DDMSVAR for Ox: a Software for Time Series Modeling with Duration Dependent Markov-Switching Vector Autoregressions

Matteo M. Pelagatti*
Department of Statistics
Università degli Studi di Milano-Bicocca

Abstract

Duration dependent Markov-switching VAR (from now on DDMS-VAR) models are time series models with data generating process consisting in a mixture of two VAR processes, which switches according to a two-state Markov chain with transition probabilities depending on how long the process has been in a state. Interesting applications of this class of models have been carried out in business cycle analysis and in finance. In the present paper we introduce DDMSVAR for Ox, a software written by the author, for the analysis of time series by means of DDMS-VAR models. Possible uses of this software are shown through applications with real data.

1 Introduction and motivation

Since the path-opening paper of Hamilton (1989), many applications of the Markov switching autoregressive model (MS-AR) to the analysis of business cycle have demonstrated its usefulness particularly in dating the cycle in an "objective" way. The basic MS-AR model has, nevertheless, some limitations: (i) it is univariate, (ii) the probability of transition from one state to the other (or to the other ones) is constant. Since business cycles are fluctuations of the aggregate economic activity, which express themselves through the comovements of many macroeconomic variables, point (i) is not a negligible weakness. The

*Addresses: web: http://www.statistica.unimib.it/utenti/p_matteo/ (the presented software can be downloaded from here), e-mail: matteo.pelagatti@unimib.it

multivariate generalization of the MS model was carried out by Krolzig (1997), in his excellent work on the MS-VAR model. As far as point (ii) is concerned, it is reasonable to believe that the probability of exiting a contraction is not the same at the very beginning of this phase as after, say, several months. Some authors, such as Diebold, Rudebusch & Sichel (1993), Watson (1994), have found evidence of duration dependence in the U.S. business cycles, and therefore, as Diebold et al. (1993) point out, the MS model can result miss-specified. In order to face the latter limitation, Durland & McCurdy (1994) introduced the the duration-dependent Markov switching autoregression, designing an alternative filter for the unobservable state variable. In Pelagatti (2002*a*) the duration-dependent switching model is generalized in a multivariate manner, and it is shown how the standard tools of MS-AR model, such as Hamilton's filter and Kim's smoother can be used to model also duration dependence. While Durland & McCurdy (1994) carry out their inference on the model exploiting standard (asymptotic) maximum likelihood theory, here a multi-move Gibbs sampler is implemented to allow Bayesian (but also finite sample likelihood) inference. The advantages of this technique are that (a) it does not relay on asymptotics[1], and in latent variable models, where the difference observations - parameters is low, *asymptopia* can be very far to reach, (b) inference on the latent variables is not conditional on the estimated parameters, but incorporates also the parameters' variability.

The duration-dependent Markov switching VAR model (DDMS-VAR) is defined in section 2 and the relative Gibbs sampler in section 3; section 4 briefly illustrates the features of DDMSVAR for Ox, and an application of the model and of the software is carried out in section 5.

2 The model

The duration-dependent MS-VAR model[2] is defined by

$$\begin{aligned} \mathbf{y}_t &= \boldsymbol{\mu}_0 + \boldsymbol{\mu}_1 S_t + \mathbf{A}_1(\mathbf{y}_{t-1} - \boldsymbol{\mu}_0 - \boldsymbol{\mu}_1 S_{t-1}) + \dots \\ &\quad + \mathbf{A}_p(\mathbf{y}_{t-p} - \boldsymbol{\mu}_0 - \boldsymbol{\mu}_1 S_{t-p}) + \boldsymbol{\varepsilon}_t \end{aligned} \tag{1}$$

[1]Actually MCMC techniques do relay on asymptotic results, but the size of the sample is under control of the researcher and some diagnostics on convergence are available, although this is a field still under development. Here it is meant that the reliability of the inference does not depend on the sample size of the real-world data.

[2]Using Krolzig's terminology, we are defining a duration dependent MSM(2)-VAR, that is, Markov-Switching in Mean VAR with two states.

where $\mathbf{y}_t$ is a vector of observable variables, S_t is a binary (0-1) unobservable random variable following a Markov chain with varying transition probabilities, $\mathbf{A}_1, \ldots, \mathbf{A}_p$ are coefficient matrices of a stable VAR process, and ε_t is a gaussian (vector) white noise with covariance matrix $\mathbf{\Sigma}$.

In order to achieve duration dependence for S_t, a Markov chain is built for the pair (S_t, D_t), where D_t is the duration variable defined as follows:

$$D_t = \begin{cases} D_{t-1} + 1 & \text{if } S_t = S_{t-1},\ D_{t-1} < \tau \\ 1 & \text{if } S_t \neq S_{t-1} \\ D_{t-1} & \text{if } S_t = S_{t-1},\ D_{t-1} = \tau \end{cases} ; \qquad (2)$$

an example of a possible sample path of the Markov chain (S_t, D_t) is shown in table 1. A maximum value, τ, for the duration variable

t	1	2	3	4	5	6	7	8	9	10	11	12
S_t	1	1	1	1	0	0	0	1	0	0	0	0
D_t	3	4	5	6	1	2	3	1	1	2	3	4

Table 1: A possible realization of processes S_t and D_t.

D_t must be fixed, so that the Markov chain (S_t, D_t) is defined on the finite state space

$$\{(0,1), (1,1), (0,2), (1,2), \ldots, (0,\tau), (1,\tau)\},$$

with finite dimensional transition matrix[3]

$$\mathbf{P} = \begin{bmatrix} 0 & p_{0|1}(1) & 0 & p_{0|1}(2) & 0 & p_{0|1}(3) & \ldots & 0 & p_{0|1}(\tau) \\ p_{1|0}(1) & 0 & p_{1|0}(2) & 0 & p_{1|0}(3) & 0 & \ldots & p_{1|0}(\tau) & 0 \\ p_{0|0}(1) & 0 & 0 & 0 & 0 & 0 & \ldots & 0 & 0 \\ 0 & p_{1|1}(1) & 0 & 0 & 0 & 0 & \ldots & 0 & 0 \\ 0 & 0 & p_{0|0}(2) & 0 & 0 & 0 & \ldots & 0 & 0 \\ 0 & 0 & 0 & p_{1|1}(2) & 0 & 0 & \ldots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & 0 & \ldots & p_{0|0}(\tau) & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \ldots & 0 & p_{1|1}(\tau) \end{bmatrix}$$

where $p_{i|j}(d) = \Pr(S_t = i | S_{t-1} = j, D_{t-1} = d)$.

When $D_t = \tau$, only four events are given non-zero probabilities:

$$\begin{array}{ll} (S_t = i, D_t = \tau)|(S_{t-1} = i, D_{t-1} = \tau) & i = 0,1 \\ (S_t = i, D_t = 1)|(S_{t-1} = j, D_{t-1} = \tau) & i \neq j, \quad i,j = 0,1 \end{array},$$

[3]The transition matrix is here designed so that the rows, and not the columns, sum to ones.

with the interpretation that, when the economy has been in state i at least τ times, the additional periods in which the state remains i influence no more the probability of transition.

As pointed out by Hamilton (1994, section 22.4), it is always possible to write the likelihood function of $\mathbf{y}_t$, depending only on the state variable at time t, even though in the model a p-order autoregression is present; this can be done using the extended state variable $S_t^* = (D_t, S_t, S_{t-1}, \ldots, S_{t-p})$, which comprehends all the possible combinations of the states of the economy in the last p periods. In table 2 the state space of non-negligible states[4] S_t^*, with $p = 4$ and $\tau = 5$, is shown. If $\tau \geq p$ the maximum number of non-negligible states is given by $u = \sum_{i=1}^{p} 2^i + 2(\tau - p)$. The transition matrix $\mathbf{P}^*$ of the

	D_t	S_t	S_{t-1}	S_{t-2}	S_{t-3}	S_{t-4}		D_t	S_t	S_{t-1}	S_{t-2}	S_{t-3}	S_{t-4}
1	1	0	1	0	0	0	17	2	0	0	1	0	0
2	1	0	1	0	0	1	18	2	0	0	1	0	1
3	1	0	1	0	1	0	19	2	0	0	1	1	0
4	1	0	1	0	1	1	20	2	0	0	1	1	1
5	1	0	1	1	0	0	21	2	1	1	0	0	0
6	1	0	1	1	0	1	22	2	1	1	0	0	1
7	1	0	1	1	1	0	23	2	1	1	0	1	0
8	1	0	1	1	1	1	24	2	1	1	0	1	1
9	1	1	0	0	0	0	25	3	0	0	0	1	0
10	1	1	0	0	0	1	26	3	0	0	0	1	1
11	1	1	0	0	1	0	27	3	1	1	1	0	0
12	1	1	0	0	1	1	28	3	1	1	1	0	1
13	1	1	0	1	0	0	29	4	0	0	0	0	1
14	1	1	0	1	0	1	30	4	1	1	1	1	0
15	1	1	0	1	1	0	31	5	0	0	0	0	0
16	1	1	0	1	1	1	32	5	1	1	1	1	1

Table 2: State space of $S_t^* = (D_t, S_t, S_{t-1}, \ldots, S_{t-p})$ for $p = 4$, $\tau = 5$.

Markov chain S_t^* is a $(u \times u)$ matrix, although rather sparse, having a maximum number 2τ of independent non-zero elements.

In order to reduce the number (2τ) of elements in $\mathbf{P}^*$ to be estimated, a more parsimonious probit specification is used. Consider the linear model

$$S_t^\bullet = [\beta_1 + \beta_2 D_{t-1}]S_{t-1} + [\beta_3 + \beta_4 D_{t-1}](1 - S_{t-1}) + \epsilon_t \tag{3}$$

with $\epsilon_t \sim \mathcal{N}(0,1)$, and $S_t^\bullet$ latent variable defined by

$$\Pr(S_t^\bullet \geq 0 | S_{t-1}, D_{t-1}) = \Pr(S_t = 1 | S_{t-1}, D_{t-1}) \tag{4}$$
$$\Pr(S_t^\bullet < 0 | S_{t-1}, D_{t-1}) = \Pr(S_t = 0 | S_{t-1}, D_{t-1}). \tag{5}$$

[4] "Negligible states" stands here for 'states always associated with zero probability'. For example the state $(D_t = 5, S_t = 1, S_{t-1} = 0, S_{t-2} = s_2, S_{t-3} = s_3, S_{t-4} = s_4)$, where s_2, s_3 and s_4 can be either 0 or 1, is negligible as it is not possible for S_t to have been 5 periods in the same state, if the state at time $t-1$ is different from the state at time t.

It's easy to show that it holds

$$p_{1|1}(d) = \Pr(S_t = 1|S_{t-1} = 1, D_{t-1} = d) = 1 - \Phi(-\beta_1 - \beta_2 d) \quad (6)$$

$$p_{0|0}(d) = \Pr(S_t = 0|S_{t-1} = 0, D_{t-1} = d) = \Phi(-\beta_3 - \beta_4 d) \quad (7)$$

where $d = 1, \ldots, \tau$, and $\Phi(.)$ is the standard normal cumulative distribution function. Now the four parameters β completely define the transition matrix $\mathbf{P}^*$.

3 Bayesian inference on the model's parameters

Bayesian inference on the model's unknowns $(\boldsymbol{\mu}, \mathbf{A}, \boldsymbol{\Sigma}, \boldsymbol{\beta}, \{(S_t, D_t)\}_{t=1}^T)$, where $\boldsymbol{\mu} = (\boldsymbol{\mu}_0', \boldsymbol{\mu}_1')'$ and $\mathbf{A} = (\mathbf{A}_1, \ldots, \mathbf{A}_p)$, is carried out using MCMC methods.

The used prior distribution is

$$p(\boldsymbol{\mu}, \mathbf{A}, \boldsymbol{\Sigma}, \boldsymbol{\beta}, (S_0, D_0)) = p(\boldsymbol{\mu})p(\mathbf{A})p(\Sigma)p(\boldsymbol{\beta})p(S_0, D_0),$$

where

$$\boldsymbol{\mu} \sim \mathcal{N}(\mathbf{m}_0, \mathbf{M}_0), \quad (8)$$

$$\text{vec}(\mathbf{A}) \sim \mathcal{N}(\mathbf{a}_0, \mathbf{A}_0), \quad (9)$$

$$p(\boldsymbol{\Sigma}) = |\boldsymbol{\Sigma}|^{-\frac{1}{2}(\text{rank}(\boldsymbol{\Sigma})+1)}, \quad (10)$$

$$\boldsymbol{\beta} \sim \mathcal{N}(\mathbf{b}_0, \mathbf{B}_0), \quad (11)$$

$$(12)$$

and $p(S_0, D_0)$ is a probability function that assigns a prior probability to every element of the state-space of (S_0, D_0). Alternatively it is possible to let $p(S_0, D_0)$ be the ergodic probability function of the Markov chain $\{S_t, D_t\}$.

Let $\boldsymbol{\theta}_i$, $i = 1, \ldots, I$, be a partition of the set $\boldsymbol{\theta}$ containing all the unknowns of the model, and $\boldsymbol{\theta}_{-i}$ represent the set $\boldsymbol{\theta}$ without the elements in $\boldsymbol{\theta}_i$. In order to implement a Gibbs sampler to sample from the joint posterior distribution of all the unknowns of the model, it is sufficient to find the full conditional posterior distribution $p(\boldsymbol{\theta}_i|\boldsymbol{\theta}_{-i}, \mathbf{Y})$, with $\mathbf{Y} = (\mathbf{y}_1, \ldots, \mathbf{y}_T)$ and $i = 1, \ldots, I$. A Gibbs sampler iteration is a generation of random numbers from $p(\boldsymbol{\theta}_i|\boldsymbol{\theta}_{-i}, \mathbf{Y})$, $i = 1, \ldots, I$, where the elements of $\boldsymbol{\theta}_{-i}$ are substituted with the most recent generated values. Since, under mild conditions the Markov chain defined for $\boldsymbol{\theta}^{(i)}$, where $\boldsymbol{\theta}^{(i)}$ is the value of $\boldsymbol{\theta}$ generated at the i^{th} iteration of the

Gibbs sampler, converges to its stationary distribution, and that this stationary distribution is the "true" posterior distribution $p(\boldsymbol{\theta}|\mathbf{Y})$, it is sufficient to fix an initial burn-in period of M iterations, such that the Markov chain may "forget" the starting values $\boldsymbol{\theta}^{(0)}$ (furnished by the researcher), to sample form (an approximation of) the joint posterior distribution. The samples obtained for each element of $\boldsymbol{\theta}$ are samples from the marginal posterior distribution of each parameters.

Gibbs Sampling Steps

Step 1. Generation of $\{S_t^*\}_{t=1}^T$

We use an implementation of the multi-move Gibbs sampler originally proposed by Carter & Kohn (1994), which, suppressing the conditioning on the other parameters from the notation, exploits the equality

$$p(S_1^*, \ldots, S_T^*|\mathbf{Y}_T) = p(S_T^*|\mathbf{Y}_T) \prod_{t=1}^{T-1} p(S_t^*|S_{t+1}^*, \mathbf{Y}_t), \tag{13}$$

with $\mathbf{Y}_t = (\mathbf{y}_1, \ldots, \mathbf{y}_t)$.

Let $\hat{\boldsymbol{\xi}}_{t|i}$ be the vector containing the probabilities of S_t^* being in each state (the first element is the probability of being in state 1, the second element is the probability of being in state 2, and so on) given $\mathbf{Y}_i$ and the model's parameters. Let $\boldsymbol{\eta}_t$ be the vector containing the likelihood of each state given $\mathbf{Y}_t$ and the model's parameters, whose generic element is

$$(2\pi)^{-n/2}|\boldsymbol{\Sigma}|^{-1/2} \exp\left\{-\frac{1}{2}(\mathbf{y}_t - \hat{\mathbf{y}}_t)'\boldsymbol{\Sigma}^{-1}(\mathbf{y}_t - \hat{\mathbf{y}}_t)\right\},$$

where

$$\hat{\mathbf{y}}_t = \boldsymbol{\mu}_0 + \boldsymbol{\mu}_1 S_t + \mathbf{A}_1(\mathbf{y}_{t-1} - \boldsymbol{\mu}_0 - \boldsymbol{\mu}_1 S_{t-1}) + \ldots + \mathbf{A}_p(\mathbf{y}_{t-p} - \boldsymbol{\mu}_0 - \boldsymbol{\mu}_1 S_{t-p})$$

changes value according to the state of S_t^*.

The filtered probabilities of the states can be calculated using the Hamilton's (1994, section 22.4) filter

$$\hat{\boldsymbol{\xi}}_{t|t} = \frac{\hat{\boldsymbol{\xi}}_{t|t-1} \odot \boldsymbol{\eta}_t}{\hat{\boldsymbol{\xi}}_{t|t-1}' \boldsymbol{\eta}_t}$$

$$\hat{\boldsymbol{\xi}}_{t+1|t} = \mathbf{P}\hat{\boldsymbol{\xi}}_{t|t}$$

with the symbol $\odot$ indicating element by element multiplication. The filter is completed with the prior probabilities vector $\hat{\boldsymbol{\xi}}_{1|0}$, that, as already said, will be set equal to the vector of ergodic (or stationary) probabilities.

To sample from the distribution of $\{S_t^*\}_1^T$ given the full information set $\mathbf{Y}_T$, it can be shown that it holds

$$\Pr(S_t^* = i | S_{t+1}^* = k, \mathbf{Y}_t) = \frac{p_{i|k}\hat{\xi}_{t|t}^{(i)}}{\sum_{j=1}^m p_{j|k}\hat{\xi}_{t|t}^{(j)}},$$

where $p_{j|k}$ indicates the transition probability of moving to state j from state k (element (j,k) in the transition matrix $\mathbf{P}^*$) and $\xi_{t|t}^{(j)}$ is the j-th element of vector $\boldsymbol{\xi}_{t|t}$. In matrix notation the same can be written as

$$\hat{\boldsymbol{\xi}}_{t|(S_{t+1}^*=k,\mathbf{Y}_T)} = \frac{\mathbf{P}'_{[k,.]} \odot \hat{\boldsymbol{\xi}}_{t|t}}{\mathbf{P}_{[k,.]}\hat{\boldsymbol{\xi}}_{t|t}} \tag{14}$$

where $\mathbf{P}_{[k,.]}$ identifies the k-th row of the transition matrix $\mathbf{P}$.

Now all the probability functions in equation (13) have been given a form, and the states can, thus, be generated starting from the filtered probability $\hat{\boldsymbol{\xi}}_{T|T}$ and proceeding backward $(T-1,\ldots,1)$, using equation (14) where k is to be substituted with the last generated value (s_{t+1}^*).

Once a set of sampled $\{S_t^*\}$ has been generated, it is automatically available a sample for $\{S_t\}$ and $\{D_t\}$.

The advantage of using the described multi-move Gibbs sampler, compared to the single move Gibbs sampler that can be implemented as in Carlin, Polson & Stoffer (1992), or using the excellent software BUGS, is that the whole vector of states is sampled at once, improving significantly the speed of convergence of the Gibbs samper's chain to its ergodic distribution.

Step 2. Generation of $(\mathbf{A}, \boldsymbol{\Sigma})$

Conditionally on $\{S_t\}_{t=1}^T$ and $\boldsymbol{\mu}$ equation 1 is just a multivariate normal regression model for the variable $\mathbf{y}_t^* = \mathbf{y}_t - \boldsymbol{\mu}_0 - \boldsymbol{\mu}_1 S_t$, whose parameters, given the discussed prior distribution, have the following posterior distributions, known in literature. Let $\mathbf{X}$ be the matrix, whose t^{th} column is

$$\mathbf{x}_t = \begin{pmatrix} \mathbf{y}_t^* \\ \mathbf{y}_{t-1}^* \\ \vdots \\ \mathbf{y}_{t-p}^* \end{pmatrix},$$

for $t = 1,\ldots,T$, and let $\mathbf{Y}^* = (\mathbf{y}_1^*,\ldots,\mathbf{y}_T^*)$.

The posterior for $(\text{vec}(\mathbf{A}), \boldsymbol{\Sigma})$ is, suppressing the conditioning on the other parameters, the normal-inverse Wishart distribution

$$
\begin{aligned}
&p(\text{vec}(\mathbf{A}), \boldsymbol{\Sigma}|\mathbf{Y}, \mathbf{X}) = p(\text{vec}(\mathbf{A})|\boldsymbol{\Sigma}, \mathbf{Y}, \mathbf{X})p(\boldsymbol{\Sigma}|\mathbf{Y}, \mathbf{X}) \\
&p(\boldsymbol{\Sigma}|\mathbf{Y}, \mathbf{X}) \text{ density of a } \mathcal{IW}_k(\mathbf{V}, n-m) \\
&p(\text{vec}(\mathbf{A})|\boldsymbol{\Sigma}, \mathbf{Y}, \mathbf{X}) \text{ density of a } \mathcal{N}(\mathbf{a}_1, \mathbf{A}_1),
\end{aligned}
$$

with

$$
\begin{aligned}
&\mathbf{V} = \mathbf{Y}^*\mathbf{Y}^{*\prime} - \mathbf{Y}^*\mathbf{X}'(\mathbf{X}\mathbf{X}')^{-1}\mathbf{X}\mathbf{Y}^{*\prime} \\
&\mathbf{A}_1 = (\mathbf{A}_0^{-1} + \mathbf{X}\mathbf{X}'\boldsymbol{\Sigma}^{-1})^{-1} \\
&\mathbf{a}_1 = \mathbf{A}_1[\mathbf{A}_0^{-1}\mathbf{a}_0 + (\mathbf{X} \otimes \boldsymbol{\Sigma}^{-1})\text{vec}(\mathbf{Y})].
\end{aligned}
$$

Step 3. Generation of $\boldsymbol{\mu}$

Conditionally on $\mathbf{A}$ and $\boldsymbol{\Sigma}$, by multiplying both sides of equation (2) times

$$\mathbf{A}(L) = (\mathbf{I} - \mathbf{A}_1 L - \ldots - \mathbf{A}_p L^p),$$

where L is the lag or backward operator, we obtain

$$\mathbf{A}(B)\mathbf{y}_t = \boldsymbol{\mu}_0\mathbf{A}(1) + \boldsymbol{\mu}_1\mathbf{A}(B)S_t + \boldsymbol{\varepsilon}_t,$$

which is a multivariate normal linear regression model with known variance $\boldsymbol{\Sigma}$, and can be treated as shown at step 2., with respect to the specified prior for $\boldsymbol{\mu}$.

Step 4. Generation of $\boldsymbol{\beta}$

Conditionally on $\{S_t^*\}_{t=1}^T$, $\boldsymbol{\beta}$ contains the parameters of the porbit model described in section 2. Albert & Chib (1993*a*) have proposed a method based on a data augmentation algorithm to simulate values for the parameters of a probit model. Given the parameter vector $\boldsymbol{\beta}$ of last iteration of the Gibbs sampler, generate the latent variables $\{S_t^\bullet\}$ from the respective truncated normal densities

$$
\begin{aligned}
S_t^\bullet|(S_t = 0, \mathbf{x}_t, \boldsymbol{\beta}) &\sim \mathcal{N}(\mathbf{x}_t'\boldsymbol{\beta}, 1)\mathbb{I}_{(-\infty,0)} && (15) \\
S_t^\bullet|(S_t = 1, \mathbf{x}_t, \boldsymbol{\beta}) &\sim \mathcal{N}(\mathbf{x}_t'\boldsymbol{\beta}, 1)\mathbb{I}_{[0,\infty)} && (16) \\
& && (17)
\end{aligned}
$$

with

$$
\begin{aligned}
&\boldsymbol{\beta} = (\beta_1, \beta_2, \beta_3, \beta_4)' && (18) \\
&\mathbf{x}_t = (S_{t-1}, D_{t-1}, (1 - S_{t-1}), (1 - S_{t-1})D_{t-1})' && (19) \\
& && (20)
\end{aligned}
$$

and $\mathbb{I}_{(a,b)}$ indicator variable used to denote truncation.

With the generated $S_t^{\bullet}$'s the probit regression equation (3) becomes, again, a normal linear model with known variance.

The former Gibbs samper's steps were numbered from 1 to 4, but any ordering of the steps would eventually bring to the same ergodic distribution.

4 The software

In Pelagatti (2002*b*) a software library for Aptech Gauss for DDMS-VAR model had been written and left freely downloadable from the author's internet site. The library was downloaded by many researcher, but some found difficulties in using it. We rewrote the library with some corrections and add-ons for Ox, and made it user friendly exploiting the Modelbase class and OxPack. The DDMSVAR software can be used in three different ways: i) as a software library, ii) as an object class, iii) as an OxPack menu driven package. The full documentation will be soon available, together with the DDMSVAR software at the address *www.statistica.unimib.it/utenti/p_matteo/*; here we give a brief description of the software and illustrate it with an application.

4.1 OxPack version

The easiest way to use DDMSVAR is adding the package (at the moment the DDMSVAR03.oxo file) to OxPack giving DDMSVAR as class name, and following the standard steps.

Formulate

Open a database, choose the time series to model and give them the Y variable status. If you wish to specify an initial series of state variables, this series has to be included in the database and, once selected in the model variables' list, give it the State variable init status; otherwise DDMSVAR assigns the state variables initial values automatically.

Model settings

Chose the order of the VAR model (p), the maximal duration (tau), which must be at least[5] 2, and write a comma separated list of percentiles of the marginal posterior distributions, that you want to read

[5]If you wish to estimate a classical MS-VAR model, choose tau = 2 and use priors for the parameters β_2 and β_4 that put an enormous mass of probability around 0. This will prevent the duration variable from having influence in the probit regression. The maximal

in the output (default is 2.5,50,97.5).

Estimate/Options

At the moment only the illustrated Gibbs sampler is implemented, but EM algorithm based maximum-likelihood estimation is in the to-do list for the next versions of the software. So choose the sample of data to model and press Options.... The options window is divided in three areas.

ITERATIONS

Now you have to choose the number of iteration of the Gibbs sampler, and the number of burn in iteration, that is, the amounts of start iterations that will not be used for estimation, because too much influenced by the arbitrary starting values. Of course the latter must be smaller than the former.

PRIORS & INITIAL VALUES

If you want to specify prior means and variances of the parameters to estimate, do it in a .in7 or .xls database following these rules: prior means and variances for the vectorization of the autoregressive matrix $\mathbf{A} = [\mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_p]$ must be in fields with names mean_a and var_a; prior means and variances for the mean vectors $\boldsymbol{\mu}_0$ and $\boldsymbol{\mu}_1$ must be in fields with names mean_mu0, var_mu0, mean_mu1 and var_mu1; the fields for the vector $\boldsymbol{\beta}$ are to be named mean_beta and var_beta. The file name is to be specified with extension. If you don't specify the file, DDMSVAR uses priors that are vague for typical applications.

The file containing the initial values for the Gibbs sampler needs also to be a database in .in7 or .xls format, with fields a for vec($\mathbf{A}$), mu0 for $\boldsymbol{\mu}_0$, mu1 for $\boldsymbol{\mu}_1$, sigma for vech($\boldsymbol{\Sigma}$) and beta for $\boldsymbol{\beta}$. If no file is specified, DDMSVAR assigns initial values automatically.

SAVING OPTIONS

In order to save the Gibbs sample generated by DDMSVAR, specify a file name (you don't need to write the extension, at the moment the only format available is .in7) and check Save also state series if the specified file should contain also the samples of the state variables. Check Probabilities of state 0 in *filename.ext* to save the probabilities of $\{S_t = 0\}_{t=1}^{T}$ in the database, from which the time series are taken.

Programme's Output

Since Gibbs sampling may take a long time, after five iterations the programme prints an estimate of the waiting time. The user is informed of the progress of the process every 100 iterations.

usable value for tau depends only on the power of your computer, but have care that the dimensions of the transition matrix $u \times u$ don't grow too much, or the waiting time may become unbearable.

At the end of the iteration process, the estimated means, standard deviations (in the output named standard errors), percentiles of the marginal posterior distributions are given.

The output consists also in a number of graphs:

1. probabilities of S_t being in state 0 and 1,
2. mean and percentiles of the transition probabilities distributions with respect to the duration,
3. autocorrelation function of every sampled parameter (the faster it dies out, the higher the speed of the Gibbs sampler in exploring the posterior distribution's support, and the smaller is the number of iteration needed to achieve the same estimate's precision),
4. kernel density estimates of the marginal posterior distributions,
5. Gibbs sample graphs (to check if the burn in period is long enough to ensure that the initial values have been "forgot"),
6. running means, to visually check the convergence of the Gibbs sample means.

4.2 The DDMSVAR() object class

The second simplest way to use our programme is creating an instance of the object DDMSVAR and using its member functions. The best way to illustrate the most relevant member functions of the class DDMSVAR is showing a sample program and commenting it.

```
#include "DDMSVAR.ox"
main() {
    decl dd = new DDMSVAR();

    dd->LoadIn7("USA4.in7");
    dd->Select(Y_VAR, {"DLIP", 0, 0, "DLEMP", 0, 0,
                       "DLTRADE", 0, 0, "DLINCOME",0 ,0});
    dd->Select(S_VAR,{"NBER", 0, 0});
    dd->SetSelSample(1960, 1, 2001, 8);

    dd->SetVAROrder(0);
    dd->SetMaxDuration(60);
    dd->SetIteration(21000);
    dd->SetBurnIn(1000);
    dd->SetPosteriorPercentiles(<0.05,50,99.5>);
    dd->SetPriorFileName("prior.in7");
    dd->SetInitFileName("init.in7");
```

```
    dd->SetSampleFileName("prova.in7",TRUE);

    dd->Estimate();

    dd->StatesGraph("states.eps");
    dd->DurationGraph("duration.eps");
    dd->Correlograms("acf.eps", 100);
    dd->Densities("density.eps");
    dd->SampleGraphs("sample.eps");
    dd->RunningMeans("means.eps");
}
```

`dd` is declared as instance of the object DDMSVAR. The first four member functions are an inheritance of the class Database and will not be commented here[6]. Notice only that the variable selected in the `S_VAR` group must contain the initial values for the state variable time series. Nevertheless, if no series is selected as `S_VAR`, DDMSVAR calculates initial values for the state variables automatically.

`SetVAROrder(const iP)` sets the order of the VAR model to the integer value `iP`.

`SetMaxDuration(const iTau)` sets the maximal duration to the integer value `iTau`.

`SetIteration(const iIter)` sets the number of Gibbs sampling iterations to the integer value `iIter`.

`SetBurnIn(const iBurn)` sets the number of burn in iterations to the integer value `iBurn`.

`SetPosteriorPercentiles(const vPerc)` sets the percentiles of the posterior distributions that have to be printed in the output. `vPerc` is a row vector containing the percentiles (in %).

`SetPriorFileName(const sFileName)`,
`SetInitFileName(const sFileName)` are optional; they are used to specify respectively the file containing the prior means and variances of the parameters and the file with the initial values for the Gibbs sampler (see the previous subsection for the format that the two files need to have). If they are not used, priors are vague and initial values are automatically calculated.

[6]See Doornik (2001)

`SetSampleFileName(const sFileName, const bSaveS)` is optional; if used it sets the file name for saving the Gibbs sample and if `bSaveS` is `FALSE` the state variables are not saved, otherwise they are saved in the same file `sFileName`. `sFileName` does not need the extension, since the only available format is .in7.

`Estimate()` carries out the iteration process and generates the textual output (if run within GiveWin-OxRun it does also the graphs). After 5 iteration the user is informed of the expected waiting time and every 100 iterations also about the progress of the Gibbs sampler.

`StatesGraph(const sFileName)`,
`DurationGraph(const sFileName)`,
`Correlograms(const sFileName, const iMaxLag)`,
`Densities(const sFileName)`,
`SampleGraphs(const sFileName)`,
`RunningMeans(const sFileName)` are optional and used to save the graphs described in the last subsection. `sFileName` is a string containing the file name with extension (.emf, .wmf, .gwg, .eps, .ps) and `iMaxLag` is the maximum lag for which the autocorrelation funtcion should be calculated.

4.3 DDMSVAR software library

The last and most complicated (but also flexible) way to use the software is as library of functions. The DDMS-VAR library consists in 25 functions, but the user need to know only the following 10. Throughout the function list, it is used the nomenclature below.

p	scalar	order of vector autoregression (VAR(p))
tau	scalar	maximal duration (τ)
k	scalar	number of time series in the model
T	scalar	number of observations of the k time series
u	scalar	dimension of the state space of $\{S_t^*\}$ ($u = \sum_{i=1}^{p} 2^i + 2(\tau - p)$)
Y	$(k \times T)$	matrix of observation vectors ($\mathbf{Y}_T$)
s	$(T \times 1)$	vector of current state variable (S_t)
mu0	$(k \times 1)$	vector of means when the state is 0 ($\boldsymbol{\mu}_0$)
mu1	$(k \times 1)$	vector of mean-increments when the state is 1 ($\boldsymbol{\mu}_1$)
A	$(k \times pk)$	VAR matrices side by side ($[\mathbf{A}_1, \ldots, \mathbf{A}_p]$)
Sig	$(k \times k)$	covariance matrix of VAR error ($\boldsymbol{\Sigma}$)
SS	$(u \times p{+}2)$	state space of the complete Markov chain $\{S^*\}$ (tab. 2)
pd	(tau $\times$ 4)	matrix of the probabilities $[p_{00}(d), p_{01}(d), p_{10}(d), p_{11}(d)]$

P	$(u \times u)$	transition matrix relative to SS ($\mathbf{P}^*$)
xi_flt	$(u \times T-p)$	filtered probabilities ($[\hat{\boldsymbol{\xi}}_{t\|t}]$)
eta	$(u \times T-p)$	matrix of likelihoods ($[\eta_t]$)

`ddss(p,tau)`
Returns the state space SS (see table 2).

`A_sampler(Y,s,mu0,mu1,p,a0,pA0)`
Carry out step 2. of the Gibbs sampler, returning a sample point from the posterior of vec($\mathbf{A}$) with `a0` and `pA0` being respectively the prior mean vector and the prior precision matrix (inverse of covariance matrix) of vec($\mathbf{A}$).

`mu_sampler(Y,s,p,A,Sig,m0,pM0)`
Carry out step 3. of the Gibbs sampler, returning a sample point from the posterior of $[\boldsymbol{\mu}_0', \boldsymbol{\mu}_1']'$ with `m0` and `pM0` being respectively the prior mean vector and the prior precision matrix (inverse of covariance matrix) of $[\boldsymbol{\mu}_0', \boldsymbol{\mu}_1']'$.

`probitdur(beta,tau)`
Returns the matrix `pd` containing the transition probabilities for every duration $d = 1, 2, \ldots, \tau$.

$$\text{pd} = \begin{pmatrix} p_{0|0}(1) & p_{0|1}(1) & p_{1|0}(1) & p_{1|1}(1) \\ p_{0|0}(2) & p_{0|1}(2) & p_{1|0}(2) & p_{1|1}(2) \\ \vdots & \vdots & \vdots & \vdots \\ p_{0|0}(\tau) & p_{0|1}(\tau) & p_{1|0}(\tau) & p_{1|1}(\tau) \end{pmatrix}.$$

`ddtm(SS,pd)`
Puts the transition probabilities `pd` into the transition matrix relative to the chain with state space `SS`.

`ergodic(P)`
Returns the vector `xi0` of ergodic probabilities of the chain with transition matrix `P`.

`msvarlik(Y,mu0,mu1,Sig,A,SS)`
Returns `eta`, matrix of T columns of likelihood contributions for every possible state in `SS`.

`ham_flt(xi0,P,eta)`
Returns `xi_flt`, matrix of T columns of filtered probabilities of being

in each state in `SS`.

`state_sampler(xi_flt,P)`
Carry out step 1. of the Gibbs sampler. It returns a sample time series of values drawn from the chain with state space `SS`, transition matrix `P` and filtered probabilities `xi_flt`.

`new_beta(s,X,lastbeta,diffuse,b,B0)`
Carry out step 4. of the Gibbs sampler. It returns a new sample point from the posterior of the vector $\boldsymbol{\beta}$, given the dependent variables in `X`, where the generic row is given by (19).

The discussed software and sample programs can be downloaded from www.statistica.unimib.it/utenti/p_matteo/[7]

5 A real world application

The model and the software illustrated in the previous sections have been applied to 100 times the difference of the logarithm of the four time series, on which the NBER relays to date the U.S. business cycle, dating from January 1960 to August 2001: a) industrial production (IP), b) total nonfarm-employment (EMP), c) total manufacturing and trade sales in million of 1996$ (TRADE), d) personal income less transfer payments in billions of 1996$ (INCOME).

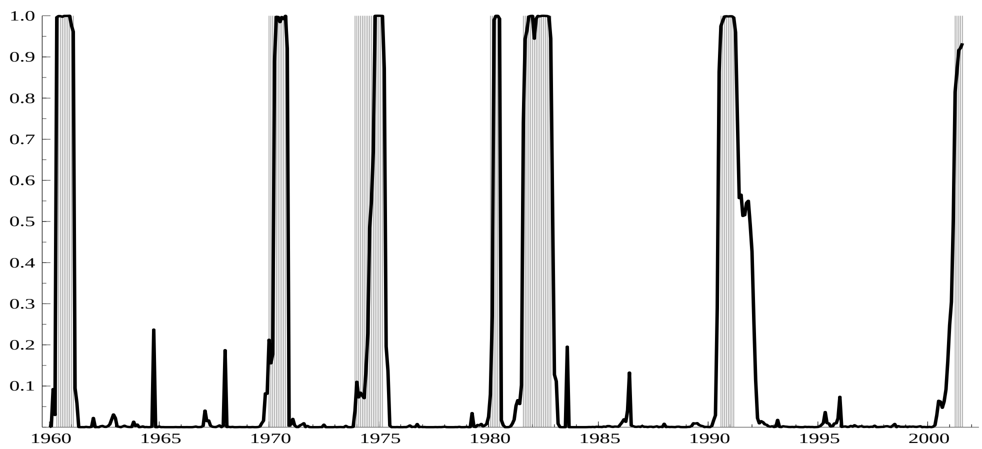
The model, with $\tau = 60$ and $p = 2$ did not work too well, while the results in absence of the VAR part ($\tau = 60$, $p = 0$) are rather encouraging[8]. Summaries of the marginal posterior distributions, based on a Gibbs sample of 21000 points, of which the first 1000 were discarded, are shown in the table below, while figure 1 compares the probability of the U.S. economy being in recession resulting from the estimated model with the official NBER dating of the business cycles: the signal "probability of being in recession" extracted by the model here presented matches rather well the official dating, and is less noisy than the signal extracted by Hamilton (1989). Figure 2 shows how the duration of a state (recession or expansion) influences the transition probabilities: while the probability of moving from a recession into an expansion seems to be influenced by the duration of the recession, the

[7]The software is freely available and usable (at your own risk): the only condition is that the present article is to be cited in any works in which the DDMSVAR software have been used.

[8]This is probably due to the fact that the duration dependent MS model is a stationary process, which, therefore, can be approximated with an autoregerssive model: so the duration dependent switching part and the VAR part try to "explain" almost the same features of the series, and the model is not too well identified.

probability of falling into a recession appears to be independent of the length of the expansion.

Figure 1: (Smoothed) probability of recession (line) and NBER dating (gray shade)

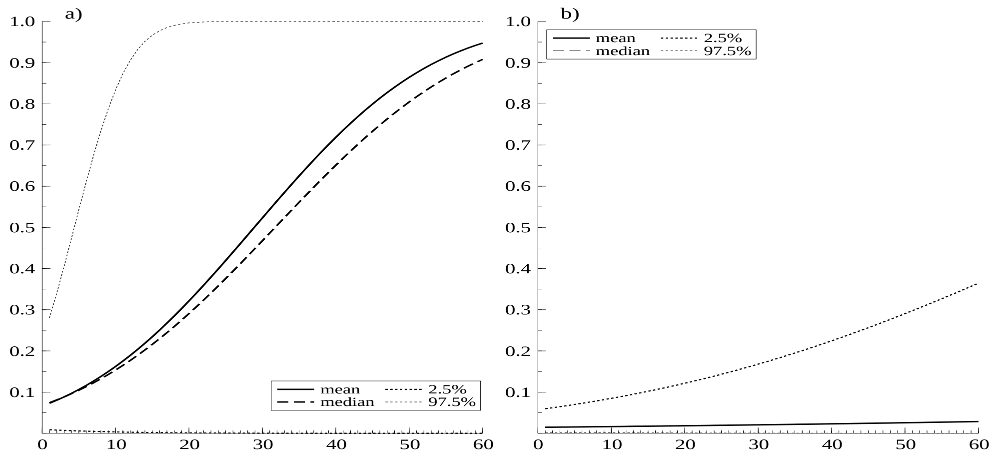


	Prior		Posterior				
Parameter	mean	var	mean	s.d.	2.5%	50%	97.5%
μ_0 IP	-0.300	1.000	-0.684	0.133	-0.943	-0.681	-0.430
μ_0 EMP	-0.300	1.000	-0.182	0.034	-0.246	-0.183	-0.113
μ_0 TRADE	-0.300	1.000	-0.410	0.126	-0.666	-0.407	-0.171
μ_0 INCOME	-0.300	1.000	-0.098	0.050	-0.199	-0.098	0.002
μ_1 IP	1.500	1.000	1.119	0.136	0.858	1.117	1.388
μ_1 EMP	1.500	1.000	0.425	0.032	0.361	0.425	0.486
μ_1 TRADE	1.500	1.000	0.800	0.134	0.539	0.798	1.070
μ_1 INCOME	1.500	1.000	0.441	0.052	0.339	0.441	0.544
β_1	0.000	5.000	2.183	0.335	1.578	2.165	2.887
β_2	0.000	5.000	-0.005	0.008	-0.021	0.004	0.010
β_3	0.000	5.000	-1.507	0.401	-2.341	-1.491	-0.755
β_4	0.000	5.000	0.052	0.053	-0.047	0.066	0.172

6 Conclusions

The model proved to have a good capability of discerning recessions and expansions, as the probabilities of recession tend to assume very low or very high values. Our probabilities of recession (between Jan-1960 and Aug-2001) have a correlation of 0.83 with the NBER classification, and using a 0.5-rule to determine the state of the economy,

Figure 2: Probability of moving a) from a recession into an expansion after d months of recession b) from an expansion to a recession after d months of expansion



only 4.2% (21 out of 499) of our states differ from the NBER states. It also interesting to notice how the last recession is picked up by our model, with a perfect synchronization to the NBER classification, with data until August 2001, while the official NBER announcement of the peak of March 2001 dates November 26, 2001[9].

The results on the duration-dependence of the business cycles are similar to those of Diebold & Rudebusch (1990), Diebold et al. (1993), Sichel (1991) and Durland & McCurdy (1994): recessions are duration dependent, while expansions seem to be not duration dependent.

As far as the software is concerned, we conclude with some comments and a to-do list.

First of all, the version of the software used for this article is 0.3, where the 0. stands for "under construction". Version 1.0 will be available as soon as the things in the to-do list below are implemented and we feel that the code is sufficiently checked and optimized.

The Gibbs sampling approach has many advantages but also a big disadvantage: the former are i) it allows prior information to be exploited, ii) it avoids the computational problems that can arise with

[9]Refer to the Internet page http://www.nber.org/cycles/november2001/. It's not possible to draw conclusions on the compared speed of the NBER and of our model to classify a new cycle because we are not aware of the delay and the precision, with which the NBER receives the first data on the four times series.

maximum likelihood estimation pointed out by Hamilton (1994), iii) it does not relay on asymptotic inference (read note 1.), iv) the inference on the state variables is not conditional on the set of estimated parameters. The big disadvantage is a long computation time. The 21000 Gibbs sampler iterations generated for last section's results took more than 15 hours[10].

The next versions of DDMSVAR should

1. contain estimation of the standard error of the Gibbs sample mean (Geweke's numeric standard error),
2. calculate some Gibbs sampler convergence diagnostics,
3. permit an interactive use of the Gibbs sample (for example, it should be possible to run k_1 iterations, see the output and decide to generate other k_2 iterations and calculate a new output, etc.),
4. allow the Gibbs sample graphs and the running means to be available also at run-time,
5. implement EM-algorithm based maximum likelihood estimates as alternative to the Gibbs sampler.

Suggestion are welcome.

References

Albert, J. H. & Chib, S. (1993*a*), 'Bayesian analysis of binary and polychotomous responce data', *Journal of the American Statistical Association* **88**, 669–79.

Albert, J. H. & Chib, S. (1993*b*), 'Bayesian inference via Gibbs sampling of autoregressive time series subject to Markov mean and variance shifts', *Journal of Business and Economic Statistics* **1**, 1–15.

Carlin, B. P., Polson, N. G. & Stoffer, D. S. (1992), 'A Monte Carlo approach to nonnormal and nonlinear state-space modeling', *Journal of the American Statistical Association* **87**, 493–500.

Carter, C. K. & Kohn, R. (1994), 'On Gibbs sampling for state space models', *Biometrika* **81**, 541–53.

Carter, C. K. & Kohn, R. (1996), 'Markov chain Monte Carlo in conditionally Gaussian state space models', *Biometrika* **83**, 589–601.

Casella, G. & George, E. I. (1992), 'Explaining the Gibbs sampler', *The American Statistician* **46**, 167–173.

[10]On a IBM Thinkpad provided with a Pentium III 800Mhz processor and 384Mb RAM, running Microsoft Windows 2000.

Diebold, F. & Rudebusch, G. (1990), 'A nonparametric investigation of duration dependence in the American business cycle', *Journal of Political Economy* **98**, 596–616.

Diebold, F., Rudebusch, G. & Sichel, D. (1993), Further evidence on business cycle duration dependence, *in* J. Stock & M. Watson, eds, '*Business Cycles, Indicators and Forcasting*', The University of Chicago Press.

Doornik, J. A. (2001), *Ox. An object-oriented matrix programming language*, Timberlake Consultants Ltd.

Durland, J. & McCurdy, T. (1994), 'Duration–dependent transitions in a Markov model of U.S. GNP growth', *Journal of Business and Economic Statistics* **12**, 279–288.

Filardo, A. J. & Gordon, S. F. (1998), 'Business cycle durations', *Journal of Econometrics* **85**, 99–123.

Gamerman, D. (1997), *Markov Chain Monte Carlo: Stochastic Simulation for Bayesian Inference*, Chapman and Hall.

Gelfand, A. E. & Smith, A. F. M. (1990), 'Sampling–based approaches to calculating marginal densities', *Journal of the American Statistical Association* **89**, 398–409.

Gilks, W. R., Richardson, S. & Spiegelhalter, D. J. (1996), *Markov Chain Monte Carlo in Practice*, Chapman and Hall.

Hamilton, J. D. (1989), 'A new approach to the economic analysis of nonstationary time series and the business cycle', *Econometrica* **57**, 357–384.

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

Kim, C. J. & Nelson, C. R. (1999), *State-space models with regime switching: classical and Gibbs-sampling approches with applications*, The MIT Press.

Krolzig, H.-M. (1997), *Markov-Switching Vector Autoregressions. Modelling, Statistical Inference and Application to Business Cycle Analysis*, Springer-Verlag.

Pelagatti, M. M. (2001), 'Gibbs sampling for a duration dependent Markov switching model with an application to the U.S. business cycle', *Working paper of the Department of Statistics, Università degli Studi di Milano-Bicocca* **QD 2001/2**.

Pelagatti, M. M. (2002*a*), 'Duration-dependent Markov-switching VAR models with applications to the business cycle analysis', *Proceedings of the XLI Scientific Meeting of the Italian Statistical Society* .

Pelagatti, M. M. (2002*b*), Markov Chain Monte Carlo Methods for Dynamic Models with Applications to the Business Cycle Analysis, PhD thesis, Università degli Studi di Milano.

Sichel, D. E. (1991), 'Business cycle duration dependence: A parametric approach', *Review of Economics and Statistics* **73**, 254–256.

Tierney, L. (1994), 'Markov chains for exploring posterior distributions', *The Annals of Statistics* **22**, 1701–1762.

Watson, J. (1994), 'Business cycle durations and postwar stabilization of the U.S. economy', *American Economic Review* **84**, 24–46.

Appendix

Figure 3: Kernel density estimates and correlograms of $\boldsymbol{\mu}_0$.

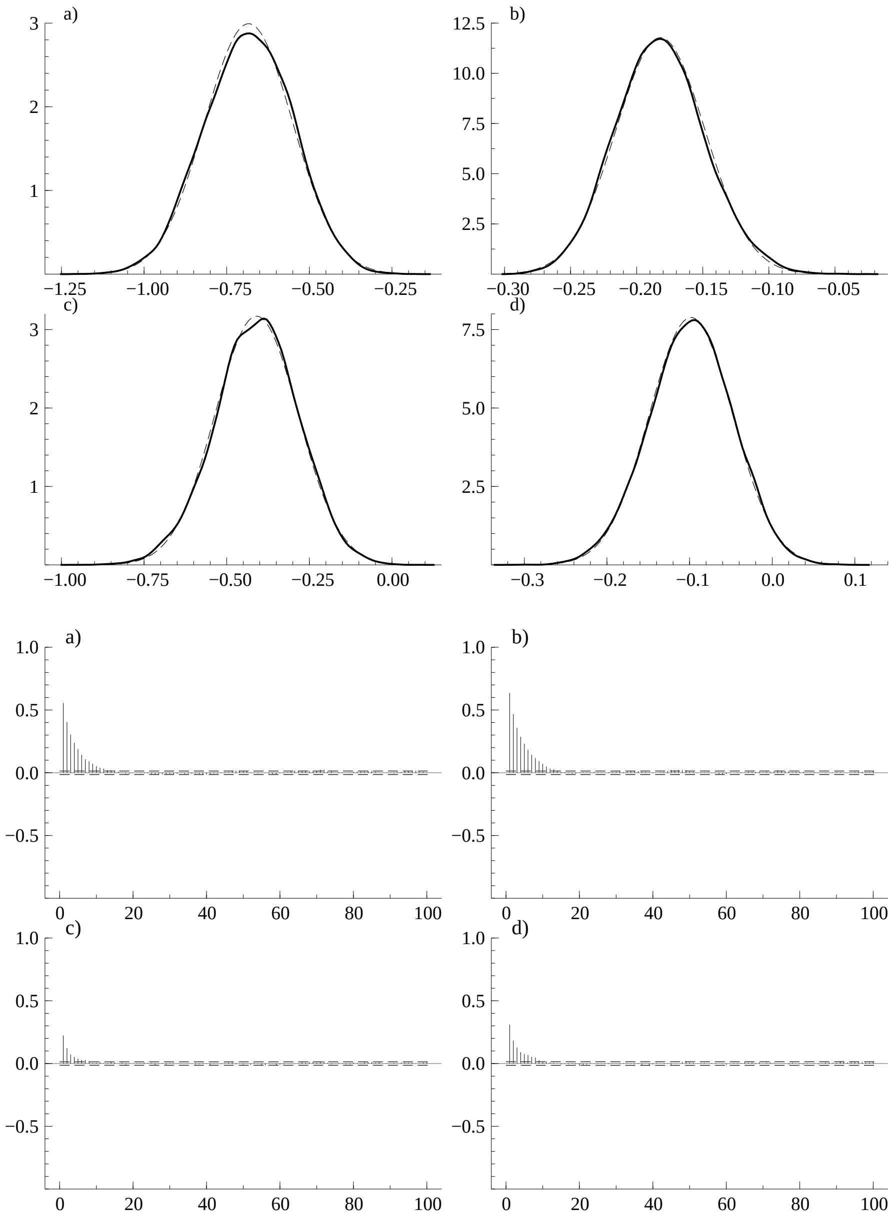


Figure 4: Kernel density estimates and correlograms of $\boldsymbol{\mu}_1$.

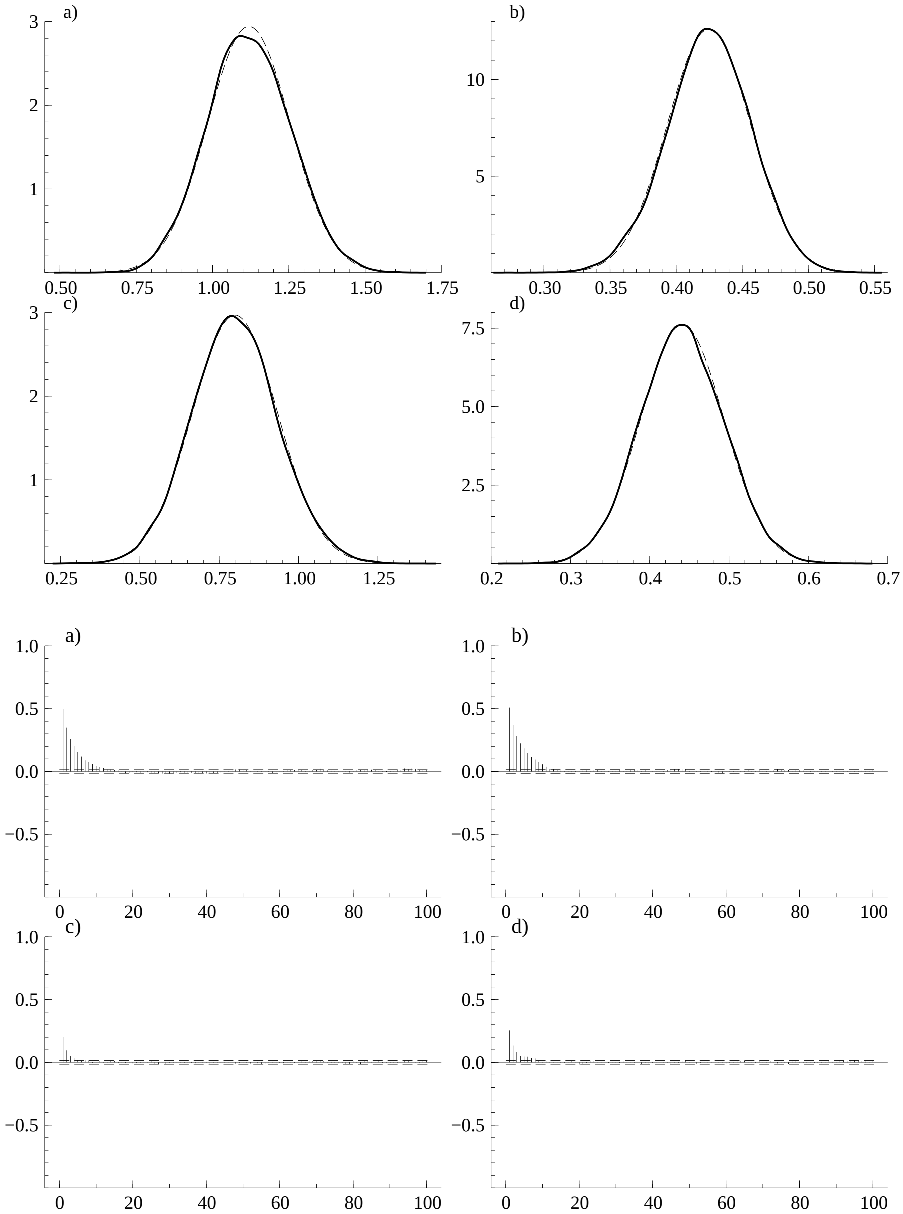


Figure 5: Kernel density estimates and correlograms of $\boldsymbol{\beta}$.

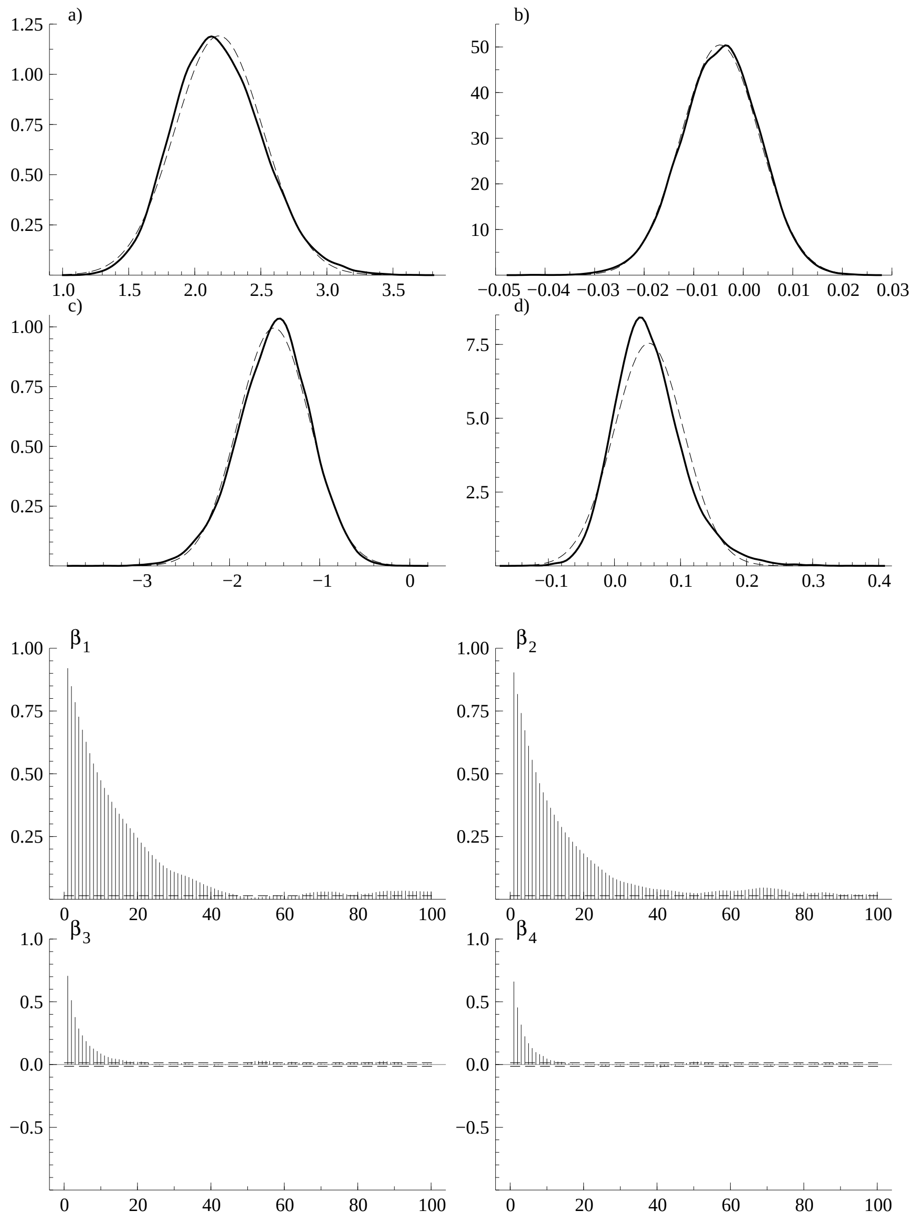


Figure 6: Kernel density estimates and correlograms of $\mathbf{\Sigma}$.