UNSUPervised Learning of Asymmetrical HigH-ORDER AToRegressive STOChastic Volatility Model

Ivan Gorynin, Emmanuel Monfrini and Wojciech Pieczynski

SAMOVAR, Telecom SudParis, CNRS, Université 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:

\[
L_{n+1} = \sum_{j=1}^{p} \phi_j L_{n+1-j} + \psi W_n + \sigma V_{n+1}; \quad (1a)
\]

\[
Y_n = W_n \exp \left( \frac{L_n}{2} + \frac{\mu}{2} \right), \quad (1b)
\]

where \( \{ V_n \}_{n \geq 1}, \{ W_n \}_{n \geq 1} \) are standard Gaussian white noises. The parameter vector \( \theta = (\phi_1, \ldots, \phi_p, \psi, \sigma, \mu) \) is fixed, unknown and is to be estimated. The equation (1a) describes an autoregressive model assumed to be stationary (see Section 3 for a stationarity condition). The process \( \{ L_n \}_{n \geq 1} \) is hidden while \( \{ Y_n \}_{n \geq 1} \) is observable, as in the hidden Markov model framework (HMM, see, e.g., [2]). Moreover, \( \{ L_m \}_{m \geq n} \) 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 A-ARSV$(p)$). Unsupervised learning of A-ARSV$(p)$ model consists in estimating of \( \theta \) from \( \{ y_n \}_{n=1}^{N} \).

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 \( \{ y_n \}_{n=1}^{N} \).

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 indices 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 \( \theta \) that would reproduce the observed time series \( \{ y_n \}_{n=1}^{N} \). In other words, it maximizes the following function:

\[
f(\theta) = \log p_{\theta} (y_{1..N}) = \sum_{n=1}^{N} \log p_{\theta} (y_n | y_{1..n-1}),
\]

where \( y_{1..n} \) stands for \( \{ y_i \}_{i=1}^{n} \) for any \( n \). Regarding model (1), we have

\[
p_{\theta} (y_n | y_{1..n-1}) = \int h_{\theta} (y_n, l_n) p_{\theta} (l_n | y_{1..n-1}) \, dl_n, \quad (2)
\]
with \( h_\theta (y_n, l_n) = p_\theta (y_n | l_n) \), or equivalently

\[
h_\theta (y_n, l_n) = \frac{1}{\sqrt{2\pi}} \exp \left( -\frac{l_n}{2} - \frac{\mu}{2} - \frac{y_n^2}{2} \exp(-l_n - \mu) \right). \tag{3}\]

The exact predictive distribution \( p_\theta (l_n | y_{1:n-1}) \) is unavailable in a-ARSV\( (p) \) model cf. [9], therefore the exact value of \( p_\theta (y_n | y_{1:n-1}) \) is not computable for a given \( \theta \), neither is the exact value of \( f(\theta) \).

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

1. Compute

\[
\hat{I}_{n|n-1} (\theta) = \mathbb{E}_\theta [l_n | y_{1:n-1}]; \tag{4a}\]
\[
\hat{\sigma}^2_{n|n-1} (\theta) = \mathbb{E}_\theta [(l_n - \hat{I}_{n|n-1} (\theta))^2 | y_{1:n-1}] \tag{4b}\]

with an ADF cf. Section 3;

2. Let \( \tilde{p}_\theta (l_n | y_{1:n-1}) = \mathcal{N} (l_n; \hat{I}_{n|n-1} (\theta), \hat{\sigma}^2_{n|n-1} (\theta)) \), i.e. \( \tilde{p}_\theta (l_n | y_{1:n-1}) \) is a Gaussian distribution with mean \( \hat{I}_{n|n-1} (\theta) \) and variance \( \hat{\sigma}^2_{n|n-1} (\theta) \).

3. Use Gaussian quadrature to approximate \( p_\theta (y_n | y_{1:n-1}) \)

\[
\tilde{p}_\theta (y_n | y_{1:n-1}) = \int h_\theta (y_n, l_n) \tilde{p}_\theta (l_n | y_{1:n-1}) \, dl_n \approx \sum_{i=1}^M h_\theta (y_n, \hat{I}_{n|n-1} (\theta) + \tilde{\sigma}_{n|n-1} (\theta) \xi_i) \pi_i. \tag{5}\]

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

\[
\hat{f}(\theta) = \sum_{n=1}^N \log \left( \sum_{i=1}^M h_\theta (y_n, \hat{I}_{n|n-1} (\theta) + \tilde{\sigma}_{n|n-1} (\theta) \xi_i) \pi_i \right). \tag{6}\]

We obtain the related QML parameter estimate by solving numerically

\[
\hat{\theta} = \arg \max_\theta \hat{f}(\theta). \tag{7}\]

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 \( \{L_n\}_{n \geq 1} \) is not Markovian, we begin by reformulating the system (1) as a classic HMM. Let us consider the following state vector \( X_n \) in \( \mathbb{R}^{p+1} \):

\[
X_n = \left[ \begin{array}{c} L_{n+1} - \nu + 1 \vspace{1mm} \cdots \vspace{1mm} L_{n+1} - \nu + 1 \vspace{1mm} \cdots \vspace{1mm} \cdots \vspace{1mm} \cdots \vspace{1mm} W_{n} \end{array} \right]^T. \tag{8}\]

By the stationarity assumption of \( \{L_n\}_{n \geq 1} \) and since \( \{W_n\}_{n \geq 1} \) is a white noise, we see that \( \{X_n\}_{n \geq 1} \) is a stationary AR (1) process:

\[
X_{n+1} = AX_n + BU_n; \tag{9a}\]

\[
A = \begin{bmatrix} 0 & I_{p-1} & 0_{p+1} \\
0 & 0 & 0 \end{bmatrix}; \tag{9b}\]

\[
BB^T = \begin{bmatrix} 0 & \ldots & 0 \\
\vdots & \ddots & \vdots \\
0 & \ldots & \psi^2 + \sigma^2 & \psi \end{bmatrix}, \tag{9c}\]

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

\[
A\Gamma A^T + BB^T = \Gamma. \tag{10}\]

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

\[
\Gamma(k+1) = A\Gamma(k)A^T + BB^T, \tag{11}\]

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

\[
\lim_{k \to \infty} \Gamma(k) = \Gamma. \tag{12}\]

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

\[
\hat{p}_\theta (x_n | y_{1:n}) = \mathcal{N} (x_n; \hat{x}_{n|n}(\theta), \hat{\Gamma}_{n|n}(\theta)); \tag{12a}\]

\[
\tilde{p}_\theta (x_n | y_{1:n-1}) = \mathcal{N} (x_n; \tilde{x}_{n|n-1}(\theta), \tilde{\Gamma}_{n|n-1}(\theta)). \tag{12b}\]
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:

\[
\hat{x}_{n|n}(\theta) = \int x_n \omega(x_n, \theta) dx_n; \quad (13a)
\]

\[
\hat{\Gamma}_{n|n}(\theta) = \int x_n x_n^\top \omega(x_n, \theta) dx_n - \hat{x}_{n|n}(\theta) \hat{x}_{n|n}^\top(\theta); \quad (13b)
\]

\[
\omega(x_n, \theta) = \frac{\varphi_{\theta}(y_n | x_n) \hat{p}_{\theta}(x_n | y_{1:n-1})}{d_n(\theta)}; \quad (13c)
\]

\[
d_n(\theta) = \int \varphi_{\theta}(y_n | x_n) \hat{p}_{\theta}(x_n | y_{1:n-1}) dx_n. \quad (13d)
\]

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 $X_n$ into $R_n$ and $Z_n$:

\[
R_n = [L_{(n+1) - p+1} \ldots L_{(n+1) - 2} L_{n+1}]^\top; \quad (14a)
\]

\[
Z_n = [I_n W_n]^\top. \quad (14b)
\]

We suggest the following reasoning given the system state at $n - 1$. $Y_n$ is deterministic given $Z_n$, cf. (1b), therefore $R_n$ and $Y_n$ are independent given $Z_n$. We can use this to begin by updating the mean $\hat{x}_{n|n-1}(\theta)$ and variance $\hat{\Sigma}_{n|n-1}(\theta)$ parameters of $Z_n$, then to derive the mean and variance parameters of $X_n$ by appropriate linear and quadratic transformations, since $X_n$ 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 $Z_n$ 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 $\hat{x}_{n|n-1}(\theta)$, $\hat{\Sigma}_{n|n-1}(\theta)$ and $\hat{\Gamma}_{n|n-1}(\theta)$ from $\hat{x}_{n|n-1}(\theta)$ and $\hat{\Gamma}_{n|n-1}(\theta)$ cf. (14), such that

\[
\hat{p}_{\theta}(z_n | y_{1:n-1}) = \mathcal{N}(z_n; \hat{x}_{n|n-1}(\theta), \hat{\Sigma}_{n|n-1}(\theta)); \quad (19a)
\]

\[
\hat{\Gamma}_{n|n-1}(\theta) = \text{Cov}_{\theta}(X_n, Z_n | y_{1:n-1}). \quad (19b)
\]

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

\[
\hat{\alpha}_{n}(\theta) = \hat{\Gamma}_{n|n-1}(\theta) \hat{\Sigma}_{n|n-1}^{-1}(\theta); \quad (16a)
\]

\[
\hat{\beta}_{n}(\theta) = \hat{x}_{n|n-1}(\theta) - \hat{\alpha}_{n}(\theta) \hat{x}_{n|n-1}(\theta); \quad (16b)
\]

\[
\hat{\gamma}_{n}(\theta) = \hat{\Gamma}_{n|n-1}(\theta) - \hat{\alpha}_{n}(\theta) \hat{\Gamma}_{n|n-1}^\top(\theta). \quad (16c)
\]

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

\[
\hat{z}_{n|n}(\theta) = \sum_{i=1}^{M} \varphi_{\theta}(y_n, \hat{z}_{n|n-1}(\theta) + \hat{\nu}_{n|n-1}(\theta) \xi_i) \omega(\pi_n, \theta);
\]

\[
\hat{\Sigma}_{n|n}(\theta) = \sum_{i=1}^{M} \left( \varphi_{\theta}(y_n, \hat{z}_{n|n-1}(\theta) + \hat{\nu}_{n|n-1}(\theta) \xi_i) \times \right)
\]

\[
g_{\theta}(y_n, \hat{z}_{n|n-1}(\theta) + \hat{\nu}_{n|n-1}(\theta) \xi_i) \omega(\pi_n, \theta) - \hat{z}_{n|n}(\theta) \hat{z}_{n|n}^\top(\theta), \quad (19c)
\]

where

\[
\omega(\pi_n, \theta) = \frac{h_{\theta}(y_n, \hat{z}_{n|n-1}(\theta) + \hat{\nu}_{n|n-1}(\theta) \xi_i) \pi_n}{d_n(\theta)};
\]

\[
d_n(\theta) = \sum_{i=1}^{M} h_{\theta}(y_n, \hat{z}_{n|n-1}(\theta) + \hat{\nu}_{n|n-1}(\theta) \xi_i) \pi_i,
\]

with

\[
\hat{\nu}_{n|n-1}(\theta) = \hat{x}_{n|n-1}(\theta)[p]; \quad (19a)
\]

\[
\hat{\nu}_{n|n-1}^2(\theta) = \hat{\Gamma}_{n|n-1}(\theta)[p, p]; \quad (19b)
\]

\[
g_{\theta}(y_n, l_n) = [l_n y_n \exp \left(-\frac{l_n}{2} - \frac{y_n^2}{2} \right)]^\top. \quad (19c)
\]

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

\[
\hat{x}_{n|n}(\theta) = \hat{\alpha}_{n}(\theta) \hat{x}_{n|n}(\theta) + \hat{\beta}_{n}(\theta); \quad (20a)
\]

\[
\hat{\Gamma}_{n|n}(\theta) = \hat{\alpha}_{n}(\theta) \hat{\Sigma}_{n|n}(\theta) \hat{\Sigma}_{n|n}(\theta) + \hat{\Gamma}_{n|n}(\theta). \quad (20b)
\]

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

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

1. Initialize $\hat{x}_{0|0}(\theta) = 0$, $\hat{\Gamma}_{0|0}(\theta) = \Gamma$;

2. For each $n$ in 1..$N$,
   - Perform the prediction step (12);
   - Obtain $\hat{\Gamma}_{n|n-1}(\theta)$ and $\hat{\nu}_{n|n-1}^2(\theta)$ from (19);
   - Run the updating step cf. Algorithm 1.

3. Evaluate $\hat{f}(\theta)$ 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.
4.1. Experiments on synthetic data

We consider the A-ARSV(7) model with the parameter
\[
\theta_0 = (1.73, -1.55, 1.32, -1.31, 1.6, -1.62, 0.77, 0.3, 0.3 \sqrt{1 - \rho^2}, 0).
\]

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 \(\rho\) designs the degree of the asymmetry of the model and takes different values according to the test case (see Table 1).

We generate a data set \(\{y_n\}_{n=1}^{1000}\), then we estimate the hidden state sequence \(\{l_n\}_{n=1}^{1000}\) by the following estimators:

- “GARCH” estimator computes the coefficients of the GARCH\( (7,1)\) model cf. [11] from \(\{y_n\}_{n=1}^{1000}\), 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 A-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 \(\hat{\theta}_0\) 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 = \(\frac{1}{1000} \sum_{n=1}^{1000} (l_n - \hat{l}_n)^2\)).

<table>
<thead>
<tr>
<th>Estimator</th>
<th>MSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>GARCH</td>
<td>1.31</td>
</tr>
<tr>
<td>A-ARSV((1))</td>
<td>0.87</td>
</tr>
<tr>
<td>ARSV((7))</td>
<td>0.70</td>
</tr>
<tr>
<td>A-ARSV((7))</td>
<td>0.71</td>
</tr>
<tr>
<td>A-ARSV((7)*)</td>
<td>0.70</td>
</tr>
</tbody>
</table>

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

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

\[
BIC = -2 \log p(y_{1..N}|\hat{\theta}, \mathcal{M}) + k \log N.
\]

Here, \(p(y_{1..N}|\hat{\theta}, \mathcal{M})\) denotes the maximized value of the likelihood function in the model \(\mathcal{M}\) 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 \(p\).

![BIC values of the ARSV\((p)\), A-ARSV\((p)\) models estimated from the EEM, A-ARSV\((p)\) model is preferable to ARSV\((p)\); the A-ARSV\((p)\) model is preferable to ARSV\((p)\) for \(p \geq 2\).](image)

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


