

Time Series Modeling with Duration Dependent Markov-Switching Vector Autoregressions: MCMC Inference, Software and Applications

MATTEO M. PELAGATTI¹

Department of Statistics, Università degli Studi di Milano-Bicocca

E-mail: matteo.pelagatti@unimib.it

Summary Duration dependent Markov-switching VAR (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. In the present paper I propose a MCMC-based methodology to carry out inference on the model's parameters and introduce DDMSVAR for Ox, a software written by the author for the analysis of time series by means of DDMS-VAR models. An application of the methodology to the U.S. business cycle concludes the article.

Keywords: *Markov-switching, business cycle, Gibbs sampling, duration dependence, vector autoregression*

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 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 several months. Some authors, such as Diebold and Rudebusch (1990), Diebold et al. (1993) and Watson (1994) have found evidence of duration dependence in the U.S. business cycles, and therefore, as Diebold et al. (1993) point out, the standard MS model results miss-specified. In order to face the latter limitation, Durland and McCurdy (1994) introduced the duration-dependent Markov switching autoregression, designing an alternative filter for the unobservable state variable. In the present article 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

¹An earlier version of this paper was presented at the 1st OxMetrics User Conference, London 2003. I would like to thank Prof. David Hendry for useful comments and his encouragement. This work was partially supported by a grant of the Italian Ministry of Education, University and Research (MIUR).

filter and Kim's smoother can be used to model also duration dependence. While Durland and McCurdy (1994) carry out their inference on the model by exploiting maximum likelihood estimation, 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 rely on asymptotics¹, and in latent variable models, where the unknowns are many, *asymptopia* can be very far to reach, (b) inference on the latent variables is not conditional on the estimated parameters, but incorporates also the uncertainty on the parameters' values.

The work is organized as follows: the duration-dependent Markov switching VAR model (DDMS-VAR) is defined in section 2, while the MCMC-based inference is explained in section 3; section 4 briefly illustrates the features of DDMSVAR for Ox, and an application of the model and of the software to the U.S. business cycle is carried out in section 5.

2. THE MODEL

The duration-dependent MS-VAR model² is defined by

$$\begin{aligned} \mathbf{y}_t = & \boldsymbol{\mu}_0 + \boldsymbol{\mu}_1 S_t + \mathbf{A}_1(\mathbf{y}_{t-1} - \boldsymbol{\mu}_0 - \boldsymbol{\mu}_1 S_{t-1}) + \dots \\ & + \mathbf{A}_p(\mathbf{y}_{t-p} - \boldsymbol{\mu}_0 - \boldsymbol{\mu}_1 S_{t-p}) + \boldsymbol{\varepsilon}_t \end{aligned} \quad (2.1)$$

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, $\mathbf{A}_1, \dots, \mathbf{A}_p$ are coefficient matrices of a stable VAR process, and $\boldsymbol{\varepsilon}_t$ is a gaussian (vector) white noise with covariance matrix $\boldsymbol{\Sigma}$.

In order to achieve duration dependence for S_t , the pair (S_t, D_t) is considered, where D_t is the duration variable defined by

$$D_t = \begin{cases} 1 & \text{if } S_t \neq S_{t-1} \\ D_{t-1} + 1 & \text{if } S_t = S_{t-1} \text{ and } D_{t-1} < \tau \\ D_{t-1} & \text{if } S_t = S_{t-1} \text{ and } D_{t-1} = \tau \end{cases} \quad (2.2)$$

It is easy to see that (S_t, D_t) is also a Markov chain, since conditioning on (S_{t-1}, D_{t-1}) makes (S_t, D_t) independent of (S_{t-k}, D_{t-k}) with $k = 2, 3, \dots$. An example of a possible sample path of (S_t, D_t) is shown in table 1. The value τ is the maximum that the duration

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 the process (S_t, D_t) .

variable D_t can reach and must be fixed so that the Markov chain (S_t, D_t) is defined on

¹Actually MCMC techniques do rely 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.

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

the finite state space

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

with finite dimensional transition matrix³

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

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:

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

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 $S_t^* = (D_t, S_t, S_{t-1}, \dots, S_{t-p})$, 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 states⁴ 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 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

$$S_t^\bullet = [\beta_1 + \beta_2 D_{t-1}] S_{t-1} + [\beta_3 + \beta_4 D_{t-1}] (1 - S_{t-1}) + \epsilon_t \quad (2.3)$$

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

$$\Pr(S_t^\bullet \geq 0 | S_{t-1}, D_{t-1}) = \Pr(S_t = 1 | S_{t-1}, D_{t-1}) \quad (2.4)$$

$$\Pr(S_t^\bullet < 0 | S_{t-1}, D_{t-1}) = \Pr(S_t = 0 | S_{t-1}, D_{t-1}). \quad (2.5)$$

³The transition matrix is here designed so that the elements of each column, and not of each row, sum to one.

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 .

	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 $S_t^* = (D_t, S_t, S_{t-1}, \dots, S_{t-p})$ for $p = 4$, $\tau = 5$.

It's easy to show that it holds

$$\begin{aligned} p_{1|1}(d) &= \Pr(S_t = 1 | S_{t-1} = 1, D_{t-1} = d) = \\ &= 1 - \Phi(-\beta_1 - \beta_2 d) \end{aligned} \quad (2.6)$$

$$p_{0|0}(d) = \Pr(S_t = 0 | S_{t-1} = 0, D_{t-1} = d) = \Phi(-\beta_3 - \beta_4 d) \quad (2.7)$$

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

3. BAYESIAN INFERENCE ON THE MODEL'S UNKNOWNNS

In this section it is shown how Bayesian inference on the model's unknownns

$$\boldsymbol{\theta} = (\boldsymbol{\mu}, \mathbf{A}, \boldsymbol{\Sigma}, \boldsymbol{\beta}, \{(S_t, D_t)\}_{t=1}^T),$$

where $\boldsymbol{\mu} = (\boldsymbol{\mu}'_0, \boldsymbol{\mu}'_1)'$ and $\mathbf{A} = (\mathbf{A}_1, \dots, \mathbf{A}_p)$, can be carried out using MCMC methods.

3.1. Priors

In order to exploit conditional conjugacy, the prior joint distribution used is

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

$p(\cdot)$ denoting the generic density or probability function, where

$$\boldsymbol{\mu} \sim \mathcal{N}(\mathbf{m}_0, \mathbf{M}_0), \quad (3.8)$$

$$\text{vec}(\mathbf{A}) \sim \mathcal{N}(\mathbf{a}_0, \mathbf{A}_0), \quad (3.9)$$

$$p(\boldsymbol{\Sigma}) \propto |\boldsymbol{\Sigma}|^{-\frac{1}{2}(\text{rank}(\boldsymbol{\Sigma})+1)}, \quad (3.10)$$

$$\boldsymbol{\beta} \sim \mathcal{N}(\mathbf{b}_0, \mathbf{B}_0), \quad (3.11)$$

$$(3.12)$$

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 $\{(S_t, D_t)\}$.

3.2. Gibbs sampling in short

Let θ_i , $i = 1, \dots, I$, be a partition of the set θ containing all the unknowns of the model, and θ_{-i} represent the set θ without the elements in θ_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 $p(\theta_i | \theta_{-i}, \mathbf{Y})$, with $\mathbf{Y} = (\mathbf{y}_1, \dots, \mathbf{y}_T)$ and $i = 1, \dots, I$. A Gibbs sampler step is the generation of a random variate from $p(\theta_i | \theta_{-i}, \mathbf{Y})$, $i = 1, \dots, I$, where the elements of θ_{-i} are substituted with the newest values previously generated. Since, under mild regularity conditions, the Markov chain defined for $\theta^{(i)}$, where $\theta^{(i)}$ is the value of θ generated at the i^{th} iteration of the Gibbs sampler, converges to its stationary distribution, and this stationary distribution is the “true” posterior distribution $p(\theta | \mathbf{Y})$, it is sufficient to fix an initial burn-in period of M iterations, such that the Markov chain virtually “forgets” the arbitrary starting values $\theta^{(0)}$, to sample from (an approximation of) the joint posterior distribution. The values obtained for each element of θ are samples from the marginal posterior distribution of each parameters.

3.3. Gibbs sampling steps

Step 1. Generation of $\{S_t^\}_{t=1}^T$* It is used an implementation of the multi-move Gibbs sampler originally proposed by Carter and Kohn (1994), which, suppressing the conditioning on the other parameters from the notation, exploits the identity

$$p(S_1^*, \dots, S_T^* | \mathbf{Y}_T) = p(S_T^* | \mathbf{Y}_T) \prod_{t=1}^{T-1} p(S_t^* | S_{t+1}^*, \mathbf{Y}_t), \quad (3.13)$$

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

Let $\hat{\xi}_{t|r}$ 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}_r 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

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

changes value according to the state of S_t^* .

The filtered probabilities of the states can be calculated using Hamilton’s filter

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

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

with the symbol \odot indicating element by element multiplication. The filter is completed with the prior probabilities vector $\hat{\xi}_{1|0}$, that, as already noticed, can be set equal to the vector of ergodic probabilities of the Markov chain $\{S_t^*\}$.

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

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

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

$$\hat{\boldsymbol{\xi}}_{t|(S_{t+1}^*=i, \mathbf{Y}_T)} = \frac{\mathbf{p}_{i\cdot} \odot \hat{\boldsymbol{\xi}}_{t|t}}{\mathbf{p}'_{i\cdot} \hat{\boldsymbol{\xi}}_{t|t}} \quad (3.14)$$

where $\mathbf{p}'_{i\cdot}$ denotes the i -th row of the transition matrix \mathbf{P}^* .

Now all the probability functions in equation (3.13) 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 ($T-1, \dots, 1$), using equation (3.14) where i is to be substituted with the last generated value s_{t+1}^* .

Once a set of sampled $\{S_t^*\}_{t=1}^T$ has been generated, it is automatically available a sample for $\{S_t\}_{t=1}^T$ and $\{D_t\}_{t=1}^T$.

The advantage of using the described multi-move Gibbs sampler, compared to the single move Gibbs sampler that can be implemented as in Carlin et al. (1992), or using the software BUGS, is that the whole vector of states is sampled at once, improving significantly the speed of convergence of the Gibbs sampler's chain to its ergodic distribution. Kim and Nelson (1999, section 10.3), in their outstanding monograph on state-space models with regime switching, use a single-move Gibbs sampler (12000 sample points) to achieve (almost) the same goal as in this paper, but my experience with the slow convergence properties of the single-move sampler does not convince me on the reliability of their estimates.

Step 2. Generation of $(\mathbf{A}, \boldsymbol{\Sigma})$ Conditionally on $\{S_t\}_{t=1}^T$ and $\boldsymbol{\mu}$ equation (2.1) is just a multivariate normal (auto-)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

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

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

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

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

Step 3. Generation of $\boldsymbol{\mu}$ Conditionally on \mathbf{A} and $\boldsymbol{\Sigma}$, by multiplying both sides of equation (2) times

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

where L is the lag operator, we obtain

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

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

Step 4. Generation of $\boldsymbol{\beta}$ Conditionally on $\{S_t^*\}_{t=1}^T$, consider the probit model described in section 2. Albert and Chib (1993) have proposed a method based on a data augmentation algorithm to draw from the posterior of 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

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

with

$$\begin{aligned}\boldsymbol{\beta} &= (\beta_1, \beta_2, \beta_3, \beta_4)' \\ \mathbf{x}_t &= (S_{t-1}, D_{t-1}, (1 - S_{t-1}), (1 - S_{t-1}) D_{t-1})'\end{aligned}\tag{3.15}$$

and $\mathbb{I}_{\{\cdot, \cdot\}}$ indicator function used to denote truncation.

With the generated S_t^\bullet 's the probit regression equation (2.3) becomes, again, a normal linear model with known variance.

The former Gibbs sampler'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

DDMSVAR for Ox⁵ is a software for time series modeling with DDMS-VAR processes that can be used in three different ways: (i) as a menu driven package⁶, (ii) as an Ox object class, (iii) as a software library for Ox. The DDMSVAR software is freely available⁷ at the author's internet site www.statistica.unimib.it/utenti/p_matteo/. In this section I give a brief description of the software and in next section I illustrate its use with a real-world application.

⁵Ox (Doornik, 2001) is an object-oriented matrix programming language freely available for the academic community in its console version.

⁶If run with the commercial version of Ox (OxProfessional).

⁷The software is freely available and usable (at your own risk): the only condition is that the present article should be cited in any work in which the DDMSVAR software is used.

4.1. OxPack version

The easiest way to use DDMSVAR is adding the package⁸ to OxPack giving DDMSVAR as class name. The following steps must be followed to load the data, specify the model and estimate it.

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 variable's initial values automatically.

Model settings

Choose the order of the VAR model (**p**), the maximal duration (**tau**), which must be at least⁹ 2, and write a comma separated list of percentiles of the marginal posterior distributions, that you want to read 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

Here you 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 $\mathbf{A} = [\mathbf{A}_1, \mathbf{A}_2, \dots, \mathbf{A}_p]$ 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 $\text{vec}(\mathbf{A})$, **mu0** for $\boldsymbol{\mu}_0$, **mu1** for $\boldsymbol{\mu}_1$, **sigma** for $\text{vech}(\boldsymbol{\Sigma})$ 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

⁸At the moment the DDMSVAR03.oxo file.

⁹If 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 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.

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 smoothed probabilities $\{\Pr(S_t = 0 | \mathbf{Y}_T)\}_{t=1}^T$ in the database from which the time series are taken.

Program's Output

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

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 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 the software 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 here¹⁰. 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.

`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`.

¹⁰See Doornik (2001).

`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 function 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 notation below.

<code>p</code>	scalar	order of vector autoregression ($\text{VAR}(p)$)
<code>tau</code>	scalar	maximal duration (τ)
<code>k</code>	scalar	number of time series in the model
<code>T</code>	scalar	number of observations of the k time series
<code>u</code>	scalar	dimension of the state space of $\{S_t^*\}$ $(u = \sum_{i=1}^p 2^i + 2(\tau - p))$
<code>Y</code>	$(k \times T)$	matrix of observation vectors (\mathbf{Y}_T)
<code>s</code>	$(T \times 1)$	vector of current state variable (S_t)
<code>mu0</code>	$(k \times 1)$	vector of means when the state is 0 ($\boldsymbol{\mu}_0$)
<code>mu1</code>	$(k \times 1)$	vector of mean-increments when the state is 1 ($\boldsymbol{\mu}_1$)
<code>A</code>	$(k \times pk)$	VAR matrices side by side ($[\mathbf{A}_1, \dots, \mathbf{A}_p]$)
<code>Sig</code>	$(k \times k)$	covariance matrix of VAR error ($\boldsymbol{\Sigma}$)
<code>SS</code>	$(u \times p+2)$	state space of the complete Markov chain $\{S^*\}$ (tab. 2)
<code>pd</code>	$(\text{tau} \times 4)$	matrix of the probabilities $[p_{00}(d), p_{01}(d), p_{10}(d), p_{11}(d)]$
<code>P</code>	$(u \times u)$	transition matrix relative to SS (\mathbf{P}^*)
<code>xi_fit</code>	$(u \times T-p)$	filtered probabilities ($[\hat{\xi}_{t t}]$)
<code>eta</code>	$(u \times T-p)$	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 $\text{vec}(\mathbf{A})$ with `a0` and `pA0` being respectively the prior mean vector and the prior precision matrix (inverse of covariance matrix) of $\text{vec}(\mathbf{A})$.

`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 $d = 1, 2, \dots, \tau$.

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

`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`.

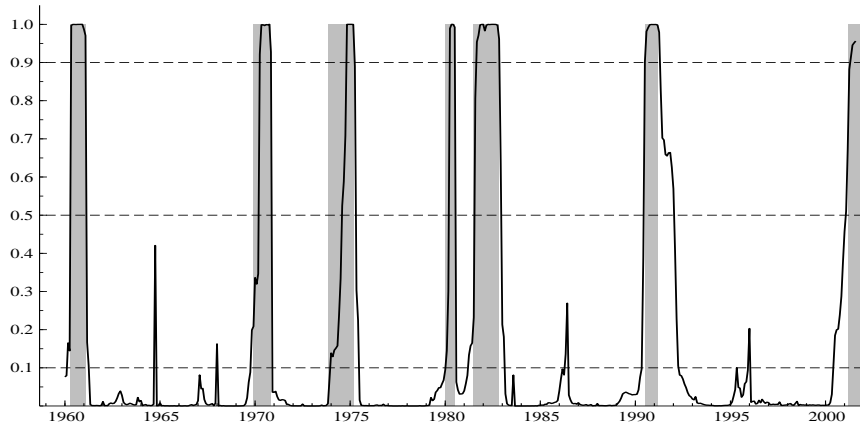
`new_beta(s,X,lastbeta,diffuse,b,B0)`

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 (3.15). If `diffuse` $\neq 0$, a diffuse prior is used.

5. DURATION DEPENDENCE IN THE U.S. BUSINESS CYCLE

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 relies to date the U.S. business cycle, dating from January 1960 to August 2001: i) industrial production (IP), ii) total nonfarm-employment (EMP), iii) total manufacturing and trade sales in million of 1996\$ (TRADE), iv) personal income less transfer payments in billions of 1996\$ (INCOME).

The model, with $p = 2$ did not work too well, while the results in absence of the VAR part ($p = 0$) and maximal duration $\tau = 120$ (10yrs) are rather encouraging: this is

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

probably due to the fact that the duration dependent MS model is a stationary process, which, therefore, can be arbitrarily well approximated by an autoregressive process. As a result the duration dependent switching part and the VAR part of the model try to “explain” almost the same features (autocorrelation) of the series, and the model is not too well identified.

The inference on the model unknowns is based on a Gibbs sample of 21000 points, the first 1000 of which were discarded. In appendix the correlograms and the kernel density estimates for each parameter are reported. All the correlograms die out before the 100th lag, thus the choice of a burn-in sample of 1000 points seems quite reasonable¹¹.

An earlier experiment with $\tau = 60$ (5yrs) was carried out, but it is not reported here: the results were quite similar to the ones reported below and the conclusions the same.

Summaries of the marginal posterior distributions are shown in table 3, while figure 1 compares the probability of the U.S. economy being in recession resulting from the estimated model with the official NBER dating: the signal “probability of being in recession” extracted by the model here presented matches the official dating rather well, and is less noisy than the signal extracted by Hamilton (1989), based on the IP series only. The NBER dating seems to be best matched if, every time the model’s probability of being in recession exceeds 0.5, the peak date is set equal the time the line crosses a low probability level (say 0.1) from below and the trough date is set equal the time the probability line crosses a high probability level (say 0.9) from above. NBER trough dates seem to be matched more frequently by the model than the peaks.

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

¹¹The Gibbs sample is available for further inspections, by sending an e-mail to the author.

Table 3. Description of the prior and posterior distributions of the model's parameters.

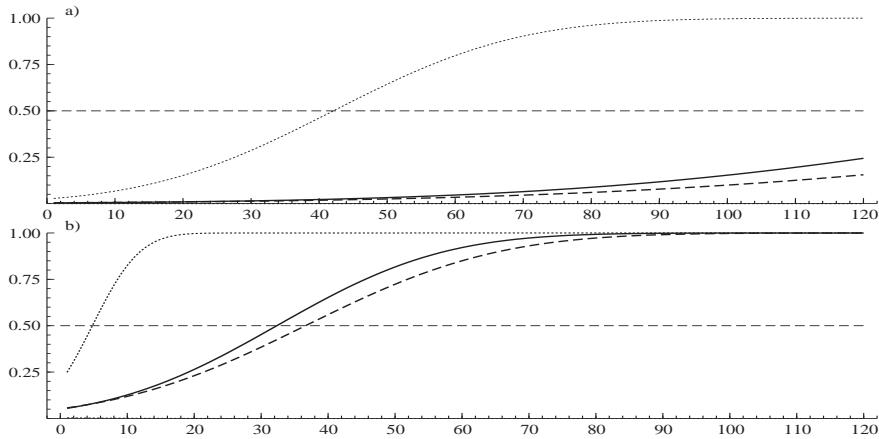
Parameter	Prior		Posterior				
	mean	var	mean	s.d.	2.5%	50%	97.5%
μ_0 IP	0.000	4.000	0.4292	0.04221	0.3519	0.4271	0.5187
μ_0 EMP	0.000	4.000	0.2425	0.01257	0.2213	0.2415	0.2698
μ_0 TRADE	0.000	4.000	0.3888	0.05460	0.2849	0.3875	0.4978
μ_0 INCOME	0.000	4.000	0.3422	0.02273	0.3024	0.3406	0.3926
μ_1 IP	0.000	4.000	-1.0056	0.1520	-1.2907	-1.0080	-0.6914
μ_1 EMP	0.000	4.000	-0.3962	0.0457	-0.4725	-0.4013	-0.2844
μ_1 TRADE	0.000	4.000	-0.7360	0.1453	-1.0193	-0.7366	-0.4500
μ_1 INCOME	0.000	4.000	-0.4087	0.0578	-0.5233	-0.4081	-0.2967
β_1	0.000	5.000	1.6526	0.4490	0.8584	1.6239	2.6200
β_2	0.000	5.000	-0.0511	0.0552	-0.1794	-0.0443	0.0350
β_3	0.000	5.000	-2.6732	0.4208	-3.6189	-2.6341	-1.9601
β_4	0.000	5.000	0.0165	0.0120	0.0002	0.0135	0.0466

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 and, the resulting dating of the U.S. business cycle is very close to the official one.

As far as duration-dependence is concerned, my results are similar to those of Diebold and Rudebusch (1990), Diebold et al. (1993), Sichel (1991) and Durland and McCurdy (1994): U.S. recessions are duration dependent, while expansions seem to be not duration dependent. This could be simply due to the fact that governments are interested in exiting

Figure 2. Mean (solid), median (dash) and 95% credible interval (dots) of the posterior distribution of the 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



contractions, while the opposite is not true, and the policies they put in practice in order to achieve this goal seem effective.

The DDMSVAR software has demonstrated to work fine, even though I must recognize that it is far from being fully optimized: there is too much looping in the code for an interpreted, although very efficient, language as Ox. Future versions will be more efficient.

The Gibbs sampling approach has many advantages but also a big disadvantage: the former are that (i) it allows prior information to be exploited, (ii) it avoids the computational problems pointed out by Hamilton (1994) that can arise with maximum likelihood estimation, (iii) it does not rely 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 13 hours¹².

REFERENCES

- Albert, J. H. and S. Chib (1993). Bayesian analysis of binary and polychotomous response data. *Journal of the American Statistical Association* 88, 669–79.
- Carlin, B. P., N. G. Polson, and D. S. Stoffer (1992). A Monte Carlo approach to nonnormal and nonlinear state-space modeling. *Journal of the American Statistical Association* 87, 493–500.
- Carter, C. K. and R. Kohn (1994). On Gibbs sampling for state space models. *Biometrika* 81, 541–53.
- Diebold, F. and G. Rudebusch (1990). A nonparametric investigation of duration dependence in the American business cycle. *Journal of Political Economy* 98, 596–616.
- Diebold, F., G. Rudebusch, and D. Sichel (1993). Further evidence on business cycle duration dependence. In J. Stock and M. Watson (Eds.), *Business Cycles, Indicators and Forecasting*. The University of Chicago Press.
- Doornik, J. A. (2001). *Ox. An object-oriented matrix programming language*. Timberlake Consultants Ltd.
- Durland, J. and T. McCurdy (1994). Duration-dependent transitions in a Markov model of U.S. GNP growth. *Journal of Business and Economic Statistics* 12, 279–288.
- 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. and C. R. Nelson (1999). *State-space models with regime switching: classical and Gibbs-sampling approaches with applications*. The MIT Press.
- Krolzig, H.-M. (1997). *Markov-Switching Vector Autoregressions. Modelling, Statistical Inference and Application to Business Cycle Analysis*. Springer-Verlag.
- Sichel, D. E. (1991). Business cycle duration dependence: a parametric approach. *Review of Economics and Statistics* 73, 254–256.
- Watson, J. (1994). Business cycle durations and postwar stabilization of the U.S. economy. *American Economic Review* 84, 24–46.

¹²On a Pentium IV 2Ghz processor with 256Mb RAM, running Microsoft Windows 2000.

APPENDIX

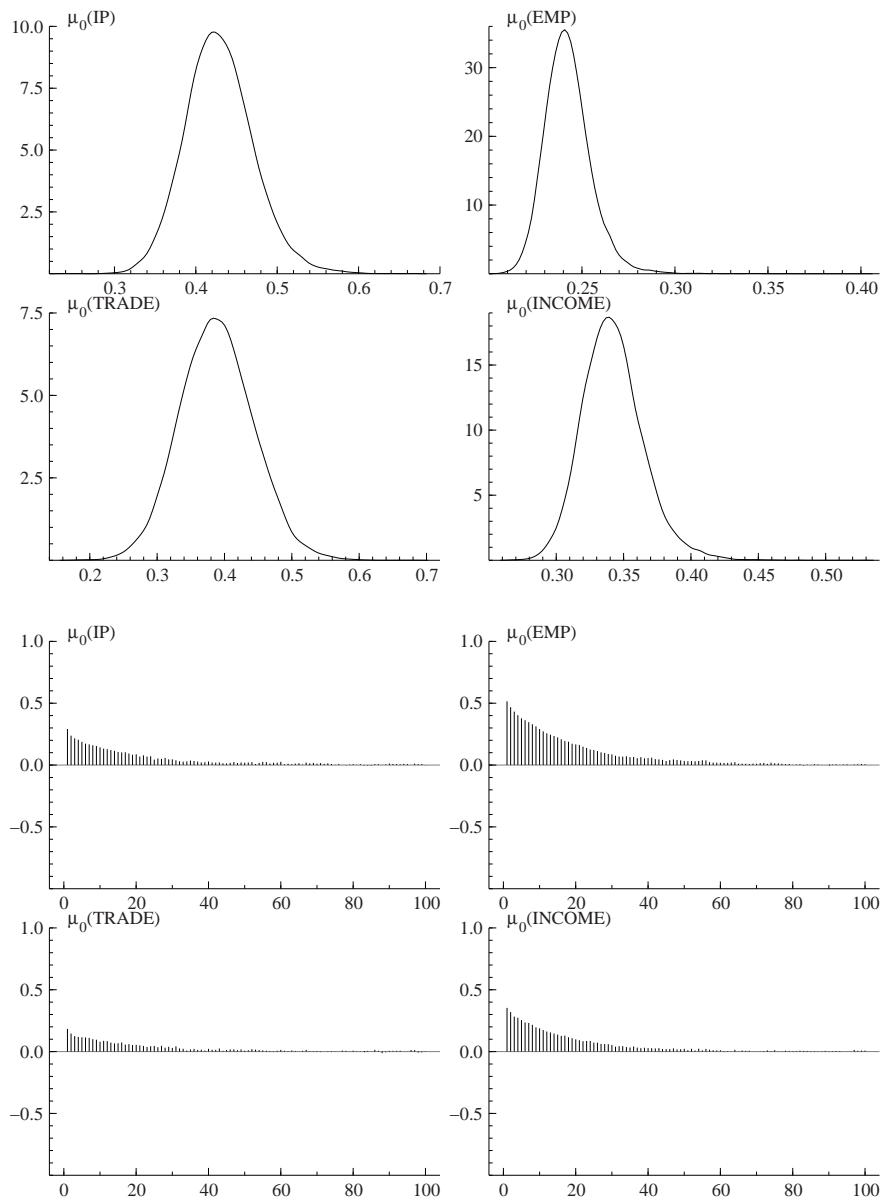
Figure 3. Kernel density estimates and correlograms of μ_0 .

Figure 4. Kernel density estimates and correlograms of μ_1 .

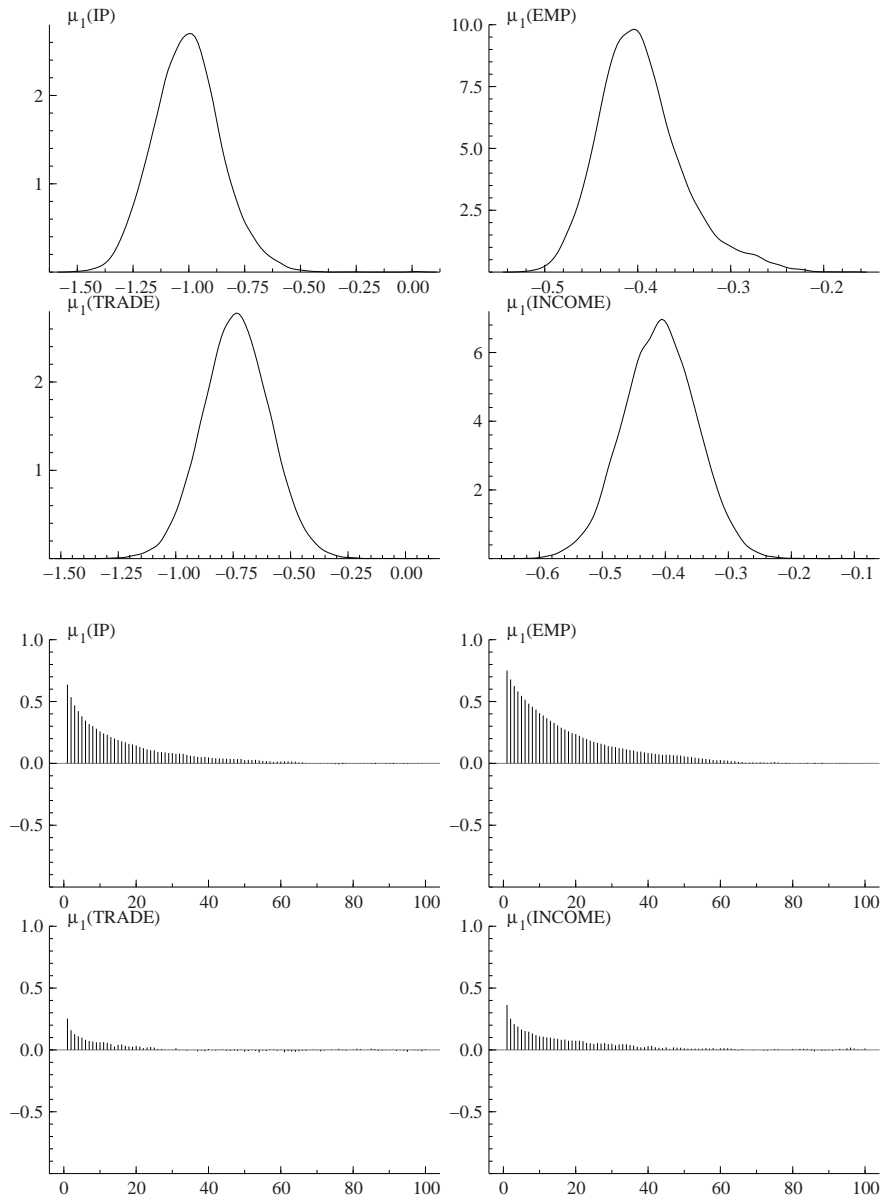


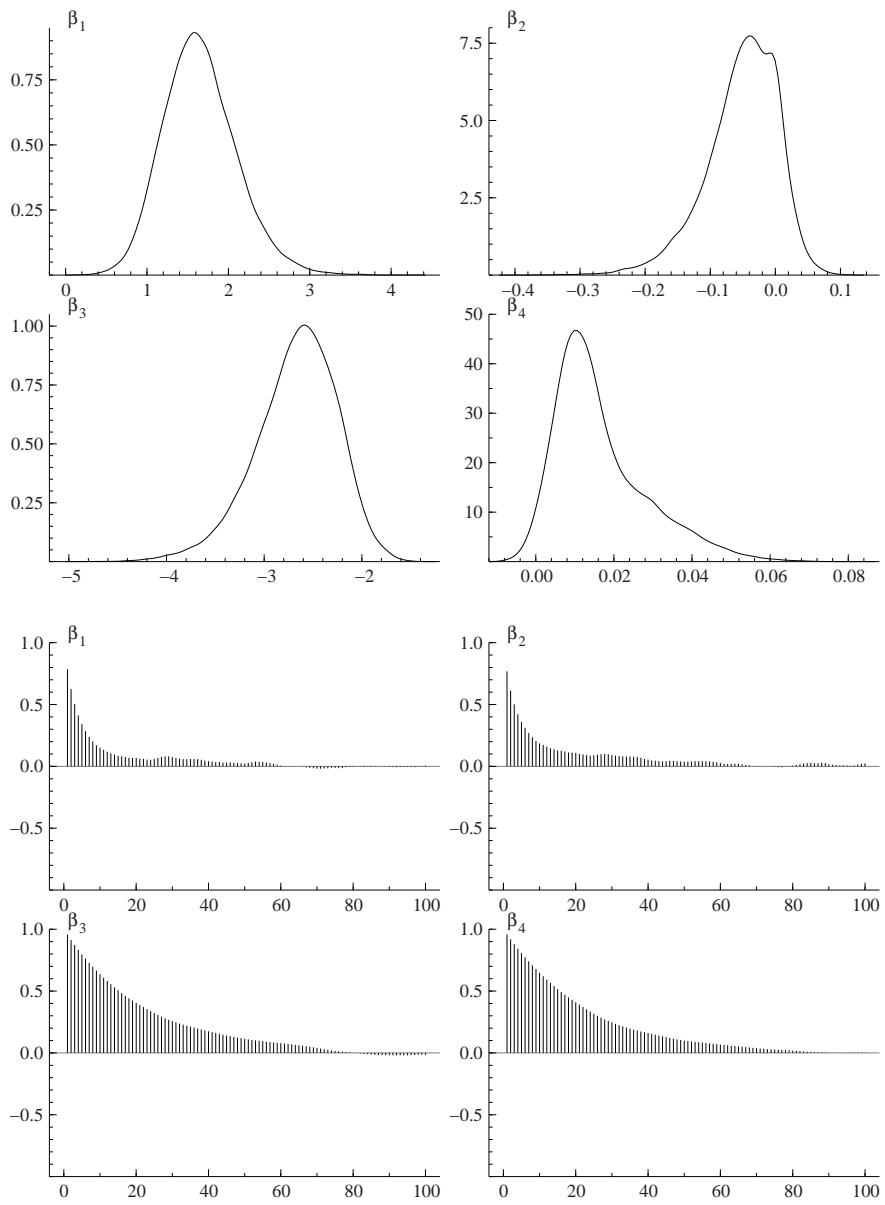
Figure 5. Kernel density estimates and correlograms of β .

Figure 6. Kernel density estimates and correlograms of Σ .

