

Fast Smoothing in Switching Approximations of non-Linear and non-Gaussian Models

Ivan Gorynin

*SAMOVAR, Telecom Sudparis, CNRS,
Université Paris-Saclay
9, Rue Charles Fourier,
91011 Evry Cedex, France.*

Stéphane Derrode*

*École Centrale de Lyon,
LIRIS, CNRS UMR 5205,
36, Av. Guy de Collongue,
69134 Ecully, France.*

Emmanuel Monfrini

*SAMOVAR, Telecom Sudparis, CNRS,
Université Paris-Saclay
9, Rue Charles Fourier,
91011 Evry Cedex, France.*

Wojciech Pieczynski

*SAMOVAR, Telecom Sudparis, CNRS,
Université Paris-Saclay
9, Rue Charles Fourier,
91011 Evry Cedex, France.*

Abstract

Statistical smoothing in general non-linear non-Gaussian systems is a challenging problem. A new smoothing method based on approximating the original system by a recent switching model has been introduced. Such switching model allows fast and optimal smoothing. The new algorithm is validated through an application on stochastic volatility and dynamic beta models. Simulation experiments indicate its remarkable performances and low processing cost. In practice, the proposed approach can overcome the limitations of particle smoothing methods and may apply where their usage is discarded.

Keywords: Smoothing in non-linear systems, Stochastic volatility, Optimal statistical smoother, Conditionally Gaussian linear state-space models, Conditionally Markov switching hidden linear models.

Highlights

- Derivation of an exact statistical smoother for a sub-class of switching models;
- Any stationary dynamical system may be approximated by a switching one in which exact smoothing is available;
- Numerical examples related to the stochastic volatility and dynamic beta models.

*Corresponding author, Phone: int+33 4 72 18 64 45, Fax: int+33 4 72 18 64 43

Email addresses: ivan.gorynin@telecom-sudparis.eu (Ivan Gorynin), stephane.derrode@ec-lyon.fr (Stéphane Derrode), emmanuel.monfrini@telecom-sudparis.eu (Emmanuel Monfrini), wojciech.pieczynski@telecom-sudparis.eu (Wojciech Pieczynski)

URL: www-public.tem-tsp.eu/~igorynin/ (Ivan Gorynin), perso.ec-lyon.fr/derrode.stephane/ (Stéphane Derrode), www-public.tem-tsp.eu/~monfrini/ (Emmanuel Monfrini), www-public.int-evry.fr/~pieczyn/ (Wojciech Pieczynski)

1. Introduction

Let us consider two random sequences $\mathbf{X}_1^N = (\mathbf{X}_1, \dots, \mathbf{X}_N)$ and $\mathbf{Y}_1^N = (\mathbf{Y}_1, \dots, \mathbf{Y}_N)$. \mathbf{X}_1^N are hidden, while \mathbf{Y}_1^N are observable. For all $n \in \{1, \dots, N\}$, \mathbf{X}_n and \mathbf{Y}_n respectively take their values in \mathbb{R}^a and \mathbb{R}^b .

We deal with computing $\mathbb{E}[\mathbf{X}_n | \mathbf{y}_1^N]$ and $\mathbb{E}[\mathbf{X}_n \mathbf{X}_n^T | \mathbf{y}_1^N]$ for all $n \in \{1, \dots, N\}$ and we propose a method that applies when $(\mathbf{X}_1^N, \mathbf{Y}_1^N)$ is a general stationary hidden Markov model (HMM), which can possibly be neither Gaussian nor linear. In fact, it works even in a more general framework called pairwise Markov models (PMMs) for which $p(\mathbf{x}_1^N, \mathbf{y}_1^N)$ is defined by $p(\mathbf{x}_1, \mathbf{y}_1)$ and the recursion

$$(\mathbf{X}_{n+1}, \mathbf{Y}_{n+1}) = f(\mathbf{X}_n, \mathbf{Y}_n, \mathbf{W}_{n+1}), \quad (1)$$

where $\mathbf{W}_1, \dots, \mathbf{W}_N$ are appropriate independent variables. Fast exact filtering and smoothing are generally not computationally feasible in such general models and various approximation methods are used. Among them, the particle filters and smoothers [1, 2] are often used in practice [3, 4, 5]. However, they are particularly CPU intensive.

We remind that the distribution $p(\mathbf{x}_n^{n+1}, \mathbf{y}_n^{n+1})$ resulting from (1) do not depend on n because the PMM is assumed to be stationary. Therefore, the whole distribution $p(\mathbf{x}_1^N, \mathbf{y}_1^N)$ derives from $p(\mathbf{x}_1^2, \mathbf{y}_1^2)$.

Following [6, 7], we propose to approximate the latter distribution by a Gaussian mixture of K^2 components

$$p(\mathbf{x}_1^2, \mathbf{y}_1^2) \approx \sum_{1 \leq i, j \leq K} c_{ij} p_{ij}(\mathbf{x}_1^2, \mathbf{y}_1^2), \quad (2)$$

knowing that such an approximation can be made under mild conditions and with an arbitrarily high precision if K is sufficiently large. Let us interpret c_{ij} as a discrete probability distribution of pair $(R_1, R_2) : c_{ij} = p(R_1 = i, R_2 = j)$. Then $p(\mathbf{x}_1^2, \mathbf{y}_1^2)$ can be seen as a marginal distribution of

$$p(\mathbf{x}_1^2, \mathbf{r}_1^2, \mathbf{y}_1^2) = p(\mathbf{r}_1^2) p(\mathbf{x}_1^2, \mathbf{y}_1^2 | \mathbf{r}_1^2), \quad (3)$$

and $p(\mathbf{x}_1^2, \mathbf{y}_1^2) = \sum_{1 \leq r_1, r_2 \leq K} p(\mathbf{r}_1^2) p(\mathbf{x}_1^2, \mathbf{y}_1^2 | \mathbf{r}_1^2)$ suggests to set, using (2), $p(\mathbf{x}_1^2, \mathbf{y}_1^2 | \mathbf{r}_1^2 = (i, j)) = p_{ij}(\mathbf{x}_1^2, \mathbf{y}_1^2)$.

Let $\mathbf{R}_1^N = (R_1, \dots, R_N)$ be a stationary discrete random sequence, R_n taking its values in the set of classes $\Omega = \{1, \dots, K\}$. We remind that (3) entirely defines the distribution of the stationary Markov triplet $\mathbf{T}_1^N = (\mathbf{X}_1^N, \mathbf{R}_1^N, \mathbf{Y}_1^N)$. Now, exact filtering and smoothing are computable in particular stationary Markov triplet models called ‘‘stationary conditionally Gaussian observed Markov switching models’’ (SCGOMSMs) (see *e.g.* [7, 8]). Indeed, SCGOMSMs are particular ‘‘conditionally Markov switching models’’ (CMSHLMs) in which fast filtering is feasible [9], and we show in the paper that fast smoothing also is. The interest of using (2) to approximate the general model (1) has already been demonstrated for filtering (*c.f.* [10, 7]) and the aim of our paper is to show its interest for smoothing.

How to find the approximation (2)? The core idea is to sample realizations using (1), consider them as being produced by a SCGOMSM, and to use an ‘‘expectation-maximization’’ (EM) method to estimate the SCGOMSM’s parameters. Indeed, considering SCGOMSM as a classic hidden

Markov chain, with $\mathbf{R}_1^N = (R_1, \dots, R_N)$ hidden and $(\mathbf{X}_1^N, \mathbf{Y}_1^N)$ observed, one can use the sample obtained with (1) to estimate all SCGOMSM parameters with some general method like EM.

Our method was applied to three recent dynamical non-Gaussian systems. Reported experiments results show its effectiveness.

Organization of the paper is the following. We show in Section 2 that fast exact smoothing is feasible in the general CMSHLM model. We recall in Section 3 the SCGOMSM model, specifying why it is a particular CMSHLM. SCGOMSM is then used as an approximation of (1), as specified in Section 4. Some experimental results within the context of stochastic volatility [11, 12, 13, 14, 15, 16, 17, 18, 19] and the dynamic beta regression [20] are presented in Section 5. The last Section exposes conclusions and perspectives.

2. Exact Smoothing in Conditionnaly Markov Switching Hidden Linear Models

This section is devoted to our original framework for filtering and smoothing.

Definition (“Model 1”). *Let $\mathbf{X}_1^N, \mathbf{R}_1^N$ and \mathbf{Y}_1^N be random sequences as specified above. The triplet $\mathbf{T}_1^N = (\mathbf{X}_1^N, \mathbf{R}_1^N, \mathbf{Y}_1^N)$ is said to be a “Conditionally Markov switching hidden linear model” if it verifies*

$$\begin{aligned} \mathbf{T}_1^N \text{ is Markov with } p(r_{n+1}, \mathbf{y}_{n+1} | \mathbf{x}_n, r_n, \mathbf{y}_n) &= p(r_{n+1}, \mathbf{y}_{n+1} | r_n, \mathbf{y}_n); \\ \mathbf{X}_{n+1} &= \mathbf{F}_{n+1}(\mathbf{R}_n^{n+1}, \mathbf{Y}_n^{n+1})\mathbf{X}_n + \mathbf{G}_{n+1}(\mathbf{R}_n^{n+1}, \mathbf{Y}_n^{n+1})\mathbf{W}_{n+1} + \mathbf{H}_{n+1}(\mathbf{R}_n^{n+1}, \mathbf{Y}_n^{n+1}), \end{aligned} \quad (4)$$

with $\mathbf{F}_{n+1}(\mathbf{R}_n^{n+1}, \mathbf{Y}_n^{n+1})$, $\mathbf{G}_{n+1}(\mathbf{R}_n^{n+1}, \mathbf{Y}_n^{n+1})$ matrices of appropriate dimensions, \mathbf{W}_1^N is a white noise and $\mathbf{H}_{n+1}(\mathbf{R}_n^{n+1}, \mathbf{Y}_n^{n+1})$ vectors of appropriate dimension.

Figure 1 (a) presents the dependency graph of the Model 1.

We can state the following result, which is an achievement of our prior works:

Proposition 1. *Let $\mathbf{T}_1^N = (\mathbf{X}_1^N, \mathbf{R}_1^N, \mathbf{Y}_1^N)$ be from the Model 1. Then, for each $n = 1, \dots, N$,*

$$\mathbb{E}[\mathbf{X}_n | \mathbf{y}_1^N] = \sum_{r_n} p(r_n | \mathbf{y}_1^N) \mathbb{E}[\mathbf{X}_n | r_n, \mathbf{y}_1^n]; \quad (5)$$

$$\mathbb{E}[\mathbf{X}_n \mathbf{X}_n^\top | \mathbf{y}_1^N] = \sum_{r_n} p(r_n | \mathbf{y}_1^N) \mathbb{E}[\mathbf{X}_n \mathbf{X}_n^\top | r_n, \mathbf{y}_1^n], \quad (6)$$

both expectations being computable with a complexity linear in N .

PROOF. Let us show (5) and (6). By hypothesis (4), for all n in $\{1, \dots, N-1\}$, the variables \mathbf{X}_n and \mathbf{Y}_{n+1} are independent conditionally on $(R_n, \mathbf{Y}_n) = (r_n, \mathbf{y}_n)$. It follows that the variables \mathbf{X}_n and $(R_{n+1}^N, \mathbf{Y}_{n+1}^N)$ are also independent conditionally on $(R_n, \mathbf{Y}_n) = (r_n, \mathbf{y}_n)$. Thus, $p(\mathbf{x}_n | r_n, \mathbf{y}_n^N) = p(\mathbf{x}_n | r_n, \mathbf{y}_n)$. Since $p(\mathbf{x}_n | \mathbf{y}_1^N) = \sum_{r_n} p(r_n | \mathbf{y}_1^N) p(\mathbf{x}_n | r_n, \mathbf{y}_1^n)$, we have (5) and (6).

Let us show that the complexity is linear in N . Since for all n in $\{1, \dots, N-1\}$,

$$\mathbb{E}[\mathbf{X}_{n+1} | r_{n+1}, \mathbf{y}_1^{n+1}] = \sum_{r_n} p(r_n | r_{n+1}, \mathbf{y}_1^{n+1}) (\mathbf{F}_{n+1}(\mathbf{r}_n^{n+1}, \mathbf{y}_n^{n+1}) \mathbb{E}[\mathbf{X}_n | r_n, \mathbf{y}_1^n] + \mathbf{H}_{n+1}(\mathbf{r}_n^{n+1}, \mathbf{y}_n^{n+1})) \quad (7)$$

and

$$\begin{aligned} \mathbb{E} \left[\mathbf{X}_{n+1} \mathbf{X}_{n+1}^\top \mid r_{n+1}, \mathbf{y}_1^{n+1} \right] &= \sum_{r_n} p(r_n \mid r_{n+1}, \mathbf{y}_1^{n+1}) (\mathbf{F}_{n+1}(\mathbf{r}_n^{n+1}, \mathbf{y}_n^{n+1}) \mathbb{E} \left[\mathbf{X}_n \mathbf{X}_n^\top \mid r_n, \mathbf{y}_1^n \right] \mathbf{F}_{n+1}^\top(\mathbf{r}_n^{n+1}, \mathbf{y}_n^{n+1}) \\ &+ \mathbf{F}_{n+1}(\mathbf{r}_n^{n+1}, \mathbf{y}_n^{n+1}) \mathbb{E} \left[\mathbf{X}_n \mid r_n, \mathbf{y}_1^n \right] \mathbf{H}_{n+1}^\top(\mathbf{r}_n^{n+1}, \mathbf{y}_n^{n+1}) + \mathbf{H}_{n+1}(\mathbf{r}_n^{n+1}, \mathbf{y}_n^{n+1}) \mathbb{E} \left[\mathbf{X}_n^\top \mid r_n, \mathbf{y}_1^n \right] \mathbf{F}_{n+1}^\top(\mathbf{r}_n^{n+1}, \mathbf{y}_n^{n+1}) \\ &+ \mathbf{G}_{n+1}(\mathbf{r}_n^{n+1}, \mathbf{y}_n^{n+1}) \mathbf{G}_{n+1}^\top(\mathbf{r}_n^{n+1}, \mathbf{y}_n^{n+1}) + \mathbf{H}_{n+1}(\mathbf{r}_n^{n+1}, \mathbf{y}_n^{n+1}) \mathbf{H}_{n+1}^\top(\mathbf{r}_n^{n+1}, \mathbf{y}_n^{n+1})), \end{aligned} \quad (8)$$

$\mathbb{E} \left[\mathbf{X}_n \mid r_n, \mathbf{y}_1^n \right]$ in (5) and $\mathbb{E} \left[\mathbf{X}_n \mathbf{X}_n^\top \mid r_n, \mathbf{y}_1^n \right]$ in (6) can be computed recursively.

Besides, it follows from hypothesis (4) that $\mathbf{V}_1^N = (\mathbf{R}_1^N, \mathbf{Y}_1^N)$ is Markovian. We can therefore calculate the needed probabilities $p(r_n \mid r_{n+1}, \mathbf{y}_1^{n+1}) = \frac{p(r_{n+1}, \mathbf{y}_{n+1} \mid r_n, \mathbf{y}_n) p(r_n \mid \mathbf{y}_1^n)}{\sum_{r_n^*} p(r_{n+1}, \mathbf{y}_{n+1} \mid r_n^*, \mathbf{y}_n) p(r_n^* \mid \mathbf{y}_1^n)}$ since

$p(r_{n+1}, \mathbf{y}_{n+1} \mid r_n, \mathbf{y}_n)$ are known and $p(r_n \mid \mathbf{y}_1^n)$, $p(r_n \mid \mathbf{y}_1^N)$ can be computed by using the classical “forward” and “backward” probabilities $\alpha_n(r_n) = p(r_n, \mathbf{y}_1^n)$, $\beta_n(r_n) = p(\mathbf{y}_{n+1}^N \mid \mathbf{v}_n)$. More precisely, we have:

$$\begin{aligned} \alpha_1(r_1) &= p(\mathbf{v}_1); \\ \alpha_{n+1}(r_{n+1}) &= \sum_{r_n \in \Omega} \alpha_n(r_n) p(\mathbf{v}_{n+1} \mid \mathbf{v}_n); \end{aligned} \quad (9)$$

$$\begin{aligned} \beta_N(r_N) &= 1; \\ \beta_n(r_n) &= \sum_{r_{n+1} \in \Omega} \beta_{n+1}(r_{n+1}) p(\mathbf{v}_{n+1} \mid \mathbf{v}_n). \end{aligned} \quad (10)$$

Then

$$p(r_n \mid \mathbf{y}_1^n) = \frac{\alpha_n(r_n)}{\sum_{r_n^* \in \Omega} \alpha_n(r_n^*)}, \quad (11)$$

and

$$p(r_n \mid \mathbf{y}_1^N) = \frac{\alpha_n(r_n) \beta_n(r_n)}{\sum_{r_n^* \in \Omega} \alpha_n(r_n^*) \beta_n(r_n^*)}. \quad (12)$$

Remark 1. *There is no particular assumption on the marginal distribution of \mathbf{W}_1^N . Besides, matrices $\mathbf{G}_{n+1}(\mathbf{R}_n^{n+1}, \mathbf{Y}_n^{n+1})$ are not necessary to compute the smoother; they are only necessary to compute its variance.*

Remark 2. *The fact that the variables \mathbf{X}_n and $(\mathbf{R}_{n+1}^N, \mathbf{Y}_{n+1}^N)$ are independent given $(R_n, \mathbf{Y}_n) = (r_n, \mathbf{y}_n)$ could appear as somewhat limiting. However, this kind of assumptions is widespread. For example, in the very classic hidden Markov chain $(\mathbf{R}_1^N, \mathbf{Y}_1^N)$ the variables R_n and \mathbf{Y}_{n+1} are independent conditionally on $R_{n+1} = r_{n+1}$, but they are not independent without this conditioning and it is well known that \mathbf{Y}_{n+1}^N can bring a large deal of information about R_n .*

Remark 3. Let us consider a general stationary Markov triplet $\mathbf{T}_1^N = (\mathbf{X}_1^N, \mathbf{R}_1^N, \mathbf{Y}_1^N)$. Its distribution is defined by $p(\mathbf{x}_1^2, \mathbf{r}_1^2, \mathbf{y}_1^2)$, which is defined by the distribution $p(\mathbf{r}_1^2, \mathbf{y}_1^2)$ and the distributions $p(\mathbf{x}_1^2 | \mathbf{r}_1^2, \mathbf{y}_1^2)$. In the Model 1, the distribution $p(\mathbf{r}_1^2, \mathbf{y}_1^2)$ can be of any kind; in particular, it can be defined in a very general framework by using copulas [21]. In such a general context one may obtain a Model 1 by taking $p(\mathbf{x}_1^2 | \mathbf{r}_1^2, \mathbf{y}_1^2)$ Gaussian such that $p(\mathbf{x}_1 | r_1, \mathbf{y}_1) = p(\mathbf{x}_1 | r_1, \mathbf{y}_1)$.

3. Stationnary Conditionnally Gaussian Observed Markov Switching Models

Let us consider a stationary Model 1 $\mathbf{T}_1^N = (\mathbf{X}_1^N, \mathbf{R}_1^N, \mathbf{Y}_1^N)$ such that $p(\mathbf{x}_1^2, \mathbf{y}_1^2 | \mathbf{r}_1^2)$ (which is equal to $p(\mathbf{x}_n^{n+1}, \mathbf{y}_n^{n+1} | \mathbf{r}_n^{n+1})$ for all n in $\{1, \dots, N-1\}$ by the stationarity assumption) is Gaussian. Let $\mathbf{Z}_n = (\mathbf{X}_n^\top, \mathbf{Y}_n^\top)^\top$. Let us assume that the mean vectors and the covariance matrices of multivariate normal distributions

$$p(\mathbf{x}_1^2, \mathbf{y}_1^2 | \mathbf{r}_1^2) = \mathcal{N}(\mathbf{z}_1, \mathbf{z}_2; \mathbf{\Upsilon}(\mathbf{r}_1^2), \mathbf{\Xi}(\mathbf{r}_1^2)) \quad (13)$$

have a particular structure:

$$\mathbf{\Upsilon}(\mathbf{r}_1^2) = \begin{bmatrix} \mathbb{E}[\mathbf{Z}_1 | r_1] \\ \mathbb{E}[\mathbf{Z}_2 | r_2] \end{bmatrix} = \begin{bmatrix} \mathbf{M}(r_1) \\ \mathbf{M}(r_2) \end{bmatrix} \quad (14)$$

$$\mathbf{\Xi}(\mathbf{r}_1^2) = \begin{bmatrix} \mathbf{S}(r_1) & \mathbf{\Sigma}(\mathbf{r}_1^2) \\ \mathbf{\Sigma}^\top(\mathbf{r}_1^2) & \mathbf{S}(r_2) \end{bmatrix}. \quad (15)$$

Remark 4. According to (13) - (15) we may state that for all n in $\{1, \dots, N-1\}$, $p(\mathbf{x}_n, \mathbf{y}_n | r_n, r_{n+1}) = p(\mathbf{x}_n, \mathbf{y}_n | r_n)$, and thus $p(r_{n+1} | \mathbf{x}_n, r_n, \mathbf{y}_n) = p(r_{n+1} | r_n)$. This ensures that in the subcase of the Model 1 we consider the chain \mathbf{R}_1^N is Markovian.

Definition (“Model 2”). A stationary Model 1 $\mathbf{T}_1^N = (\mathbf{X}_1^N, \mathbf{R}_1^N, \mathbf{Y}_1^N)$ will be said “stationary conditionally Gaussian observed Markov switching model” (SCGOMSMs, or Model 2) if it verifies (13) - (15), and if it has the following property:

$$p(\mathbf{y}_2 | \mathbf{x}_1, \mathbf{r}_1^2, \mathbf{y}_1) = p(\mathbf{y}_2 | \mathbf{r}_1^2, \mathbf{y}_1). \quad (16)$$

Figure 1 (b) presents the dependency graph of the Model 2. In contrast with the dependency graph of the Model 1, we have $p(r_{n+1} | r_n, \mathbf{y}_n) = p(r_{n+1} | r_n)$, what removes the line between \mathbf{Y}_n and R_{n+1} .

To apply the exact smoothing algorithm previously described to a Model 2 defined by (13)-(15), let us remind the link among the matrices from equation (4), i.e. $\mathbf{F}_{n+1}(\mathbf{r}_n^{n+1}, \mathbf{y}_n^{n+1})$, $\mathbf{G}_{n+1}(\mathbf{r}_n^{n+1}, \mathbf{y}_n^{n+1})$, $\mathbf{H}_{n+1}(\mathbf{r}_n^{n+1}, \mathbf{y}_n^{n+1})$, the Model 2 matrices $\mathbf{\Xi}(\mathbf{r}_n^{n+1})$, $\mathbf{\Upsilon}(\mathbf{r}_n^{n+1})$, and the observed variables \mathbf{y}_1^N . For each n in $\{1, \dots, N-1\}$ and \mathbf{r}_n^{n+1} in $\{1, \dots, K\}^2$, let

$$\mathbf{A}(\mathbf{r}_n^{n+1}) = \mathbf{\Sigma}^\top(\mathbf{r}_n^{n+1}) \mathbf{S}^{-1}(r_n), \quad (17)$$

$$\mathbf{Q}(\mathbf{r}_n^{n+1}) = \mathbf{S}(r_{n+1}) - \mathbf{\Sigma}^\top(\mathbf{r}_n^{n+1}) \mathbf{S}^{-1}(r_n) \mathbf{\Sigma}(\mathbf{r}_n^{n+1}). \quad (18)$$

Let us set:

$$\mathbf{Q}(\mathbf{r}_n^{n+1}) = \begin{bmatrix} \mathbf{Q}_1(\mathbf{r}_n^{n+1}) & \mathbf{Q}_2(\mathbf{r}_n^{n+1}) \\ \mathbf{Q}_3(\mathbf{r}_n^{n+1}) & \mathbf{Q}_4(\mathbf{r}_n^{n+1}) \end{bmatrix}.$$

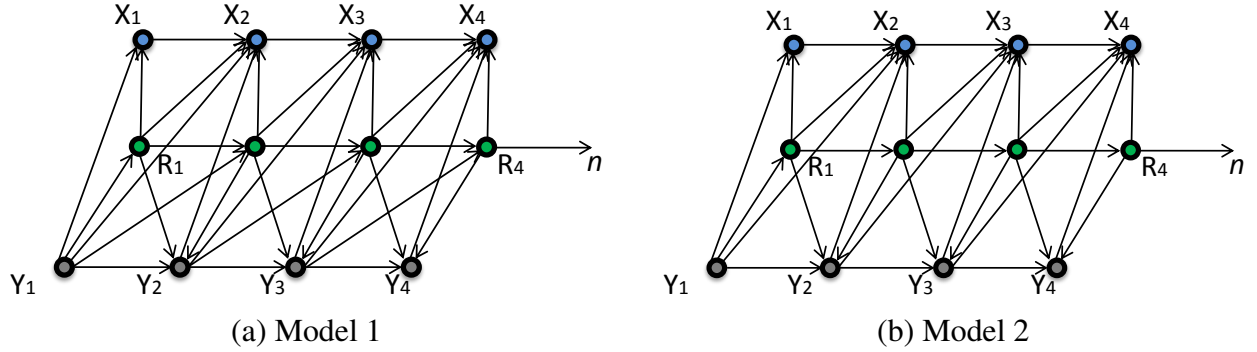


Figure 1: Dependency graphs of Models 1 and 2. \mathbf{X}_1^N , \mathbf{R}_1^N and \mathbf{Y}_1^N are represented by blue, green and gray dots respectively.

Since $\mathbf{Q}(\mathbf{r}_n^{n+1})$ is a positive-definite matrix, $\mathbf{Q}(\mathbf{r}_n^{n+1}) = \mathbf{B}(\mathbf{r}_n^{n+1})\mathbf{B}^\top(\mathbf{r}_n^{n+1})$, for some matrix $\mathbf{B}(\mathbf{r}_n^{n+1})$. Thus, the discrete time process \mathbf{Z}_1^N satisfies the following recursion equation

$$\mathbf{Z}_{n+1} = \mathbf{A}(R_n^{n+1})(\mathbf{Z}_n - \mathbf{M}(R_n)) + \mathbf{B}(R_n^{n+1})\mathbf{W}_{n+1} + \mathbf{M}(r_{n+1}),$$

where $\mathbf{W}_1, \dots, \mathbf{W}_N$ are independent standard Gaussian vectors. Condition (16) implies that the matrix $\mathbf{A}(\mathbf{r}_n^{n+1})$ is of the form

$$\mathbf{A}(\mathbf{r}_n^{n+1}) = \begin{bmatrix} \mathbf{A}_1(\mathbf{r}_n^{n+1}) & \mathbf{A}_2(\mathbf{r}_n^{n+1}) \\ \mathbf{0} & \mathbf{A}_4(\mathbf{r}_n^{n+1}) \end{bmatrix}, \quad (19)$$

with matrices $\mathbf{A}_1(\mathbf{r}_n^{n+1}) \in \mathbb{R}^{a \times a}$, $\mathbf{A}_2(\mathbf{r}_n^{n+1}) \in \mathbb{R}^{a \times b}$ and $\mathbf{A}_4(\mathbf{r}_n^{n+1}) \in \mathbb{R}^{b \times b}$. The distribution

$$p(\mathbf{x}_{n+1}, \mathbf{y}_{n+1} | \mathbf{x}_n, \mathbf{r}_n^{n+1}, \mathbf{y}_n)$$

is then a multivariate normal probability density function with covariance matrix $\mathbf{Q}(\mathbf{r}_n^{n+1})$ and mean vector

$$\mathbf{A}(\mathbf{r}_n^{n+1}) \begin{bmatrix} \mathbf{x}_n \\ \mathbf{y}_n \end{bmatrix} + \begin{bmatrix} \mathbf{N}_1(\mathbf{r}_n^{n+1}) \\ \mathbf{N}_2(\mathbf{r}_n^{n+1}) \end{bmatrix} = \begin{bmatrix} \mathbf{A}_1(\mathbf{r}_n^{n+1})\mathbf{x}_n + \mathbf{A}_2(\mathbf{r}_n^{n+1})\mathbf{y}_n + \mathbf{N}_1(\mathbf{r}_n^{n+1}) \\ \mathbf{A}_4(\mathbf{r}_n^{n+1})\mathbf{y}_n + \mathbf{N}_2(\mathbf{r}_n^{n+1}) \end{bmatrix},$$

where we set $\mathbf{N}_1(\mathbf{r}_n^{n+1}) = \mathbf{M}_1(r_{n+1}) - \mathbf{A}_1(\mathbf{r}_n^{n+1})\mathbf{M}_1(r_n) - \mathbf{A}_2(\mathbf{r}_n^{n+1})\mathbf{M}_2(r_n)$ and $\mathbf{N}_2(\mathbf{r}_n^{n+1}) = \mathbf{M}_2(r_{n+1}) - \mathbf{A}_4(\mathbf{r}_n^{n+1})\mathbf{M}_2(r_n)$ with $\mathbf{M}_1(r_n) = \mathbb{E}[\mathbf{X}_n | r_n]$ and $\mathbf{M}_2(r_n) = \mathbb{E}[\mathbf{Y}_n | r_n]$.

Using the classic Gaussian conditioning rule, the distribution $p(\mathbf{x}_{n+1}, \mathbf{y}_{n+1} | \mathbf{x}_n, \mathbf{r}_n^{n+1}, \mathbf{y}_n)$ gives then the distribution $p(\mathbf{x}_{n+1} | \mathbf{x}_n, \mathbf{r}_n^{n+1}, \mathbf{y}_n^{n+1})$, which is a normal distribution with mean vector

$$\mathbf{Q}_2(\mathbf{r}_n^{n+1})\mathbf{Q}_4^{-1}(\mathbf{r}_n^{n+1})(\mathbf{y}_{n+1} - \mathbf{A}_4(\mathbf{r}_n^{n+1})\mathbf{y}_n - \mathbf{N}_2(\mathbf{r}_n^{n+1})) + \mathbf{A}_1(\mathbf{r}_n^{n+1})\mathbf{x}_n + \mathbf{A}_2(\mathbf{r}_n^{n+1})\mathbf{y}_n + \mathbf{N}_1(\mathbf{r}_n^{n+1}),$$

and covariance matrix

$$\mathbf{Q}_1(\mathbf{r}_n^{n+1}) - \mathbf{Q}_2(\mathbf{r}_n^{n+1})\mathbf{Q}_4^{-1}(\mathbf{r}_n^{n+1})\mathbf{Q}_3(\mathbf{r}_n^{n+1}). \quad (20)$$

Thus the corresponding matrices $\mathbf{F}_{n+1}(\mathbf{r}_n^{n+1}, \mathbf{y}_n^{n+1})$, $\mathbf{H}_{n+1}(\mathbf{r}_n^{n+1}, \mathbf{y}_n^{n+1})$ and $\mathbf{G}_{n+1}(\mathbf{r}_n^{n+1}, \mathbf{y}_n^{n+1})$, are:

$$\mathbf{F}_{n+1}(\mathbf{r}_n^{n+1}, \mathbf{y}_n^{n+1}) = \mathbf{A}_1(\mathbf{r}_n^{n+1}); \quad (21)$$

$$\mathbf{H}_{n+1}(\mathbf{r}_n^{n+1}, \mathbf{y}_n^{n+1}) = \mathbf{N}_1(\mathbf{r}_n^{n+1}) + \mathbf{A}_2(\mathbf{r}_n^{n+1})\mathbf{y}_n + \mathbf{Q}_2(\mathbf{r}_n^{n+1})\mathbf{Q}_4^{-1}(\mathbf{r}_n^{n+1})(\mathbf{y}_{n+1} - \mathbf{A}_4(\mathbf{r}_n^{n+1})\mathbf{y}_n - \mathbf{N}_2(\mathbf{r}_n^{n+1})), \quad (22)$$

$$\mathbf{G}_{n+1}(\mathbf{r}_n^{n+1}, \mathbf{y}_n^{n+1})\mathbf{G}_{n+1}^T(\mathbf{r}_n^{n+1}, \mathbf{y}_n^{n+1}) = \mathbf{Q}_1(\mathbf{r}_n^{n+1}) - \mathbf{Q}_2(\mathbf{r}_n^{n+1})\mathbf{Q}_4^{-1}(\mathbf{r}_n^{n+1})\mathbf{Q}_3(\mathbf{r}_n^{n+1}). \quad (23)$$

They do not depend on n due to the stationarity assumption.

We can therefore make use of Proposition 1 to set up a fast exact smoother for Model 2.

Remark 5. *One can use Model 2 as an alternative to the classic “conditionally Gaussian linear state-space models” (CGLSSMs), in which fast Bayesian smoothing is not computationally feasible [22]. More precisely, any stationary CGLSSM is given by $p(\mathbf{r}_1^2)$, $p(\mathbf{x}_1^2|\mathbf{r}_1^2)$ and $p(\mathbf{y}_1|\mathbf{r}_1)$. If these distributions are the only information about some physical system, then a Model 2 can capture them as well [23]. Specifically, $p(\mathbf{r}_1^2)$, $p(\mathbf{x}_1^2|\mathbf{r}_1^2)$ and $p(\mathbf{y}_1^2|\mathbf{r}_1^2)$ would be the same in CGLSSM and Model 2, and only $p(\mathbf{x}_2, \mathbf{y}_1|\mathbf{r}_1^2)$ would differ. Of course, by the stationarity assumption, the last statement remains valid if we replace, for all n in $\{1, \dots, N-1\}$, 1 by n and 2 by $n+1$.*

4. Approximating non-linear non-Gaussian Models

Since we have shown the computational interest of approximating the process (1) by a Model 2 in order to run an efficient smoothing procedure, we should now explain how to obtain an approximation (2). In other words, how can we find the parameters $(c_{ij}, \Upsilon_{ij}, \Xi_{ij})_{1 \leq i, j \leq K}$, for a given K . To this end, our method requires to be able to simulate realizations of the PMM process (1) in order to generate a training sample $(\mathbf{x}_1^M, \mathbf{y}_1^M)$ to learn the parameters of the Model 2. In practice, the feasibility of a random sampling within the PMM framework is not a restrictive assumption, e.g. in the case of (1), it would be enough to be able to sample \mathbf{W}_1^N .

Regarding the time consumed during parameters’ inference, the number of parameters of the Gaussian mixture increases quadratically with K . Specifically, we have K mean vectors $\mathbf{M}(i) \in \mathbb{R}^{(a+b)}$, K variance matrices $\mathbf{S}(i) \in \mathbb{R}^{(a+b) \times (a+b)}$, K^2 cross-covariance matrices $\Sigma(i, j) \in \mathbb{R}^{(a+b) \times (a+b)}$ and $(K^2 - 1)$ mixture weight scalars c_{ij} .

The complexity of the smoothing procedure is also quadratic in K . Hence, the value of K trades off between quality and time burden. However, we show in next Section that a good smoothing performance is achievable with low values of K , namely four or five.

We suggest to use a variant of the EM algorithm described in [7] to achieve the inference. The EM algorithm usually performs well in conditionally Gaussian switching models. The idea proposed in [10] is to consider an artificial sample $(\mathbf{x}_1^M, \mathbf{y}_1^M)$ generated by the model (1) and then to consider it as a sample issued from a Model 2 $(\mathbf{X}_1^N, \mathbf{R}_1^N, \mathbf{Y}_1^N)$, which is then considered as a hidden Markov chain with latent \mathbf{R}_1^N . The EM algorithm is a great way to estimate the parameters of interest $(c_{ij}, \Upsilon_{ij}, \Xi_{ij})_{1 \leq i, j \leq K}$, however any alternative parameter estimation scheme may be used instead. Besides, for practical purposes, our original EM implementation proposed

in [7] estimates directly $\mathbf{A}(\mathbf{r}_n^{n+1})$, $\mathbf{Q}(\mathbf{r}_n^{n+1})$, $\mathbf{F}(\mathbf{r}_n^{n+1})$, $\mathbf{H}(\mathbf{r}_n^{n+1})$ and $\mathbf{G}(\mathbf{r}_n^{n+1})$ for each value of pair \mathbf{r}_n^{n+1} instead of $(c_{ij}, \Upsilon_{ij}, \Xi_{ij})_{1 \leq i, j \leq K}$.

Let us sum up our new smoothing method by the following algorithm, which thus contains two stages: parameter estimation (or identification of the Model 2) stage, and smoothing stage. Let us insist on the fact that Model 2 identification stage is independent from the data to be smoothed and depends only on the general model (1).

Algorithm 1. *Model 2 smoothing algorithm*

- *Given model (1), Model 2 identification stage:*
 1. *Simulate a training sample $(\mathbf{x}_1^M, \mathbf{y}_1^M)$ within the PMM (1) framework;*
 2. *Apply the EM algorithm to $(\mathbf{x}_1^M, \mathbf{y}_1^M)$ in order to estimate parameters $\mathbf{A}(\mathbf{r}_n^{n+1})$, $\mathbf{Q}(\mathbf{r}_n^{n+1})$, $\mathbf{F}(\mathbf{r}_n^{n+1})$, $\mathbf{H}(\mathbf{r}_n^{n+1})$ and $\mathbf{G}(\mathbf{r}_n^{n+1})$ for each value of pair \mathbf{r}_n^{n+1} ;*
- *Given input data \mathbf{y}_1^N , smoothing stage:*
 1. *Compute recursively $p(r_n | \mathbf{y}_1^n)$, $\mathbb{E}[\mathbf{X}_n | r_n, \mathbf{y}_1^n]$, and $\mathbb{E}[\mathbf{X}_n \mathbf{X}_n^T | r_n, \mathbf{y}_1^n]$ using (11), (7), (8);*
 2. *Compute $\forall n \in \{1, \dots, N\}$, $p(r_n | \mathbf{y}_1^N)$ using (12);*
 3. *Compute the smoothed output using (5) and (6).*

Remark 6. *From Section 3 (see also [7]), we know that the Model 2 may be alternatively represented as:*

$$\begin{aligned} \mathbf{Y}_{n+1} &= \mathbf{D}(\mathbf{r}_n^{n+1})\mathbf{Y}_n + \mathbf{H}(\mathbf{r}_n^{n+1}) + \mathbf{\Lambda}(\mathbf{r}_n^{n+1})\mathbf{V}_{n+1}; \\ \mathbf{X}_{n+1} &= \mathbf{A}(\mathbf{r}_n^{n+1})\mathbf{X}_n + \mathbf{B}(\mathbf{r}_n^{n+1})\mathbf{Y}_n + \mathbf{C}(\mathbf{r}_n^{n+1})\mathbf{Y}_{n+1} + \mathbf{F}(\mathbf{r}_n^{n+1}) + \mathbf{\Pi}(\mathbf{r}_n^{n+1})\mathbf{U}_{n+1}, \end{aligned}$$

for some parameters $\mathbf{D}(\mathbf{r}_n^{n+1})$, $\mathbf{H}(\mathbf{r}_n^{n+1})$, $\mathbf{\Lambda}(\mathbf{r}_n^{n+1})$, $\mathbf{A}(\mathbf{r}_n^{n+1})$, $\mathbf{B}(\mathbf{r}_n^{n+1})$, $\mathbf{C}(\mathbf{r}_n^{n+1})$, $\mathbf{F}(\mathbf{r}_n^{n+1})$, $\mathbf{\Pi}(\mathbf{r}_n^{n+1})$ and standard Gaussian vectors $\mathbf{U}_1, \mathbf{V}_1, \dots, \mathbf{U}_N, \mathbf{V}_N$. In fact, the EM algorithm in [7] estimates a particular autoregression coefficients for the pair $(\mathbf{X}_1^N, \mathbf{Y}_1^N)$ in the Markov-switching context. Thus, the structure imposed by the Model 2 does not result in any difficulties in the M-step of the algorithm, and we still obtain updates in closed form.

Remark 7. *The Model 2 identification stage is the most time-consuming one since the number of elementary operations is proportional to $M \times Q \times K^2$ where Q is the number of iterations of EM. However, these operations do not require the real-world flow \mathbf{y}_1^N and they can be accomplished in advance. The smoothing stage has a complexity linear in N and thus is very fast.*

A Matlab implementation of our smoothing algorithm is downloadable at <http://www-public.it-sudparis.eu/~igorynin/SCGOMSMs>). An implementation of the corresponding EM algorithm is also available.

5. Experiments

In this Section, we consider three sets of experiments related to the smoothing in various non-linear non-Gaussian systems.

5.1. Dynamic beta models

The dynamic beta regression allows modeling monthly unemployment rate [20]. More precisely, let \mathbf{Y}_n be the unemployment rate at time n , the dynamic beta model for \mathbf{Y}_n is (c.f. [24]):

$$\begin{aligned} \mathbf{Y}_n &\sim \text{Beta}\left(\frac{1}{c(1 + \exp(\mathbf{X}_n))}, \frac{\exp(\mathbf{X}_n)}{c(1 + \exp(\mathbf{X}_n))}\right); \\ \mathbf{X}_{n+1} &= \mu + \phi(\mathbf{X}_n - \mu) + \sigma\mathbf{U}_{n+1}, \end{aligned} \quad (25)$$

where μ, ϕ, σ and c are fixed and $\mathbf{U}_1, \dots, \mathbf{U}_N$ are independent standard Gaussian vectors. We recall that for two positives reals α and β , $\text{Beta}(\alpha, \beta)$ denotes the beta distribution:

$$\text{Beta}(x; \alpha, \beta) = \begin{cases} \frac{x^{\alpha-1}(1-x)^{\beta-1}\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} & \text{if } x \in [0, 1] \\ 0 & \text{otherwise} \end{cases}, \quad (26)$$

where Γ denotes the Gamma function $\Gamma(x) = \int_0^\infty t^{x-1} \exp(-t) dt$.

If $|\phi| < 1$ and $\mathbf{X}_1 \sim \mathcal{N}(\mu, \frac{\sigma^2}{1-\phi^2})$, then the autoregressive process of $\mathbf{X}_1, \dots, \mathbf{X}_N$ is stationary c.f. [25], as well as $(\mathbf{X}_1, \mathbf{Y}_1), \dots, (\mathbf{X}_N, \mathbf{Y}_N)$. Therefore, we define $\sigma_0 = \frac{\sigma}{\sqrt{1-\phi^2}}$.

The conditional distribution of \mathbf{Y}_n is generally skewed. Besides, we have:

$$\mathbb{E}[\mathbf{Y}_n | \mathbf{X}_n] = \frac{1}{1 + \exp(\mathbf{X}_n)}; \quad (27)$$

$$\text{Var}[\mathbf{Y}_n | \mathbf{X}_n] = \frac{\exp(\mathbf{X}_n)}{(1 + \exp(\mathbf{X}_n))^2} \left(1 - \frac{1}{c + 1}\right), \quad (28)$$

which means that one can see c as a ‘‘noise level’’ of the observation of \mathbf{X}_n made through \mathbf{Y}_n . When $c = 0$, \mathbf{Y}_n is a deterministic bijective function of \mathbf{X}_n , and when c tends to infinity, the conditional variance of \mathbf{Y}_n tends to its maximum. See Figure 2 for an illustration.

The parameter ϕ is the lag-one autocorrelation of the latent process.

The dynamic beta regression is a particular case of the dynamic generalized linear model [24, 26], where the latent process is Gaussian autoregressive and the observational distribution belongs to the exponential family.

The state estimation in model (25) is an established part of econometric and social analyses. We calibrated this model to a real-world data¹. The rounded values of the parameters are $\mu = -2.82$, $\phi = 0.95$, $\sigma_0 = 0.17$ and $c = 0.005$. In order to test the robustness of Model 2 smoothing algorithm in the case of model (25), we consider estimating $\mathbf{X}_1, \dots, \mathbf{X}_N$ from $\mathbf{Y}_1, \dots, \mathbf{Y}_N$ when observed variables arise from (25) for various values of c and ϕ .

We use our method with different number of states K to estimate the latent variables from the $N = 1000$ observable ones, and we report our results in terms of relative mean square error

¹United States monthly unemployment rate from March 2002 to December 2015, available from the Bureau of Labor Statistics at <http://data.bls.gov/timeseries/LNS14000000>

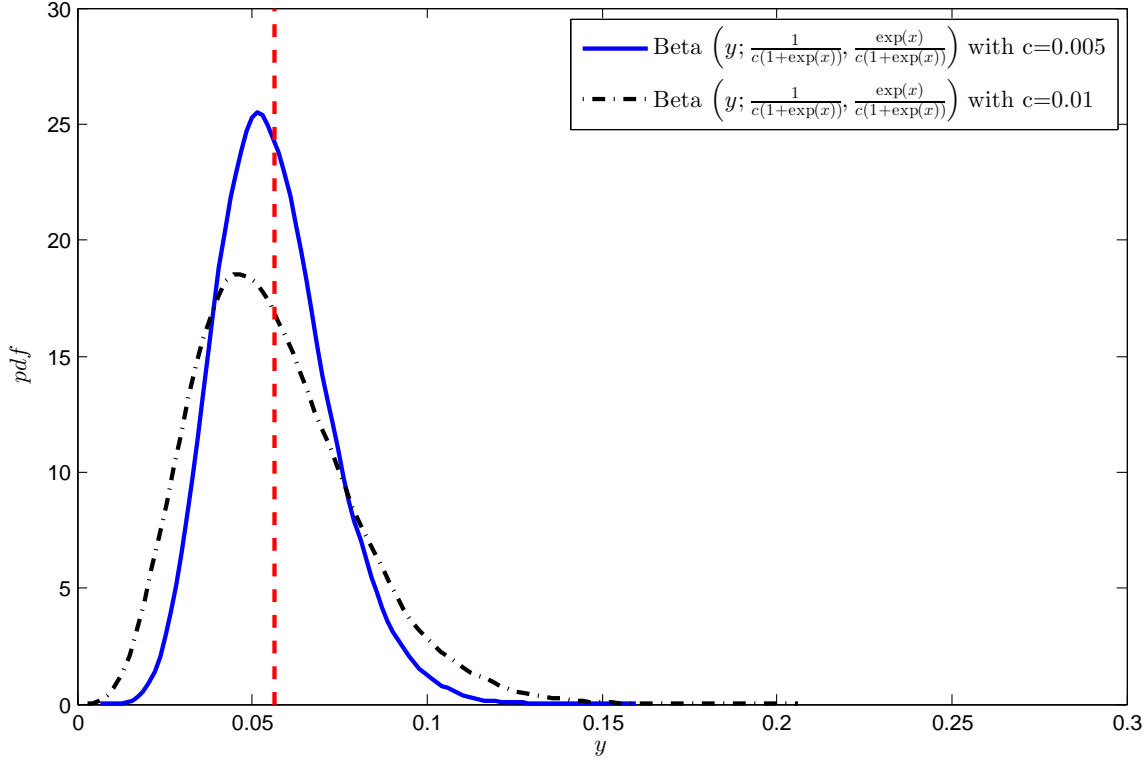


Figure 2: Distribution of \mathbf{Y}_n given $\mathbf{x}_n = -2.82$, for different values of the “noise level” c . The vertical red line locates the common mean of both distributions.

(RMSE) for the mean of 100 independent experiments. The RMSE is relative to the variance of the marginal distribution of \mathbf{X}_n which is σ_0^2 . The results are in Table 1.

The dimensions of the latent variables and the observable ones are $a = b = 1$, the training sample size is $M = 20000$, and $Q = 100$ is the number of EM iterations. For comparison purpose, a similar outcome using a particle smoother (PS) and particle filter (PF) with $m = 1500$ particles (*c.f.* [3, 4, 5]) is also given. We use a fixed-lag particle smoother [27], which computes $\mathbb{E}[\mathbf{X}_n | \mathbf{Y}_1^{n+T}]$ for $T = 5$. We find out that using greater values of T needs more particles to cope with the degeneracy phenomenon, but does not change the RMSE value. We thus consider that $\mathbb{E}[\mathbf{X}_n | \mathbf{Y}_1^{n+T}]$ is a good approximation of $\mathbb{E}[\mathbf{X}_n | \mathbf{Y}_1^N]$.

We observe that for moderate values of K (*e.g.*, $K = 5$), the accuracy of the Model 2 smoother is satisfactory. When the latent process is highly persistent (ϕ close to 1) and when the “noise level” c is significant, one needs a greater number of states to estimate the latent process accurately.

Remark 8. *The complexity of the particle smoother is $N \times m \times T$ while the complexity of our method is $N \times K^2$. In practice, the computation time of our method is quite the same as the one consumed by a particle smoother using K^2 particles, which is rather a small number of particles. As a consequence, one may use a large value of K if needed.*

Table 1: The RMSE of smoothing in model (25) with $\mu = -2.82$, $\sigma_0 = 0.17$ and four different values of lag-one autocorrelation ϕ and noise level c coefficients. The RMSE values for asymptotically optimal particle filter (PF) and particle smoother (PS) are present as a reference.

	ϕ	c	K				PS	PF
			2	3	5	7		
1	0.95	0.005	0.38	0.32	0.31	0.29	0.29	0.41
2	0.95	0.01	0.54	0.43	0.39	0.38	0.38	0.50
3	0.99	0.005	0.40	0.21	0.16	0.16	0.16	0.22
4	0.99	0.01	0.42	0.37	0.31	0.24	0.23	0.28

5.2. Asymmetric stochastic volatility (ASV) model

The Model 2 smoothing algorithm has been successfully applied (*c.f.* [6]) in the context of the classic Stochastic Volatility (SV) model (see [11, 12]) which is the root of more advanced models. The classic SV model assumes that the stock log-return process evolves according to

$$\mathbf{Y}_{n+1} = \beta \exp(\mathbf{X}_{n+1}/2)\mathbf{V}_{n+1}, \quad (29)$$

with the log-variance process verifying

$$\mathbf{X}_{n+1} = \mu + \phi(\mathbf{X}_n - \mu) + \sigma\mathbf{U}_{n+1}, \quad (30)$$

where μ, ϕ, σ and β are fixed and $\mathbf{U}_1, \mathbf{V}_1, \dots, \mathbf{U}_N, \mathbf{V}_N$ are independent standard Gaussian vectors. The SV model is thus a hidden Markov model, specified by a state equation and a measurement equation

$$\mathbf{X}_{n+1} = g(\mathbf{X}_n, \mathbf{U}_{n+1}) \text{ and } \mathbf{Y}_{n+1} = h(\mathbf{X}_{n+1}, \mathbf{V}_{n+1}), \quad (31)$$

which is a particular case of (1).

In the asymmetric stochastic volatility (ASV) model (see [14, 15, 16]), we have:

$$\mathbf{X}_{n+1} = \mu + \phi(\mathbf{X}_n - \mu) + \sigma(\rho\mathbf{V}_n + \lambda\mathbf{U}_{n+1}); \quad (32)$$

$$\mathbf{Y}_{n+1} = \beta \exp(\mathbf{X}_{n+1}/2)\mathbf{V}_{n+1}, \quad (33)$$

where $\mu, \phi, \sigma, \lambda, \rho$ and β are fixed and $\mathbf{U}_1, \mathbf{V}_1, \dots, \mathbf{U}_N, \mathbf{V}_N$ are independent standard Gaussian vectors. Since $p(\mathbf{x}_{n+1} | \mathbf{x}_n, \mathbf{y}_n) \neq p(\mathbf{x}_{n+1} | \mathbf{x}_n)$, ASV model is not of the form (31) and thus is not a classic HMM. However, (1) is still verified and it is possible to generate a training sample which is required by our method. The results are presented in Table 2, on the same methodology basis as previously.

5.3. Markov-switching stochastic volatility (MSSV) model

The Markov switching stochastic volatility (MSSV) model (see [17, 18, 19]) reads as follows:

$$\mathbf{X}_{n+1} = \gamma_1 + \sum_{j=2}^q \gamma_j \mathbb{1}_{[j; +\infty]}(\mathbf{S}_{n+1}) + \phi\mathbf{X}_n + \sigma\mathbf{U}_{n+1}; \quad (34)$$

$$\mathbf{Y}_{n+1} = \exp(\mathbf{X}_{n+1}/2)\mathbf{V}_{n+1}, \quad (35)$$

Table 2: The MSE of smoothing in the ASV model with $\mu = 0.5, \beta = 0.5, \phi = 0.5$ and five different values of λ^2 and ρ such that $\lambda^2 + \rho^2 = 1$ for a unitary unconditional variance of \mathbf{x} .

	K						
	ρ	λ^2	2	3	5	7	PS
1	-0.90	0.19	0.23	0.21	0.20	0.20	0.19
2	-0.80	0.36	0.36	0.34	0.32	0.32	0.32
3	-0.50	0.75	0.57	0.55	0.55	0.55	0.54
4	-0.30	0.91	0.65	0.63	0.62	0.62	0.62
5	-0.00	1.00	0.70	0.67	0.66	0.66	0.66

Table 3: MSE of smoothing in the MSSV model with $k = 2, \gamma_1 = -5.0, \gamma_2 = -3.0, \sigma^2 = 0.1, \phi = 0.5$ and three different values of p_{11} and p_{22} .

	K							
	p_{11}	p_{22}	2	3	5	7	PS	SPS
1	0.99	0.985	0.02	0.02	0.02	0.02	0.66	0.02
2	0.85	0.25	0.71	0.38	0.38	0.38	0.75	0.38
3	0.5	0.5	0.45	0.42	0.42	0.42	0.82	0.42

where $\mathbb{1}_{\mathcal{A}}(\cdot)$ is the indicator function of the set \mathcal{A} , \mathbf{S}_1^N is a stationary discrete Markov chain with q states, $p(\mathbf{s}_{n+1} | \mathbf{x}_1^n, \mathbf{y}_1^n, \mathbf{s}_1^n) = p(\mathbf{s}_{n+1} | \mathbf{s}_n)$ and $\gamma_1, \dots, \gamma_q, \phi, \sigma$ are fixed and $\mathbf{U}_1, \mathbf{V}_1, \dots, \mathbf{U}_N, \mathbf{V}_N$ are independent standard Gaussian vectors. Following the simulation study in [19], we set $q = 2$, $p_{11} = p(\mathbf{s}_{n+1} = 1 | \mathbf{s}_n = 1)$ and $p_{22} = p(\mathbf{s}_{n+1} = 2 | \mathbf{s}_n = 2)$. As a random sampling is straightforward within the framework of MSSV [17], our smoothing algorithm remains applicable. Table 3 shows its results for the MSSV parameters given in [19].

Since PS (for which (34) is approximated by (30) identifying respective ergodic means of \mathbf{X}_n) fails for MSSV model, we propose to compare our method to a switching variant, SPS, of PS which also computes $\mathbb{E}[\mathbf{X}_n | \mathbf{Y}_1^{n+T}]$ for $T = 5$ with $m = 1500$ particles (*c.f.* [19]).

In all cases, we note that if K is large enough, the smoothed output of our method is as good as the statistically optimal one, produced by the PS (or SPS). Our smoothing procedure is riskless from the weight degeneracy phenomenon frequently encountered in particle methods (see [1, 2]) and seems to be robust even in the case of the switching models.

6. Conclusion

We presented a new method to estimate the latent variables in non-linear and non-Gaussian systems. Our method is very general, works under slight conditions and produces an outcome quite close to the optimal one, as illustrated in our experiments. Once the Model 2 is identified, our smoothing procedure is as fast as the classic Kalman smoothing in Gaussian systems.

To conclude, let us mention two perspectives. The first one is to consider different and more complex stochastic volatility models [5, 14, 15, 16]; the second one is to consider more advanced families of switching models allowing fast exact smoothing.

References

- [1] O. Cappé, E. Moulines, T. Rydén, *Inference in Hidden Markov Models*, Springer-Verlag, 2005.
- [2] A. Doucet, A. Johansen, *A tutorial on particle filtering and smoothing: Fifteen years later*, Eds. London, U.K., Oxford Univ. Press, 2011.
- [3] D. Duffie, J. Pan, K. J. Singleton, Transform analysis and asset pricing for affine jump diffusions, *Econometrica* 68 (6) (2000) 1343–1376.
- [4] B. Eraker, Do stock prices and volatility jump? Reconciling evidence from spot and option prices, *The Journal of Finance* 59 (2004) 1367–1404.
- [5] C.-J. Kim, C. R. Nelson, *State-space models with regime switching*, MIT Press, 1999.
- [6] I. Gorynin, S. Derrode, E. Monfrini, W. Pieczynski, Exact fast smoothing in switching models with application to stochastic volatility model, in: *Proc. of the EUSIPCO'15, Nice, France*, 924–928, 2015.
- [7] I. Gorynin, S. Derrode, E. Monfrini, W. Pieczynski, Fast filtering in switching approximations of nonlinear Markov systems with applications to stochastic volatility, *IEEE Transactions on Automatic Control* 62 (2) (2017) 853–862.
- [8] N. Abbassi, D. Benboudjema, S. Derrode, W. Pieczynski, Optimal filter approximations in conditionally Gaussian pairwise Markov switching models, *IEEE Transactions on Automatic Control* 60 (4) (2015) 1104 – 1109.
- [9] W. Pieczynski, Exact filtering in conditionally Markov switching hidden linear models, *Comptes Rendus Mathématique* 349 (9-10) (2011) 587–590.
- [10] S. Derrode, W. Pieczynski, Fast filter in nonlinear systems with application to stochastic volatility model, in: *Proc. of the EUSIPCO'14, Lisbon, Portugal*, 290–294, 2014.
- [11] E. Ghysels, A. Harvey, E. Renault, Stochastic Volatility, *Handbook of Statistics* 14 (1995) 119–192.
- [12] E. Jacquier, N. G. Polson, P. Rossi, Bayesian analysis of stochastic volatility models, *Journal of Business & Economic Statistics* 12 (4) (1994) 371–389.
- [13] N. Shephard, S. Kim, S. Chib, Stochastic volatility: likelihood inference and comparison with ARCH models, *Review of Economic Studies* 65 (3) (1998) 361–393.
- [14] N. Y. Nikolaev, E. Smirnov, L. M. de Menezes, Nonlinear filtering of asymmetric stochastic volatility models and value-at-risk estimation, in: *Proc. of the IEEE Conf. Computational Intelligence for Financial Engineering and Economics (CIFER-2014), London, U.K.*, 310–317, 2014.
- [15] Y. Omori, T. Watanabe, Block sampler and posterior mode estimation for asymmetric stochastic volatility models, *Computational Statistics & Data Analysis* 52 (6) (2008) 2892–2910.
- [16] M. Takashi, Y. Omori, T. Watanabe, Estimating stochastic volatility models using daily returns and realized volatility simultaneously, *Computational Statistics & Data Analysis* 53 (6) (2009) 2404–2426.
- [17] M. K. P. So, K. Lam, W. K. Li, A stochastic volatility model with Markov switching, *Journal of Business & Economic Statistics* 16 (2) (1998) 244–253.
- [18] R. Susmel, M. Kalimipalli, Regime-switching stochastic volatility and short-term interest rates, *CEMA Working Papers: Serie Documentos de Trabajo*. 197, Universidad del CEMA, 2001.
- [19] Y. Bao, C. Chiarella, B. Kang, Particle filters for Markov switching stochastic volatility models, *Research Paper Series* 299, Quantitative Finance Research Centre, University of Technology, Sydney, 2012.
- [20] C. Da-Silva, H. Migon, L. Correia, Dynamic Bayesian beta models, *Computational Statistics & Data Analysis* 55 (6) (2011) 2074 – 2089.
- [21] S. Derrode, W. Pieczynski, Unsupervised data classification using pairwise Markov chains with automatic copulas selection, *Computational Statistics & Data Analysis* 63 (1) (2013) 81–98.
- [22] Y. Petetin, F. Desbouvries, A class of fast exact Bayesian filters in dynamical models with jumps, *IEEE Trans. on Signal Processing* 62 (14) (2014) 3643–3653.
- [23] S. Derrode, W. Pieczynski, Exact fast computation of optimal filter in Gaussian switching linear systems, *IEEE Signal Processing Letters* 20 (7) (2013) 701–704.

- [24] H. Lopes, R. Tsay, Particle filters and Bayesian inference in financial econometrics, *Journal of Forecasting* 30 (1) (2011) 168–209.
- [25] D. A. Dickey, W. A. Fuller, Distribution of the estimators for autoregressive time series with a unit root, *Journal of the American statistical association* 74 (366a) (1979) 427–431.
- [26] M. West, P. Harrison, H. Migon, Dynamic generalized linear models and Bayesian forecasting, *Journal of the American Statistical Association* 80 (389) (1985) 73–83.
- [27] M. Briers, A. Doucet, S. Maskell, Smoothing algorithms for state-space models, *Annals of the Institute of Statistical Mathematics* 62 (1) (2010) 61–89.