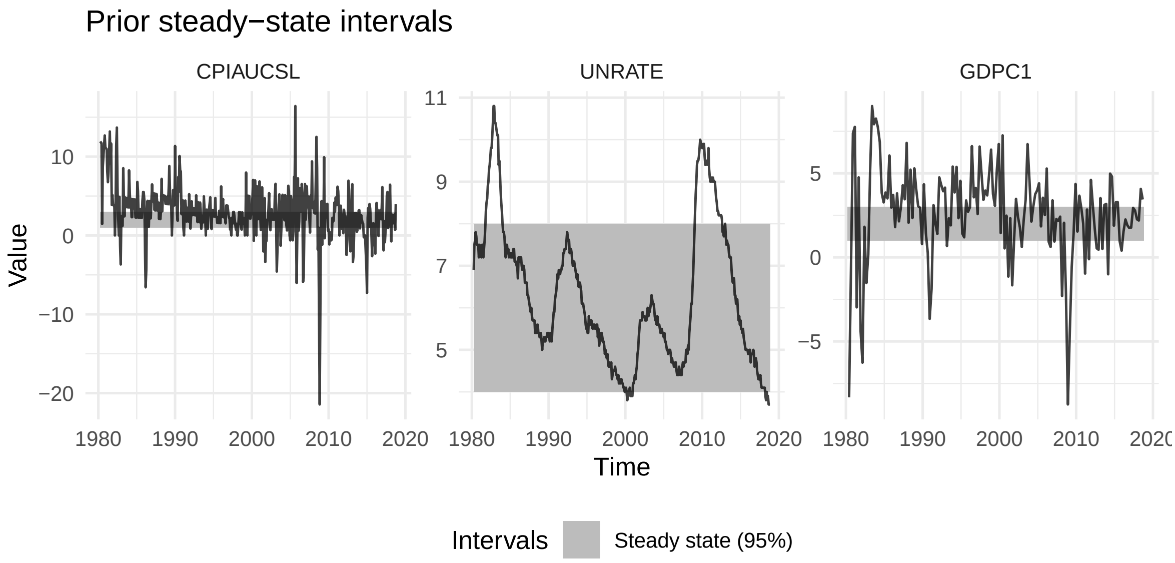


Figure 1: Prior steady-state intervals

```
R> prior_intervals <- matrix(c(1, 3,
+                              4, 8,
+                              1, 3), ncol = 2, byrow = TRUE)
R> moments <- interval_to_moments(prior_intervals)
R> prior <- update_prior(prior,
+                        d = "intercept",
+                        prior_psi_mean = moments$prior_psi_mean,
+                        prior_psi_Omega = moments$prior_psi_Omega)
```

The argument `d` should generally be a matrix, but because intercept-only applications are common passing only the string `"intercept"` is allowed. The prior steady-state intervals can be visualized by calling `plot` on the prior object. The intervals used here are displayed in Figure 1.

```
R> plot(prior)
```

We will next make forecasts 24 months ahead and must therefore update the prior to accomodate this request. The argument `n_fcst` takes the number of forecasts desired in terms of months, i.e. `n_fcst = 24` corresponds to two years.

```
R> prior <- update_prior(prior, n_fcst = 24)
```

The prior is now fully specified also for the steady-state prior. A summary of the specification can be obtained from the `summary` method implemented for the prior object:

```
R> summary(prior)

PRIOR SUMMARY
----------------------------
```

```
General specification:
  Y: 3 variables, 464 time points
  aggregation: average
  freq: 2 monthly and 1 quarterly variable
  prior_Pi_AR1: 0 0 0
  lambda1: 0.2
  lambda2: 0.5
  lambda3: 1
  lambda4: 10000
  block_exo: 0 block exogenous variables
  n_lags: 4
  n_fcst: 24
  n_burnin: 1000
  n_reps: 1000
----------------------------
Steady-state prior:
  d: intercept
  d_fcst: intercept
  prior_psi_mean: prior mean vector of steady states
  prior_psi_Omega: prior covariance matrix of steady states
  check_roots: FALSE
----------------------------
Hierarchical steady-state prior:
  s: -1000
  c0: 0.01
  c1: 0.01
----------------------------
Common stochastic volatility:
  prior_phi: mean = 0.9, var = 0.1
  prior_sigma2: mean = 0.01, df = 4
----------------------------
Factor stochastic volatility:
  n_fac: <missing>
  n_cores: 1
----------------------------
Other:
  verbose: FALSE
```

4.2. Estimating the Model

The prior is prepared and its necessary components have been provided and so estimation is possible by calling the `estimate_mfbvar` function. In calling the function, either `prior = "minn"`, `prior = "ss"` or `prior = "ssng"` should be provided to indicate which regression prior to use. The argument `variance` determines the form of the error covariance matrix. We first consider the steady-state prior with and without hierachical shrinkage, and with an inverse Wishart prior for the error covariance matrix.

```
R> mod_ss_iw   <- estimate_mfbvar(prior, prior = "ss", variance = "iw")
R> mod_ssng_iw <- estimate_mfbvar(prior, prior = "ssng", variance = "iw")
```

To estimate models with stochastic volatility, only the variance argument needs to be changed. For the factor stochastic volatility model, the number of factors must also be provided.

```
R> mod_ss_csv  <- estimate_mfbvar(prior, prior = "ss", variance = "csv")
R> mod_ss_fsv <- estimate_mfbvar(prior, prior = "ss", variance = "fsv",
+                                n_fac = 1)
```

Temporary arguments can be added to the call to estimate_mfbvar, like n_fac = 1 above. The purpose of first creating a prior object and then estimating the model is illustrated in the preceding code snippet. For two models with different priors but similar settings, we can leverage the same object (i.e., prior). Estimation of multiple models is thereby simplified as we can reuse the previous settings.

4.3. Processing the Results

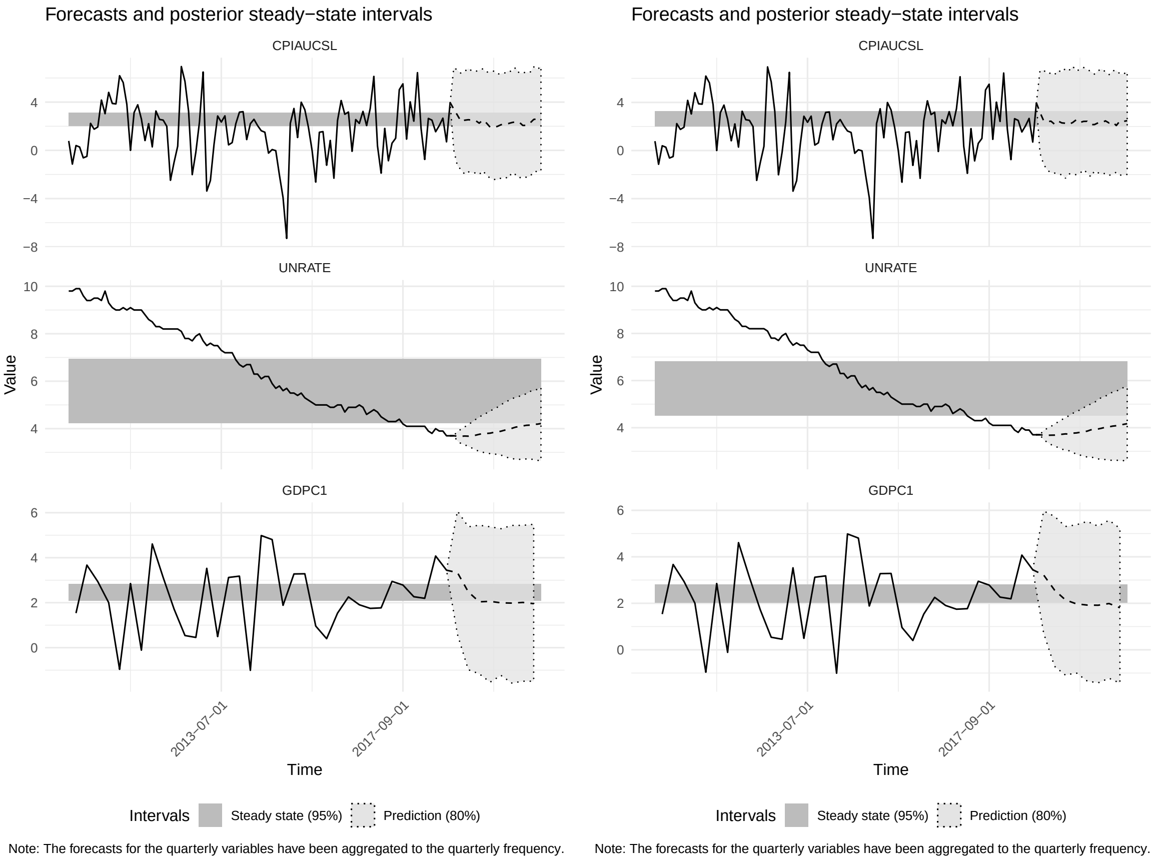
The principle of tidy data (Wickham 2014) has been preserved in creating a method for the predict function.

```
R> predict(mod_ss_iw, pred_bands = 0.8)
```

```
# A tibble: 60 x 6
   variable  time fcst_date    lower median upper
   <chr>    <dbl> <date>       <dbl>  <dbl> <dbl>
 1 CPIAUCSL   463 2018-10-01  3.96     3.96  3.96
 2 CPIAUCSL   464 2018-11-01  0.0232   3.42  6.85
 3 CPIAUCSL   465 2018-12-01 -1.29     2.92  6.67
 4 CPIAUCSL   466 2019-01-01 -1.60     2.40  6.38
 5 CPIAUCSL   467 2019-02-01 -1.98     2.50  6.55
 6 CPIAUCSL   468 2019-03-01 -1.78     2.54  6.80
 7 CPIAUCSL   469 2019-04-01 -1.87     2.51  6.58
 8 CPIAUCSL   470 2019-05-01 -1.87     2.47  6.54
 9 CPIAUCSL   471 2019-06-01 -2.01     2.25  6.67
10 CPIAUCSL   472 2019-07-01 -1.70     2.51  6.76
# ... with 50 more rows
```

The forecasts produced by a mixed-frequency VAR are at the monthly frequency. Because the frequency of interest for quarterly variables is the quarterly frequency, forecasts are aggregated by default in the predict method. To visualize forecasts and posterior intervals for the steady states (if available), plot can be called directly on the output of estimate_mfbvar.

```
R> plot(mod_ss_iw, plot_start = "2010-01-01", nrow_facet = 3)
R> plot(mod_ssng_iw, plot_start = "2010-01-01", nrow_facet = 3)
```



Note: The forecasts for the quarterly variables have been aggregated to the quarterly frequency.

Note: The forecasts for the quarterly variables have been aggregated to the quarterly frequency.

(a) Steady-state prior

(b) Hierarchical steady-state prior

Figure 2: Forecasts and posterior steady-state intervals

Figure 2 displays the forecasts and posterior steady-state intervals obtained using the steady-state prior with and without hierarchical shrinkage. The posterior intervals are similar, with a narrower interval for unemployment using the hierarchical specification, and with almost indistinguishable differences for inflation and GDP. Because of the similarity between the models, the forecasts from both models in Figure 2 show that the assessment made by the model is that the economy is in a relatively steady state with a stable rate of growth in the future. Unemployment is expected to rise slowly and return to the steady state in the long run. By default, the plotting method displays the forecasts for the quarterly variables on the quarterly scale, as in the output from `predict`.

In order to see the difference between constant and time-varying error covariance matrices, we next compare the distributions of the GDP forecasts. By letting `pred_bands = NULL` be an argument to `predict`, we obtain the entire set of forecasts from the model. Combining them and plotting the distributions using the **ggridges** package (Wilke 2018) allows us to easily compare the distributions.

```
R> pred_iw <- predict(mod_ss_iw, pred_bands = NULL)
R> pred_csv <- predict(mod_ss_csv, pred_bands = NULL)
R> pred_fsv <- predict(mod_ss_fsv, pred_bands = NULL)
```

```
R> pred_df <- bind_rows("Inverse Wishart" = pred_iw,
+                        "Common stochastic volatility" = pred_csv,
+                        "Factor stochastic volatility" = pred_fsv,
+                        .id = "Variance") %>%
+    filter(variable == "GDPC1")
R> ggplot(pred_df, aes(y = factor(fcst_date), x = fcst, fill = Variance)) +
+    ggridges::stat_density_ridges(quantile_lines = TRUE,
+                                  quantiles = 2, alpha = 0.5) +
+    labs(x = "US GDP Growth",
+         y = "Date of Forecast") +
+    coord_cartesian(xlim = c(-5, 10)) +
+    theme_minimal() +
+    scale_fill_brewer(palette = "YlGnBu")
```

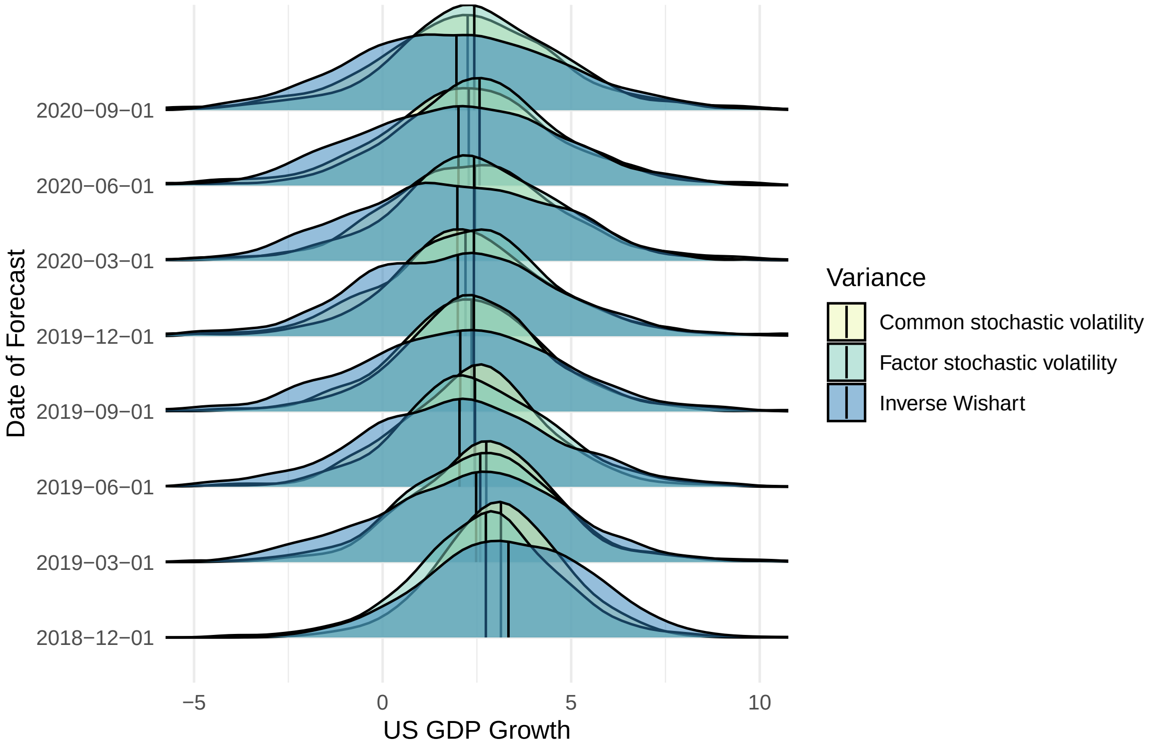
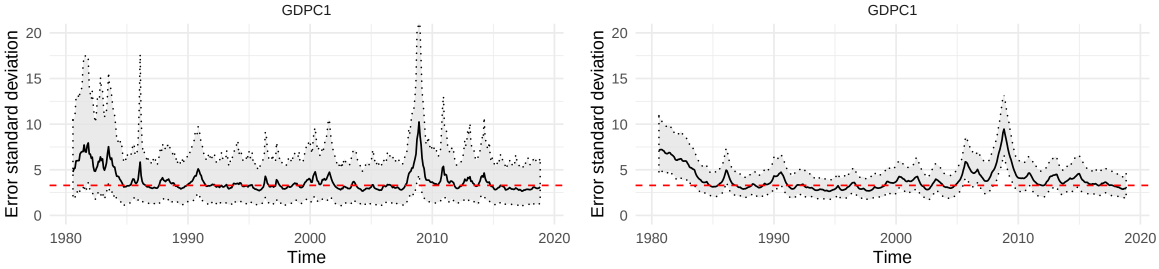


Figure 3: Distributions of forecasts produced using the steady-state prior with constant or time-varying error covariance. The vertical lines represent the medians.

Figure 3 shows that stochastic volatility leads to narrower predictive distributions for the current forecast. In terms of the central tendencies, however, the differences are negligible. The resemblance is greater between the predictive distributions obtained from the two models using stochastic volatility than between any of the models with stochastic volatility and the model with constant volatility. Thus, the choice of stochastic volatility may be less important than whether to include it at all.

The reason for the somewhat wider distribution obtained using constant volatility is that the error covariance matrix that is obtained is a compromise between regimes of high and low



(a) Factor stochastic volatility (b) Common stochastic volatility

Figure 4: Standard deviation of the error term in the equation for GDP growth. The black solid and red dashed lines are the medians from the models with factor stochastic and constant volatility, respectively. The bands are obtained from the 95 % posterior point-wise intervals.

volatility. Using stochastic volatility, there is no need to make this compromise. To see this point more clearly, the error standard deviations implied by the stochastic volatility models can be plotted using the `varplot` function.

```
R> const_vol <- median(sqrt(mod_ss_iw$Sigma[3, 3, ]))
R>
R> varplot(mod_ss_fsv, variables = "GDPC1") +
+    geom_hline(yintercept = const_vol ,
+                color = "red", linetype = "dashed") +
+    coord_cartesian(ylim = c(0, 20))
R>
R> varplot(mod_ss_csv, variables = "GDPC1") +
+    geom_hline(yintercept = const_vol,
+                color = "red", linetype = "dashed") +
+    coord_cartesian(ylim = c(0, 20))
```

Figure 4 shows that the standard deviation of the GDP error term has varied substantially. Periods of particularly high volatility are evident in the early 1980s, 1990s and 2000s as well as during the recent financial crisis at the end of the 2000s. The estimated volatilities are similar between the two models. They both capture the same peaks with the sole difference being that the model with common stochastic volatility estimated a peak in volatility around 2005 that the model with factor stochastic volatility did not. Volatility has in recent years been lower than usual. The line displaying the standard deviation from the model with constant volatility appears to capture a baseline, which the time-varying volatility is currently deviating somewhat from. For this reason, we see that the distribution of the forecasts in Figure 3 is narrower under a stochastic volatility specification.

4.4. Marginal Data Density

A generic function `mdd()` is provided that facilitates estimation of the marginal data density if the inverse Wishart prior is used for Σ. The marginal data density for the model using the steady-state prior with constant volatility is

```
R> mdd(mod_ss_iw)
```

```
[1] -1302.976
```

The marginal data density in itself is not particularly informative. One of its roles, however, is that it can be used as a way of selecting hyperparameters. By searching over a grid of values, the pair (λ_1, λ_3) that maximizes the marginal data density can be used. Such a grid search is embarrassingly parallel and can be computed using the **parallel** package (R Core Team 2019).

```
R> library("parallel")
R> par_fun <- function(lambda1, prior) {
+    set.seed(2019)
+    mod_par <- estimate_mfbvar(prior, prior = "ss", variance = "iw",
+                               lambda1 = lambda1, lambda3 = 1)
+    mdd(mod_par)
+  }
R>
R> cl <- makeCluster(4)
R> clusterEvalQ(cl, library("mfbvar"))
R> lambda1_seq <- seq(0.05, 1, by = 0.05)
R> result <- parSapply(cl, lambda1_seq,
+                      par_fun, prior = prior)
R> stopCluster(cl)
```

We are here fixing the lag decay to $\lambda_3 = 1$ and consider the grid of values $\{0.05, 0.1, \ldots, 0.95, 1.00\}$ for λ_1. Figure 5 displays the results.

```
R> plot_df <- tibble(lambda1 = lambda1_seq,
+          mdd = result)
R> ggplot(plot_df, aes(x = lambda1, y = mdd)) +
+    geom_line() +
+    geom_point(data = filter(plot_df, mdd == max(mdd))) +
+    labs(y = "Marginal data density (log)",
+         x = bquote(lambda[1])) +
+    theme_minimal()
```

The highest value of the marginal data density is obtained for $\lambda_1 = 0.4$, indicating a lower degree of shrinkage in the model than what has been used in the previous estimations. However, the marginal data density is relatively flat as a function of λ_1 around 0.2–0.5, thereby demonstrating a certain degree of indifference with respect to λ_1 in this neighborhood.

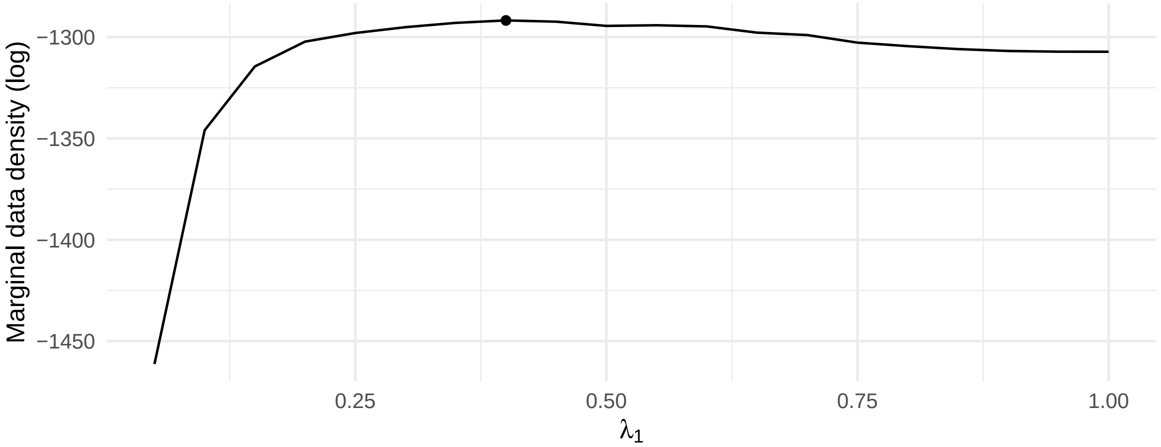


Figure 5: Logarithm of the marginal data density as a function of λ_1 with $\lambda_3 = 1$. The point shows the maximum point at $\lambda_1 = 0.4$.

5. Conclusion

The **mfbvar** package introduces a user-friendly interface for estimating mixed-frequency vector autoregressions using Bayesian techniques. As such, it fills a void and provides additional functionality compared to existing packages such as **midasr** (Ghysels *et al.* 2016) and **BMR** (O'Hara 2017).

We have discussed the models that can be estimated in the **mfbvar** and the workflow of the package. An application to a small mixed-frequency VAR using monthly inflation and unemployment, and quarterly GDP growth was used to illustrate the functionality of the package. Aside from the features directly visible to the user, the package also provides header files for the key steps of the MCMC algorithm used for estimating the models. These can easily be imported and used by other packages, and thus we hope to make also the use of other, more customized mixed-frequency VARs more frequent.

Acknowledgments

This work has been supported by Jan Wallanders och Tom Hedelius stiftelse samt Tore Browaldhs stiftelse, grant number P2016-0293:1.

References

Aguilar O, West M (2000). "Bayesian Dynamic Factor Models and Portfolio Allocation." *Journal of Business & Economic Statistics*, **18**(3), 338–357. doi:10.1080/07350015.2000.10524875.

Allaire J, Francois R, Ushey K, Vandenbrouck G, Geelnard M, Intel (2019). *RcppParallel: Parallel Programming Tools for 'Rcpp'*. R package version 4.4.3, URL https://CRAN.R-project.org/package=RcppParallel.

Ankargren S, Bjellerup M, Shahnazarian H (2017). "The Importance of the Financial System for the Real Economy." *Empirical Economics*, **53**(4), 1553–1586. doi:10.1007/s00181-016-1175-4.

Ankargren S, Jonéus P (2019a). "Simulation smoothing for large mixed-frequency VARs." *Preprint available on arXiv*. arXiv:1907.01075.

Ankargren S, Jonéus P (2019b). "Estimating large mixed-frequency Bayesian VAR Models." *Preprint available on arXiv*. arXiv:1912.02231.

Ankargren S, Unosson M, Yang Y (2018). "A Mixed-Frequency Bayesian Vector Autoregression with a Steady-State Prior." *Working Paper No. 2018:2*, Department of Statistics, Uppsala University.

Ankargren S, Unosson M, Yang Y (2019). "A Flexible Mixed-Frequency Vector Autoregression with a Steady-State Prior." *Preprint available on arXiv*. arXiv:1911.09151.

Baffigi A, Golinelli R, Parigi G (2004). "Bridge models to forecast the euro area GDP." *International Journal of Forecasting*, **20**(3), 447–460. doi:10.1016/S0169-2070(03)00067-0.

Bańbura M, Giannone D, Reichlin L (2010). "Large Bayesian Vector Auto Regressions." *Journal of Applied Econometrics*, **25**(1), 71–92. doi:10.1002/jae.1137.

Bhattacharya A, Chakraborty A, Mallick BK (2016). "Fast sampling with Gaussian scale mixture priors in high-dimensional regression." *Biometrika*, **103**(4), 985–991. doi:10.1093/biomet/asw042.

Carriero A, Clark TE, Marcellino M (2016). "Common Drifting Volatility in Large Bayesian VARs." *Journal of Business & Economic Statistics*, **34**(3), 375–390. doi:10.1080/07350015.2015.1040116.

Chib S (1995). "Marginal Likelihood from the Gibbs Output." *Journal of the American Statistical Association*, **90**(432), 1313–1321. ISSN 1537274X. doi:10.1080/01621459.1995.10476635.

Clark TE (2011). "Real-Time Density Forecasts From Bayesian Vector Autoregressions With Stochastic Volatility." *Journal of Business & Economic Statistics*, **29**(3), 327–341. doi:10.1198/jbes.2010.09248.

Clark TE, Ravazzolo F (2015). "Macroeconomic Forecasting Performance under Alternative Specifications of Time-Varying Volatility." *Journal of Applied Econometrics*, **30**(4), 551–575. doi:10.1002/jae.2379.

Cogley T, Sargent TJ (2005). "Drifts and Volatilities: Monetary Policies and Outcomes in the Post WWII US." *Review of Economic Dynamics*, **8**(2), 262–302. doi:10.1016/j.red.2004.10.009.

D'Agostino A, Gambetti L, Giannone D (2013). "Macroeconomic Forecasting and Structural Change." *Journal of Applied Econometrics*, **28**(1), 82–101. doi:10.1002/jae.1257.

Dieppe A, Legrand R, van Roye B (2016). "The BEAR Toolbox." *Working Paper No. 1934*, European Central Bank.

Durbin J, Koopman SJ (2002). "A Simple and Efficient Simulation Smoother for State Space Time Series Analysis." *Biometrika*, **89**(3), 603–615. doi:10.1093/biomet/89.3.603.

Eddelbuettel D, Francois R (2011). "Seamless R and C++ Intgration with Rcpp." *Journal of Statistical Software*, **40**(8). doi:10.18637/jss.v040.i08.

Eddelbuettel D, Sanderson C (2014). "RcppArmadillo: Accelerating R with High-Performance C++ Linear Algebra." *Computational Statistics and Data Analysis*, **71**, 1054–1063. doi:10.1016/j.csda.2013.02.005.

Foroni C, Marcellino M (2013). "A Survey of Econometric Methods for Mixed-Frequency Data." doi:10.2139/ssrn.2268912.

Fuentes-Albero C, Melosi L (2013). "Methods for Computing Marginal Data Densities from the Gibbs Output." *Journal of Econometrics*, **175**(2), 132–141. doi:10.1016/j.jeconom.2013.03.002.

Gelfand AE, Smith AFM, Lee Tm (1992). "Bayesian Analysis of Constrained Parameter and Truncated Data Problems Using Gibbs Sampling." *Journal of the American Statistical Association*, **87**(418), 523–532. doi:10.2307/2290286.

Ghysels E, Kvedaras V, Zemlys V (2016). "Mixed Frequency Data Sampling Regression Models: The R Package midasr." *Journal of Statistical Software*, **72**(4). doi:10.18637/jss.v072.i04.

Ghysels E, Sinko A, Valkanov R (2007). "MIDAS Regressions: Further Results and New Directions." *Econometric Reviews*, **26**(1), 53–90. doi:10.1080/07474930600972467.

Götz TB, Hauzenberger K (2018). "Large Mixed-Frequency VARs with a Parsimonious Time-Varying Parameter Structure." *Discussion Paper No. 40*, Deutsche Bundesbank.

Griffin JE, Brown PJ (2010). "Inference with Normal-Gamma Prior Distributions in Regression Problems." *Bayesian Analysis*, **5**(1), 171–188. ISSN 19360975. doi:10.1214/10-BA507.

Hamilton JD (1994). *Time Series Analysis*. Princeton University Press, Princeton, NJ.

Hörmann W, Leydold J (2014). "Generating generalized inverse Gaussian random variates." *Statistics and Computing*, **24**(4), 547–557. ISSN 1573-1375. doi:10.1007/s11222-013-9387-3. URL https://doi.org/10.1007/s11222-013-9387-3.

Huber F, Feldkircher M (2019). "Adaptive Shrinkage in Bayesian Vector Autoregressive Models." *Journal of Business & Economic Statistics*, **37**(1), 27–39. doi:10.1080/07350015.2016.1256217.

Jarociński M (2015). "A Note on Implementing the Durbin and Koopman Simulation Smoother." *Computational Statistics & Data Analysis*, **91**, 1–3. doi:10.1016/j.csda.2015.05.001.

Jarociński M, Smets F (2008). "House Prices and the Stance of Monetary Policy." *Working Paper No. 891*, European Central Bank.

Karlsson S (2013). *Forecasting with Bayesian Vector Autoregression*, volume 2. Elsevier B.V. doi:10.1016/B978-0-444-62731-5.00015-4.

Kastner G (2016). "Dealing with Stochastic Volatility in Time Series Using the R Package stochvol." *Journal of Statistical Software*, **69**(5). doi:10.18637/jss.v069.i05.

Kastner G (2019). *factorstochvol: Bayesian Estimation of (Sparse) Latent Factor Stochastic Volatility Models*. R package version 0.9, URL https://cran.r-project.org/package=factorstochvol.

Kastner G, Frühwirth-Schnatter S (2014). "Ancillarity-Sufficiency Interweaving Strategy (ASIS) for Boosting MCMC Estimation of Stochastic Volatility Models." *Computational Statistics and Data Analysis*, **76**, 408–423. doi:10.1016/j.csda.2013.01.002.

Kastner G, Frühwirth-Schnatter S, Freitas Lopes H (2017). "Efficient Bayesian Inference for Multivariate Factor Stochastic Volatility Models." *Journal of Computational and Graphical Statistics*, **26**(4), 905–917. doi:10.1080/10618600.2017.1322091.

Kastner G, Huber F (2018). "Sparse Bayesian Vector Autoregressions in Huge Dimensions." *Preprint available on arXiv*. arXiv:1704.03239.

Kim S, Shephard N, Chib S (1998). "Stochastic Volatility: Likelihood Inference and Comparison with ARCH Models." *The Review of Economic Studies*, **65**(3), 361–393. ISSN 0034-6527. doi:10.1111/1467-937X.00050.

Kleen O (2018). *alfred: Downloading Time Series from ALFRED Database for Various Vintages*. R package version 0.1.6, URL https://CRAN.R-project.org/package=alfred.

Litterman RB (1986). "A Statistical Approach to Economic Forecasting." *Journal of Business & Economic Statistics*, **4**(1), 1–4. doi:10.1080/07350015.1986.10509485.

Louzis DP (2019). "Steady-state modeling and macroeconomic forecasting quality." *Journal of Applied Econometrics*, **34**(2), 285–314. doi:10.1002/jae.2657.

Mariano RS, Murasawa Y (2003). "A New Coincident Index of Business Cycles Based on Monthly and Quarterly Series." *Journal of Applied Econometrics*, **18**(4), 427–443. ISSN 08837252. doi:10.1002/jae.695.

McCausland WJ, Miller S, Pelletier D (2011). "Simulation smoothing for state-space models: A computational efficiency analysis." *Computational Statistics & Data Analysis*, **55**(1), 199–212. doi:10.1016/j.csda.2010.07.009.

O'Hara K (2017). "Bayesian Macroeconometrics in R." URL http://www.kthohr.com/bmr.html.

Österholm P (2010). "The Effect on the Swedish Real Economy of the Financial Crisis." *Applied Financial Economics*, **20**(4), 265–274. doi:10.1080/09603100903357408.

Primiceri GE (2005). "Time Varying Structural Vector Autoregressions and Monetary Policy." *The Review of Economic Studies*, **72**(3), 821–852. doi:10.1111/j.1467-937X.2005.00353.x.

R Core Team (2019). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. URL `https://www.R-project.org/`.

Roberts GO, Rosenthal JS (2009). "Examples of Adaptive MCMC." *Journal of Computational and Graphical Statistics*, **18**(2), 349–367. `doi:10.1198/jcgs.2009.06134`.

Rue H (2001). "Fast sampling of Gaussian Markov random fields." *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, **63**, 325–339.

Sanderson C, Curtin R (2016). "Armadillo: A Template-Based C++ Library for Linear Algebra." *The Journal of Open Source Software*, **1**(2). `doi:10.21105/joss.00026`.

Schorfheide F, Song D (2015). "Real-Time Forecasting with a Mixed-Frequency VAR." *Journal of Business & Economic Statistics*, **33**(3), 366–380. `doi:10.1080/07350015.2014.954707`.

Stock JH, Watson MW (2001). "Vector Autoregressions." *Journal of Economic Perspectives*, **15**(4), 101–115. `doi:10.1257/jep.15.4.101`.

Tusell F (2011). "Kalman Filtering in R." *Journal of Statistical Software*, **39**(2). `doi:10.18637/jss.v039.i02`.

Villani M (2009). "Steady State Priors for Vector Autoregressions." *Journal of Applied Econometrics*, **24**(4), 630–650. `doi:10.1002/jae.1065`.

Wickham H (2014). "Tidy Data." *Journal of Statistical Software*, **59**(10). `doi:10.18637/jss.v059.i10`.

Wickham H (2015). *Advanced R*. Chapman and Hall/CRC, Philadelphia, PA.

Wilke CO (2018). *ggridges: Ridgeline Plots in 'ggplot2'*. R package version 0.5.1, URL `https://CRAN.R-project.org/package=ggridges`.

Affiliation:

Sebastian Ankargren, Yukai Yang
Department of Statistics
Uppsala University
P.O. Box 513, 751 20 Uppsala
Sweden
E-mail: `sebastian.ankargren@statistics.uu.se`
URL: `http://ankargren.github.io/`
and
Yukai Yang
Center for Data Analytics
Stockholm School of Economics
P.O. Box 6501, 113 83 Stockholm
Sweden
E-mail: `yukai.yang@statistics.uu.se`
URL: `http://yukai-yang.github.io/`