Journal of Statistical Software

MMMMMM YYYY, Volume VV, Issue II. *doi: 10.18637/jss.v000.i00*

VARshrink: An R Software Package for Shrinkage Estimation for Vector Autoregressive Models

Namgil Lee
Kangwon National University

Heon-Young Yang
Kangwon National University

Sung-Ho Kim
KAIST

Abstract

We introduce an R software package, **VARshrink**, for shrinkage estimation of vector autoregressive (VAR) models. We note that other R packages for VAR shrinkage are mostly for Bayesian VAR models using informative priors, which are parametric methods. On the other hand, the **VARshrink** is an integrative R package delivering nonparametric, parametric, and semiparametric methods in a unified and consistent manner. The **VARshrink** currently contains four VAR shrinkage estimation methods, which are the multivarate ridge regression, a nonparametric shrinkage method, a full Bayesian approach using a noninformative prior, and a semiparametric Bayesian approach using an informative prior. We provide a common formulation for VAR models so that all the shrinkage methods can be run by a simple interface function named `VARshrink()`. We explained basic principles of the shrinkage methods with a focus on the parameters involved for estimation and control. Especially, we clearly presented a mathematical expression for shrinkage estimators of the VAR parameters inferred by each shrinkage method, where shrinkage intensity parameters were denoted by λ and λ_v. We provided sample R codes for demonstration and numerical experiments to compare performances of the shrinkage methods.

Keywords: vector autoregression (VAR), high-dimensionality, shrinkage, multivariate time series..

1. Introduction

Let $\mathbf{y}_t = (y_{t1}, y_{t2}, \ldots, y_{tD})^\top$ denote a $D \times 1$ vector of D endogenous variables. A vector autoregressive (VAR) model of order p can be expressed as

$$\mathbf{y}_t = \sum_{i=1}^{p} \mathbf{A}_i \mathbf{y}_{t-i} + \mathbf{c} + \boldsymbol{\epsilon}_t, \tag{1}$$

where $\mathbf{A}_i$ is a $D \times D$ coefficient matrix ($i = 1, 2, \ldots, p$), $\mathbf{c}$ is a $D \times 1$ constant vector, $\boldsymbol{\epsilon}_t = (\epsilon_{t1}, \epsilon_{t2}, \ldots, \epsilon_{tD})^\top$ is a $D \times 1$ noise vector (Hamilton 1994; Tsay 2005).

1.1. Backgrounds

Recently, several R packages have been developed for parameter estimation and forecasting using stochastic time series models. The **forecast** package provides various methods and tools for univariate time series models such as the ARIMA and ETS models (Hyndman, Athanasopoulos, Bergmeir, Caceres, Chhay, O'Hara-Wild, Petropoulos, Razbash, Wang, Yasmeen, R Core Team, Ihaka, Reid, Shaub, Tang, and Zhou 2018). The **MTS** package has been developed for a wide variety of multivariate linear time series models and multivariate volatility models such as the VARMA, multivariate EWMA, and low-dimensional BEKK models (Tsay and Wood 2018). The **vars** package provides methods for multivariate time series analysis using the VAR, SVAR, and SVEC models (Pfaff and Stigler 2018).

In this study, we focus on the shrinkage estimation of VAR model parameters. Shrinkage estimation methods have been playing crucial roles in high-dimensional statistical modeling; see, e.g., Beltrachini, von Ellenrieder, and Muravchik (2013), Böhm and von Sachs (2009), Fiecas and Ombao (2011) and Ledoit and Wolf (2004). For VAR models, several shrinkage estimation methods have been suggested such as a nonparametric shrinkage method (Opgen-Rhein and Strimmer 2007b), Bayesian VARs using informative priors (Bańbura, Giannone, and Reichlin 2010; Doan, Litterman, and Sims 1984; Koop and Korobilis 2010; Litterman 1986), Bayesian VARs using noninformative priors (Ni and Sun 2005; Sun and Ni 2004), a semiparametric Bayesian approach adopting a modified K-fold cross validation (Lee, Choi, and Kim 2016).

Due to its popularity in macroeconomic time series analysis, several Bayesian VAR methods have been implemented in R packages. For example, the function `BVAR()` in **MTS** implements a Bayesian VAR method using an informative prior (Tsay and Wood 2018); the package **bvarsv** implements Bayesian VAR models with stochastic volatility and time-varying parameters (Krueger 2015; Koop and Korobilis 2010; Primiceri 2005).

On the other hand, other types of VAR shrinkage methods including nonparametric approaches have been implemented for other purposes than multivariate time series analysis in R. For instance, the function `cov.shrink()` in the package **corpcor** was implemented to compute shrinkage estimates of covariances, and it was applied to estimate VAR coefficients in Opgen-Rhein and Strimmer (2007b) (Schäfer, Opgen-Rhein, Zuber, Ahdesmäki, Silva, and Strimmer 2017). In addition, VAR models can be reformulated into multivariate regression problems so that penalized least squares methods have been used for shrinkage estimation of VAR parameters, e.g., the functions `lm.gls()` for generalized least squares and `lm.ridge()` for ridge regression in the package **MASS** (Ripley, Venables, Bates, Hornik, Gebhardt, and Firth 2018); the function `glmnet()` for Lasso and Elastic-Net regularized generalized linear models in the package **glmnet** (Friedman, Hastie, Tibshirani, Simon, Narasimhan, and Qian 2018); the function `linearRidge()` for ridge regression in the package **ridge** (Moritz and Cule 2018).

1.2. Main Purpose

While Bayesian approaches have been widely used in the literature, we note that nonparametric and semiparametric approaches have advantages in the case of high-dimensional VARs

with more than several hundreds of time series variables due to their relatively low computational costs (Opgen-Rhein and Strimmer 2007b). Despite of relatively high computational costs, Bayesian approaches can impose correct assumptions on the multivariate time series data flexibly, such as VAR roots near unity and correlations between noise processes (Lee *et al.* 2016). In this sense, a semiparametric approach can be a trade-off between nonparametric and parametric approaches (Lee *et al.* 2016).

In this study, we developed an integrative R package, **VARshrink**, for implementing nonparametric, parametric, and semiparametric approaches for shrinkage estimation of VAR models. By providing a simple interface function to the different types of approaches, the performances of the methods can be easily compared. We also provide model selection criteria such as AIC and BIC, which can be used to compare several VAR models and to select an optimal lag order.

This paper is organized as follows. In Section 2, we explain the general distribution assumption for the noise term for Bayesian approaches. We also present the formulation of VAR models in a multivariate regression problem, which simplifies implementation of the package. In Section 3, we describe the common interface function and the four shrinkage estimation methods for VAR models included in the package, which are the multivariate ridge regression, a nonparametric shrinkage method, a full Bayesian approach using a noninformative prior, and a semiparametric Bayesian approach using an informative prior. We clearly present closed form expressions for the shrinkage estimators inferred by the shrinkage methods, so that we can indicate the role of the shrinkage intensity parameters in each method. In Section 4, we present numerical experiments using benchmark data and simulated data for comparing performances of the shrinkage methods. Discussion and Conclusions are provided in Section 5.

2. Models

In the VAR model in (1), we assume that the noise vectors are independent and identically distributed from a scale mixture of multivariate normal distributions, including multivariate normal distributions, multivariate Student t-distributions (Ni and Sun 2005), and multivariate Laplace distributions (MLD) (Eltoft, Kim, and Lee 2006). By incorporating scale mixture distributions in the model assumption, we can control outliers and produce robust estimates (West 1984). In specific, we consider that a noise vector can be expressed as a product

$$\boldsymbol{\epsilon}_t = \mathbf{z}_t q_t^{-1/2}, \tag{2}$$

where $\mathbf{z}_t \sim \mathrm{N}_D(\mathbf{0}, \boldsymbol{\Sigma})$ is a $D \times 1$ vector having a multivariate normal distribution with the mean vector of zeros and the covariance matrix of $\boldsymbol{\Sigma}$, and $q_t \sim \mathrm{Gamma}(\nu/2, \nu/2)$ is a random variable having a gamma distribution with shape $\alpha = \nu/2$ and rate $\beta = \nu/2$. The following statements describe characteristics of the distributions in more detail.

- If $1 < \nu < \infty$, then $\boldsymbol{\epsilon}_t$ has a multivariate Student t-distribution with the degree of freedom ν. The probability density function (p.d.f.) for $\boldsymbol{\epsilon}_t$ can be expressed by

$$f(\boldsymbol{\epsilon}_t) = g\left(\left\|\mathbf{V}^{-1/2}\boldsymbol{\epsilon}_t\right\|^2\right)|\mathbf{V}|^{-1/2}, \tag{3}$$

  with

$$g(x) \propto \left(1 + \frac{x}{\nu}\right)^{-(\nu+D)/2}.$$

Define

$$h(x) = -2\frac{\mathrm{d}}{\mathrm{d}x}\log g(x) = \frac{\nu + D}{\nu + x}. \tag{4}$$

- If $\nu \to \infty$, the distribution for $\boldsymbol{\epsilon}_t$ converges to the multivariate normal distribution, $\mathrm{N}_D(\mathbf{0}, \boldsymbol{\Sigma})$. The p.d.f. for $\boldsymbol{\epsilon}_t$ can be expressed by (3) with $g(x) \propto \exp(-x/2)$ and $h(x) = 1$.

- If $\nu = 1$, the distribution for $\boldsymbol{\epsilon}_t$ converges to the multivariate Cauchy distribution. The p.d.f. for $\boldsymbol{\epsilon}_t$ can be expressed by (3) with $g(x) \propto (1 + x)^{-(1+D)/2}$ and $h(x) = (1 + D)/(1 + x)$.

- If $\nu \to 0$, the distribution cannot be obtained in a simple closed form, but the p.d.f. for $\boldsymbol{\epsilon}_t$ can still be written as $g(x) \propto x^{-D/2}$ and $h(x) = D/x$.

  In this case, the expression for q_t during the iteration process in the semiparametric Bayesian approach (Lee *et al.* 2016), is equivalent to the IRLS procedure for L1 norm minimization (Chartrand and Yin 2008); see Section 3.4 for more detail. In addition, we note that the p.d.f. $f^{MLD}(\boldsymbol{\epsilon}_t)$ of the multivariate Laplace distribution (MLD) can be expressed as in (3) with

$$g^{MLD}(x) \propto \frac{K_{(D-2)/2}(\sqrt{2x/\beta})}{(\sqrt{\beta x/2})^{(D-2)/2}}$$

  for some $\beta > 0$ (Eltoft *et al.* 2006). Since $K_\alpha(z)$, the modified Bessel function of the second kind, can be approximated by $K_\alpha(z) \approx \Gamma(\alpha)2^{\alpha-1}z^{-\alpha}$ for small values of $0 < z \leq \sqrt{\alpha + 1}$, the p.d.f. of MLD can be approximated by $g^{MLD}(x) \propto x^{-D/2+1}$ for small values of $0 < x \leq \sqrt{D\beta/4}$. Therefore, we consider that the multivariate Student t-distribution is approximately close to the MLD when $\nu \to 0$.

We can rewrite the model equation (1) in a general form as

$$\mathbf{y}_t = \boldsymbol{\Psi}^\top \mathbf{x}_t + \boldsymbol{\epsilon}_t, \tag{5}$$

which considers the following cases:

(a) In the case that the constant vector $\mathbf{c}$ should be estimated, $\boldsymbol{\Psi} = (\mathbf{c}, \mathbf{A}_1, \mathbf{A}_2, \cdots, \mathbf{A}_p)^\top$ is a $(Dp+1) \times D$ matrix of coefficients and $\mathbf{x}_t = (1, \mathbf{y}_{t-1}^\top, \mathbf{y}_{t-2}^\top, \cdots, \mathbf{y}_{t-p}^\top)^\top$ is a $(Dp+1) \times 1$ vector of lagged variables.

(b) In the case that $\mathbf{c} = \mathbf{0}$, $\boldsymbol{\Psi} = (\mathbf{A}_1, \mathbf{A}_2, \cdots, \mathbf{A}_p)^\top$ is a $Dp \times D$ matrix of coefficients and $\mathbf{x}_t = (\mathbf{y}_{t-1}^\top, \mathbf{y}_{t-2}^\top, \cdots, \mathbf{y}_{t-p}^\top)^\top$ is a $Dp \times 1$ vector of lagged variables.

(c) In the case that $\mathbf{c} = (\mathbf{I}_D - \sum_{i=1}^p \mathbf{A}_i)\boldsymbol{\mu}_\mathbf{y}$, then the equation (1) is rewritten by

$$\mathbf{y}_t^\mathrm{c} = \boldsymbol{\Psi}^\top \mathbf{x}_t^\mathrm{c} + \boldsymbol{\epsilon}_t,$$

  where $\mathbf{y}_t^\mathrm{c} = \mathbf{y}_t - \boldsymbol{\mu}_\mathbf{y}$ and $\mathbf{x}_t^\mathrm{c} = (\mathbf{y}_{t-1}^{\mathrm{c}\top}, \mathbf{y}_{t-2}^{\mathrm{c}\top}, \cdots, \mathbf{y}_{t-p}^{\mathrm{c}\top})^\top$. In this case, the sample mean vector $\bar{\mathbf{y}} = T^{-1}\sum_{t=1}^T \mathbf{y}_t$ is used for an estimate of the mean vector $\boldsymbol{\mu}_\mathbf{y}$. For notational simplicity, we denote $\mathbf{y}_t^\mathrm{c}$ by $\mathbf{y}_t$ and $\mathbf{x}_t^\mathrm{c}$ by $\mathbf{x}_t$ in the following sections.

For estimation of VAR parameters from the observed time series data $\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_T$, we define the data matrices based on (5) as

$$\mathbf{Y} = \begin{pmatrix} \mathbf{y}_{p+1}^\top \\ \mathbf{y}_{p+2}^\top \\ \vdots \\ \mathbf{y}_T^\top \end{pmatrix} \in \mathbb{R}^{(T-p)\times D}, \quad \mathbf{X} = \begin{pmatrix} \mathbf{x}_{p+1}^\top \\ \mathbf{x}_{p+2}^\top \\ \vdots \\ \mathbf{x}_T^\top \end{pmatrix}. \tag{6}$$

Then, we can derive a matrix equation

$$\mathbf{Y} = \mathbf{X}\mathbf{\Psi} + \mathbf{E}, \tag{7}$$

with $\mathbf{E} = (\boldsymbol{\epsilon}_{p+1}, \ldots, \boldsymbol{\epsilon}_T)^\top$.

3. Shrinkage Estimation Methods

In this section, we will describe the shrinkage estimation methods implemented in this package. From the observed time series data $\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_T$, the methods estimate the VAR coefficient matrix $\mathbf{\Psi}$ alone or both of the $\mathbf{\Psi}$ and $\mathbf{\Sigma}$ for the equation (5), or, equivalently (7). We provide a common R function interface `VARshrink()` for running the estimation methods, which is defined by

```
VARshrink <- function(Y, p = 1, const_type = c('const', 'none', 'mean'),
                  method = c('ridge', 'ns', 'fbayes', 'sbayes'),
                  lambda = NULL, lambda_var = NULL, dof = Inf, ...)
```

The input arguments and the output value are described as follows.

Inputs:

- `Y`: $T \times D$ time series data
- `p`: lag order
- `const_type`
  1) `'const'` - Estimate the constant vector, $\mathbf{c}$, as described in item (a) below Eq. (5).
  2) `'none'` - Estimate without the constant vector, i.e., $\mathbf{c} = \mathbf{0}$, as described in item (b) below Eq. (5).
  3) `'mean'` - Use the sample mean, $\bar{\mathbf{y}}$, to center the time series data, and estimate the constant vector, as described in item (c) below Eq. (5).
- `method`
  1) `'ridge'` - multivariate ridge regression
  2) `'ns'` - nonparametric shrinkage method
  3) `'fbayes'` - full Bayesian approach using a noninformative prior

4) `'sbayes'` - semiparametric Bayesian approach using an informative prior

- `lambda, lambda_var`: shrinkage intensity parameters. See descriptions in the following subsections for the use of shrinkage intensity parameters in each method.
- `dof`: degree of freedom of multivariate t distribution for noise. If `dof=Inf`, it means multivariate normal distribution.

Output: an object of class '`VARshrinkest`'. See Appendix A.

3.1. Multivariate Ridge Regression

The ridge regression method is a kind of penalized least squares (PLS) methods, which produces a biased estimate of the VAR coefficient (Hoerl and Kennard 1970). Formally speaking, the ridge regression estimator of $\mathbf{\Psi}$ can be obtained by minimizing the penalized sum of squared prediction errors (SSPE) as

$$\widehat{\mathbf{\Psi}}^{\mathrm{R}}(\lambda) = \arg\min_{\mathbf{\Psi}} \ \frac{1}{T-p} \left\| \mathbf{Y} - \mathbf{X}\mathbf{\Psi} \right\|_F^2 + \lambda \left\| \mathbf{\Psi} \right\|_F^2, \tag{8}$$

where $\|\mathbf{A}\|_F = \sqrt{\sum_i \sum_j a_{ij}^2}$ is the Frobenius norm of a matrix $\mathbf{A}$, and $\lambda \geq 0$ is called the regularization parameter or the shrinkage parameter. The ridge regression estimator $\widehat{\mathbf{\Psi}}^{\mathrm{R}}(\lambda)$ can be expressed in the closed form

$$\widehat{\mathbf{\Psi}}^{\mathrm{R}}(\lambda) = \left(\mathbf{X}^\top \mathbf{X} + (T-p)\lambda \mathbf{I} \right)^{-1} \mathbf{X}^\top \mathbf{Y}, \qquad \lambda \geq 0. \tag{9}$$

The shrinkage parameter λ for the ridge regression can be automatically determined by using the generalized cross-validation (GCV) (Golub, Heath, and Wahba 1979). The GCV selects the value of λ where the GCV score given below is minimized:

$$GCV(\lambda) = \frac{1}{T-p} \left\| (\mathbf{I} - \mathbf{H}(\lambda)\mathbf{Y}) \right\|_{\mathrm{F}}^2 \Bigg/ \left[\frac{1}{T-p} Trace(\mathbf{I} - \mathbf{H}(\lambda)) \right]^2, \tag{10}$$

where $\mathbf{H}(\lambda) = \mathbf{X}^\top \left(\mathbf{X}^\top \mathbf{X} + (T-p)\lambda \mathbf{I} \right)^{-1} \mathbf{X}^\top$.

In this package, the interface to the shrinkage estimation methods is provided by the function `VARshrink(method = 'ridge', ...)` or the internal function `lm_multiv_ridge()`. If the input argument `lambda` is set at a fixed λ value or a vector of candidate λ values, then corresponding GCV score is computed automatically for each λ value, and the VAR coefficients with smallest GCV score is selected. If `lambda=NULL`, then the default value of `c(0.01, 0.05, 0.1, 0.5, 1, 5, 10, 50, 100)` is used.

For example, simulated time series data of length $T = 100$ were generated based on a multivariate normal distribution for noise and a VAR model with $p = 1$, $D = 2$, $\mathbf{A}_1 = 0.5\mathbf{I}_2$, $\mathbf{c} = \mathbf{0}$, and $\mathbf{\Sigma} = 0.1^2\mathbf{I}_2$ as follows:

```
R> set.seed(100)
R> myCoef = list(A = list(matrix(c(0.5, 0, 0, 0.5), 2, 2)), c = c(0.2, 0.7))
R> myModel = VARparam(Coef = myCoef, Sigma = diag(0.1^2, 2), dof = Inf)
R> Y = simVARmodel(numT = 100, model = myModel, burnin = 10)
```

Then, the multivariate ridge regression can be run for VAR models as follows:

```
R> EstimRidge = VARshrink(Y, p = 1, const = 'const', method = 'ridge',
+                         lambda = NULL)
R> EstimRidge

VARshrink Estimation Results:
=============================
Call:
VARshrink(Y = Y, p = 1, const_type = "const", method = "ridge",
    lambda = NULL)

Estimated coefficients:
    Y1.lag1  Y2.lag1    const
Y1 0.261073 0.102723 0.156051
Y2 0.121217 0.527687 0.605825

$Sigma for noise:
         Y1       Y2
Y1 0.008358 0.000913
Y2 0.000913 0.011597

$dof for noise: Inf (estimated: FALSE )
$lambda: 0.01 0.05 0.1 0.5 1 5 10 50 100 500 (estimated: TRUE )
$GCV: 0.02006 0.02194 0.02382 0.06302 0.14841 0.83128 1.25956 1.90499 2.01992
2.11962
```

3.2. Nonparametric Shrinkage (NS)

The nonparametric shrinkage (NS) estimation method for VAR models, proposed by Opgen-Rhein and Strimmer (2007b), produces an estimate of $\mathbf{\Psi}$ based on a James-Stein type shrinkage of sample covariance matrices (Opgen-Rhein and Strimmer 2007a; Schäfer and Strimmer 2005).

We will briefly describe the NS method. In the NS method, the data matrices $\mathbf{X}$ and $\mathbf{Y}$ are centered so that each column has the mean of zero. Let $\mathbf{Z} = [\mathbf{X}, \mathbf{Y}]$ be a combined data matrix. The sample covariance matrix of $\mathbf{Z}$ is a partitioned matrix

$$\mathbf{S}_{\mathrm{ZZ}} = \frac{1}{T-p-1}\mathbf{Z}^\top\mathbf{Z} = \begin{bmatrix} \mathbf{S}_{\mathrm{XX}} & \mathbf{S}_{\mathrm{XY}} \\ \mathbf{S}_{\mathrm{XY}}^\top & \mathbf{S}_{\mathrm{YY}} \end{bmatrix}, \tag{11}$$

where $\mathbf{S}_{\mathrm{XX}} = (T-p-1)^{-1}\mathbf{X}^\top\mathbf{X}$, $\mathbf{S}_{\mathrm{XY}} = (T-p-1)^{-1}\mathbf{X}^\top\mathbf{Y}$, and $\mathbf{S}_{\mathrm{YY}} = (T-p-1)^{-1}\mathbf{Y}^\top\mathbf{Y}$. The matrix $\mathbf{S}_{\mathrm{ZZ}}$ can be decomposed into variances and correlations, i.e.,

$$\mathbf{S}_{\mathrm{ZZ}} = \mathbf{D}_{\mathrm{Z}}^{1/2}\mathbf{R}_{\mathrm{ZZ}}\mathbf{D}_{\mathrm{Z}}^{1/2}, \tag{12}$$

where $\mathbf{R}_{\mathrm{ZZ}}$ is the sample correlation matrix and $\mathbf{D}_{\mathrm{Z}} = \mathrm{diag}(s_{11}, s_{22}, \ldots, s_{Dp+D,Dp+D})$ is a diagonal matrix with diagonal elements of sample variances. Shrinkage estimates of the

correlation matrix and the variances can be written as

$$\widehat{\mathbf{R}}_{\mathrm{ZZ}} = (1-\lambda)\mathbf{R}_{\mathrm{ZZ}} + \lambda\mathbf{I} \quad \text{and} \quad \widehat{\mathbf{D}}_{\mathrm{Z}} = \mathrm{diag}(\hat{s}_{11}, \hat{s}_{22}, \ldots, \hat{s}_{Dp+D,Dp+D}) \tag{13}$$

with

$$\hat{s}_{ii} = (1-\lambda_v)s_{ii} + \lambda_v s_{\mathrm{med}}, \quad i = 1, 2, \ldots, Dp + Dp, \tag{14}$$

where s_{med} is a median of all the sample variances s_{ii}, and $0 \leq \lambda, \lambda_v \leq 1$ are shrinkage parameters. The estimated covariance matrix can be computed by

$$\widehat{\mathbf{S}}_{\mathrm{ZZ}}(\lambda, \lambda_v) = \widehat{\mathbf{D}}_{\mathrm{Z}}^{1/2}\widehat{\mathbf{R}}_{\mathrm{ZZ}}\widehat{\mathbf{D}}_{\mathrm{Z}}^{1/2}. \tag{15}$$

The values of the shrinkage parameters λ and λ_v are determined by the James-Stein type shrinkage method, which we call the NS method, described in (Opgen-Rhein and Strimmer 2007a; Schäfer and Strimmer 2005).

Note that the ordinary least squares estimate of $\mathbf{\Psi}$ can be written as $\widehat{\mathbf{\Psi}}^{\mathrm{OLS}} = \mathbf{S}_{\mathrm{XX}}^{-1}\mathbf{S}_{\mathrm{XY}}$. Similarly, the NS estimate of $\mathbf{\Psi}$ can be written by

$$\widehat{\mathbf{\Psi}}^{\mathrm{N}}(\lambda, \lambda_v) = \widehat{\mathbf{S}}_{\mathrm{XX}}^{-1}\widehat{\mathbf{S}}_{\mathrm{XY}}, \qquad 0 \leq \lambda, \lambda_v \leq 1. \tag{16}$$

where $\widehat{\mathbf{S}}_{\mathrm{XX}}$ and $\widehat{\mathbf{S}}_{\mathrm{XY}}$ are the blocks of the estimated covariance matrix, $\widehat{\mathbf{S}}_{\mathrm{ZZ}}(\lambda, \lambda_v)$, i.e.,

$$\widehat{\mathbf{S}}_{\mathrm{ZZ}}(\lambda, \lambda_v) = \begin{bmatrix} \widehat{\mathbf{S}}_{\mathrm{XX}} & \widehat{\mathbf{S}}_{\mathrm{XY}} \\ \widehat{\mathbf{S}}_{\mathrm{XY}}^{\top} & \widehat{\mathbf{S}}_{\mathrm{YY}} \end{bmatrix}. \tag{17}$$

In this package **VARshrink**, the function `VARshrink(method = 'ns', ...)` provides an interface to the NS method. In specific, the package **corpcor** (Schäfer *et al.* 2017) includes the R function `cov.shrink()`, which can determine λ and λ_v and estimate the covariance matrix $\widehat{\mathbf{S}}_{\mathrm{ZZ}}(\lambda, \lambda_v)$. The function `VARshrink()` in the **VARshrink** package infers the NS estimates of VAR coefficients, $\widehat{\mathbf{\Psi}}^{\mathrm{N}}(\lambda, \lambda_v)$, by using the covariance matrix $\widehat{\mathbf{S}}_{\mathrm{ZZ}}(\lambda, \lambda_v)$. If the input arguments `lambda` and `lambda_var` are set as `lambda = NULL` and `lambda_var = NULL`, then λ and λ_v are determined automatically. For example,

```
R> EstimNS = VARshrink(Y, p = 1, const = 'const', method = 'ns',
+                       lambda = NULL, lambda_var = NULL)
R> EstimNS

VARshrink Estimation Results:
==============================
Call:
VARshrink(Y = Y, p = 1, const_type = "const", method = "ns",
    lambda = NULL, lambda_var = NULL)

Estimated coefficients:
    Y1.lag1   Y2.lag1    const
Y1 0.387697 -0.012579 0.267390
Y2 0.015823  0.195101 1.120196
```

```
$Sigma for noise:
          Y1       Y2
Y1 0.007826 0.000770
Y2 0.000770 0.010463

$dof for noise: Inf (estimated: FALSE )
$lambda: 0.182356 (estimated: TRUE )
$lambda_var: 1 (estimated: TRUE )
```

3.3. Full Bayesian Shrinkage

Ni and Sun (2005) and Sun and Ni (2004) studied Bayesian estimation methods using noninformative priors for VAR models, where they used Markov Chain Monte Carlo (MCMC) methods for estimating coefficient matrices, noise covariance matrices, and other hyperparameters in Bayesian VAR models. In Ni and Sun (2005), various Bayesian estimators were compared using several types of loss functions, noninformative priors, and multivariate normal and Student t-distributions. Among them, Bayesian estimators using a certain type of noninformative priors showed higher accuracies than the other Bayesian estimators in simulated experiments. The noninformative prior for coefficient matrices was called the shrinkage prior, and the prior for noise covariance matrix was called the reference prior.

In the package **VARshrink**, we implemented Bayesian estimators using the shrinkage prior and the reference prior, which are the noninformative priors that yielded the highest accuracies in the simulation results in Ni and Sun (2005). As a Bayesian estimator of VAR parameters, the minimizer of the posterior expectation of quadratic loss is computed, which is the mean of the posterior distribution. Additionally, the minimizer of the posterior expectation of LINEX loss is also computed (Ni and Sun 2005; Zellner 1986). In this section, we will explain the model assumptions in more detail.

First, the noise vectors are independent and identically distributed from multivariate Student t-distributions with the degree of freedom ν, i.e., $\boldsymbol{\epsilon}_t \sim t_\nu(\mathbf{0}, \boldsymbol{\Sigma})$. It can be described using a hierarchical structure of the t-distribution as in (2), by introducing latent variables $q_{p+1}, \ldots, q_T$, i.e.,

$$\begin{aligned} (\boldsymbol{\epsilon}_t | q_t) &\sim \mathrm{N}_D(\mathbf{0}, q_t^{-1}\boldsymbol{\Sigma}), \\ q_t &\sim \mathrm{Gamma}(\nu/2, \nu/2), \end{aligned} \tag{18}$$

where $\mathrm{Gamma}(\alpha, \beta)$ is a gamma distribution with shape α and rate β.

Second, we denote the vectorization of a matrix $\boldsymbol{\Psi} \in \mathbb{R}^{(J/D)\times D}$ by $\boldsymbol{\psi} = \mathrm{vec}(\boldsymbol{\Psi}) \in \mathbb{R}^J$. Here the length J is the number of elements in the VAR coefficient matrix $\boldsymbol{\Psi}$, i.e., $J = D^2p$ or $J = D(Dp+1)$. The shrinkage prior for $\boldsymbol{\psi} \in \mathbb{R}^J$ can be written by $\pi_{\mathrm{S}}(\boldsymbol{\psi}) \propto \|\boldsymbol{\psi}\|^{-(J-2)}$, where J is either $J = D^2p$ or $J = D(Dp+1)$. By introducing a latent variable $\lambda > 0$, the shrinkage prior can also be expressed as a scale mixture of multivariate normals as[1]

$$\begin{aligned} (\boldsymbol{\psi} | \lambda) &\sim \mathrm{N}_J(\mathbf{0}, \lambda^{-1}\mathbf{I}_J), \\ \pi(\lambda) &\propto 1. \end{aligned} \tag{19}$$

[1] Ni and Sun (2005) used the letter δ to denote $\delta = \lambda^{-1}$.

Third, the reference prior for $\mathbf{\Sigma}$ can be written as

$$\pi_{\mathrm{R}}(\mathbf{\Sigma}) \propto |\mathbf{\Sigma}|^{-1} \prod_{1 \le i \le j \le D} (\lambda_i - \lambda_j)^{-1}, \tag{20}$$

where $\lambda_1 > \lambda_2 > \cdots > \lambda_D$ are eigenvalues of $\mathbf{\Sigma}$.

Note that no hyperparameters are involved in the shrinkage prior and the reference prior, since they are noninformative priors. The Gibbs MCMC method makes use of the hierarchical structure described in the above Bayesian VAR models. The Gibbs MCMC samples the parameters $(\boldsymbol{\psi}, \lambda, \mathbf{\Sigma}, \mathbf{Q}, \nu)$ with $\mathbf{Q} = \mathrm{diag}(q_{p+1}, \ldots, q_T)$ from conditional posterior distributions (Ni and Sun 2005). We remark that the mean of the conditional posterior distribution, $\pi(\boldsymbol{\psi}|\lambda, \mathbf{\Sigma}, \mathbf{Q}, \nu; \mathbf{Y})$ of $\boldsymbol{\psi}$ can be written by

$$\widehat{\boldsymbol{\psi}}^{F}(\lambda) = \left[\left(\mathbf{\Sigma}^{-1} \otimes \left(\mathbf{X}^{\top}\mathbf{Q}\mathbf{X}\right)\right) + \lambda \mathbf{I}_J\right]^{-1} \mathrm{vec}\left(\mathbf{X}^{\top}\mathbf{Q}\mathbf{Y}\mathbf{\Sigma}^{-1}\right), \qquad \lambda > 0. \tag{21}$$

Note that if $\mathbf{\Sigma} = \mathbf{I}_D$ and $q_{p+1} = \cdots = q_T = 1$, then the representation is same to the ridge regression estimator. That is, Bayesian shrinkage estimators have more flexible and abundant expressions, even though more computational effort is required to estimate more parameters.

In the package **VARshrink**, the function `VARshrink(method = 'fbayes', ...)` plays the role as an interface to the full Bayesian shrinkage method. The internal function `lm_full_Bayes_SR()` performs the full Bayesian shrinkage estimation using the shrinkage prior and the reference prior. The shrinkage parameter λ cannot be set at a fixed value, i.e., the arguments `lambda` and `lambda_var` do not work here. Instead, we can estimate the mean of the posterior distribution, $\hat{\lambda}^{-1} = \mathbb{E}\left[\lambda^{-1}|\mathbf{Y}\right]$ during the MCMC process (Ni and Sun 2005). There are several arguments to be specified as follows.

- `dof`: If `dof = Inf`, then we apply multivarate normal distribution, and do not estimate $\mathbf{Q}$. If `dof` is a finite value, then we apply the multivariate t-distribution with a fixed degree of freedom $\nu =$ `dof`, and estimate $\mathbf{Q}$. If `dof = NULL`, we estimate both the degree of freedom ν and $\mathbf{Q}$. In this case, the package **ars** is required for sampling the parameter ν from its conditional posterior distribution. The default value is `dof = Inf`.

- `burnincycle`, `mcmccycle`: The Gibbs MCMC method samples a series of parameter values of length `burnincycle + mcmccycle`. `burnincycle` is the number of sampled parameter values to be discarded in the begining, and `mcmccycle` is the number to be attained and used for computing the parameter estimates. By default, we set `burnincycle = 1000` and `mcmccycle = 2000`.

For example, we can call the full Bayesian shrinkage method with a fixed $\nu = 6$ as follows.

```
R> EstimFB1 = VARshrink(Y, p = 1, const = 'const', method = 'fbayes', dof = 6,
+                       burnincycle = 1000, mcmccycle = 2000)
R> EstimFB1

VARshrink Estimation Results:
=============================
Call:
VARshrink(Y = Y, p = 1, const_type = "const", method = "fbayes",
```

```
    dof = 6, burnincycle = 1000, mcmccycle = 2000)

Estimated coefficients:
    Y1.lag1   Y2.lag1    const
Y1 0.441881 -0.030132 0.268776
Y2 0.043424  0.255240 1.022728

$Sigma for noise:
         Y1      Y2
Y1 0.006308 0.00033
Y2 0.000330 0.00840

$dof for noise: 6 (estimated: FALSE )
$lambda: 1.334666 (estimated: TRUE )
```

Instead, we can estimate ν by the argument `dof = NULL` after installing and loading the package **ars** by `library(ars)` as follows:

```
R> EstimFB2 = VARshrink(Y, p = 1, const = 'const', method = 'fbayes', dof = NULL,
+                       burnincycle = 1000, mcmccycle = 2000)
R> EstimFB2

VARshrink Estimation Results:
============================
Call:
VARshrink(Y = Y, p = 1, const_type = "const", method = "fbayes",
    dof = NULL, burnincycle = 1000, mcmccycle = 2000)

Estimated coefficients:
    Y1.lag1   Y2.lag1    const
Y1 0.444540 -0.023891 0.259280
Y2 0.038404  0.257215 1.022204

$Sigma for noise:
         Y1       Y2
Y1 0.006491 0.000404
Y2 0.000404 0.009076

$dof for noise: 8.263929 (estimated: TRUE )
$lambda: 1.407535 (estimated: TRUE )
```

3.4. Semiparametric Bayesian Shrinkage

Whereas full Bayesian shrinkage methods estimate all the hyperparameters including latent variables via MCMC methods, semiparametric Bayesian methods estimate some of the hyperparameters by a certain nonparametric method and estimate the rest by a Bayesian framework. The semiparametric approach is advantageous especially when the dimensionality of the model is so high that standard MCMC methods are not computationally tractable.

Lee *et al.* (2016) assumed scale mixtures of multivariate normal distributions for noise vectors as in (2) and (18), i.e.,

$$\begin{aligned}(\boldsymbol{\epsilon}_t|q_t) &\sim \mathrm{N}_D(\mathbf{0}, q_t^{-1}\boldsymbol{\Sigma}),\\ q_t &\sim \mathrm{Gamma}(\nu/2, \nu/2).\end{aligned} \tag{22}$$

The prior distribution for the model coefficients, $\boldsymbol{\psi} = \mathrm{vec}(\boldsymbol{\Psi})$, was set as multivariate normal distributions, similarly to (19). Here, we can choose either the conjugate prior (CJ) and non-conjugate (NCJ) prior as

(i) the conjugate prior for $\boldsymbol{\psi} \in \mathbb{R}^J$ can be expressed by

$$(\boldsymbol{\psi}|\lambda, \boldsymbol{\Sigma}) \sim \mathrm{N}_J\left(\mathbf{0}, \frac{1-\lambda}{(T-p-1)\lambda}\boldsymbol{\Sigma} \otimes \mathbf{I}\right), \qquad 0 < \lambda < 1, \tag{23}$$

and

(ii) the non-conjugate prior for $\boldsymbol{\psi} \in \mathbb{R}^J$ can be expressed by

$$(\boldsymbol{\psi}|\lambda) \sim \mathrm{N}_J\left(\mathbf{0}, \frac{1-\lambda}{(T-p-1)\lambda}\mathbf{I}_J\right), \qquad 0 < \lambda < 1. \tag{24}$$

The noise covariance matrix $\boldsymbol{\Sigma}$ is included in the conjugate prior distribution whereas it is not included in the non-conjugate prior distribution. The non-conjugate prior is quite similar to the full Bayesian formulation presented in (19). However, the main difference is that, in the semiparametric Bayesian approach, the shrinkage parameter λ should be estimated explicitly via a nonparametric method, but in the full Bayes approach, it is a latent variable which should be sampled and estimated implicitly via an MCMC method.

The prior distribution for $\boldsymbol{\Sigma}$ was set as an inverse Wishart distribution as

$$(\boldsymbol{\Sigma}|\mathbf{L}_0, m_0) \sim \mathrm{InvWishart}(\mathbf{L}_0, m_0), \qquad \mathbf{L}_0 \succ \mathbf{0}, m_0 > D-1. \tag{25}$$

where $\mathbf{L}_0 \succ \mathbf{0}$ means that $\mathbf{L}_0$ is positive definite.

Once the shrinkage parameter λ is set at a fixed value, the other parameters, $\boldsymbol{\psi}, \boldsymbol{\Sigma}$, and $\mathbf{Q}$ can be estimated iteratively in a Bayesian framework efficiently. Briefly speaking, we consider to estimate the parameters $\boldsymbol{\psi}$ and $\boldsymbol{\Sigma}$ at the maximum point (i.e., the mode) of the marginal posterior density function $\pi(\boldsymbol{\psi}, \boldsymbol{\Sigma}|\lambda; \mathbf{Y})$. In the case of the non-conjugate prior, the mode, $(\widehat{\boldsymbol{\psi}}^{\mathrm{S}}(\lambda), \widehat{\boldsymbol{\Sigma}}^{\mathrm{S}}(\lambda))$, can be written by

$$\widehat{\boldsymbol{\psi}}^{S}(\lambda) = \left[\left((\widehat{\boldsymbol{\Sigma}}^{\mathrm{S}})^{-1} \otimes \left(\mathbf{X}^\top \widehat{\mathbf{Q}}^{\mathrm{S}} \mathbf{X}\right)\right) + \frac{(T-p-1)\lambda}{1-\lambda}\mathbf{I}_J\right]^{-1} \mathrm{vec}\left(\mathbf{X}^\top \widehat{\mathbf{Q}}^{\mathrm{S}} \mathbf{Y} (\widehat{\boldsymbol{\Sigma}}^{\mathrm{S}})^{-1}\right), \tag{26}$$

and

$$\widehat{\boldsymbol{\Sigma}}^{\mathrm{S}}(\lambda) = \frac{1}{m_0 + T + D + 1}\left(\mathbf{L}_0 + \mathbf{Y}^\top \widehat{\mathbf{Q}} \mathbf{Y} - \mathbf{Y}^\top \widehat{\mathbf{Q}} \mathbf{X} \widehat{\boldsymbol{\Psi}}^{\mathrm{S}}(\lambda)\right), \tag{27}$$

for $0 < \lambda < 1$, where $\widehat{\mathbf{Q}}^{\mathrm{S}} = \mathrm{diag}(\hat{q}_{p+1}, \ldots, \hat{q}_T)$ is a diagonal matrix which can be obtained during the iteration process.[2] The values $\widehat{\mathbf{Q}}^{\mathrm{S}}$ can also be expressed by

$$\hat{q}_t = h\left(\left\|(\widehat{\boldsymbol{\Sigma}}^{\mathrm{S}})^{-1/2}\hat{\boldsymbol{\epsilon}}_t\right\|^2\right), \qquad t = p+1, \ldots, T, \tag{28}$$

[2]We note that the expression in (26) is equivalent to the expression in (21) of the full Bayesian framework.

where $h(x)$ is defined in (4) depending on the noise distribution, and $\hat{\boldsymbol{\epsilon}}_t$ is the residual, $\hat{\boldsymbol{\epsilon}}_t = \mathbf{y}_t - \hat{\boldsymbol{\Psi}}^{\mathrm{S}\top}\mathbf{x}_t$ (Lee *et al.* 2016).

The shrinkage parameter λ is determined via an internal nonparametric method, which is called the parameterized cross validation (PCV). This algorithm can be considered as a modified K-fold cross validation, especially for estimating the shrinkage parameter of VAR models; see, e.g., Lee *et al.* (2016) for more detail.

In addition, the semiparametric Bayesian shrinkage method adopted the idea of shrinking both the correlations and variances from the NS method. As a result, there are two shrinkage parameters λ and λ_v, where $0 \leq \lambda \leq 1$ is used for the shrinkage estimation of the VAR coefficient matrix Ψ and noise covariance matrix $\boldsymbol{\Sigma}$, while $0 \leq \lambda_v \leq 1$ is used for the shrinkage estimation of the variances of the variables $y_{td}, d = 1, \ldots, D$.

In the package **VARshrink**, the function `VARshrink(method = 'sbayes', ...)` can be used to run the semiparametric Bayesian shrinkage method. It acts as an interface to the internal function `lm_semi_Bayes_PCV()`. There are several input arguments to these functions as follows.

- `dof`: If `dof = Inf`, we use the multivariate normal distribution for noise vectors. Otherwise, `dof` can be set as a finite number ν between 0 and ∞ for the multivariate Studnet t-distribution with degree of freedom ν. If `dof = NULL`, then ν is automatically selected by K-fold cross validation, with $K =$ `num_fold`. The default value is `dof = Inf`.
- `lambda, lambda_var`: `lambda = NULL` and `lambda_var = NULL` implies that the shrinkage parameters λ and λ_v are estimated automatically. We consider that $0 \leq \lambda, \lambda_v \leq 1$.
- `prior_type`: Either `prior_type = 'NCJ'` or `prior_type = 'CJ'`, which implies nonconjugate prior or conjugate prior. The default value is 'NCJ'.
- `num_folds`: Number of folds for the parameterized cross validation method for determining λ value. The default value is `num_folds = 5`. It works only when `lambda = NULL` or `dof = NULL`.
- `m0`: The value of the hyperparameter m_0 of the inverse Wishart prior of $\boldsymbol{\Sigma}$. The default value is `m0 =` D. On the other hand, the other hyperparameter $\mathbf{L}_0$ of the inverse Wishart prior is set by $\mathbf{L}_0 = (m_0 + D + 1)\mathbf{I}_D$.

For example, we can run the semiparametric shrinkage method as follows.

```
R> EstimSB1 = VARshrink(Y, p = 1, const = 'const', method = 'sbayes', dof = 6,
+                       lambda = NULL, lambda_var = NULL, prior_type = 'NCJ',
+                       num_folds = 5, m0 = ncol(Y))
R> EstimSB1

VARshrink Estimation Results:
=============================
Call:
VARshrink(Y = Y, p = 1, const_type = "const", method = "sbayes",
    lambda = NULL, lambda_var = NULL, dof = 6, prior_type = "NCJ",
    num_folds = 5, m0 = ncol(Y))
```

```
Estimated coefficients:
    Y1.lag1   Y2.lag1    const
Y1 0.447771 -0.013516 0.248414
Y2 0.046828  0.319148 0.907047

$Sigma for noise:
         Y1       Y2
Y1 0.006252 0.000561
Y2 0.000561 0.008902

$dof for noise: 6 (estimated: FALSE )
$lambda: 0.0007981101 (estimated: TRUE )
$lambda_var: 1 (estimated: TRUE )
```

We can also let the software package to choose the degree of freedom parameter ν by setting `dof = NULL`. In this case, the package uses simply a K-fold cross validation to find an optimal value of ν. However, since the K-fold cross validation procedure is computationally slow, it is not recommended at the moment.

```
R> EstimSB2 = VARshrink(Y, p = 1, const = 'const', method = 'sbayes', dof = NULL,
+                       lambda = NULL, lambda_var = NULL, prior_type = 'NCJ',
+                       num_folds = 5, m0 = ncol(Y))
R> EstimSB2


VARshrink Estimation Results:
============================
Call:
VARshrink(Y = Y, p = 1, const_type = "const", method = "sbayes",
    lambda = NULL, lambda_var = NULL, dof = NULL, prior_type = "NCJ",
    num_folds = 5, m0 = ncol(Y))

Estimated coefficients:
    Y1.lag1   Y2.lag1    const
Y1 0.542009 -0.083560 0.307348
Y2 0.120585  0.276913 0.932485

$Sigma for noise:
         Y1       Y2
Y1 0.004528 0.000480
Y2 0.000480 0.006901

$dof for noise: 0.2 (estimated: TRUE )
$lambda: 0.0007981101 (estimated: TRUE )
$lambda_var: 1 (estimated: TRUE )
```

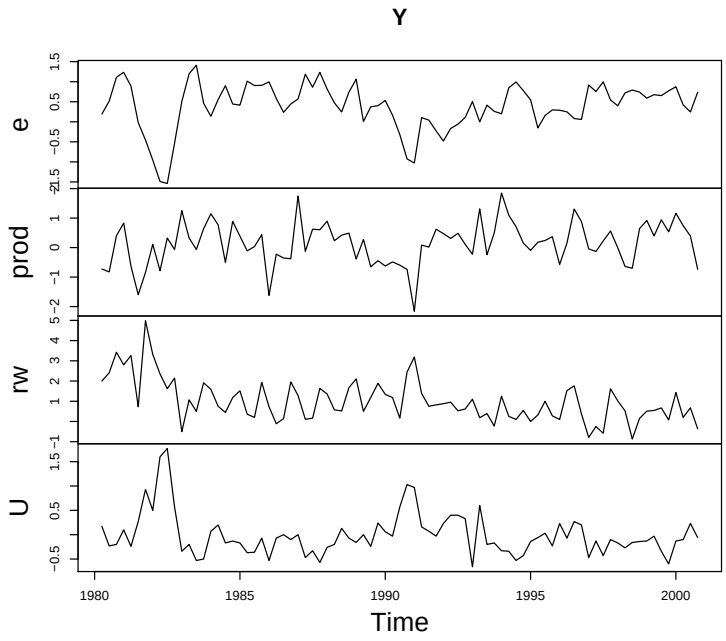

Figure 1: The benchmark data set obtained by differencing the Canada data.

4. Numerical Experiments

In this section, we apply the shrinkage estimation methods in the package **VARshrink** to a benchmark data set and simulated data sets. We also demonstrate tools for comparison of estimated VAR models.

4.1. Benchmark Data

The Canada data set is a benchmark macroeconomic data included in the package **vars**. It contains four time series variables representing employment(`e`), labour productivity(`prod`), real wage(`rw`), and unemployment rate(`U`), with 84 observations. The data were differenced to remove trend, which yielded $T = 83$. Figure 1 shows the obtained data set.

```
R> data(Canada, package = "vars")
R> Y = diff(Canada)
R> plot(Y, cex.lab = 1.3)
R> set.seed(100)
```

The shrinkage methods are run with the option `const = 'const'` since the mean of the data has not been corrected. We set the lag order p by `p` $= 1, 2, 3$ to compare several VAR models. For the semiparametric Bayes method, the degree of freedom ν was automatically selected among $\nu = 2, 5, \infty$ to reduce the computational cost. In addition, we ran the semiparametric Bayes method using the original K-fold cross validation for choosing λ and λ_v.

```
R> EstimRidge10 = VARshrink(Y, p = 1, const = 'const', method = 'ridge')
R> EstimRidge20 = VARshrink(Y, p = 2, const = 'const', method = 'ridge')
```

```
R> EstimRidge30 = VARshrink(Y, p = 3, const = 'const', method = 'ridge')
R> EstimNS10 = VARshrink(Y, p = 1, const = 'const', method = 'ns')
R> EstimNS20 = VARshrink(Y, p = 2, const = 'const', method = 'ns')
R> EstimNS30 = VARshrink(Y, p = 3, const = 'const', method = 'ns')
R> EstimFB10 = VARshrink(Y, p = 1, const = 'const', method = 'fbayes',
+                        dof = NULL, burnincycle = 1000, mcmccycle = 2000)
R> EstimFB20 = VARshrink(Y, p = 2, const = 'const', method = 'fbayes',
+                        dof = NULL, burnincycle = 1000, mcmccycle = 2000)
R> EstimFB30 = VARshrink(Y, p = 3, const = 'const', method = 'fbayes',
+                        dof = NULL, burnincycle = 1000, mcmccycle = 2000)
R> EstimSB10 = VARshrink(Y, p = 1, const = 'const', method = 'sbayes',
+                       dof = c(2,5,Inf), lambda = NULL, lambda_var = NULL,
+                       prior_type = 'NCJ', num_folds = 5, m0 = ncol(Y))
R> EstimSB20 = VARshrink(Y, p = 2, const = 'const', method = 'sbayes',
+                        dof = c(2,5,Inf), lambda = NULL, lambda_var = NULL,
+                        prior_type = 'NCJ', num_folds = 5, m0 = ncol(Y))
R> EstimSB30 = VARshrink(Y, p = 3, const = 'const', method = 'sbayes',
+                        dof = c(2,5,Inf), lambda = NULL, lambda_var = NULL,
+                        prior_type = 'NCJ', num_folds = 5, m0 = ncol(Y))
```

We can compare several models by computing AIC and BIC. The following results indicate that VAR models with the larger lag order value $p = 3$ is better than $p = 1, 2$, and that the NS method produced better VAR coefficients than those of the other methods.

```
> cat("======== Model Comparison ========\n")
> cat("AIC:\n")
> cat("Method:  p = 1    p = 2    p = 3  \n")
> cat("ridge : ", round(AIC(EstimRidge10),1), "  ", round(AIC(EstimRidge20),1),
+     "  ", round(AIC(EstimRidge30),1), "\n")
> cat("ns    : ", round(AIC(EstimNS10),1), "  ", round(AIC(EstimNS20),1),
+     "  ", round(AIC(EstimNS30),1), "\n")
> cat("fbayes: ", round(AIC(EstimFB10),1), "  ", round(AIC(EstimFB20),1),
+     "  ", round(AIC(EstimFB30),1), "\n")
> cat("sbayes: ", round(AIC(EstimSB10),1), "  ", round(AIC(EstimSB20),1),
+     "  ", round(AIC(EstimSB30),1), "\n")
> cat("BIC:\n")
> cat("Method:  p = 1    p = 2    p = 3  \n")
> cat("ridge : ", round(BIC(EstimRidge10),1), "  ", round(BIC(EstimRidge20),1),
+     "  ", round(BIC(EstimRidge30),1), "\n")
> cat("ns    : ", round(BIC(EstimNS10),1), "  ", round(BIC(EstimNS20),1),
+     "  ", round(BIC(EstimNS30),1), "\n")
> cat("fbayes: ", round(BIC(EstimFB10),1), "  ", round(BIC(EstimFB20),1),
+     "  ", round(BIC(EstimFB30),1), "\n")
> cat("sbayes: ", round(BIC(EstimSB10),1), "  ", round(BIC(EstimSB20),1),
+     "  ", round(BIC(EstimSB30),1), "\n")

======== Model Comparison ========
AIC:
```

```
Method:  p = 1    p = 2    p = 3
ridge :  424.3    371.7    356.6
ns    :  425    373.5    348.8
fbayes:  412.4    372.5    358.8
sbayes:  429.4    405    382
BIC:
Method:  p = 1    p = 2    p = 3
ridge :  412.3    350.3    326
ns    :  413.1    352.3    318.4
fbayes:  400.4    351.1    328.2
sbayes:  417.4    383.5    351.4
```

According to the results presented above, the NS method with $p = 3$ yielded the smallest AIC and BIC values. The estimated parameters can be presented as follows for further analysis:

```
R> EstimNS30

VARshrink Estimation Results:
==============================
Call:
VARshrink(Y = Y, p = 3, const_type = "const", method = "ns")

Estimated coefficients:
        e.lag1 prod.lag1   rw.lag1    U.lag1    e.lag2 prod.lag2   rw.lag2    U.lag2
e     0.500899  0.159727 -0.028909 -0.106343 -0.072892  0.084403 -0.051388 -0.006745
prod -0.031393  0.165472  0.022353 -0.388469 -0.155635  0.016392 -0.136856  0.006587
rw   -0.071347 -0.115559  0.173494  0.338034  0.259075 -0.326972  0.082810 -0.060387
U    -0.338903 -0.118026  0.025811  0.036542 -0.034963 -0.056927  0.093335 -0.089206
        e.lag3 prod.lag3   rw.lag3    U.lag3     const
e    -0.037403  0.030356 -0.034558  0.089603  0.299914
prod -0.058647  0.036091 -0.081384  0.255961  0.421743
rw    0.161740 -0.019658  0.092154 -0.045359  0.536371
U     0.066059 -0.041878  0.050852 -0.060372 -0.025126

$Sigma for noise:
             e      prod        rw         U
e     0.148484 -0.030983 -0.034448 -0.078742
prod -0.030983  0.451176  0.055321  0.014014
rw   -0.034448  0.055321  0.775603  0.064139
U    -0.078742  0.014014  0.064139  0.092005

$dof for noise: Inf (estimated: FALSE )
$lambda: 0.156342 (estimated: TRUE )
$lambda_var: 0.1542602 (estimated: TRUE )
```

4.2. Comparative Simulation

For comparison of estimation accuracies of the shrinkage methods, we built a sample R code

for conducting comparative simulations. We generated multivariate time series data of length $T = 20, 40, 80, 160$ from $D = 20$ dimensional VAR models of order $p = 1$. We note that when $T = 20$, the sample size $T - p = 19$ becomes smaller than the dimensionality $D = 20$. We set the VAR coefficient matrix, $\mathbf{A}_1$, to have the diagonal entries of 0.6 and off-diagonal entries of zeros except for D entries randomly selected on the lower triangular part of the matrix. The nonzero off-diagonal entries were uniformly randomly drawn from the interval $[-1, -0.2] \cup [0.2, 1]$. Since the coefficient matrix is lower triangular, the diagonal entries determine its eigenvalues, which determine weak stationarity (Hamilton 1994; Tsay 2005). The noise $\boldsymbol{\epsilon}_t$ was randomly sampled from the multivariate normal distribution whose covariance matrix has diagonal entries of 1 and off-diagonal entries of 0.5. The sum of squared errors (SSEs) of estimated VAR coefficients, $\widehat{\mathbf{A}}_1$, were computed by

$$\sum_{i=1}^{D}\sum_{j=1}^{D}\left(\left(\widehat{\mathbf{A}}_1\right)_{ij} - \left(\mathbf{A}_1\right)_{ij}\right)^2. \tag{29}$$

We repeated the experiments 50 times and drew boxplots to compare the SSEs of the shrinkage methods.

```
R> set.seed(100)
R> methods <- c("ridge", "ns", "fbayes", "sbayes")
R> p <- 1; d <- 20; NumTimePTS <- c(20, 40, 80, 160)
R> const_vector <- c(rep(0.2, 5), rep(0.7, 15))
R> Sig <- diag(0.5, d) + matrix(0.5, d, d)
R> nrep <- 50
R> resu_SSE <- array(0, c(nrep, length(methods), length(NumTimePTS)),
+                    dimnames = list(1:nrep, methods, NumTimePTS))
R> for (idnumT in 1:length(NumTimePTS)) {
+   numT = NumTimePTS[idnumT]
+   cat("======= T: ", numT, "=======\n")
+   for (r in 1:nrep) {
+     Ad <- createVARCoefs_ltriangular(p = p, d = d, diag_val = 0.6,
+             num_nonzero = d, const_vector = const_vector, range_max = 1)
+     Md <- VARparam(Coef = Ad, Sigma = Sig, dof = Inf)
+     Y <- simVARmodel(numT = numT, model = Md, burnin = 20)
+     EstimRG <- VARshrink(Y, p, 'const', method = 'ridge')
+     EstimNS <- VARshrink(Y, p, 'const', method = 'ns')
+     EstimFB <- VARshrink(Y, p, 'const', method = 'fbayes', dof = NULL,
+                          burnincycle = 1000, mcmccycle = 2000)
+     EstimSB <- VARshrink(Y, p, 'const', method = 'sbayes', dof = c(2,5,Inf),
+                          prior_type = 'NCJ', num_folds = 5, m0 = ncol(Y))
+     resu_SSE[r, 1, idnumT] <- calcSSE_VARcoef(EstimRG$varparam, Md)
+     resu_SSE[r, 2, idnumT] <- calcSSE_VARcoef(EstimNS$varparam, Md)
+     resu_SSE[r, 3, idnumT] <- calcSSE_VARcoef(EstimFB$varparam, Md)
+     resu_SSE[r, 4, idnumT] <- calcSSE_VARcoef(EstimSB$varparam, Md)
+   }
+   save(resu_SSE, file = paste0("comparisons_d", d, ".Rdata"))
+ }
```

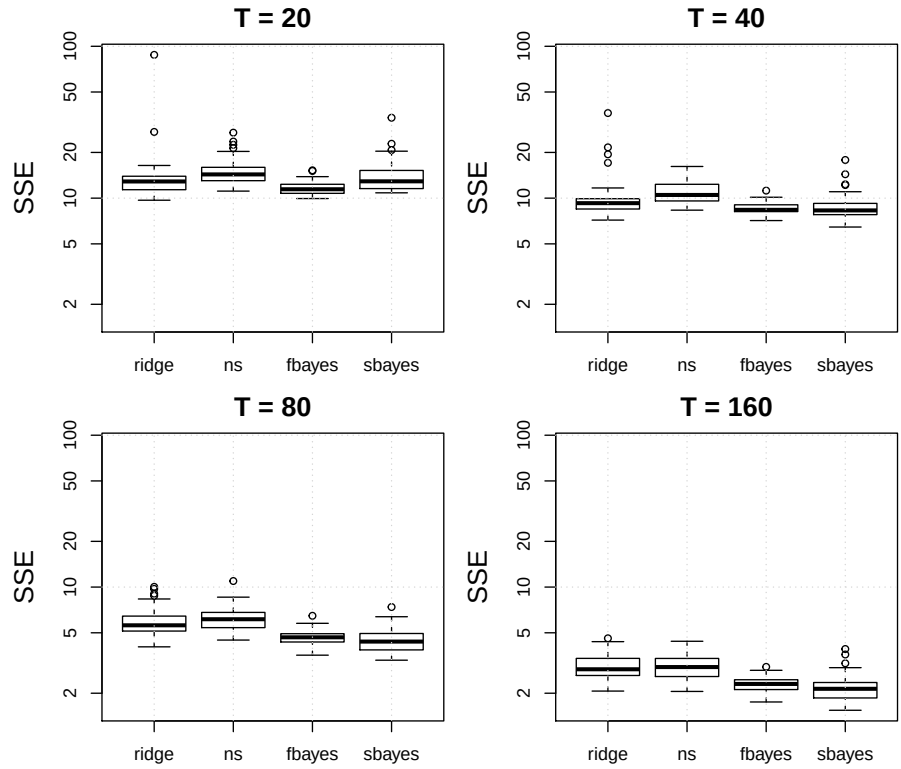

Figure 2: Comparison of sum of squared errors (SSEs) of the VAR coefficients estimated by the four shrinkage methods. The multivariate time series data were randomly generated from VAR models of order $p = 1$ and dimensionality $D = 20$, and the noise term was randomly drawn from a multivariate normal distribution.

```
R> par(mfrow=c(2, 2))
R> for (idnumT in 1:length(NumTimePTS)) {
+    boxplot(resu_SSE[,, idnumT], log = "y", cex.lab = 1.5, cex.main = 1.5,
+          ylab = "SSE", xlab = "", ylim = c(min(resu_SSE), max(resu_SSE)),
+          main = paste("T =", NumTimePTS[idnumT]))
+    grid()
+ }
```

Figure 2 shows the boxplots of the SSEs of the VAR coefficients estimated by the four shrinkage methods. While the full Bayesian approach (`fbayes`) and the semiparametric Bayesian approach (`sbayes`) yielded the best accuracies among the four methods, more simulated experiments are necessary to compare the methods under various conditions.

5. Conclusions

We implemented an R software package **VARshrink** for shrinkage estimation of VAR model parameters. The shrinkage methods included in this package are the multivariate ridge regression (Hoerl and Kennard 1970; Golub *et al.* 1979), a nonparametric shrinkage method (Opgen-Rhein and Strimmer 2007b), a full Bayesian shrinkage method (Ni and Sun 2005),

and a semiparametric Bayesian shrinkage method (Lee *et al.* 2016). An advantage of this package is the integration of the nonparametric, parametric, and semiparametric methods in one software via a common interface function `VARshrink()`, which makes it much simple to use various types of shrinkage methods. Moreover, the package **VARshrink** implemented nonparametric approaches, Bayesian approaches using noninformative priors, and semiparametric Bayesian approaches, which have not been widely implemented in R language in terms of shrinkage estimation for VAR models. We demonstrated that the different types of shrinkage methods could be compared via such model selection criteria as AIC and BIC by benchmark time series data analysis.

We note that shrinkage methods are quite crucial especially for high dimensional VAR models because of the large number of parameters. Bayesian approaches have been developed widely for VAR models in the literature for various purposes, however, the computational time for carrying out MCMC processes can be intractably high for high dimensional VAR models. For this reason, it is important to use nonparametric and semiparametric shrinkage estimation methods together to produce computationally feasible estimates of model parameters. The R package **VARshrink** is the first step to an integrative and general types of software packages for VAR models. Due to such efficiencies and an easy interface, the shrinkage estimation methods are applicable in wide areas such as econometrics, neuroscience, biology, social sciences, and biomedical engineering. Moreover, this package can be extended to include other shrinkage methods, for example, Bayesian shrinkage methods using several types of different prior distributions.

References

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.

Beltrachini L, von Ellenrieder N, Muravchik CH (2013). "Shrinkage Approach for Spatiotemporal EEG Covariance Matrix Estimation." *IEEE Transactions on Signal Processing*, **61**(7), 1797–1808. doi:10.1109/TSP.2013.2238532.

Böhm H, von Sachs R (2009). "Shrinkage estimation in the frequency domain of multivariate time series." *Journal of Multivariate Analysis*, **100**(5), 913 – 935. doi:10.1016/j.jmva.2008.09.009.

Chartrand R, Yin W (2008). "Iteratively reweighted algorithms for compressive sensing." In *2008 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 3869–3872. doi:10.1109/ICASSP.2008.4518498.

Doan T, Litterman R, Sims C (1984). "Forecasting and conditional projection using realistic prior distributions." *Econometric Reviews*, **3**(1), 1–100. doi:10.1080/07474938408800053.

Eltoft T, Kim T, Lee TW (2006). "On the multivariate Laplace distribution." *IEEE Signal Processing Letters*, **13**(5), 300–303. doi:10.1109/LSP.2006.870353.

Fiecas M, Ombao H (2011). "The generalized shrinkage estimator for the analysis of functional connectivity of brain signals." *The Annals of Applied Statistics*, **5**(2A), 1102–1125. doi:10.1214/10-AOAS396.

Friedman J, Hastie T, Tibshirani R, Simon N, Narasimhan B, Qian J (2018). "glmnet: Lasso and Elastic-Net Regularized Generalized Linear Models." R Package Version 2.0-16. Available at https://cran.r-project.org/web/packages/glmnet.

Golub GH, Heath M, Wahba G (1979). "Generalized cross-validation as a method for choosing a good ridge parameter." *Technometrics*, **21**(2), 215–223. doi:10.1080/00401706.1979.10489751.

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

Hoerl AE, Kennard RW (1970). "Ridge regression: Biased estimation for nonorthogonal problems." *Technometrics*, **12**(1), 55–67. doi:10.1080/00401706.1970.10488634.

Hyndman R, Athanasopoulos G, Bergmeir C, Caceres G, Chhay L, O'Hara-Wild M, Petropoulos F, Razbash S, Wang E, Yasmeen F, R Core Team, Ihaka R, Reid D, Shaub D, Tang Y, Zhou Z (2018). "forecast: Forecasting Functions for Time Series and Linear Models." R Package Version 8.4. Available at https://cran.r-project.org/web/packages/forecast.

Koop G, Korobilis D (2010). "Bayesian multivariate time series methods for empirical macroeconomics." *Foundations and Trends in Econometrics*, **3**(4), 267–358. doi:10.1561/0800000013.

Krueger F (2015). "bvarsv: Bayesian Analysis of a Vector Autoregressive Model with Stochastic Volatility and Time-Varying Parameters." R Package Version 1.1. Available at https://cran.r-project.org/web/packages/bvarsv.

Ledoit O, Wolf M (2004). "A well-conditioned estimator for large-dimensional covariance matrices." *Journal of Multivariate Analysis*, **88**(2), 365 – 411. doi:10.1016/S0047-259X(03)00096-4.

Lee N, Choi H, Kim SH (2016). "Bayes shrinkage estimation for high-dimensional VAR models with scale mixture of normal distributions for noise." *Computationl Statistics & Data Analysis*, **101**, 250–276. doi:10.1016/j.csda.2016.03.007.

Litterman RB (1986). "Forecasting with Bayesian Vector Autoregressions: Five Years of Experience." *Journal of Business & Economic Statistics*, **4**(1), 25–38. doi:10.2307/1391384.

Moritz S, Cule E (2018). "ridge: Ridge Regression with Automatic Selection of the Penalty Parameter." R Package Version 2.3. Available at https://cran.r-project.org/web/packages/ridge.

Ni S, Sun D (2005). "Bayesian Estimates for Vector Autoregressive Models." *Journal of Business & Economic Statistics*, **23**(1), 105–117. doi:10.1198/073500104000000622.

Opgen-Rhein R, Strimmer K (2007a). "Accurate ranking of differentially expressed genes by a distribution-free shrinkage approach." *Statistical Applications in Genetics and Molecular Biology*, **6**(1), 9. doi:10.2202/1544-6115.1252.

Opgen-Rhein R, Strimmer K (2007b). "Learning causal networks from systems biology time course data: an effective model selection procedure for the vector autoregressive process." *BMC Bioinformatics*, **8**(2), S3. doi:10.1186/1471-2105-8-S2-S3.

Pfaff B, Stigler M (2018). "vars: VAR Modelling." R Package Version 1.5-3. Available at https://cran.r-project.org/web/packages/vars.

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.

Ripley B, Venables B, Bates DM, Hornik K, Gebhardt A, Firth D (2018). "MASS: Support Functions and Datasets for Venables and Ripley's MASS." R Package Version 7.3-51.1. Available at https://cran.r-project.org/web/packages/MASS.

Schäfer J, Opgen-Rhein R, Zuber V, Ahdesmäki M, Silva APD, Strimmer K (2017). "corpcor: Efficient estimation of covariance and (partial) correlation." R Package Version 1.6.9. Available at http://CRAN.R-project.org/package=corpcor.

Schäfer J, Strimmer K (2005). "A shrinkage approach to large-scale covariance matrix estimation and implications for functional genomics." *Statistical Applications in Genetics and Molecular Biology*, **4**(1), 32. doi:10.2202/1544-6115.1175.

Sun D, Ni S (2004). "Bayesian analysis of vector-autoregressive models with noninformative priors." *Journal of Statistical Planning and Inference*, **121**(2), 291 – 309. doi:10.1016/S0378-3758(03)00116-2.

Tsay RS (2005). *Analysis of Financial Time Series*. Wiley series in probability and statistics, 2nd edition. John Wiley & Sons, Hoboken, NJ. ISBN 978-0-471-69074-0.

Tsay RS, Wood D (2018). "MTS: All-Purpose Toolkit for Analyzing Multivariate Time Series (MTS) and Estimating Multivariate Volatility Models." R Package Version 1.0. Available at https://cran.r-project.org/web/packages/MTS.

West M (1984). "Outlier models and prior distributions in Bayesian linear regression." *Journal of the Royal Statistical Society. Series B (Methodological)*, **46**(3), 431–439.

Zellner A (1986). "Bayesian estimation and prediction using asymmetric loss functions." *Journal of the American Statistical Association*, **81**(394), 446–451. doi:10.1080/01621459.1986.10478289.

A. Class Definitions

In this package, we defined R classes '`VARparam`' and '`varshrinkest`'. These classes are defined as S3 classes with lists of predefined attributes. Table A summarizes the R classes together with their class attributes.

B. Functions

We defined several functions for this package. Description of some of them are given as follows.

- `simVARmodel(numT, model, burnin = 0)`

  Generate simulated multivariate time series data using the given VAR model

  Inputs:

  `numT`: number of observed time points, T.

  `model`: a VAR model, of class '`VARparam`'

  `burnin`: number of initial points which are not included in the final values.

  Outputs:

  a matrix of size $T \times D$

- `testStationarityVARmodel(model)`

  Test stationarity of a VAR model.

  Inputs:

  `model`: a VAR model, of class '`VARparam`'

  Outputs:

  `TRUE` if the given VAR model is weakly stationary (Hamilton 1994; Tsay 2005), `FALSE` otherwise.

- `calcSSE_VARmodel(model1, model2, include_const_vector = FALSE)`

  Calculate sum of squared errors (SSEs) to measure a difference between coefficients of two VAR models

  Inputs:

  `model1, model2`: two VAR models, of class '`VARparam`'

  Outputs:

  The SSE value

Affiliation:

Namgil Lee, Heon-Young Yang
Department of Information Statistics
Kangwon National University
Chuncheon, Gangwon 24341, South Korea
E-mail: namgil.lee@kangwon.ac.kr
and
Sung-Ho Kim
Department of Mathematical Sciences
KAIST
Daejeon 305-701, South Korea
E-mail: sung-ho.kim@kaist.edu

Journal of Statistical Software http://www.jstatsoft.org/
published by the Foundation for Open Access Statistics http://www.foastat.org/
MMMMMM YYYY, Volume VV, Issue II *Submitted:* yyyy-mm-dd
doi:10.18637/jss.v000.i00 *Accepted:* yyyy-mm-dd

Table 1: Summary of the classes in the package

Name	Attribute	Description
'VARparam'	$Coef	A list with components $A and $c. $A is a list of length p representing the coefficient matrices $\mathbf{A}_1, \ldots, \mathbf{A}_p$, and $c is the constant vector $\mathbf{c}$.
	$Sigma	Covariance matrix $\mathbf{\Sigma}$ for the distribution for the noise vector $\boldsymbol{\epsilon}_t$.
	$dof	Degree of freedom for multivariate t-distribution for noise vector $\boldsymbol{\epsilon}_t$.
'VARshrinkest'	$varparam	Estimated VAR parameters, of class 'VARparam'
	$se.varparam	Standard error of $varparam, of class 'VARparam'. Available for method = 'fbayes'.
	$residuals	$(T-p) \times D$ matrix of residuals
	$df.residual	Degree of freedom for residuals
	$p	lag order
	$K	number of rows of $\mathbf{\Psi}$ in (5)
	$obs	number of observations, $T-p$
	$totobs	total number of observations, T
	$call	the call to the VARshrink()
	$const_type	the argument const_type
	$method	the argument method
	$lambda	λ value
	$lambda.estimated	TRUE if $lambda was estimated
	$lambda_var	λ_v value
	$lambda_var.estimated	TRUE if $lambda_var was estimated
	$delta	δ value for method = 'fbayes'
	$se.delta	standard error of $delta value
	$LINEXvarparam	Estimated VAR parameters by the full Bayesian approach using LINEX loss function, of class 'VARparam'. Available for method = 'fbayes'.
	$se.LINEXvarparam	Standard error of $LINEXvarparam, of class 'VARparam'.