The 42nd IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 5-9 March 2017, New Orleans, LA, USA
DOI: 10.1109/ICASSP.2017.7953064, Electronic ISSN: 2379-190X

UNSUPERVISED LEARNING OF ASYMMETRIC HIGH-ORDER AUTOREGRESSIVE STOCHASTIC VOLATILITY MODEL

Ivan Gorynin, Emmanuel Monfrini and Wojciech Pieczynski

SAMOVAR, Telecom SudParis, CNRS, Universit Paris-Saclay, 9, rue Charles Fourrier, 91000 Evry, France.

ABSTRACT

The object of this paper is to introduce a new estimation algorithm specifically designed for the latent high-order autoregressive models. It implements the concept of the filter-based maximum likelihood. Our approach is fully deterministic and is less computationally demanding than the traditional Monte Carlo Markov chain techniques. The simulation experiments and real-world data processing confirm the interest of our approach.

Index Terms— Asymmetric stochastic volatility, Gaussian quadrature, Unsupervised learning, BIC criterion, Gaussian filter.

1. INTRODUCTION

Let p be a positive integer, the p-order asymmetric autoregressive stochastic volatility model [1] A-ARSV(p) is defined as follows:

$$\mathrm{L}_{n+1} = \sum_{j=1}^{p} \phi_j \mathrm{L}_{n+1-j} + \psi \mathrm{W}_n + \sigma \mathrm{V}_{n+1}; \tag{1a}$$

$$\mathrm{Y}_n = \mathrm{W}_n \exp\left(\frac{\mathrm{L}_n}{2} + \frac{\mu}{2}\right), \tag{1b}$$

where $\{\mathrm{V}_n\}_{n\geq 1}$, $\{\mathrm{W}_n\}_{n\geq 1}$ are standard Gaussian white noises. The parameter vector $\boldsymbol{\theta} = (\phi_1, .., \phi_p, \psi, \sigma, \mu)$ is fixed, unknown and is to be estimated. The equation (1a) describes an autoregressive model assumed to be stationary (*see* Section 3 for a statinoarity condition). The process $\{\mathrm{L}_n\}_{n\geq 1}$ is hidden while $\{\mathrm{Y}_n\}_{n\geq 1}$ is observable, as in the hidden Markov model framework (HMM, *see*, *e.g.*, [2]). Moreover, $\{\mathrm{L}_m\}_{m\geq n}$ behaves differently depending on the value taken by Y_n, that is why this is an asymmetric model. We denote the subclass of the A-ARSV(p) models for which $\psi = 0$ by ARSV(p). Unsupervised learning of A-ARSV(p) model consists in estimating of $\boldsymbol{\theta}$ from $\{\mathrm{y}_n\}_{n=1}^{N}$.

In the literature, one finds the estimation algorithms for ARSV(1) (*see*, *e.g.*, [3]), A-ARSV(1) (*see*, *e.g.*, [4]) and ARSV(p) (*see*, *e.g.*, [1]) models. The object of the paper is to describe a new deterministic parameter estimation method for A-ARSV(p) model for $p \geq 1$. Note that our algorithm also allows estimating ARSV(1), A-ARSV(1) and ARSV(p) models since they are nested in A-ARSV(p) model.

Our idea is to define a quasi-likelihood function from an assumed-density filter (ADF, *see*, *e.g.*, [5, 6]). Next, we use an optimization algorithm [7] to compute the quasi-maximum likelihood parameter estimates from the observed series $\{\mathrm{y}_n\}_{n=1}^{N}$.

The novelty of this work consists in designing an ADF for the high-order latent asymmetric models, then combining it with the Gaussian quadrature [8] to define a quasi-likelihood function. As a result, our method avoids any random sampling unlike the Monte Carlo Markov chain methods.

Our primary motivation is to provide a new, simple and robust unsupervised learning algorithm for A-ARSV(p) models. Another motivation arises from the lack of use of high-order A-ARSV(p) models in the current academic literature. However, as we will see in the section devoted to experiments, there are stock indexes for which the use of such models is advisable.

The object of the next section is to define our parameter estimation method. The third section focuses on the implementation of the assumed-density filter used in our approach. The fourth section demonstrates the effectiveness of our method through experiments on both synthetic and real-world data. The fifth section concludes the paper.

2. PARAMETER ESTIMATION

The maximum-likelihood estimation is a well-known method of estimating the parameters of a statistical model. It searches for the most likely value of the unknown parameter vector $\boldsymbol{\theta}$ that would reproduce the observed time series $\{\mathrm{y}_n\}_{n=1}^{N}$. In other words, it maximizes the following function:

$$f(\boldsymbol{\theta}) = \log p_{\boldsymbol{\theta}}(\mathrm{y}_{1..N}) = \sum_{n=1}^{N} \log p_{\boldsymbol{\theta}}(\mathrm{y}_n | \mathrm{y}_{1..n-1}),$$

where $\mathrm{y}_{1..n}$ stands for $\{\mathrm{y}_i\}_{i=1}^{n}$ for any n. Regarding model (1), we have

$$p_{\boldsymbol{\theta}}(\mathrm{y}_n | \mathrm{y}_{1..n-1}) = \int h_{\boldsymbol{\theta}}(\mathrm{y}_n, \mathrm{l}_n)\, p_{\boldsymbol{\theta}}(\mathrm{l}_n | \mathrm{y}_{1..n-1})\, \mathrm{dl}_n, \tag{2}$$

with $h_{\boldsymbol{\theta}}(\mathrm{y}_n, \mathrm{l}_n) = p_{\boldsymbol{\theta}}(\mathrm{y}_n \,|\mathrm{l}_n\,)$, or equivalently

$$h_{\boldsymbol{\theta}}(\mathrm{y}_n, \mathrm{l}_n) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{\mathrm{l}_n}{2} - \frac{\mu}{2} - \frac{\mathrm{y}_n^2}{2}\exp(-\mathrm{l}_n - \mu)\right). \tag{3}$$

The exact predictive distribution $p_{\boldsymbol{\theta}}(\mathrm{l}_n \,|\mathrm{y}_{1..n-1}\,)$ is unavailable in A-ARSV(p) model *cf.* [9], therefore the exact value of $p_{\boldsymbol{\theta}}(\mathrm{y}_n \,|\mathrm{y}_{1..n-1}\,)$ is not computable for a given $\boldsymbol{\theta}$, neither is the exact value of $f(\boldsymbol{\theta})$.

The general quasi-maximum likelihood (QML) method consists in approximating (2), which results in approximating $f(\boldsymbol{\theta})$. We consider the following construction:

1. Compute

$$\widehat{\mathrm{l}}_{n|n-1}(\boldsymbol{\theta}) = \mathbb{E}_{\boldsymbol{\theta}}\left[\mathrm{l}_n \,|\mathrm{y}_{1..n-1}\,\right]; \tag{4a}$$

$$\widehat{\nu}^2_{n|n-1}(\boldsymbol{\theta}) = \mathbb{E}_{\boldsymbol{\theta}}\left[(\mathrm{l}_n - \widehat{\mathrm{l}}_{n|n-1}(\boldsymbol{\theta}))^2 \,|\mathrm{y}_{1..n-1}\,\right] \tag{4b}$$

   with an ADF *cf.* Section 3;

2. Let $\widehat{p}_{\boldsymbol{\theta}}(\mathrm{l}_n \,|\mathrm{y}_{1..n-1}\,) = \mathscr{N}\left(\mathrm{l}_n; \widehat{\mathrm{l}}_{n|n-1}(\boldsymbol{\theta}), \widehat{\nu}^2_{n|n-1}(\boldsymbol{\theta})\right)$, *i.e.* $\widehat{p}_{\boldsymbol{\theta}}(\mathrm{l}_n \,|\mathrm{y}_{1..n-1}\,)$ is a Gaussian distribution with mean $\widehat{\mathrm{l}}_{n|n-1}(\boldsymbol{\theta})$ and variance $\widehat{\nu}^2_{n|n-1}(\boldsymbol{\theta})$.

3. Use Gaussian quadrature to approximate $p_{\boldsymbol{\theta}}(\mathrm{y}_n \,|\mathrm{y}_{1..n-1}\,)$ by

$$\widehat{p}_{\boldsymbol{\theta}}(\mathrm{y}_n \,|\mathrm{y}_{1..n-1}\,) = \int h_{\boldsymbol{\theta}}(\mathrm{y}_n, \mathrm{l}_n)\, \widehat{p}_{\boldsymbol{\theta}}(\mathrm{l}_n \,|\mathrm{y}_{1..n-1}\,)\, \mathrm{dl}_n \approx \sum_{i=1}^{M} h_{\boldsymbol{\theta}}\left(\mathrm{y}_n, \widehat{\mathrm{l}}_{n|n-1}(\boldsymbol{\theta}) + \widehat{\nu}_{n|n-1}(\boldsymbol{\theta})\xi_i\right)\pi_i. \tag{5}$$

Let us recall that the Gaussian quadrature is a numeric integration technique. The points $\{\xi_i\}_{i=1}^M$ in $\mathbb{R}$ and their corresponding weight factors are tabulated *e.g.* in [10] for different values of M.

Finally, the quasi-likelihood function that we define is

$$\widehat{f}(\boldsymbol{\theta}) = \sum_{n=1}^{N} \log\left(\sum_{i=1}^{M} h_{\boldsymbol{\theta}}\left(\mathrm{y}_n, \widehat{\mathrm{l}}_{n|n-1}(\boldsymbol{\theta}) + \widehat{\nu}_{n|n-1}(\boldsymbol{\theta})\xi_i\right)\pi_i\right). \tag{6}$$

We obtain the related QML parameter estimate by solving numerically

$$\widehat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} \widehat{f}(\boldsymbol{\theta}) \tag{7}$$

with an implementation of the Broyden-Fletcher-Goldfarb-Shanno optimization routine described in [7].

3. ASSUMED-DENSITY FILTERING

In this section, we detail the computation of (4) with the ADF. Since $\{\mathrm{L}_n\}_{n\geq 1}$ is not Markovian, we begin by reformulating the system (1) as a classic HMM. Let us consider the following state vector $\mathbf{X}_n$ in $\mathbb{R}^{p+1}$:

$$\mathbf{X}_n = \begin{bmatrix}\mathrm{L}_{(n+1)-p+1} & \mathrm{L}_{(n+1)-p+2} & \dots & \mathrm{L}_{n+1} & \mathrm{W}_n\end{bmatrix}^{\top}. \tag{8}$$

By the stationarity assumption of $\{\mathrm{L}_n\}_{n\geq 1}$ and since $\{\mathrm{W}_n\}_{n\geq 1}$ is a white noise, we see that $\{\mathbf{X}_n\}_{n\geq 1}$ is a stationary AR (1) process:

$$\mathbf{X}_{n+1} = \boldsymbol{A}\mathbf{X}_n + \boldsymbol{B}\mathbf{U}_n; \tag{9a}$$

$$\boldsymbol{A} = \begin{bmatrix} 0 & & & & \\ \vdots & & \boldsymbol{I}_{p-1} & & \\ 0 & & & & \mathbf{0}_{p+1} \\ \phi_p & \phi_{p-1} & \dots & \phi_1 & \\ 0 & 0 & \dots & 0 & \end{bmatrix}; \tag{9b}$$

$$\boldsymbol{B}\boldsymbol{B}^{\top} = \begin{bmatrix} 0 & \dots & 0 & 0 \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \dots & \psi^2 + \sigma^2 & \psi \\ 0 & \dots & \psi & 1 \end{bmatrix}, \tag{9c}$$

where $\{\mathbf{U}_n\}_{n\geq 1}$ is standard Gaussian white noise in $\mathbb{R}^{p+1}$. Therefore, the probability density function (pdf) of $\mathbf{X}_n$ is zero-mean Gaussian. Let $\boldsymbol{\Gamma}$ be the variance of $\mathbf{X}_n$, then

$$\boldsymbol{A}\boldsymbol{\Gamma}\boldsymbol{A}^{\top} + \boldsymbol{B}\boldsymbol{B}^{\top} = \boldsymbol{\Gamma}. \tag{10}$$

In practice, one considers the following recursion to solve (10) in $\boldsymbol{\Gamma}$:

$$\boldsymbol{\Gamma}(k+1) = \boldsymbol{A}\boldsymbol{\Gamma}(k)\boldsymbol{A}^{\top} + \boldsymbol{B}\boldsymbol{B}^{\top}, \tag{11}$$

where $\boldsymbol{\Gamma}(1)$ is the identity matrix. If all the absolute eigenvalues of $\boldsymbol{A}$ are strictly smaller than 1 then the stationary solution exists and we have

$$\lim_{k\to\infty} \boldsymbol{\Gamma}(k) = \boldsymbol{\Gamma}.$$

Our filtering algorithm implements the assumed-density principle. The underlying idea is to assume that for each n, the filtering pdf $p_{\boldsymbol{\theta}}(\mathbf{x}_n \,|\mathrm{y}_{1..n}\,)$ and the one step-ahead predictive pdf $p_{\boldsymbol{\theta}}(\mathbf{x}_n \,|\mathrm{y}_{1..n-1}\,)$ have a pre-specified form. We approximate the filtering and predictive densities by Gaussian distributions:

$$\widehat{p}_{\boldsymbol{\theta}}(\mathbf{x}_n \,|\mathrm{y}_{1..n}\,) = \mathscr{N}\left(\mathbf{x}_n; \widehat{\mathbf{x}}_{n|n}(\boldsymbol{\theta}), \widehat{\boldsymbol{\Gamma}}_{n|n}(\boldsymbol{\theta})\right);$$

$$\widehat{p}_{\boldsymbol{\theta}}(\mathbf{x}_n \,|\mathrm{y}_{1..n-1}\,) = \mathscr{N}\left(\mathbf{x}_n; \widehat{\mathbf{x}}_{n|n-1}(\boldsymbol{\theta}), \widehat{\boldsymbol{\Gamma}}_{n|n-1}(\boldsymbol{\theta})\right).$$

ADF performs prediction and update steps for each n.

In the prediction step, one takes the expectation of (9a) given the information available at $n-1$ to obtain the parameters of the predictive pdf:

$$\widehat{\mathbf{x}}_{n|n-1}(\boldsymbol{\theta}) = \boldsymbol{A}\widehat{\mathbf{x}}_{n-1|n-1}(\boldsymbol{\theta}); \tag{12a}$$

$$\widehat{\boldsymbol{\Gamma}}_{n|n-1}(\boldsymbol{\theta}) = \boldsymbol{A}\widehat{\boldsymbol{\Gamma}}_{n-1|n-1}(\boldsymbol{\theta})\boldsymbol{A}^{\top} + \boldsymbol{B}\boldsymbol{B}^{\top}. \tag{12b}$$

The updating step incorporates the n-th measurement y_n into the predictive pdf to obtain the filtering pdf at time n. The parameters of this updated pdf derive from the general Bayesian framework:

$$\widehat{\mathbf{x}}_{n|n}(\boldsymbol{\theta}) = \int \mathbf{x}_n \omega(\mathbf{x}_n, \boldsymbol{\theta}) \mathrm{d}\mathbf{x}_n; \tag{13a}$$

$$\widehat{\boldsymbol{\Gamma}}_{n|n}(\boldsymbol{\theta}) = \int \mathbf{x}_n \mathbf{x}_n^\top \omega(\mathbf{x}_n, \boldsymbol{\theta}) \mathrm{d}\mathbf{x}_n - \widehat{\mathbf{x}}_{n|n}(\boldsymbol{\theta}) \widehat{\mathbf{x}}_{n|n}^\top(\boldsymbol{\theta}); \tag{13b}$$

$$\omega(\mathbf{x}_n, \boldsymbol{\theta}) = \frac{p_{\boldsymbol{\theta}}\left(\mathrm{y}_n \,|\mathbf{x}_n\right) \widehat{p}_{\boldsymbol{\theta}}\left(\mathbf{x}_n \,|\mathrm{y}_{1..n-1}\right)}{d_n(\boldsymbol{\theta})}; \tag{13c}$$

$$d_n(\boldsymbol{\theta}) = \int p_{\boldsymbol{\theta}}\left(\mathrm{y}_n \,|\mathbf{x}_n\right) \widehat{p}_{\boldsymbol{\theta}}\left(\mathbf{x}_n \,|\mathrm{y}_{1..n-1}\right) \mathrm{d}\mathbf{x}_n. \tag{13d}$$

We see that the formulas above require an integration over $\mathbb{R}^{p+1}$. However, we remark that the context of the A-ARSV(p) model allows replacing this multidimensional integration by an integration over $\mathbb{R}$. In fact, let us consider the following partitioning of $\mathbf{X}_n$ into $\mathbf{R}_n$ and $\mathbf{Z}_n$:

$$\mathbf{R}_n = \begin{bmatrix} \mathrm{L}_{(n+1)-p+1} & \dots & \mathrm{L}_{(n+1)-2} & \mathrm{L}_{n+1} \end{bmatrix}^\top \tag{14a}$$

$$\mathbf{Z}_n = \begin{bmatrix} \mathrm{L}_n & \mathrm{W}_n \end{bmatrix}^\top. \tag{14b}$$

We suggest the following reasoning given the system state at $n-1$. Y_n is deterministic given $\mathbf{Z}_n$ *cf.* (1b), therefore $\mathbf{R}_n$ and Y_n are independent given $\mathbf{Z}_n$. We can use this to begin by updating the mean $\widehat{\mathbf{z}}_{n|n}(\boldsymbol{\theta})$ and variance $\widehat{\boldsymbol{\Sigma}}_{n|n}(\boldsymbol{\theta})$ parameters of $\mathbf{Z}_n$, then to derive the mean and variance parameters of $\mathbf{X}_n$ by appropriate linear and quadratic transformations, since $\mathbf{X}_n$ is assumed to be Gaussian. Moreover, there is a deterministic link between W_n, Y_n and L_n *cf.* (1b), so we can compute the parameters of $\mathbf{Z}_n$ given Y_n by taking the expectation of an appropriate function of L_n. The following original algorithm implements this idea:

Algorithm 1 ***The updating step***

1. *Extract* $\widehat{\mathbf{z}}_{n|n-1}(\boldsymbol{\theta})$, $\widehat{\boldsymbol{\Sigma}}_{n|n-1}(\boldsymbol{\theta})$ *and* $\widehat{\boldsymbol{C}}_{n|n-1}(\boldsymbol{\theta})$ *from* $\widehat{\mathbf{x}}_{n|n-1}(\boldsymbol{\theta})$ *and* $\widehat{\boldsymbol{\Gamma}}_{n|n-1}(\boldsymbol{\theta})$ cf. (14)*, such that*

$$\widehat{p}_{\boldsymbol{\theta}}\left(\mathbf{z}_n \,|\mathrm{y}_{1..n-1}\right) = \mathcal{N}\left(\mathbf{z}_n; \widehat{\mathbf{z}}_{n|n-1}(\boldsymbol{\theta}), \widehat{\boldsymbol{\Sigma}}_{n|n-1}(\boldsymbol{\theta})\right);$$

$$\widehat{\boldsymbol{C}}_{n|n-1}(\boldsymbol{\theta}) = \mathrm{Cov}_{\boldsymbol{\theta}}\left(\mathbf{X}_n, \mathbf{Z}_n \,|\mathrm{y}_{1..n-1}\right).$$

2. *Use the classic Gaussian conditioning formulas to derive the coefficients of linear and quadratic transformations corresponding to partitioning* (14)*:*

$$\widehat{\boldsymbol{\alpha}}_n(\boldsymbol{\theta}) = \widehat{\boldsymbol{C}}_{n|n-1}(\boldsymbol{\theta}) \widehat{\boldsymbol{\Sigma}}_{n|n-1}^{-1}(\boldsymbol{\theta}); \tag{16a}$$

$$\widehat{\boldsymbol{\beta}}_n(\boldsymbol{\theta}) = \widehat{\mathbf{x}}_{n|n-1}(\boldsymbol{\theta}) - \widehat{\boldsymbol{\alpha}}_n(\boldsymbol{\theta}) \widehat{\mathbf{z}}_{n|n-1}(\boldsymbol{\theta}); \tag{16b}$$

$$\widehat{\boldsymbol{\gamma}}_n(\boldsymbol{\theta}) = \widehat{\boldsymbol{\Gamma}}_{n|n-1}(\boldsymbol{\theta}) - \widehat{\boldsymbol{\alpha}}_n(\boldsymbol{\theta}) \widehat{\boldsymbol{C}}_{n|n-1}^\top(\boldsymbol{\theta}). \tag{16c}$$

3. *Compute* $\widehat{\mathbf{z}}_{n|n}(\boldsymbol{\theta})$ *and* $\widehat{\boldsymbol{\Sigma}}_{n|n}(\boldsymbol{\theta})$ *by using the Gaussian quadrature:*

$$\widehat{\mathbf{z}}_{n|n}(\boldsymbol{\theta}) = \sum_{i=1}^{M} \boldsymbol{g}_{\boldsymbol{\theta}}\left(\mathrm{y}_n, \widehat{\mathrm{l}}_{n|n-1}(\boldsymbol{\theta}) + \widehat{\nu}_{n|n-1}(\boldsymbol{\theta})\xi_i\right) \omega(\pi_i, \boldsymbol{\theta});$$

$$\widehat{\boldsymbol{\Sigma}}_{n|n}(\boldsymbol{\theta}) = \sum_{i=1}^{M} \Big(\boldsymbol{g}_{\boldsymbol{\theta}}\left(\mathrm{y}_n, \widehat{\mathrm{l}}_{n|n-1}(\boldsymbol{\theta}) + \widehat{\nu}_{n|n-1}(\boldsymbol{\theta})\xi_i\right) \times \boldsymbol{g}_{\boldsymbol{\theta}}^\top\left(\mathrm{y}_n, \widehat{\mathrm{l}}_{n|n-1}(\boldsymbol{\theta}) + \widehat{\nu}_{n|n-1}(\boldsymbol{\theta})\xi_i\right) \omega(\pi_i, \boldsymbol{\theta}) \Big) - \widehat{\mathbf{z}}_{n|n}(\boldsymbol{\theta}) \widehat{\mathbf{z}}_{n|n}^\top(\boldsymbol{\theta}),$$

where

$$\omega(\pi_i, \boldsymbol{\theta}) = \frac{h_{\boldsymbol{\theta}}\left(\mathrm{y}_n, \widehat{\mathrm{l}}_{n|n-1}(\boldsymbol{\theta}) + \widehat{\nu}_{n|n-1}(\boldsymbol{\theta})\xi_i\right) \pi_i}{d_n(\boldsymbol{\theta})};$$

$$d_n(\boldsymbol{\theta}) = \sum_{i=1}^{M} h_{\boldsymbol{\theta}}\left(\mathrm{y}_n, \widehat{\mathrm{l}}_{n|n-1}(\boldsymbol{\theta}) + \widehat{\nu}_{n|n-1}(\boldsymbol{\theta})\xi_i\right) \pi_i,$$

with

$$\widehat{\mathrm{l}}_{n|n-1}(\boldsymbol{\theta}) = \widehat{\mathbf{x}}_{n-1|n-1}(\boldsymbol{\theta})[p]; \tag{19a}$$

$$\widehat{\nu}_{n|n-1}^2(\boldsymbol{\theta}) = \widehat{\boldsymbol{\Gamma}}_{n-1|n-1}(\boldsymbol{\theta})[p, p]; \tag{19b}$$

$$\boldsymbol{g}_{\boldsymbol{\theta}}\left(\mathrm{y}_n, \mathrm{l}_n\right) = \begin{bmatrix} \mathrm{l}_n & \mathrm{y}_n \exp\left(-\frac{\mathrm{l}_n}{2} - \frac{\mu}{2}\right) \end{bmatrix}^\top. \tag{19c}$$

4. *The values of* $\widehat{\mathbf{x}}_{n|n}(\boldsymbol{\theta})$ *and* $\widehat{\boldsymbol{\Gamma}}_{n|n}(\boldsymbol{\theta})$ *are:*

$$\widehat{\mathbf{x}}_{n|n}(\boldsymbol{\theta}) = \widehat{\boldsymbol{\alpha}}_n(\boldsymbol{\theta}) \widehat{\mathbf{z}}_{n|n}(\boldsymbol{\theta}) + \widehat{\boldsymbol{\beta}}_n(\boldsymbol{\theta}); \tag{20a}$$

$$\widehat{\boldsymbol{\Gamma}}_{n|n}(\boldsymbol{\theta}) = \widehat{\boldsymbol{\alpha}}_n(\boldsymbol{\theta}) \widehat{\boldsymbol{\Sigma}}_{n|n}(\boldsymbol{\theta}) \widehat{\boldsymbol{\alpha}}_n^\top(\boldsymbol{\theta}) + \widehat{\boldsymbol{\gamma}}_n(\boldsymbol{\theta}). \tag{20b}$$

The overall algorithm for evaluating our quasi-likelihood function at a parameter vector value $\boldsymbol{\theta}$ runs as follows.

Algorithm 2 ***The quasi-likelihood function***

```
1. Initialize $\widehat{\mathbf{x}}_{0|0}(\boldsymbol{\theta}) = \mathbf{0}$, $\widehat{\boldsymbol{\Gamma}}_{0|0}(\boldsymbol{\theta}) = \boldsymbol{\Gamma}$;
2. For each $n$ in $1..N$,
   -Perform the prediction step (12);
   -Obtain $\widehat{\mathrm{l}}_{n|n-1}(\boldsymbol{\theta})$ and $\widehat{\nu}_{n|n-1}^2(\boldsymbol{\theta})$ from (19);
   -Run the updating step cf. Algorithm 1.
3. Evaluate $\widehat{f}(\boldsymbol{\theta})$ by (6).
```

4. EXPERIMENTS

In this section, we first provide an experiment on synthetic data to validate our approach. Next, we illustrate the applicability of our method on an example of a real-world data set.

	0.0	-0.4	-0.7	-0.9
GARCH	1.31	1.30	1.32	1.30
A-ARSV(1)	0.87	0.73	0.72	0.72
ARSV(7)	0.70	0.70	0.70	0.71
A-ARSV(7)	0.71	0.61	0.49	0.40
A-ARSV(7)*	0.70	0.59	0.49	0.40

Table 1. MSE of different estimators in a synthetic setting (lower is better), for varying values of ρ. The results for A-ARSV(7)* are given as a reference.

4.1. Experiments on synthetic data

We consider the A-ARSV(7) model with the parameter

$$\boldsymbol{\theta}_0 = (1.73, -1.55, 1.32, -1.31, 1.6, -1.62, 0.77, \\ 0.3\rho, 0.3\sqrt{1-\rho^2}, 0). \tag{21}$$

To choose the values of auto-regression coefficients in (21), we have preliminary estimated several A-ARSV(p) models from the public price histories of different stock indexes. The parameter ρ designs the degree of the asymmetry of the model and takes different values according to the test case *cf.* Table 1.

We generate a data set $\{\mathrm{y}_n\}_{n=1}^{1000}$, then we estimate the hidden state sequence $\{\mathrm{l}_n\}_{n=1}^{1000}$ by the following estimators:

• "GARCH" estimator computes the coefficients of the GARCH(7,1) model *cf.* [11] from $\{\mathrm{y}_n\}_{n=1}^{1000}$, then estimates the log-volatility;

• "A-ARSV(1)" estimator implements the algorithm from [4] to estimate the A-ARSV(1) model, then uses the sequential importance sampling particle filter (SIR-PF, *see*, *e.g.*, [12]) to estimate the underlying log-volatility;

• "ARSV(7)" estimator uses our unsupervised algorithm to estimate the ARSV(7) parameters, then uses the SIR-PF.

Next, consider two estimators based on the A-ARSV(7) model:

• "A-ARSV(7)" estimator uses our unsupervised algorithm to estimate the A-ARSV(7) parameters, then uses the SIR-PF;

• "A-ARSV(7)*" uses the true parameter vector $\boldsymbol{\theta}_0$ to estimate the underlying log-volatility with the SIR-PF.

Table 1 summarizes the averaged results of these methods in terms of the mean square error (MSE, MSE $= \frac{1}{1000}\sum_{n=1}^{1000}(\mathrm{l}_n - \widehat{\mathrm{l}}_n)^2$).

Regarding the processing time, our algorithm runs approximately 5 times faster than its MCMC counterpart, but one may obtain different running times depending on the CPU, operating system, compilation and implementation details. It is not surprising that the A-ARSV(7) and A-ARSV(7)* estimators were the most accurate, thanks to their consistency with the data-generating process. However, we see that the classic estimators may be particularly suboptimal.

4.2. Experiments on stock data

Being able to determine which model, A-ARSV(p) or A-ARSV(p), and the order p that describes sufficiently well a given data is important for the practitioners. The common way to do it consists in using the BIC criterion [13]. The model with the lowest BIC would be the best at forecasting, especially in a long-sample context [14]. The BIC is defined as

$$\mathrm{BIC} = -2\log p(\mathrm{y}_{1..N}|\widehat{\boldsymbol{\theta}}, \mathscr{M}) + k\log N.$$

Here, $p(\mathrm{y}_{1..N}|\widehat{\boldsymbol{\theta}}, \mathscr{M})$ denotes the maximized value of the likelihood function in the model $\mathscr{M}$ and k is the number of free parameters. In the case of the ARSV(p) model, $k = p+2$ and $k = p+3$ in the case of the A-ARSV(p) model.

We use our method to estimate the A-ARSV(p) and ARSV(p) models from the price histories of iShares Emerging Markets (EEM) index observed in-between 01/01/2010 and 04/01/2016 (1512 observations). Figure 1 plots the BIC values that we find after estimating ARSV(p) and A-ARSV(p) models on this data for various p.

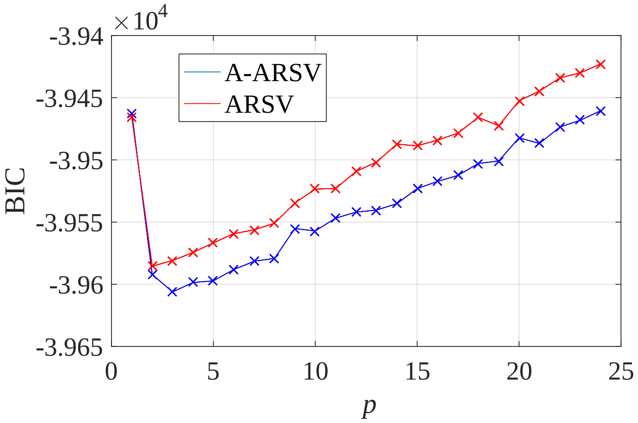

Fig. 1. BIC values of the ARSV(p), A-ARSV(p) models estimated from the EEM index (lower is better). For $p = 1$, the ARSV(p) model is preferable to A-ARSV(p); the A-ARSV(p) model is preferable to ARSV(p) for $p \geq 2$.

5. CONCLUSION

We proposed a general method of parameter estimation in the latent asymmetric autoregressive models. The solution results from maximizing a specific approximation of the likelihood function. This method is simple and robust. The order of the underlying AR process may be chosen by using the BIC criterion. Besides, our method may be easily generalized to apply in various fields involving the latent autoregressive models including *e.g.* the speech synthesis [15] and tracking [16].

6. REFERENCES

[1] M. Centedo and R. Salido, "Estimation of asymmetric stochastic volatility models for stock-exchange index returns," *International advances in economic research*, vol. 15, no. 3, pp. 70–87, 2003.

[2] A. Poritz, "Hidden Markov models: A guided tour," in *International Conference on Acoustics, Speech, and Signal Processing ICASSP-88*. IEEE, 1988, pp. 7–13.

[3] S. Kim, N. Shephard, and S. Chib, "Stochastic volatility: likelihood inference and comparison with ARCH models," *The Review of Economic Studies*, vol. 65, no. 3, pp. 361–393, 1998.

[4] A. Harvey and N. Shephard, "Estimation of an asymmetric stochastic volatility model for asset returns," *Journal of Business & Economic Statistics*, vol. 14, no. 4, pp. 429–434, 1996.

[5] T. Minka, *A family of algorithms for approximate Bayesian inference*, Ph.D. thesis, Massachusetts Institute of Technology, 2001.

[6] K. Ito and K. Xiong, "Gaussian filters for nonlinear filtering problems," *IEEE Transactions on Automatic Control*, vol. 45, no. 5, pp. 910–927, 2000.

[7] C. Zhu, R. Byrd, P. Lu, and al., "Algorithm 778: L-BFGS-B: Fortran subroutines for large-scale bound-constrained optimization," *ACM Transactions on Mathematical Software (TOMS)*, vol. 23, no. 4, pp. 550–560, 1997.

[8] E. Crouch and D. Spiegelman, "The evaluation of integrals of the form $\int f(t)\exp(-t^2)dt$: Application to logistic-normal models," *Journal of the American Statistical Association*, vol. 85, no. 410, pp. 464–469, 1990.

[9] T. Bollerslev and H. Zhou, "Estimating stochastic volatility diffusion using conditional moments of integrated volatility," *Journal of Econometrics*, vol. 109, no. 1, pp. 33–65, 2002.

[10] H. Salzer, H. Zucker, and R. Capuano, *Table of the zeros and weight factors of the first twenty Hermite polynomials*, US Government Printing Office, 1952.

[11] T. Bollerslev, "Generalized autoregressive conditional heteroskedasticity," *Journal of Econometrics*, vol. 31, no. 3, pp. 307–327, 1986.

[12] M. Arulampalam, S. Maskell, N. Gordon, and al., "A tutorial on particle filters for online nonlinear/non-Gaussian bayesian tracking," *IEEE Transactions on signal processing*, vol. 50, no. 2, pp. 174–188, 2002.

[13] A. Mcquarrie and C. Tsai, *Regression and time series model selection*, World Scientific, 1998.

[14] C. Medel, S. Salgado, and al., *Does BIC estimate and forecast better than AIC?*, Banco Central de Chile, 2012.

[15] Carl Quillen, "Autoregressive HMM speech synthesis," in *International Conference on Acoustics, Speech, and Signal Processing ICASSP-12*. IEEE, 2012, pp. 4021–4024.

[16] Shaobo Hou, Aphrodite Galata, Fabrice Caillette, Neil Thacker, and Paul Bromiley, "Real-time body tracking using a Gaussian process latent variable model," in *2007 IEEE 11th International Conference on Computer Vision*. IEEE, 2007, pp. 1–8.