

## **FINAL REPORT**

**SAS-IIF Grant Investigator: David Ardia**

**Project** Bayesian Prediction of Market Risk using Regime-Switching GARCH Models

### **A. Specific Aims**

Our research aimed at: (1) developing an R package for estimating and forecasting with regime-switching GARCH models (RSGARCH), and (2) testing the predictive performance of RSGARCH models over a large universe of assets worldwide.

### **B. Progress**

The research project is finished. During the last two years, the research team has done the following:

#### **2015**

1. Data collection;
2. Literature review;
3. R and C++ code:
  - a. Single models (72 models) implementation;
  - b. ML and MCMC estimation;
  - c. Backtesting engine.
4. Redaction of a first research paper on GARCH combination forecasting:  
[https://papers.ssrn.com/sol3/papers.cfm?abstract\\_id=2688633](https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2688633)
5. Presentation at the IIF conference in Riverside.

#### **2016**

1. R and C++ code:
  - a. Regime-switching (MSGARCH) implementation;
  - b. MS and mixture models;
  - c. Performance tests implementation;
2. Setup and backtesting on clusters;
3. Finalization of the R package MSGARCH:  
<https://cran.r-project.org/web/packages/MSGARCH/index.html>
4. Redaction of vignette describing the package:  
[https://papers.ssrn.com/sol3/papers.cfm?abstract\\_id=2845809](https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2845809)
5. Redaction of a research paper on MSGARCH forecasting:  
[https://papers.ssrn.com/sol3/papers.cfm?abstract\\_id=2918413](https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2918413)
6. Disseminate package and results via CRAN, GitHub, and SSRN websites.

### **C. Plans for 2017**

1. Improve package and vignette (via Google summer of code 2017);
  2. Improve the research papers;
  3. Presentation at conferences;
  4. Submit vignette and research papers for publication;
-

5. Disseminate R package information via the IIF newsletter.

#### **D. Publications and presentations**

##### ***Working papers***

1. “Predicting Market Risk with Combinations of GARCH-Type Models” (with J. Kolly and D.-A. Trottier). *SSRN Working paper*.
2. “Markov-Switching GARCH Models in R: The MSGARCH Package” (with K. Bluteau, K. Boudt and D.-A. Trottier). *SSRN Working paper*.
3. “Forecasting Performance of Markov-switching GARCH models: A Large-Scale Empirical Study” (with K. Bluteau, K. Boudt and L. Catania). *SSRN Working paper*.

##### ***Presentation***

1. “Predicting Market Risk with Combinations of GARCH-Type Models” (presented by J. Kolly at IIF conference 2015 in Riverside).

#### **E. Other Delays or Difficulties**

I moved to the University of Neuchâtel, Switzerland, in January 2016. Obviously, this implied administrative burden for the first months. Moreover, the final team involved in the project has changed. This explains the 2-3 months delay for the project to be finalized, *i.e.*, February 2017 instead of December 2016.

# **MSGARCH paper**

[https://papers.ssrn.com/sol3/papers.cfm?abstract\\_id=2918413](https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2918413)

# Forecasting performance of Markov-switching GARCH models: A large-scale empirical study<sup>☆</sup>

David Ardia<sup>a,b,\*</sup>, Keven Bluteau<sup>a</sup>, Kris Boudt<sup>c,d,e</sup>, Leopoldo Catania<sup>f</sup>

<sup>a</sup>*Institute of Financial Analysis, University of Neuchâtel, Neuchâtel, Switzerland*

<sup>b</sup>*Department of Finance, Insurance and Real Estate, Laval University, Québec City, Canada*

<sup>c</sup>*Solvay Business School, Vrije Universiteit Brussel*

<sup>d</sup>*Faculty of Economics and Business, Vrije Universiteit Amsterdam*

<sup>e</sup>*Quantitative Strategies, Finver*

<sup>f</sup>*University of Rome, “Tor Vergata”*

## PRELIMINARY

---

### Abstract

We perform a large-scale empirical study to evaluate the forecasting performance of Markov-switching GARCH (MSGARCH) models compared with standard single-regime specifications. We find that the need for a Markov-switching mechanism in GARCH models depends on the underlying asset class on which it is applied. For stock data, we find strong evidence for MSGARCH while this is not the case for stock indices and currencies. Moreover, Markov-switching GARCH models with a conditional (skew) Normal distribution are not able to jointly account for the switch in the parameters as well as for the excess of kurtosis exhibited from the data; hence, Markov-switching GARCH models with a (skew) Student- $t$  specification are usually required. Finally, accounting for the parameter uncertainty in predictions, via MCMC, is necessary for stock data.

**Keywords:** GARCH, MSGARCH, forecasting performance, large-scale study, Value-at-Risk, risk management

---

---

<sup>☆</sup>We are grateful to Industrielle-Alliance, International Institute of Forecasters, Google Summer of Code 2016, FQRSC (Grant # 2015-NP-179931) and Fonds de Donations at the University of Neuchâtel for financial support. We thank Félix-Antoine Fortin and Calcul Québec (clusters Briaree, Colosse, Mammouth and Parallèle II) as well as Laurent Fastnacht and the Institute of Hydrology at the University of Neuchâtel (cluster Galileo) for computational support. All computations have been performed with the R package MSGARCH (Ardia et al., 2016b,a) available from the CRAN repository at <https://cran.r-project.org/package=MSGARCH>. The views expressed in this paper are the sole responsibility of the authors. Any remaining errors or shortcomings are the authors' responsibility.

\*Corresponding author. University of Neuchâtel, Rue A.-L. Breguet 2, CH-2000 Neuchâtel, Switzerland. Phone: +41 32 718 1365.

*Email addresses:* [david.ardia@unine.ch](mailto:david.ardia@unine.ch) (David Ardia), [keven.bluteau@unine.ch](mailto:keven.bluteau@unine.ch) (Keven Bluteau), [kris.boudt@vub.ac.be](mailto:kris.boudt@vub.ac.be) (Kris Boudt), [leopoldo.catania@uniroma2.it](mailto:leopoldo.catania@uniroma2.it) (Leopoldo Catania)

## 1. Introduction

“Modern society relies on the smooth functioning of banking and insurance systems and has a collective interest in the stability of such systems.” (McNeil et al., 2005). This statement is especially true since the global financial crisis of 2008. Following the regulatory process of the Basel Accords (currently the Basel III Accords), financial institutions of leading nations are obliged to meet stringent capital requirements and rely on state-of-the-art risk management systems (Board of Governors of the Federal Reserve Systems, 2012). Better risk management practices lead to a higher stability of the economy and have obvious social benefits: Guarantying the pension plan of current and future retirees is an example. There is, therefore, a strong need for backtesting existing risk models and comparing the estimation techniques used to calibrate these models.

Modeling the volatility of financial markets is central in risk management. Research on modeling volatility dynamics using time series models has been active since the creation of the original ARCH model by Engle (1982) and its generalization by Bollerslev (1986). From there, multiple extensions of the standard ARCH scedastic function have been proposed to capture additional stylized facts observed in financial markets. These so-called GARCH-type models recognize that there may be important nonlinearities, asymmetries, and long-memory properties in the volatility process; see Bollerslev et al. (1992), Bollerslev et al. (1994) and Engle (2004) for a review.

Recent studies show that estimates of GARCH-type models can be biased by structural breaks in the volatility dynamics (see, *e.g.*, Bauwens et al., 2010, 2014). These breaks typically occur during periods of financial turmoil. Estimating a GARCH model on data displaying a structural break yields a non-stationary estimated model and implies poor risk predictions. A way to cope with this problem is provided by Markov-switching GARCH models (MSGARCH) whose parameters can change over time according to a discrete latent (*i.e.*, unobservable) variable. These models can quickly adapt to variations in the unconditional volatility level, which improves risk predictions (see, *e.g.*, Marcucci, 2005; Ardia, 2008).

The first contribution of this paper is to investigate if MSGARCH models provide risk managers and regulators with useful new methodologies for improving the risk forecasts of their portfolios.<sup>1</sup>

---

<sup>1</sup>This study focuses exclusively on GARCH and MSGARCH models. GARCH is a workhorse in financial econo-

To answer this question, we perform a large-scale backtesting experiment in which we compare the forecasting performance of single-regime and Markov-switching GARCH models. As financial institutions invest in a large set of securities over different asset classes, our study is conducted on a vast universe of stocks (*i.e.*, four hundred), eleven equity indices and eight foreign exchange rates. To our knowledge, this is the first empirical study which assesses the performance of MSGARCH models on such a large cross-section of assets. For single-regime and regime-switching specifications, the stochastic models considered account for different reactions in the conditional volatility of past asset returns. More precisely, we consider the symmetric GARCH model (Bollerslev, 1986) as well as asymmetric GJR model (Glosten et al., 1993). These stochastic specifications are integrated into the MSGARCH framework with the approach of Haas et al. (2004). For the conditional distributions (which are regime-dependent), we use the symmetric and skewed versions of the Normal and Student- $t$  distributions, using the approach by Fernández and Steel (1998). Overall, this leads to sixteen models. Thus, differently from Hansen and Lunde (2005), who compare a large number of GARCH-type models on few series, we focus on few GARCH and MSGARCH models and a large number of series. The forecasting performance is tested for each time series and is based on 2,000 out-of-sample daily (percentage) log-returns. The backtesting period ranges from (approximatively) 2005 to 2016. We take a risk management perspective and assess the statistical and economical performance of the various models in forecasting the left-tail (*i.e.*, losses) of the conditional distribution of the assets' returns.

GARCH and MSGARCH models are traditionally estimated by the Maximum Likelihood (ML) technique; see for instance Haas et al. (2004), Marcucci (2005) and Augustyniak (2014). However, several recent studies have shown the advantages of the Bayesian approach (see, *e.g.*, Ardia, 2008; Ardia and Hoogerheide, 2010; Bauwens et al., 2010, 2014). In particular, appropriate Markov chain

---

metrics, has been investigated for decades, and is widely used by practitioners. MSGARCH is the most natural and straightforward extension to GARCH. It is therefore interesting to see if it adds any value to the toolkit of a risk manager. Extensions to stochastic volatility (SV) models (Taylor, 1994; Jacquier et al., 1994) or realized measures volatility models (RV) such as HEAVY (Shephard and Sheppard, 2010) or Realized GARCH (Hansen et al., 2011) are of course possible. However, SV models are sensitive to the implementation, as pointed out by Bos (2012). RV models require high-frequency data to deliver daily volatility forecasts. Backtesting RV models over a universe of hundred of stocks, as done in our study, is a challenging task. Moreover, to the best of our knowledge, RV models are used by (some) volatility-trading hedge funds, but are not standard risk models implemented by major banks or financial institutions. We, therefore, leave it for further research.

Monte Carlo (MCMC) procedures can explore the joint posterior distribution of the model parameters, and avoid local maxima (*i.e.*, non-convergence or convergence to wrong values) encountered via ML estimation. Moreover, parameter uncertainty is naturally integrated into the risk forecasts via the predictive distribution. As shown by Hoogerheide et al. (2012) in the context of single-regime GARCH models, integrating parameter uncertainty with the Bayesian approach is key to successfully forecast more accurately the left-tail of the return distribution, that is, the losses.

The second contribution of this paper is thus to investigate the advantages of the Bayesian approach compared with the traditional ML technique for GARCH and MSGARCH models. In particular, we test if integrating the parameter uncertainty translates into better risk measures forecasts. As for the ML estimation, the backtest experiment is performed for the large universe of stocks and indices, thus providing more significant results for practitioners. We, therefore, extend the study by Hoogerheide et al. (2012) both on the data and on the model dimensions. We rely on the adaptive sampler by Vihola (2012) for the MCMC estimation of the various models. To cope with the large computing time of the experiment, all computations are performed in parallel on several large clusters with the MSGARCH package (Ardia et al., 2016b), which efficiently implements the various models in C++.

Overall, our empirical results can be summarized as follows. First, the need for a Markov-switching mechanism in GARCH models depends on the underlying asset class on which it is applied. For stock data, we find strong evidence in favor of MSGARCH while this is not the case for stock indices and currencies. This can be explained by the large (un)conditional kurtosis observed for the log-returns of stock data. Second, Markov-switching GARCH models with a conditional (skew) Normal distribution are not able to jointly account for the switch in the parameters as well as for the excess of kurtosis exhibited from the data; hence, Markov-switching GARCH models with a (skew) Student- $t$  specification are usually required. Finally, accounting for the parameter uncertainty (*i.e.*, integrating the parameter uncertainty into the predictive distribution) via MCMC is necessary for stock data.

The paper proceeds as follows. Model specification, estimation, and forecasting are presented in Section 2. The datasets, the backtesting design, and the empirical results are discussed in Section 3.

Section 4 concludes.

## 2. Markov-switching GARCH models

### 2.1. Model specification

Let  $y_t \in \mathbb{R}$  be the (percentage) log-return of a financial asset at time  $t$ . Our general Markov-switching GARCH specification assumes that:

$$y_t \mid (s_t = k, \mathcal{I}_{t-1}) \sim \mathcal{D}(0, h_{k,t}, \boldsymbol{\xi}_k), \quad (1)$$

where  $\mathcal{D}(0, h_{k,t}, \boldsymbol{\xi}_k)$  is a continuous distribution with zero mean, time-varying variance  $h_{k,t}$ , and additional shape parameters gathered in the vector  $\boldsymbol{\xi}_k$ . Furthermore, we assume that the integer-valued stochastic variable  $s_t$ , defined on the discrete space  $\{1, \dots, K\}$ , evolves according to an unobserved first order ergodic homogeneous Markov chain with transition probability matrix  $\mathbf{P} \equiv \{p_{i,j}\}_{i,j=1}^K$ , with  $p_{i,j} \equiv \mathbb{P}[s_t = j \mid s_{t-1} = i]$ . We denote by  $\mathcal{I}_{t-1}$  the information set up to time  $t-1$ , *i.e.*,  $\mathcal{I}_{t-1} \equiv \{y_{t-i}, i > 0\}$ .

Given the parametrization of  $\mathcal{D}(\cdot)$ , we have  $\mathbb{E}[y_t^2 \mid s_t = k, \mathcal{I}_{t-1}] = h_{k,t}$ , that is,  $h_{k,t}$  is the variance of  $y_t$  conditional on the realization of  $s_t$ . Note that the conditional mean of the return is assumed to be zero across time and regimes.

As in Haas et al. (2004), the conditional variance of  $y_t$  is assumed to follow a GARCH-type model. Hence, conditionally on regime  $s_t = k$ ,  $h_{k,t}$  is available as a function of past returns and the additional regime-dependent vector of parameters  $\boldsymbol{\theta}_k$ :

$$h_{k,t} \equiv \lambda(y_{t-1}, h_{k,t-1}, \boldsymbol{\theta}_k), \quad (2)$$

where  $\lambda(\cdot)$  is a  $\mathcal{I}_{t-1}$ -measurable function which defines the filter for the conditional variance and also ensures its positiveness. We further assume that  $h_{k,1} \equiv \bar{h}_k$  ( $k = 1, \dots, K$ ), where  $\bar{h}_k$  is a fixed initial volatility level for regime  $k$ , that we set equal to the long-run unconditional volatility in regime  $k$ . Depending on the shape of  $\lambda(\cdot)$ , we obtain different scedastic specifications. For instance,



if:

$$h_{k,t} \equiv \omega_k + \alpha_k y_{t-1}^2 + \beta_k h_{k,t-1}, \quad (3)$$

with  $\omega_k > 0$ ,  $\alpha_k \geq 0$  and  $\alpha_k + \beta_k < 1$  ( $k = 1, \dots, K$ ), we recover the Markov-switching GARCH model (MSGARCH ( $K$ )) of Haas et al. (2004). In this case  $\boldsymbol{\theta}_k \equiv (\omega_k, \alpha_k, \beta_k)'$ .

More flexible definitions of the filter  $\lambda(\cdot)$  can be easily incorporated in the model. In order to account for the well-known asymmetric reaction of volatility to the sign of past returns (often referred to as the *leverage effect*; see Black (1976)), we can specify a Markov-switching GJR model with  $K$  regimes exploiting the volatility specification of Glosten et al. (1993):

$$h_{k,t} \equiv \omega_k + (\alpha_k + \gamma_k \mathbb{I}\{y_{t-1} < 0\}) y_{t-1}^2 + \beta_k h_{k,t-1}, \quad (4)$$

where  $\mathbb{I}\{\cdot\}$  is the indicator function equal to one if the condition holds, and zero otherwise. In this case, the additional parameter  $\gamma_k > 0$  controls the asymmetry in the conditional volatility process. We have  $\boldsymbol{\theta}_k \equiv (\omega_k, \alpha_k, \gamma_k, \beta_k)'$ . Stationarity of the volatility process conditionally on the Markovian state is achieved by imposing  $\alpha_k + \beta_k + \kappa_k \gamma_k < 1$ , where  $\kappa_k \equiv \mathbb{P}[y_t < 0 \mid s_t = k, \mathcal{I}_{t-1}]$ . For symmetric distributions we have  $\kappa_k = 1/2$ . For skewed distributions,  $\kappa_k$  is obtained following the approach of Trottier and Ardia (2016).

As stated in the introduction, we consider different choices for  $\mathcal{D}(\cdot)$ . We take the standard Normal,  $\mathcal{N}$ , and the fat-tailed Student- $t$  distribution,  $\mathcal{S}$ . Note that since  $\mathbb{E}[y_t \mid s_t = k, \mathcal{I}_{t-1}] = 0$  for all  $k = 1, \dots, K$ , the distribution of  $y_t \mid \mathcal{I}_{t-1}$  is symmetric by construction. In order to investigate the benefits of incorporating skewness in our large-scale analysis, we also consider the skewed version of  $\mathcal{N}$  and  $\mathcal{S}$  using the mechanism of Fernández and Steel (1998). Hence, we recover the skew-Normal,  $\text{sk}\mathcal{N}$ , and the skew-Student- $t$ ,  $\text{sk}\mathcal{S}$ . Standardized skewed distributions are parametrized as in Bauwens and Laurent (2005) such that they have zero mean and unit variance; see Trottier and Ardia (2016).

Overall, our model set includes 16 different specifications recovered as combinations of:

- The number of regimes,  $K \in \{1, 2\}$ . When  $K = 1$ , we label our specification as single-regime (SR);

- The filter for the conditional volatility process: GARCH and GJR;
- The choice of the conditional distribution  $\mathcal{D}(\cdot)$ , *i.e.*,  $\mathcal{D} \in \{\mathcal{N}, \mathcal{S}, \text{sk}\mathcal{N}, \text{sk}\mathcal{S}\}$ .<sup>2</sup>

## 2.2. Estimation

We estimate the models either by ML or by MCMC techniques (Bayesian estimation). Both approaches require the evaluation of the likelihood function.

In order to write the likelihood function corresponding to the MSGARCH model specification (1), we define the vector of log-returns  $\mathbf{y} \equiv (y_1, \dots, y_T)'$  and we regroup the model parameters into the vector  $\Psi \equiv (\xi_1, \theta_1, \dots, \xi_K, \theta_K, \mathbf{P})$ . The conditional density of  $y_t$  in state  $s_t = k$  given  $\Psi$  and  $\mathcal{I}_{t-1}$  is denoted by  $f_{\mathcal{D}}(y_t | s_t = k, \Psi, \mathcal{I}_{t-1})$ .

By integrating out the state variable  $s_t$ , we can obtain the density of  $y_t$  given  $\Psi$  and  $\mathcal{I}_{t-1}$  only. The (discrete) integration is obtained as follows:

$$f(y_t | \Psi, \mathcal{I}_{t-1}) \equiv \sum_{i=1}^K \sum_{j=1}^K p_{i,j} \eta_{i,t-1} f_{\mathcal{D}}(y_t | s_t = j, \Psi, \mathcal{I}_{t-1}), \quad (5)$$

where  $\eta_{i,t-1} \equiv \mathbb{P}[s_{t-1} = i | \Psi, \mathcal{I}_{t-1}]$  is the *filtered probability* of state  $i$  at time  $t-1$  and where we recall that  $p_{i,j}$  denotes the transition probability of moving from state  $i$  to state  $j$ . The filtered probabilities  $\{\eta_{k,t}; k = 1, \dots, K; t = 1, \dots, T\}$  are obtained by an iterative algorithm similar in spirit to a Kalman filter; we refer the reader to Hamilton (1989) and Hamilton (1994, Chapter 22) for details.

Finally, the likelihood function is obtained from (5) as follows:

$$\mathcal{L}(\Psi | \mathbf{y}) \equiv \prod_{t=1}^T f(y_t | \Psi, \mathcal{I}_{t-1}). \quad (6)$$

The ML estimator  $\hat{\Psi}$  is obtained by maximizing the logarithm of (6) (or minimizing the negative logarithm value). In the case of MCMC estimation, the likelihood function is combined with a diffuse (truncated) prior  $f(\Psi)$  to build the kernel of the posterior distribution  $f(\Psi | \mathbf{y})$ . As the

---

<sup>2</sup>We also tested the asymmetric EGARCH scedastic specification (Nelson, 1991) as well as alternative fat-tailed distributions, such as the Laplace distribution. The performance results were qualitatively similar.

posterior is of an unknown form (the normalizing constant is numerically intractable), it must be approximated by simulation techniques. In our case, draws from the posterior are generated with the adaptive random-walk Metropolis sampler of Vihola (2012). We use 50,000 burn-in draws and build the posterior sample of size 1,000 with the next 50,000 draws keeping only every 50th draws to diminish autocorrelation in the chain.<sup>3</sup> For both ML and MCMC estimations, we ensure positivity and stationarity of the conditional variance in each state during the estimation.

### 2.3. Density and VaR forecasting

Generating one-step ahead density and VaR forecasts with MSGARCH models is straightforward. First, note that the one-step ahead conditional probability density function (PDF) of  $y_{t+1}$  is a mixture of  $K$  regime-dependent distributions:

$$f(y_{t+1} | \Psi, \mathcal{I}_t) \equiv \sum_{k=1}^K \pi_{k,t+1} f_{\mathcal{D}}(y_{t+1}; 0, h_{k,t+1}, \xi_k), \quad (7)$$

with mixing weights  $\pi_{k,t+1} \equiv \sum_{i=1}^K p_{i,k} \eta_{i,t}$  where  $\eta_{i,t} \equiv \mathbb{P}[s_t = i | \Psi, \mathcal{I}_t]$  ( $i = 1, \dots, K$ ) are the filtered probabilities at time  $t$ . The cumulative density function (CDF) is obtained from (7) as follows:

$$F(y | \Psi, \mathcal{I}_t) \equiv \mathbb{P}[y_{t+1} \leq y | \Psi, \mathcal{I}_t] = \int_{-\infty}^y f(y_{t+1} | \Psi, \mathcal{I}_t) dy_{t+1}. \quad (8)$$

Within the ML framework, the predictive PDF and CDF are simply computed by replacing  $\Psi$  by the ML estimator  $\hat{\Psi}$  in (7) and (8). Within the MCMC framework, we proceed differently, and we integrate out the parameter uncertainty. Given a posterior sample  $\{\Psi^{[m]}, m = 1, \dots, M\}$ , the predictive PDF is obtained as:

$$f(y_{t+1} | \mathcal{I}_t) \equiv \int_{\Psi} f(y_{t+1} | \Psi, \mathcal{I}_t) d\Psi \approx \frac{1}{M} \sum_{m=1}^M f(y_{t+1} | \Psi^{[m]}, \mathcal{I}_t), \quad (9)$$

---

<sup>3</sup>We performed several sensitivity analyses to assess the implication of the estimation setup. First, we changed the hyper-parameter values. Second, we ran longer MCMC chains. Third, we used 10,000 posterior draws instead of 1,000. Finally, we tested an alternative MCMC sampler based on adaptive mixtures of Student- $t$  distribution (Ardia et al., 2009). In all cases, the conclusions remained qualitatively similar.

and the predictive CDF is given by:

$$F(y_{t+1} | \mathcal{I}_t) \equiv \int_{-\infty}^{y_{t+1}} f(u | \mathcal{I}_t) du. \quad (10)$$

Both for the ML and MCMC estimation, the VaR is estimated as a quantile of the predictive density, by numerically inverting the predictive CDF. For instance, in the MCMC framework, the VaR at the  $\alpha$  risk level is estimated as:

$$\text{VaR}_{t+1}^\alpha \equiv \{y_{t+1} \in \mathbb{R} | F(y_{t+1} | \mathcal{I}_t) = \alpha\}. \quad (11)$$

In our empirical application, we consider the VaR at the 1% and 5% risk levels.

### 3. Large-scale empirical study

We use 1,500 log-returns (in percent) for the fit and run the backtest over 2,000 out-of-sample log-returns for a period ranging from October 10, 2008, to November 17, 2016 (data start in December 26, 2002). For each time series on which the backtest is applied, we first remove the unconditional mean and autocorrelation by using an AR(1)-filter, thus focusing on the conditional variance dynamics. Each model is estimated on a rolling window basis, and one-step ahead density forecasts are obtained. From the density, we compute the VaR at the 1% and 5% risk levels.

#### 3.1. Datasets

We test the performance of various models on several universes:

- A set of four hundred stocks on the US market, selected withing the constituents of the S&P 500 index as of November 2016;
- A set of eleven stock indices: (1) S&P 500 (US; SPX), (2) FTSE 100 (UK; FTSE), (3) CAC 40 (France; FCHI), (4) DAX 30 (Germany; GDAXI), (5) Nikkei 225 (Japan; N225), (6) Hang Seng (China, HSI), (7) Dow Jones Industrial Average (US; DJI), (8) Euro Stoxx 50 (Europe; STOXX50), (9) KOSPI (South Korea; KS11), (10) S&P/TSX Composite (Canada; GSPTSE), and (11) SMI (Switzerland; SSMI);

- A set of eight currencies: USD against CAD, DKK, NOK, AUD, CHF, GBP, JPY, and EUR.

Each dataset is expressed in local currency. For all datasets, we compute the daily percentage log–return series defined by  $y_t \equiv 100 \times (\log(P_t) - \log(P_{t-1}))$ , where  $P_t$  is the adjusted closing price (index) on day  $t$ . Data are retrieved from Datastream. The data are filtered for liquidity following Lesmond et al. (1999).

In Table 1, we report the summary statistics on the 2,000 out–of–sample log–returns for the assets in the various universes. The left part of the table presents unconditional moments, while the right part presents the average of the statistics computed over 250–day rolling windows. We note the higher volatility in the universe of stocks, followed by stock indices and currencies. All assets exhibit negative skewness, with larger values for stocks, while currencies seem to behave more symmetrically. Finally, we observe a significant kurtosis for stocks, unconditionally but also on a rolling–window basis. Fat tails are also present for stock indices and currencies, but less pronounced though. From these empirical facts, we anticipate best performance for model accounting for skewed and fat–tailed conditional distributions.

[Insert Table 1 about here.]

### 3.2. Forecasting performance tests

We assess the quality of left–tail risk forecasts via several standard tests used in financial risk management.

First, we focus on the VaR forecasts at the 1% and 5% risk levels. The first test used is the conditional coverage (CC) approach by Christoffersen (1998), the common extension of the unconditional coverage (UC) test by Kupiec (1995). This approach is based on the study of the *hit* sequence  $I_t^\alpha \equiv \mathbb{I}\{y_t \leq \text{VaR}_t^\alpha\}$ , where  $\text{VaR}_t^\alpha$  denotes the VaR prediction at time  $t$  for risk level  $\alpha$ , and  $\mathbb{I}\{\cdot\}$  is the indicator function equal to one if the condition holds, and zero otherwise. A sequence of VaR forecasts at risk level  $\alpha$  has correct conditional coverage if  $\{I_t^\alpha; t = 1, \dots, H\}$  is an independent and identically distributed sequence of Bernoulli random variables with parameter  $\alpha$ . This hypothesis can be verified by testing jointly the independence on the series and the unconditional coverage of the VaR forecasts.

The second test considered to assess the quality of VaR forecasts for risk levels at 1% and 5% is the dynamic quantile (DQ) approach by Engle and Manganelli (2004). This method jointly tests for UC and CC and has more power than previous alternatives under some form of model misspecification. The series of interest is defined as  $\{I_t^\alpha - \alpha; t = 1, \dots, H\}$ . Under correct model specification, we have the following moment conditions:  $\mathbb{E}[I_t^\alpha - \alpha] = 0$ ,  $\mathbb{E}[I_t^\alpha - \alpha | \mathcal{I}_{t-1}] = 0$ ,  $\mathbb{E}[(I_t^\alpha - \alpha)(I_{t'}^\alpha - \alpha)] = 0$  for  $t \neq t'$ ; see Engle and Manganelli (2004).

Third, we follow González-Rivera et al. (2004) and McAleer and Da Veiga (2008) and use the (tick) asymmetric linear losses induced by our VaR forecasts. Formally, given a VaR prediction at risk level  $\alpha$  for time  $t$ , the associated quantile loss (QL) is defined as:

$$\text{QL}_t^\alpha \equiv (\alpha - I_t^\alpha)(y_t - \text{VaR}_t^\alpha).$$

Evidently, QL is an asymmetric loss function that penalizes more heavily with weight  $(1 - \alpha)$  the observations for which we observe returns VaR exceedance. Quantile losses are then compared between models over the out-of-sample period. We discriminate between models using the approach by Diebold and Mariano (1995), with the heteroscedasticity and autocorrelation robust (HAC) standard error estimators of Andrews (1991) and Andrews and Monahan (1992). As our VaR forecasts are generated in some cases by nested models, the DM test statistics does not have standard distribution under the null (Diebold, 2015). We therefore use the critical values obtained from bootstrap, as detailed in Clark and McCracken (2012).

Finally, in addition to the QL function, we also consider the weighed Continuous Ranked Probability Score (wCRPS) introduced by Gneiting and Ranjan (2011) as a generalization of the well-known CRPS scoring rule (Matheson and Winkler, 1976). wCRPS is a *proper* scoring rule which permits us to compare the predictive ability of different models over a particular region of the support.<sup>4</sup> Following the notation introduced in Section 2, wCRPS for a forecast at time  $t + 1$

---

<sup>4</sup>Given a random variable  $y \in \mathbb{R}$  with continuous probability density function  $f$ , the scoring rule  $S(f, y)$  is said to be *proper* if and only if  $\mathbb{E}_f[S(f, y)] = \int_{\mathbb{R}} f(y)S(f, y)dy \leq \int_{\mathbb{R}} f(y)S(g, y)dy = \mathbb{E}_f[S(g, y)]$  for all density functions  $f$  and  $g$ .

is defined as:

$$\text{wCRPS}_{t+1} \equiv \int_{\mathbb{R}} \omega(z) (F(z | \mathcal{I}_t) - \mathbb{I}\{y_{t+1} < z\})^2 dz,$$

where  $\omega : \mathbb{R} \rightarrow \mathbb{R}^+$  is a continuous weight function which emphasizes regions of interest of the predictive distribution, such as the tails or the center. Evidently, the wCRPS measures the distance between the predicted CDF,  $F(y_{t+1} | \mathcal{I}_t)$ , and the empirical CDF represented as a step function in  $y_{t+1}$ . Averaging the wCRPS over the out-of-sample period provides the quantity at the base of our comparative analysis. Models with lower averaged wCRPS are preferred.

Since our focus is on the left tail of the returns distribution, following Gneiting and Ranjan (2011), we use  $\omega(z) \equiv 1 - \Phi(z)$ , where  $\Phi$  is the CDF of a standard Gaussian distribution. This way, discrepancies between the left tail of the returns distribution are weighed more than those referred to the right tail.<sup>5</sup>

### 3.3. Results

We now answer the general question: *Does the inclusion of a Markov-switching mechanism for the returns distribution improve VaR predictions ?* Indeed, while there is plenty of evidence concerning the benefits of accounting for skewness (De Luca et al., 2006; Franceschini and Loperfido, 2010; Luca and Loperfido, 2015), excess kurtosis (Bollerslev, 1987), asymmetries in the volatility dynamics (Nelson, 1991; Zakoian, 1994) and parameters uncertainty (Ardia et al., 2012) for volatility modeling, we aim to investigate to which extent including Markov-switching improves over single-regime models VaR predictions, while accounting for others well-known features of financial returns.

We answer this question relying on the large-scale performance study over a broad universe of assets, as previously detailed. The experiment proceeds with two approaches, which we refer to “Backtesting performance” and “Pairwise performance”.

---

<sup>5</sup>We compute wCRPS with the following approximation:

$$\text{wCRPS}_{t+1} \approx \frac{y_u - y_l}{I - 1} \sum_{i=1}^I w(y_i) (F(y_i | \mathcal{I}_t) - \mathbb{I}\{y_{t+1} < y_i\})^2,$$

where  $y_i \equiv y_l + i \times (y_u - y_l)/I$  and  $y_u$  and  $y_l$  are the upper and lower values, which defines the range of integration. The accuracy of the approximation can be increased to any desired level by  $I$ . In this paper, we set  $y_l = -100$ ,  $y_u = 100$  and  $I = 1,000$ , which work well for daily returns in percentage points.

### 3.3.1. Backtesting

We backtest the VaR predictions delivered by MSGARCH and GARCH models using CC and DQ tests on each series of 2,000 out-of-sample observations. Then, we measure the number of times we reject the null hypothesis of accurate VaR forecast at the 5% significance level for all asset classes, and we compare the results. In the case of stocks, as the universe is large and therefore prone to false positives, the frequency of rejections is corrected for Type I error using the false discovery rate (FDR) approach by Storey (2002).

Table 2 reports the comparison between Markov-switching (MS) and single-regime (SR) GARCH models estimated by MCMC or ML for all asset classes. Panels A and B summarize the results for the CC test and Panels C and D for the DQ test. In light gray, we report the significantly lowest percentage (at the 5% level) between MS and SR specifications, for a given model, estimation method and data set. This is obtained by performing a t-test between the rejection frequencies with robust estimation of the standard error. The star sign ( $\star$ ) indicates for a given model and data set if there is a significantly outperforming specification (MS vs. SR and ML vs. MCMC).

[Insert Table 2 about here.]

*CC test.* At both VaR risk levels, we find that MS specifications are favored for all datasets. Improvements of MS over SR are usually of larger magnitude when we consider Normal and skewed Normal conditional distribution.<sup>6</sup> For instance, for stock data and 1% risk level, MS GARCH  $\mathcal{N}$  estimated via MCMC reports a 1.5% rejections frequency while its SR counterpart rejects 22.75% of the time. Concerning the same MS specification but estimated via ML, the frequency of rejections increases from 1.5% to 3% indicating that in this case, MCMC inference helps to improve VaR forecast delivered by MSGARCH models. However, the evidence for MCMC inference over ML is mixed across asset classes. While for stocks data MCMC is generally favored, for stock indices and currencies we cannot see a clear picture. Interesting, we find that MS results are generally unaffected by the chosen risk level. For instance, the rejection frequencies for MS specification for at 1% and 5% risk levels are quite similar while results from SR specifications are generally affected

---

<sup>6</sup>For Markov-switching models with *conditional distribution*, we refer to the distribution of returns conditionally on past information and the realization of the Markov process.



by this choice. Overall, the lowest rejection frequencies are obtained for the GJR scedastic function together with a fat-tailed asymmetric distribution. For this particular specification, MS and SR perform equally well, but when estimated via ML for stock data.

*DQ test.* Results provided by the DQ test, both for the 1% or 5%–VaR, exhibit higher rejection frequencies. For all asset classes, an asymmetric GJR specification with skewed Student- $t$  is required. Again, MS models are preferred over SR specifications. At the 5% risk level, results for stocks indicate that MCMC outperforms ML estimation technique for most specification. Remarkably, the single-regime GJR sk $\mathcal{S}$  model performs very well again. Results are also consistent across CC and DQ tests. Specifically, for DQ we also find that accounting for parameter uncertainty is important for stock data at both risk levels; rejection frequencies for MS models estimated via MCMC are generally lower than those reported by the analogs specifications estimated via ML. For other asset classes, results with respect to the estimation procedure are mixed again, thus indicating that this result is asset-specific.

Overall, backtesting tests indicate that MSGARCH models are generally preferred over single-regime models, independently of the specification of the scedastic function. This is especially so for the universe of stocks and stock indices, while for currencies both models report very satisfactory results and essentially perform equally well. We observe that Markov-switching models provide sensible gains over single-regime models at both risk levels. For example, when the model is conditionally Normal and estimated via MCMC, the DQ test rejects 29.5% of times for MSGARCH and 58.75% of times for GARCH in the case of 5%–VaR, and 14.5% of times versus 23.75% in the case of 1%–VaR. Results are similar for the CC test. Notably, we find that for 5%–VaR, MSGARCH provides almost no rejections for the CC tests across all the considered specifications with Normal and skewed-Normal conditional distributions, while GARCH models reject the null with a rate of about 30%. Differently, if we consider the Student- $t$  and skewed-Student- $t$  cases, then MSGARCH and GARCH models perform similarly well, even if the latter provide slightly better results. Our findings indicate that assuming a fat-tailed conditional distribution for both Markov-switching and single-regime models is of primary importance and delivers excellent results at both risk levels.

Heterogeneous results with respect to the different asset classes are obviously related to the

various characteristics of the data detailed in Table 1. It is not surprising that MSGARCH models perform better for stocks, as they are characterized by a higher (un)conditional kurtosis.

### 3.3.2. Pairwise comparison

We compare now MS and SR models in terms of QL and wCRPS measures. To that end, for each model specification and asset in a universe, we compute the DM statistics of the QL and wCRPS differentials between MS and SR models and determine if it is significantly different from zero at the 1% level.<sup>7</sup> Results are presented in Table 3, where a negative (positive) value indicates outperformance (underperformance) of MSGARCH against GARCH specification. In light (dark) gray, we emphasize cases of significant outperformance of MS (SR) models. All models are estimated by MCMC.<sup>8</sup>

[Insert Table 3 about here.]

*QL test.* Results for the QL pairwise comparison do not allow to discriminate between MS and SR for the various model specifications, except for the universe of stocks in the case of the 5%–VaR. In this case, the DM statistics are significantly negative for all model specification, but the GJR skS model, which however exhibit a negative DM statistics at -1.81.

*wCRPS.* The results for wCRPS favor MS models with negative values observed for all asset classes and model specifications. In particular, significant values are observed for the universe of stocks (except the GJR skS). For the universe of stock indices, significant values are observed for GARCH with Normal and skewed Normal distributions. In the case of currencies, MSGARCH is significantly outperforming single-regime models for the GARCH with (skewed) Normal and symmetric Student- $t$  distributions.

### 3.3.3. Full pairwise model comparison

As the last step in our analysis, we analyze in more details the wCRPS performance for the full set of Markov-switching specifications, versus the single-regimes counter-parts. The analysis

---

<sup>7</sup>We take a more conservative view here, as we cannot correct for false discoveries.

<sup>8</sup>Results with models estimated by ML are similar.

is again conducted on the three universes of assets. Results are reported in Table 4. In light (dark) gray we highlight the significantly better performance of MS (SR) models (at the 1% significance level, based on a DM test).

[Insert Table 4 about here.]

Let us consider first the universe of 400 stocks. We note that the MS specification is always better than SR in the case of Normal and skewed Normal conditional distributions. This result should be principally attributed to the fact that the one-step ahead predictive distribution delivered from MS is fat-tailed even if the state dependent densities are not. Moreover, we notice that the GJR specification is important for this class of assets, as the wCRPS value decreases when considering GARCH vs. GJR or GJR vs. GARCH. Another interesting feature is the outperformance of SR when accounting for conditional fat-tails while MS is based on the normal distribution. Hence, the MS mechanism itself is not sufficient to introduce enough conditional kurtosis in the data; a fat-tailed Student- $t$  distribution is required. Indeed, it is hard to believe that the MS mechanism can contemporaneously fully account for the switch in the parameters controlling the evolution of the conditional volatility and also provide predictive distribution with remarkably fat-tails. Looking at the GJR sk $\mathcal{S}$  case, which is the most general specification we consider, we find that MS still reports lower values for wCRPS over SR, even if the difference is not statistically significant at the 1% confidence level.

Results are similar for the universe of the eleven stock indices. MS models generally report lower wCRPS values over SR. However, differences between MS and SR are less pronounced, indeed, while for stocks data the average differences with respect to GARCH  $\mathcal{N}$  are around 9 points, for stock indices average differences are around 5–6 points. Furthermore, for stock indices data the DM test rejects less frequently the null of equal predictive ability across the two specifications.

Consistently with previous results, we find that for currencies, the gain of including MS decreases substantially. However, results remain highly significant in most of the cases. Similar to other asset classes, SR models with a fat-tailed conditional distribution outperform MS with Normal and skewed Normal conditional distribution.

## 4. Conclusion

In this paper, we tested if MSGARCH models provide risk managers and regulators with useful new methodologies for improving the risk forecasts of their portfolios. As financial institutions invest in a large set of securities over different asset classes, our study is conducted on a very large universe of stocks, equity indices, and foreign exchange rates.

Our empirical results can be summarized as follows. First, the need for a Markov-switching mechanism in GARCH models depends on the underlying asset class on which it is applied. For stock data, we find strong evidence for MSGARCH while this is not the case for stock indices and currencies. This result can be explained by the large (un)conditional kurtosis observed for the log-returns in stock data. Second, Markov-switching GARCH models with a (skew) Normal distribution are not able to jointly account for the switch in the parameters as well as for the excess of kurtosis exhibited from the data; hence, Markov-switching GARCH models with a (skew) Student- $t$  specification are usually required. Finally, accounting for the parameter uncertainty (*i.e.*, integrating the parameter uncertainty into the predictive distribution) is necessary for stock data.

Our study could be extended in numerous ways. First, additional universes could be considered. In particular, we plan to add commodities and emerging markets data. As MSGARCH models are able to deal quickly with changes in the unconditional level of the volatility, contrary single-regime GARCH models, it would be interesting to investigate multi-step ahead risk forecasts. Finally, our study considered single-regime versus two-state Markov-switching specifications. It would be of interest to see if a third regime leads to superior performance, and also, if the optimal number of regime (in the Akaike, BIC or DIC sense) change over time, and is different across the data sets.

## References

- Andrews, D., 1991. Heteroskedasticity and autocorrelation consistent covariance matrix estimation. *Econometrica* 59, 817–858. doi:10.2307/2938229.
- Andrews, D., Monahan, J., 1992. An improved heteroskedasticity and autocorrelation consistent covariance matrix estimation. *Econometrica* 60, 953–966. doi:10.2307/2951574.
- Ardia, D., 2008. *Financial Risk Management with Bayesian Estimation of GARCH Models: Theory and Applications*. Springer. doi:10.1007/978-3-540-78657-3.
- Ardia, D., Baştürk, N., Hoogerheide, L.F., Van Dijk, H.K., 2012. A comparative study of Monte Carlo methods for efficient evaluation of marginal likelihood. *Computational Statistics & Data Analysis* 56, 3398–3414. doi:10.1016/j.csda.2010.09.001.
- Ardia, D., Bluteau, K., Boudt, K., Trottier, D.A., 2016a. Markov-switching GARCH models in R: The MSGARCH package. URL: <https://ssrn.com/abstract=2845809>. working paper.
- Ardia, D., Bluteau, K., Bout, K., Peterson, B., 2016b. MSGARCH: Markov Switching GARCH Models in R. URL: <https://cran.r-project.org/package=MSGARCH>. R package version 0.17.7.
- Ardia, D., Hoogerheide, L.F., 2010. Efficient Bayesian estimation and combination of GARCH-type models, in: Böcker, K. (Ed.), *Rethinking Risk Measurement and Reporting - Volume II: Examples and Applications from Finance*. Risk Books, London. chapter 1, pp. 1–18.
- Ardia, D., Hoogerheide, L.F., van Dijk, H.K., 2009. Adaptive mixture of Student-t distributions as a flexible candidate distribution for efficient simulation: The R package AdMit. *Journal of Statistical Software* 29. doi:10.18637/jss.v029.i03.
- Augustyniak, M., 2014. Maximum likelihood estimation of the Markov-switching GARCH model. *Computational Statistics & Data Analysis* 76, 61–75. doi:10.1016/j.csda.2013.01.026.
- Bauwens, L., Backer, B.D., Dufays, A., 2014. A Bayesian method of change-point estimation with recurrent regimes: Application to GARCH models. *Journal of Empirical Finance* 29, 207–229. doi:10.1016/j.jempfin.2014.06.008.
- Bauwens, L., Laurent, S., 2005. A new class of multivariate skew densities, with application to generalized autoregressive conditional heteroscedasticity models. *Journal of Business & Economic Statistics* 23, 346–354. doi:10.1198/073500104000000523.
- Bauwens, L., Preminger, A., Rombouts, J.V.K., 2010. Theory and inference for a Markov switching GARCH model. *Econometrics Journal* 13, 218–244. doi:10.1111/j.1368-423X.2009.00307.x.
- Black, F., 1976. Studies of stock price volatility changes, in: *Proceedings of the 1976 Meetings of the American Statistical Association*, pp. 177–181.
- Board of Governors of the Federal Reserve Systems, 2012. 99th Annual Report. Technical Report. Board of Governors of the Federal Reserve Systems.
- Bollerslev, T., 1986. Generalized autoregressive conditional heteroskedasticity. *Journal of Econometrics* 31, 307–327.

doi:10.1016/0304-4076(86)90063-1.

- Bollerslev, T., 1987. A conditionally heteroskedastic time series model for speculative prices and rates of return. *Review of Economics and Statistics* 69, 542–547. doi:10.2307/1925546.
- Bollerslev, T., Chou, R.Y., Kroner, K., 1992. ARCH modeling in finance: A review of the theory and empirical evidence. *Journal of Econometrics* 52, 5–59. doi:10.1016/0304-4076(92)90064-X.
- Bollerslev, T., Engle, R.F., Nelson, D.B., 1994. *Handbook of Econometrics*. North-Holland. chapter 49. pp. 2959–3038.
- Bos, C.S., 2012. Relating stochastic volatility estimation methods, in: *Handbook of Volatility Models and Their Applications*. Wiley-Blackwell, pp. 147–174. doi:10.1002/9781118272039.ch6.
- Christoffersen, P.F., 1998. Evaluating interval forecasts. *International Economic Review* 39, 841–862. doi:10.2307/2527341.
- Clark, T.E., McCracken, M.W., 2012. Reality checks and comparisons of nested predictive models. *Journal of Business & Economic Statistics* 30, 53–66. doi:10.1198/jbes.2011.10278.
- De Luca, G., Genton, M.G., Loperfido, N., 2006. *Advanced in Econometrics*. Emerald. volume 20 of 1. chapter A multivariate skew-GARCH model. pp. 33–57. doi:10.1016/S0731-9053(05)20002-6.
- Diebold, F.X., 2015. Comparing predictive accuracy, twenty years later: A personal perspective on the use and abuse of Diebold-Mariano tests. *Journal of Business & Economic Statistics* 33, 1–9. doi:10.1080/07350015.2014.983236.
- Diebold, F.X., Mariano, R.S., 1995. Comparing predictive accuracy. *Journal of Business & Economic Statistics* 13, 253–263. doi:10.1080/07350015.1995.10524599.
- Engle, R.F., 1982. Autoregressive conditional heteroscedasticity with estimates of the variance of United Kingdom inflation. *Econometrica* 50, 987–1008. doi:10.2307/1912773.
- Engle, R.F., 2004. Risk and volatility: Econometric models and financial practice. *American Economic Review* 94, 405–420. doi:10.1257/0002828041464597.
- Engle, R.F., Manganelli, S., 2004. CAViaR: Conditional autoregressive Value at Risk by regression quantiles. *Journal of Business & Economic Statistics* 22, 367–381. doi:10.1198/073500104000000370.
- Fernández, C., Steel, M.F.J., 1998. On Bayesian modeling of fat tails and skewness. *Journal of the American Statistical Association* 93, 359–371. doi:10.1080/01621459.1998.10474117.
- Franceschini, C., Loperfido, N., 2010. *Mathematical and Statistical Methods for Actuarial Sciences and Finance*. Springer Milan, Milano. chapter A skewed GARCH-type model for multivariate financial time series. pp. 143–152. doi:10.1007/978-88-470-1481-7\_15.
- Glosten, L.R., Jagannathan, R., Runkle, D.E., 1993. On the relation between the expected value and the volatility of the nominal excess return on stocks. *Journal of Finance* 48, 1779–1801. doi:10.1111/j.1540-6261.1993.tb05128.x.
- Gneiting, T., Ranjan, R., 2011. Comparing density forecasts using threshold –and quantile– weighted scoring rules. *Journal of Business & Economic Statistics* 29, 411–422. doi:10.1198/jbes.2010.08110.

- González-Rivera, G., Lee, T.H., Mishra, S., 2004. Forecasting volatility: A reality check based on option pricing, utility function, Value-at-Risk, and predictive likelihood. *International Journal of Forecasting* 20, 629–645. doi:10.1016/j.ijforecast.2003.10.003.
- Haas, M., Mittnik, S., Paoletta, M.S., 2004. A new approach to Markov-switching GARCH models. *Journal of Financial Econometrics* 2, 493–530. doi:10.1093/jjfinec/nbh020.
- Hamilton, J.D., 1989. A new approach to the economic analysis of nonstationary time series and the business cycle. *Econometrica* 57, 357–384. doi:10.2307/1912559.
- Hamilton, J.D., 1994. *Time Series Analysis*. First ed., Princeton University Press, Princeton, USA.
- Hansen, P.R., Huang, Z., Shek, H.H., 2011. Realized GARCH: A joint model for returns and realized measures of volatility. *Journal of Applied Econometrics* 27, 877–906. doi:10.1002/jae.1234.
- Hansen, P.R., Lunde, A., 2005. A forecast comparison of volatility models: does anything beat a GARCH(1,1)? *Journal of Applied Econometrics* 20, 873–889. doi:10.1002/jae.800.
- Hoogerheide, L.F., Ardia, D., Corré, N., 2012. Density prediction of stock index returns using GARCH models: Frequentist or Bayesian estimation? *Economics Letters* 116, 322–325. doi:10.1016/j.econlet.2012.03.026.
- Jacquier, E., Polson, N.G., Rossi, P.E., 1994. Bayesian analysis of stochastic volatility models. *Journal of Business & Economic Statistics* 12, 371–389. doi:10.1080/07350015.1994.10524553.
- Kupiec, P.H., 1995. Techniques for verifying the accuracy of risk measurement models. *Journal of Derivatives* 3. doi:10.3905/jod.1995.407942.
- Lesmond, D.A., Ogden, J.P., Trzcinka, C.A., 1999. A new estimate of transaction costs. *Review of Financial Studies* 12, 113–1141.
- Luca, G.D., Loperfido, N., 2015. Modelling multivariate skewness in financial returns: A SGARCH approach. *European Journal of Finance* 21, 1113–1131. doi:10.1080/1351847X.2011.640342.
- Marcucci, J., 2005. Forecasting stock market volatility with regime-switching GARCH models. *Studies in Nonlinear Dynamics & Econometrics* 9. doi:10.2202/1558-3708.1145.
- Matheson, J.E., Winkler, R.L., 1976. Scoring rules for continuous probability distributions. *Management Science* 22, 1087–1096. doi:10.1287/mnsc.22.10.1087.
- McAleer, M., Da Veiga, B., 2008. Single-index and portfolio models for forecasting Value-at-Risk thresholds. *Journal of Forecasting* 27, 217–235. doi:10.1002/for.1054.
- McNeil, A.J., Frey, R., Embrechts, P., 2005. *Quantitative Risk Management: Concepts, Techniques and Tools*. Princeton University Press.
- Nelson, D.B., 1991. Conditional heteroskedasticity in asset returns: A new approach. *Econometrica* 59, 347–370. doi:10.2307/2938260.
- Shephard, N., Sheppard, K., 2010. Realising the future: Forecasting with high-frequency-based volatility (HEAVY) models. *Journal of Applied Econometrics* 25, 197–231. doi:10.1002/jae.1158.
- Storey, J., 2002. A direct approach to false discovery rates. *Journal of the Royal Statistical Society B* 64, 479–498.

doi:10.1111/1467-9868.00346.

Taylor, S.J., 1994. Modeling stochastic volatility: A review and comparative study. *Mathematical Finance* 4, 183–204.

doi:10.1111/j.1467-9965.1994.tb00057.x.

Trottier, D.A., Ardia, D., 2016. Moments of standardized Fernández-Steel skewed distributions: Applications to the estimation of GARCH-type models. *Finance Research Letters* 18, 311–316. doi:10.1016/j.fr1.2016.05.006.

Vihola, M., 2012. Robust adaptive Metropolis algorithm with coerced acceptance rate. *Statistics and Computing* 22, 997–1008. doi:10.1007/s11222-011-9269-5.

Zakoian, J.M., 1994. Threshold heteroskedastic models. *Journal of Economic Dynamics & Control* 18, 931–955. doi:10.1016/0165-1889(94)90039-6.



**Table 1: Data summary statistics**

The table presents the summary statistics for the three asset classes used in our study. We report the standard deviation (Std), the skewness (Skew) and the kurtosis (Kurt) on an unconditional and rolling-windows basis for the 2,000 out-of-sample observations. The rolling window is based on 250 observations, and we report the average statistics over 1750 windows. For each statistic, we compute the 25th, 50th and 75th percentiles of the whole universe of assets.

	Unconditional			Rolling windows		
	Std	Skew	Kurt	Std	Skew	Kurt
<i>Panel A: Stocks</i>						
25th	1.48	−0.38	6.87	1.34	−0.30	4.92
50th	1.89	−0.13	9.19	1.68	−0.16	6.05
75th	2.31	0.12	14.08	2.03	0.00	8.55
<i>Panel B: Stock indices</i>						
25th	1.07	−0.40	6.07	1.00	−0.37	4.22
50th	1.15	−0.23	7.29	1.02	−0.21	4.66
75th	1.39	−0.17	10.29	1.28	−0.18	4.82
<i>Panel C: Currencies</i>						
25th	0.61	−0.53	4.36	0.58	−0.16	4.05
50th	0.62	−0.08	4.51	0.60	−0.01	4.16
75th	0.77	0.05	11.60	0.73	0.09	6.05

**Table 2: VaR performance results**

The table presents the percentage of rejections for the 1%-VaR (Panels A and C) and 5%-VaR (Panels B and D) coverage tests (at the 5% significance level) of the Markov-switching (MS) and single-regime (SR) models for the various universes of assets (400 stocks, 11 stock indices and 8 currencies) and estimated via MCMC or ML techniques. The considered tests are the conditional coverage test (CC, Panels A and B) by Christoffersen (1998) and the Dynamic Quantile test (DQ, Panels C and D) by Engle and Manganelli (2004). Each test is applied to a given time series for a given model with 2,000 out-of-sample observations. In the case of stocks, as the universe is large and therefore prone to false positives, the frequency of rejections is corrected for Type I error using the false discovery rate (FDR) approach by Storey (2002). In light gray, we report the lowest significant (at the 5% confidence level) percentage between Markov-switching and single-regime models, for a given data set and a given model specification. The star sign ( $\star$ ) indicates the lowest percentage (at the 5% confidence level) for a given asset class. Significance is determined via a t-test between the percentages with robust estimation of the standard error.

Model	Stocks				Stock indices				Currencies			
	MCMC		ML		MCMC		ML		MCMC		ML	
	MS	SR	MS	SR	MS	SR	MS	SR	MS	SR	MS	SR
<i>Panel A: CC 1%-VaR</i>												
GARCH $\mathcal{N}$	1.50	22.75	3.00	22.75	81.82	100.00	72.73	90.91	37.50	12.50	25.00	25.00
GARCH sk $\mathcal{N}$	0.00	12.00	1.25	13.00	0.00	72.73	0.00	54.55	0.00	12.50	0.00	12.50
GARCH $\mathcal{S}$	0.00	1.00	0.00	0.50	45.45	72.73	63.64	45.45	0.00	12.50	12.50	12.50
GARCH sk $\mathcal{S}$	0.00	0.25	0.00	0.25	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
GJR $\mathcal{N}$	0.50	14.50	0.50	12.50	45.45	90.91	54.55	90.91	12.50	25.00	25.00	25.00
GJR sk $\mathcal{N}$	0.25	6.25	0.00	2.50	0.00	54.55	0.00	45.45	25.00	0.00	0.00	0.00
GJR $\mathcal{S}$	0.00	0.00	0.00	0.50	36.36	27.27	27.27	36.36	0.00	0.00	12.50	12.50
GJR sk $\mathcal{S}$	0.00	0.00	0.00	0.25	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
<i>Panel B: CC 5%-VaR</i>												
GARCH $\mathcal{N}$	1.50 $\star$	31.75	3.00	33.00	36.36	27.27	27.27	27.27	25.00	25.00	12.50	12.50
GARCH sk $\mathcal{N}$	0.50	34.50	1.75	34.50	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
GARCH $\mathcal{S}$	1.00	2.00	2.25	2.75	54.55	45.45	54.55	63.64	0.00	0.00	0.00	0.00
GARCH sk $\mathcal{S}$	0.00	0.75	0.00	1.75	0.00	0.00	0.00	0.00	12.50	0.00	0.00	0.00
GJR $\mathcal{N}$	0.00 $\star$	29.75	0.00	29.75	18.18	9.09	18.18	18.18	25.00	0.00	12.50	12.50
GJR sk $\mathcal{N}$	0.00 $\star$	33.75	0.00	32.75	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
GJR $\mathcal{S}$	0.25	0.25	0.00	1.00	18.18	27.27	9.09	18.18	12.50	0.00	12.50	0.00
GJR sk $\mathcal{S}$	0.00	0.25	0.00	0.00	9.09	9.09	0.00	18.18	0.00	0.00	0.00	0.00
<i>Panel C: DQ 1%-VaR</i>												
GARCH $\mathcal{N}$	29.50	58.75	30.25	62.25	81.82	100.00	63.64	90.91	25.00	37.50	25.00	37.50
GARCH sk $\mathcal{N}$	21.50	53.25	28.25	57.00	36.36 $\star$	72.73	54.55	81.82	0.00 $\star$	37.50	12.50	37.50
GARCH $\mathcal{S}$	28.25 $\star$	30.00	35.00	37.00	63.64	72.73	72.73	63.64	37.50	25.00	12.50	12.50
GARCH sk $\mathcal{S}$	24.25 $\star$	26.75	31.25	32.00	45.45	45.45	45.45	45.45	25.00	12.50	12.50	12.50
GJR $\mathcal{N}$	14.75	44.00	14.75	47.75	63.64	90.91	63.64	90.91	25.00	37.50	12.50	37.50
GJR sk $\mathcal{N}$	10.25	33.75	11.50	41.25	9.09	54.55	0.00	54.55	25.00	25.00	12.50	25.00
GJR $\mathcal{S}$	9.75	13.00	13.75	19.25	36.36	27.27	54.55	54.55	25.00	25.00	25.00	25.00
GJR sk $\mathcal{S}$	11.75	12.00	12.25	17.75	9.09	9.09	0.00	0.00	12.50	25.00	12.50	25.00
<i>Panel D: DQ 5%-VaR</i>												
GARCH $\mathcal{N}$	14.50 $\star$	23.75	17.75	29.50	36.36	54.55	18.18	36.36	0.00	12.50	0.00	0.00
GARCH sk $\mathcal{N}$	7.50 $\star$	27.25	9.75	29.50	9.09	9.09	9.09	9.09	0.00	0.00	0.00	0.00
GARCH $\mathcal{S}$	5.75 $\star$	10.00	13.75	21.75	36.36	54.55	54.55	54.55	0.00	25.00	12.50	0.00
GARCH sk $\mathcal{S}$	3.00 $\star$	9.75	9.25	12.75	0.00	18.18	18.18	9.09	0.00	0.00	0.00	0.00
GJR $\mathcal{N}$	1.00 $\star$	12.75	3.50	16.50	0.00	9.09	9.09	9.09	0.00	0.00	0.00	0.00
GJR sk $\mathcal{N}$	1.00 $\star$	11.75	3.50	15.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
GJR $\mathcal{S}$	0.75	0.00	1.75	3.50	0.00	9.09	9.09	36.36	37.50	25.00	0.00	12.50
GJR sk $\mathcal{S}$	0.50	0.25	0.25	2.50	0.00	9.09	0.00	9.09	0.00	0.00	0.00	0.00

**Table 3: QL and wCRPS pairwise performance results**

The table presents the forecasting results of the QL test (at  $\alpha = 1\%$  and  $\alpha = 5\%$ ) and wCRPS test for the universes of 400 stocks, 11 stock indices and 8 currencies. We report the average Diebold–Mariano (DM) test statistics (Diebold and Mariano, 1995) computed with robust HAC standard errors, for the time series in the various universes. Negative values indicate outperformance of the Markov–switching specification compared with single–regime models. In light (dark) gray, we report statistics which are significantly negative (positive) at the 1% level. Models are estimated by MCMC.

	Stocks			Stock indices			Currencies		
	QL 1%	QL 5%	wCRPS	QL 1%	QL 5%	wCRPS	QL 1%	QL 5%	wCRPS
GARCH $\mathcal{N}$	1.01	-3.51	-9.49	-0.81	2.22	-5.11	0.27	1.16	-2.80
GARCH sk $\mathcal{N}$	0.49	-4.21	-9.75	-2.36	-1.70	-5.45	0.29	-0.06	-4.11
GARCH $\mathcal{S}$	-2.94	-2.95	-2.70	-0.47	-0.24	-2.30	1.45	0.40	-2.85
GARCH sk $\mathcal{S}$	1.38	-3.22	-2.00	-0.31	0.27	-0.92	1.05	-1.06	-1.27
GJR $\mathcal{N}$	0.64	-4.99	-9.86	0.52	0.76	-2.37	1.07	1.18	-2.25
GJR sk $\mathcal{N}$	1.10	-5.25	-10.08	-1.78	0.23	-2.95	1.52	0.33	-2.31
GJR $\mathcal{S}$	0.71	-3.70	-2.91	-1.41	-0.58	-0.73	0.35	-0.40	-0.95
GJR sk $\mathcal{S}$	0.01	-1.81	-1.63	0.01	-0.42	-2.11	-0.44	-1.51	-1.56

**Table 4: wCRPS pairwise performance results for all specifications**

The table presents the forecasting results of the wCRPS tests for the universes of 400 stocks (Panel A), 11 stock indices (Panel B) and 8 currencies (Panel C). We report the average Diebold–Mariano (DM) test statistics (Diebold and Mariano, 1995) computed with robust HAC standard errors, for the time series in the various universes. Negative values indicate outperformance of the Markov-switching specification compared with single-regime models. In light (dark) gray, we report statistics which are significantly negative (positive) at the 1% level. Models are estimated by MCMC.

MS/SR	GARCH				GJR			
	$\mathcal{N}$	sk $\mathcal{N}$	$\mathcal{S}$	sk $\mathcal{S}$	$\mathcal{N}$	sk $\mathcal{N}$	$\mathcal{S}$	sk $\mathcal{S}$
<i>Panel A: Stocks</i>								
GARCH $\mathcal{N}$	-9.49	-9.67	4.77	4.83	-7.48	-7.43	2.55	2.44
GARCH sk $\mathcal{N}$	-9.55	-9.75	4.39	4.60	-7.58	-7.56	2.36	2.27
GARCH $\mathcal{S}$	-9.80	-10.01	-2.70	-1.69	-8.34	-8.37	-1.17	-1.40
GARCH sk $\mathcal{S}$	-9.54	-9.76	-2.86	-2.00	-8.18	-8.22	-1.19	-1.44
GJR $\mathcal{N}$	-8.93	-9.02	2.86	2.99	-9.86	-10.04	4.92	4.69
GJR sk $\mathcal{N}$	-9.03	-9.19	2.57	2.73	-9.86	-10.08	4.40	4.31
GJR $\mathcal{S}$	-9.50	-9.66	-0.86	-0.61	-10.24	-10.44	-2.91	-3.45
GJR sk $\mathcal{S}$	-9.28	-9.47	-0.25	-0.02	-10.04	-10.28	-1.07	-1.63
<i>Panel B: Stock indices</i>								
GARCH $\mathcal{N}$	-5.11	-2.82	1.22	4.63	-1.16	1.27	2.29	3.39
GARCH sk $\mathcal{N}$	-6.27	-5.45	-2.56	0.81	-2.92	-0.94	-0.07	1.07
GARCH $\mathcal{S}$	-7.11	-5.21	-2.30	2.24	-2.51	-0.27	0.64	1.77
GARCH sk $\mathcal{S}$	-7.05	-6.72	-4.08	-0.92	-3.47	-1.69	-0.86	0.17
GJR $\mathcal{N}$	-3.32	-1.91	0.31	2.25	-2.37	1.68	3.64	4.71
GJR sk $\mathcal{N}$	-5.47	-4.59	-2.31	-0.42	-5.36	-2.95	-1.48	0.31
GJR $\mathcal{S}$	-5.33	-4.07	-1.88	0.11	-5.48	-2.08	-0.73	1.16
GJR sk $\mathcal{S}$	-6.66	-6.00	-3.71	-1.88	-6.75	-4.97	-3.71	-2.11
<i>Panel C: Currencies</i>								
GARCH $\mathcal{N}$	-2.80	-4.84	4.81	4.34	-3.22	-3.75	3.40	1.81
GARCH sk $\mathcal{N}$	-1.48	-4.11	5.44	6.89	-2.12	-2.88	3.95	3.21
GARCH $\mathcal{S}$	-6.64	-7.46	-2.85	-2.29	-7.13	-6.95	-2.94	-4.03
GARCH sk $\mathcal{S}$	-5.34	-7.34	-1.14	-1.27	-5.78	-6.55	-1.56	-3.20
GJR $\mathcal{N}$	-1.15	-2.85	5.11	4.40	-2.25	-2.61	4.90	3.02
GJR sk $\mathcal{N}$	-0.58	-2.77	5.61	6.38	-1.60	-2.31	5.39	4.43
GJR $\mathcal{S}$	-4.61	-5.89	-0.19	-0.08	-5.93	-5.88	-0.95	-2.06
GJR sk $\mathcal{S}$	-4.08	-5.91	0.35	0.48	-5.10	-5.70	-0.28	-1.56

# **MSGARCH vignette**

[https://papers.ssrn.com/sol3/papers2.cfm?abstract\\_id=2845809](https://papers.ssrn.com/sol3/papers2.cfm?abstract_id=2845809)

# Markov–Switching GARCH Models in R: The MSGARCH Package

**David Ardia**  
University of Neuchâtel  
Laval University

**Keven Bluteau**  
University of Neuchâtel

**Kris Boudt**  
Vrije Universiteit Brussel  
Vrije Universiteit Amsterdam

**Denis–Alexandre Trottier**  
Laval University

## PRELIMINARY

---

### Abstract

Markov-switching GARCH models have become popular to model the structural break in the conditional variance dynamics of financial time series. In this paper, we describe the R package **MSGARCH** which implements Markov-switching GARCH-type models very efficiently by using C++ object-oriented programming techniques. It allows the user to perform simulations as well as Maximum Likelihood and Bayesian estimation of a very large class of Markov-switching GARCH-type models. Risk management tools such as Value-at-Risk and Expected-Shortfall calculations are available. An empirical illustration of the usefulness of the R package **MSGARCH** is presented.

*Keywords:* GARCH, MSGARCH, Markov-switching, conditional volatility, risk management, R software.

---

## 1. Introduction

Modeling the volatility of financial markets is central in risk management. A seminal contribution in this field was the development of the GARCH model by [Bollerslev \(1986\)](#) where the volatility is a function of past asset returns. The GARCH model is today a widespread tool in risk management. However, recent studies show that estimates of GARCH models can be biased by structural breaks in the volatility dynamics ([Bauwens \*et al.\* 2010, 2014](#)). These structural breaks typically occur during periods of financial turmoil. Estimating a GARCH model on data displaying a structural break yields a non-stationary estimated model and implies poor risk predictions. A way to cope with this problem is provided by Markov-switching GARCH models (MSGARCH) whose parameters vary over time according to a latent discrete Markov process. These models can quickly adapt to variations in the unconditional volatility level, which improves risk predictions ([Ardia 2008](#)).

Following the seminal work of [Hamilton and Susmel \(1994\)](#), different parametrizations have

been proposed to account for discrete changes in the GARCH parameters, for instance by Dueker (1997), Gray (1996) and Klaassen (2002). However, these parametrizations for the conditional variance process lead to computational difficulties. Indeed, the evaluation of the likelihood function for a sample of length  $T$  in the case of  $K$  states requires the integration over all  $K^T$  possible paths, rendering the estimation infeasible.

In order to avoid any difficulties related to the past infinite history of the state variable, we adopt the parametrization due to Haas *et al.* (2004b). In their model, the authors hypothesize  $K$  separate GARCH(1,1) processes for the conditional variance of the MSGARCH process. In addition to its appealing computational aspects, the MSGARCH model of Haas *et al.* (2004b) has conceptual advantages. In effect, one reason for specifying Markov–switching models that allow for different GARCH behavior in each regime is to capture the difference in the variance dynamics in low– and high–volatility periods.

The R package **MSGARCH** aims to provide a comprehensive set of methods for the estimation, the simulation and the forecasting of MSGARCH models. Also, methods for risk management such as Value-at-Risk and Expected-Shortfall calculations are available. The R package **MSGARCH** uses object-oriented programming techniques to efficiently implement the estimation of such models, which is a computationally challenging task. This is crucial in risk management for instance, in the case of large-scale backtesting. To our knowledge, there is currently no such R package to implement **MSGARCH** models. The R package **MSGARCH** is available from the CRAN repository at <https://cran.r-project.org/package=MSGARCH> and the development version is available at <https://github.com/keblu/MSGARCH>.

In this vignette, we describe the models and the functions/methods available in the package.

## 2. Model specification

This section illustrates how to create MSGARCH specifications using the function `create.spec`.

As a first example, let us consider a GARCH conditional volatility model with two different regimes in which the conditional distribution of the log-returns is assumed to be Normal. This model can be created with the R package **MSGARCH** as follows:

```
R> spec = MSGARCH::create.spec(model = c("sGARCH", "sGARCH"),
+   distribution = c("norm", "norm"),
+   do.skew = c(FALSE, FALSE), do.mix = FALSE, do.shape.ind = FALSE)
```

The arguments are described below:

- **model**: conditional volatility model in each regime (vector). Valid models are "sGARCH", "eGARCH", "gjrGARCH", "tGARCH", and "GAS" (see Section 2.1 for details).
- **distribution**: conditional distribution in each regime (vector). Available distributions are "norm", "std", and "ged" (see Section 2.2 for details).
- **do.skew**: boolean indicating whether skewness should be allowed in the distributions.
- **do.mix**: boolean indicating whether a Mixture of GARCH processes should be used instead of a MSGARCH process.

- `do.shape.ind`: boolean indicating whether the distributions' parameters should be constrained to be the same across the different regimes.

We refer the reader to the documentation for more details; see `help("create.spec")`.

#### *Example 1: A single-regime model*

The R package **MSGARCH** also supports single-regime models as they are the building blocks for regime-switching models. The simplest specification we can build is a GARCH model with a symmetric Normal conditional distribution:

```
R> spec = MSGARCH::create.spec(model = "sGARCH", distribution = "norm",
+   do.skew = FALSE)
```

#### *Example 2: A model with heterogeneous regimes*

The user can technically create any MSGARCH specification by selecting the desired single-regime scedastic models and conditional distributions.<sup>1</sup> Here is an example of a three-state MSGARCH process where each regime is characterized by a different conditional volatility model and a different conditional distribution:

```
R> spec = MSGARCH::create.spec(model = c("sGARCH", "tGARCH", "eGARCH"),
+   distribution = c("norm", "std", "ged"),
+   do.skew = c(TRUE, FALSE, TRUE), do.mix = FALSE, do.shape.ind = FALSE)
```

#### *Summary of the specification object*

The output of the function `create.spec` is a list of class `MSGARCH_SPEC` containing various functions and variables. The relevant information is summarized with `print` or `summary`:

```
R> spec = MSGARCH::create.spec()
R> summary(spec)
```

```
[1] "Specification Type: Markov-Switching "
```

```
[1] "Specification Name: sGARCH_normal_sym sGARCH_normal_sym"
```

```
[1] "Number of parameters in each variance model: 3 3"
```

```
[1] "Number of parameters in each distribution: 0 0"
```

```
[1] "Default parameters:"
```

	alpha0_1	alpha1_1	beta_1	alpha0_2	alpha1_2	beta_2	P	P
[1,]	0.1	0.1	0.8	0.1	0.1	0.8	0.5	0.5

The specification provides information regarding the model such as the type (Markov-switching or mixture), the GARCH-type specification within each regime, the number of variance parameters in each regime as well as the number of shape parameters in each regime. The last line in the specification outputs the parameters in each regime as well as the transition probabilities. We refer the reader to the documentation for details; see `help("create.spec")`.

<sup>1</sup>As discussed in Section 3, complex models are however more difficult to estimate.



## 2.1. Conditional volatility models

The building blocks of the regime-switching models are the single-regime specifications. In this section we review the conditional volatility models available in the R package **MSGARCH**. A summary can be found in Table 1.

The log-return at time  $t$  is assumed to be of the form

$$y_t = \eta_t h_t^{1/2}, \quad (1)$$

where  $h_t^{1/2}$  is the conditional volatility at time  $t$ , and  $\{\eta_t\}_{t \geq 0} \stackrel{i.i.d.}{\sim} \mathcal{D}(0, 1, \lambda)$ . Here,  $\mathcal{D}(0, 1, \lambda)$  denotes a distribution with zero mean, unit variance, and shape parameters  $\lambda$ .

All single-regime conditional volatility models presented below are one-lag processes. This approach proved to be an effective specification to capture the volatility clustering observed in financial data. Moreover, it reduces the model's complexity.

### GARCH model

The GARCH model of [Bollerslev \(1986\)](#) is given by:

$$h_t \equiv \alpha_0 + \alpha_1 y_{t-1}^2 + \beta h_{t-1}. \quad (2)$$

To ensure positivity, we require  $\alpha_0 > 0$ ,  $\alpha_1 \geq 0$ ,  $\beta \geq 0$ . Covariance-stationarity is obtained by adding the condition  $\alpha_1 + \beta < 1$ . To create a single-regime GARCH specification we use `model = "sGARCH"` in the function `create.spec`.

### EGARCH model

The Exponential GARCH (EGARCH) of [Nelson \(1991\)](#) is given by:

$$\ln(h_t) \equiv \alpha_0 + \alpha_1 (|y_{t-1}| - \mathbb{E}[|y_{t-1}|]) + \alpha_2 y_{t-1} + \beta \ln(h_{t-1}). \quad (3)$$

This model takes into consideration the leverage effect where past negative returns have a larger influence on the conditional volatility than past positive returns of the same magnitude ([Black 1976](#); [Christie 1982](#)). The persistence of the models is captured by the coefficient  $\beta$ . We set  $\beta < 1$  to ensure stationarity. The creation of a single-regime EGARCH specification is done by using `model = "eGARCH"` in the function `create.spec`.

### GJR model

The GJR model of [Glosten et al. \(1993\)](#) is also able to capture the asymmetry in the conditional volatility process. This model is given by:

$$h_t \equiv \alpha_0 + \alpha_1 y_{t-1}^2 + \alpha_2 y_{t-1}^2 \mathbb{I}_{\{y_{t-1} < 0\}} + \beta h_{t-1}, \quad (4)$$

where  $\mathbb{I}_{\{\cdot\}}$  is the indicator function. The parameter  $\alpha_2$  controls the degree of asymmetry in the conditional volatility response to the past shock. To ensure positivity, we set  $\alpha_0 > 0$ ,  $\alpha_1 \geq 0$ ,  $\alpha_2 \geq 0$ ,  $\beta \geq 0$  (sufficient condition). To ensure covariance-stationarity we need the condition  $\alpha_1 + \alpha_2 \mathbb{E}[\eta^2 \mathbb{I}_{\{\eta < 0\}}] + \beta < 1$ . The single-regime GJR specification is created by using `model = "gjrGARCH"` in the function `create.spec`.

### *TGARCH model*

Zakoian (1994) introduces the TGARCH which has the conditional volatility as dependent variable instead of the conditional variance:

$$h_t^{1/2} \equiv \alpha_0 + \alpha_1 y_{t-1} \mathbb{I}_{\{y_{t-1} \geq 0\}} + \alpha_2 y_{t-1} \mathbb{I}_{\{y_{t-1} < 0\}} + \beta h_{t-1}^{1/2}. \quad (5)$$

For positivity we set  $\alpha_0 > 0$ ,  $\alpha_1 \geq 0$ ,  $\alpha_2 \geq 0$  and  $\beta \geq 0$ . To ensure covariance–stationarity, we require the condition  $\alpha_1^2 + \beta^2 - 2\beta(\alpha_1 + \alpha_2)\mathbb{E}[\eta \mathbb{I}_{\{\eta < 0\}}] - (\alpha_1^2 - \alpha_2^2)\mathbb{E}[\eta^2 \mathbb{I}_{\{\eta < 0\}}] < 1$  (see Francq and Zakoian 2011, Section 10.2). The single–regime TGARCH specification is created by using `model = "tGARCH"` in function `create.spec`.

### *GAS model*

Generalized Autoregressive Score models were proposed in their full generality in Creal *et al.* (2013). They provide a general framework for modeling time variation in parametric models. The GAS model can be written as:

$$h_t \equiv \alpha_0 + \alpha_1 s_{t-1} + \beta h_{t-1}, \quad (6)$$

where:

$$s_{t-1} \equiv S_{t-1} \nabla_{t-1}, \quad \nabla_{t-1} \equiv \frac{\partial \ln f(y_{t-1} | h_{t-1}, \lambda)}{\partial h_{t-1}}, \quad S_{t-1} \equiv \mathbb{E}[\nabla_{t-1} \nabla_{t-1}']^{-1}. \quad (7)$$

Here,  $f(y_{t-1} | h_{t-1}, \lambda)$  is the likelihood (PDF) of observation  $y_{t-1}$  given the volatility  $h_{t-1}$  and the distribution's shape parameters  $\lambda$ , and  $s_{t-1}$  is the score of the observation. The single–regime GAS model is created by using `model = "GAS"` in the function `create.spec`.

## 2.2. Conditional distributions

We present here the conditional distributions available in the R package **MSGARCH**. There are two functions directly related to the conditional distribution:

- **pdf**: computes the probability density function (PDF).
- **cdf**: computes the cumulative density function (CDF).

We refer the reader to the documentation manual for further details; see `help("create.spec")`.

Each distribution presented below is standardized to have zero mean and unit variance, and is symmetric about the origin.

### *The Normal distribution*

The PDF of the standard Normal distribution is:

$$f_N(z) \equiv \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}z^2}, \quad z \in \mathbb{R}. \quad (8)$$

This distribution is selected with the argument `distribution = "norm"` in `create.spec`.

### The Student- $t$ distribution

The PDF of the standardized Student- $t$  distribution is given by:

$$f_S(z; \nu) \equiv \sqrt{\frac{\nu}{\nu-2}} \frac{\Gamma(\frac{\nu+1}{2})}{\sqrt{\nu\pi} \Gamma(\frac{\nu}{2})} \left(1 + \frac{z^2}{\nu}\right)^{-\frac{\nu+1}{2}}, \quad z \in \mathbb{R}, \quad (9)$$

where  $\Gamma(\cdot)$  is the Gamma function. We impose a shape parameter  $\nu > 2$  to ensure that the second order moment exists. The kurtosis of this distribution is higher for lower  $\nu$ .<sup>2</sup> This distribution is selected with the argument `distribution = "std"` in the function `create.spec`.

### The GED distribution

The PDF of the standardized generalized error distribution (GED) is given by

$$f_{\text{GED}}(z; \nu) \equiv \frac{\nu e^{-\frac{1}{2}|z/\lambda|^\nu}}{\lambda 2^{(1+1/\nu)} \Gamma(1/\nu)}, \quad \lambda \equiv \left( \frac{\Gamma(1/\nu)}{4^{1/\nu} \Gamma(3/\nu)} \right)^{1/2}, \quad z \in \mathbb{R}, \quad (10)$$

where  $\nu > 0$  is the shape parameter.<sup>3</sup> This distribution is selected with the argument `distribution = "ged"` in the function `create.spec`.

### Skewed distributions

Fernández and Steel (1998) provide a simple way to introduce skewness into any unimodal standardized distribution. Trottier and Ardia (2016) derive the moments of the standardized Fernandez–Steel skewed distributions which are needed in the estimation of the GJR, EGARCH, and TGARCH models. We refer the reader to Trottier and Ardia (2016) for details. In this approach, the parameter  $\xi$  ( $0 < \xi < \infty$ ) measures the degree of asymmetry. To create any specification with skewed distribution we use the argument `do.skew = TRUE` in the function `create.spec`.

## 2.3. Multiple-regime specifications

We present in this section the two multiple-regime specifications available in the R package **MSGARCH**.

### Markov-switching GARCH

Let  $\{\Delta_t\}_{t \geq 0}$  be a time-homogeneous Markov chain with a finite state space  $S \equiv \{1, \dots, K\}$  and an irreducible and primitive  $K \times K$  transition matrix  $\mathbf{P}$ :

$$\mathbf{P} \equiv \begin{bmatrix} p_{1,1} & \cdots & p_{K,1} \\ \vdots & \ddots & \vdots \\ p_{1,K} & \cdots & p_{K,K} \end{bmatrix}, \quad (11)$$

<sup>2</sup>For  $\nu = \infty$ , the Student- $t$  distribution is equivalent to the Normal distribution.

<sup>3</sup>Special cases of this distribution are obtained for  $\nu = 1$  (Laplace distribution) and  $\nu = 2$  (Normal distribution). The uniform distribution is obtained in the limit  $\nu \rightarrow \infty$ .

---

**Conditional volatility models**

GARCH model (`model = "sGARCH"`)

$$h_t \equiv \alpha_0 + \alpha_1 y_{t-1}^2 + \beta h_{t-1}$$

EGARCH model (`model = "eGARCH"`)

$$\ln(h_t) \equiv \alpha_0 + \alpha_1 (|y_{t-1}| - \mathbb{E}[|y_{t-1}|]) + \alpha_2 y_{t-1} + \beta \ln(h_{t-1})$$

GJR model (`model = "gjrGARCH"`)

$$h_t \equiv \alpha_0 + \alpha_1 y_{t-1}^2 + \alpha_2 y_{t-1}^2 \mathbb{I}_{\{y_{t-1} < 0\}} + \beta h_{t-1}$$

TGARCH model (`model = "tGARCH"`)

$$h_t^{1/2} \equiv \alpha_0 + \alpha_1 y_{t-1} \mathbb{I}_{\{y_{t-1} \geq 0\}} + \alpha_2 y_{t-1} \mathbb{I}_{\{y_{t-1} < 0\}} + \beta h_{t-1}^{1/2}$$

GAS model (`model = "GAS"`)

$$h_t \equiv \alpha_0 + \alpha_1 s_{t-1} + \beta h_{t-1},$$

$$s_{t-1} \equiv S_{t-1} \nabla_{t-1}, \quad \nabla_{t-1} \equiv \frac{\partial \ln f(y_{t-1} | h_{t-1}, \lambda)}{\partial h_{t-1}}, \quad S_{t-1} \equiv \mathbb{E}[\nabla_{t-1} \nabla_{t-1}']^{-1}$$


---

**Conditional distributions**

Normal distribution (`distribution = "norm"`)

$$f_N(z) \equiv \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}z^2}, \quad z \in \mathbb{R}$$

Student-*t* distribution (`distribution = "std"`)

$$f_S(z; \nu) \equiv \sqrt{\frac{\nu}{\nu-2}} \frac{\Gamma(\frac{\nu+1}{2})}{\sqrt{\nu\pi} \Gamma(\frac{\nu}{2})} \left(1 + \frac{z^2}{\nu}\right)^{-\frac{\nu+1}{2}}, \quad z \in \mathbb{R}$$

GED distribution (`distribution = "ged"`)

$$f_{\text{GED}}(z; \nu) \equiv \frac{\nu e^{-\frac{1}{2}|z/\lambda|^\nu}}{\lambda 2^{(1+1/\nu)} \Gamma(1/\nu)}, \quad \lambda \equiv \left(\frac{\Gamma(1/\nu)}{4^{1/\nu} \Gamma(3/\nu)}\right)^{1/2}, \quad z \in \mathbb{R}$$


---

Table 1: Available single-regime specifications. The conditional volatility models are presented in the top panel and the PDF of the innovations' distributions are presented in the bottom panel. Each distribution is symmetric about the origin. Skewed versions of the distributions are obtained with the argument `do.skew = TRUE` in the function `create.spec`.

where  $0 \leq p_{i,j} \leq 1$  is the probability of a transition from state  $\Delta_{t-1} = i$  to state  $\Delta_t = j$ . By normalization it follows that  $\sum_{j=1}^K p_{i,j} = 1, \forall i \in \{1, \dots, K\}$ .

Let the log-return at time  $t$  be of the form:

$$y_t = \eta_{\Delta_t, t} h_{\Delta_t, t}^{1/2}, \quad (12)$$

where  $h_{k,t}$  is the conditional variance at time  $t$  under regime  $k \in \{1, \dots, K\}$ ,  $\{\eta_{k,t}\}_{t \geq 0} \stackrel{i.i.d.}{\sim} \mathcal{D}_k(0, 1, \lambda_k)$ ,  $\forall k \in \{1, \dots, K\}$ , and all innovations are independent. Here,  $\mathcal{D}_k(0, 1, \lambda_k)$  denotes the conditional distribution in regime  $k$ .

For the single-regime specification characterizing regime  $k$ , we let  $\theta_k$  be the parameters of the conditional variance process, and we let  $\lambda_k$  be the shape parameters of the conditional distribution  $\mathcal{D}_k$ . The realized time-series of the conditional variance process in regime  $k$  is denoted by  $\mathbf{h}_k \equiv (h_{k,1}, \dots, h_{k,T})'$ . The MSGARCH specification is constructed following the approach by Haas *et al.* (2004b), which consists of many distinct single-regime specifications evolving in parallel. We refer to Ardia (2008, Chapter 6) for more details.

#### *Example: Estimation of a Markov-switching model*

As an example, let us use the R package **MSGARCH** to create a two-state MSGARCH model for the log-returns of the S&P 500. We first create a two-state MSGARCH model,  $K = 2$ , from two single-regime GARCH processes each following a Normal distribution. We then fit the model to the `sp500` dataset which consists of the S&P 500 index closing value log-returns ranging from 1998-01-01 to 2015-12-31.

```
R> data("sp500")
R> spec = MSGARCH::create.spec(model = c("sGARCH", "sGARCH"),
+   distribution = c("norm", "norm"),
+   do.skew = c(FALSE, FALSE), do.mix = FALSE, do.shape.ind = FALSE)
R> out.mle = MSGARCH::fit.mle(spec = spec, y = sp500)
R> summary(out.mle)
```

```
[1] "Specification Type: Markov-Switching"
[1] "Specification Name: sGARCH_normal_sym sGARCH_normal_sym"
[1] "Number of parameters in each variance model: 3 3"
[1] "Number of parameters in each distribution: 0 0"
[1] "Default parameters:"
alpha0_1 alpha1_1 beta_1 alpha0_2 alpha1_2 beta_2    P    P
[1,]      0.1      0.1   0.8      0.1      0.1   0.8 0.5 0.5
[1] "DEoptim initialization: FALSE"
[1] "Fitted Parameters:"
alpha0_1 alpha1_1 beta_1 alpha0_2 alpha1_2 beta_2          P          P
[1,]  0.0016 0.02709 0.891  0.03792   0.1178 0.8777 1.976e-07 0.3686
[1] "Transition matrix:"
t = 1   t = 2
t + 1 = 1 1.976e-07 0.3686
t + 1 = 2 1.000e+00 0.6314
[1] "Stable probabilities:"
```

```

Stable probabilities
State 1          0.2693
State 2          0.7307
[1] "Unconditional volatility:"
State 1 State 2
[1,]  0.1398  2.901
Log-kernel:  -6507
AIC:  13236
BIC:  13287

```

The model is estimated by Maximum likelihood with the function `fit.mle` (see Section 3). The resulting parameters are collected in the vector `theta` where each parameter is labeled according to the model and its state. The function `transmat` is a helper function that builds the transition matrix from the fitted parameters for better readability.

The output summary returns various information regarding the fit. In the case of the ML estimation, the fitted parameters are returned. The summary also returns the stable (unconditional probabilities) and the unconditional volatility of each regime. It also reports the log-kernel and the Akaike and Bayesian information criteria computed at the ML optimum.

### Mixture of GARCH

Haas *et al.* (2004a) propose a general class of Mixture of GARCH models. They specify a Mixture of Normal distributions where the variance process of each Normal component is a GARCH process. They name this new class the MNGARCH models. A special case of this specification, named the Full and Diagonal MNGARCH, is encountered when the covariances between all components are constrained to be zero. This special case has a direct relationship with the MSGARCH model. Indeed, we can constrain the transition matrix  $\mathbf{P}$  of the MSGARCH model so that the probability  $p_{i,j}$  of a transition from state  $\Delta_{t-1} = i$  to state  $\Delta_t = j$  is the same for all  $i \in \{1, \dots, K\}$ , *i.e.*,  $p_{i,j} = p_{k,j}$ ,  $\forall i, k \in \{1, \dots, K\}$ . The transition matrix can then be described by a vector  $[p_1, \dots, p_K]$ . This constraint effectively converts the regime-switching model to a Mixture model.

For demonstration, let us repeat the experiment done in the previous section now with the argument `do.mix = TRUE`.

```

R> spec = MSGARCH::create.spec(model = c("sGARCH", "sGARCH"),
+   distribution = c("norm", "norm"),
+   do.skew = c(FALSE, FALSE), do.mix = TRUE, do.shape.ind = FALSE)
R> out.mle = MSGARCH::fit.mle(spec = spec, y = sp500)
R> summary(out.mle)

[1] "Specification Type: Mixture"
[1] "Specification Name: sGARCH_normal_sym sGARCH_normal_sym"
[1] "Number of parameters in each variance model: 3 3"
[1] "Number of parameters in each distribution: 0 0"
[1] "Default parameters:"
alpha0_1 alpha1_1 beta_1 alpha0_2 alpha1_2 beta_2 P
[1,]      0.1      0.1    0.8      0.1      0.1    0.8 0.5

```

```

[1] "DEoptim initialization: FALSE"
[1] "Fitted Parameters:"
alpha0_1 alpha1_1 beta_1 alpha0_2 alpha1_2 beta_2      P
[1,] 0.001127 0.01747 0.8976 0.02929 0.1064 0.8884 0.1954
[1] "Stable probabilities:"
Stable probabilities
State 1          0.1954
State 2          0.8046
[1] "Unconditional volatility:"
State 1 State 2
[1,] 0.1152 2.368
Log-kernel: -6518
AIC: 13201
BIC: 13246

```

We can observe that we have less parameters labelled as P (where P is the matrix containing the transition probabilities) since a Mixture of GARCH will always have less parameters than a Markov-switching GARCH process. For a Mixture of GARCH, the `transmat` function will output a probability vector rather than a probability matrix.

### *Regime-independent shape parameters*

Sometimes it is useful to have a regime-switching behavior only in the conditional variance and keep the same conditional distribution across all regimes. In this situation, we will say that the shape parameters are *regime-independent*, since all distributions  $\mathcal{D}_k$  in  $\mathbf{D}$  and shape parameters  $\lambda_k$  in  $\Lambda$  are restricted to be the same (*i.e.*, the regimes only differ in terms of the conditional variance process). This can be done by setting `do.shape.ind = TRUE`.

We now illustrate this in a two-state regime-switching GARCH model for which the conditional Student-*t* distribution is the same in both regimes.

```

R> spec = MSGARCH::create.spec(model = c("sGARCH", "sGARCH"),
+   distribution = c("std", "std"),
+   do.skew = c(FALSE, FALSE), do.mix = FALSE, do.shape.ind = TRUE)
R> out.mle = MSGARCH::fit.mle(spec = spec, y = sp500)
R> summary(out.mle)

[1] "Specification Type: Markov-Switching with Regime-Independent distribution"
[1] "Specification Name: sGARCH_student_sym sGARCH_student_sym"
[1] "Number of parameters in each variance model: 3 3"
[1] "Number of parameters in distribution: 1"
[1] "Default parameters:"
alpha0_1 alpha1_1 beta_1 alpha0_2 alpha1_2 beta_2 nu_1  P  P
[1,] 0.1 0.1 0.8 0.1 0.1 0.8 10 0.5 0.5
[1] "DEoptim initialization: FALSE"
[1] "Fitted Parameters:"
alpha0_1 alpha1_1 beta_1 alpha0_2 alpha1_2 beta_2 nu_1      P      P
[1,] 1e-04 0.02049 0.8767 0.02231 0.09955 0.8953 15.47 1.11e-16 0.1671

```

```

[1] "Transition matrix:"
t = 1  t = 2
t + 1 = 1 1.11e-16 0.1671
t + 1 = 2 1.00e+00 0.8329
[1] "Stable probabilities:"
Stable probabilities
State 1          0.1432
State 2          0.8568
[1] "Unconditional volatility:"
State 1 State 2
[1,] 0.03119  2.074
Log-kernel: -6506
AIC:  13120
BIC:  13178

```

As we can see, the output only contains one parameter `nu_1` with no regime indication instead of two parameters `nu_1` and `nu_2`.

### 3. Estimation

In the R package **MSGARCH**, estimation of single-regime and Markov-switching GARCH models can be either done by Maximum Likelihood (ML) or via Markov chain Monte Carlo (MCMC) simulation. In both cases, the key is the function `kernel`, which returns the sum of the (log-)likelihood and the prior. More precisely, we have  $\text{kernel}(\Theta) = L(\mathbf{y}|\Theta, \Lambda, \mathbf{P}) + \text{prior}(\Theta) + \text{prior}(\Lambda) + \text{prior}(\mathbf{P})$  where  $L$  is the (log-)likelihood of  $\mathbf{y}$  given the parameter  $\Theta$ ,  $\Lambda$ , and  $\mathbf{P}$ . We follow [Ardia \(2008\)](#) and use non-informative truncated Normal priors. Moreover, the prior ensures that the  $\Theta$  makes the conditional variance processes positive and stationary, that  $\Lambda$  respects the parameters bounds of all the conditional distributions, and that the sum of columns of  $\mathbf{P}$  are equal to one in the case of a Markov-switching models. If any of these conditions is not respected, the prior returns `-1e10`. For details on the ML or Bayesian estimation via MCMC techniques, we refer the reader to [Ardia \(2008\)](#).

#### *Maximum likelihood estimation*

Obtaining the ML estimator of Markov-switching specifications using a standard optimization technique can be a difficult task in practice. We therefore rely on the following sequential strategy for the estimation. First, the optimization is performed using Nelder-Mead algorithm for derivative-free optimization with the function `nmkb` from the **dfoptim** package ([Varadhan and Borchers 2016](#)). If convergence is not achieved, the optimization is restarted using the (almost) original Nelder-Mead simplex algorithm with the function `neldermead` from the **nloptr** package ([Johnson 2014](#)). Finally, if the convergence has still failed, we rely on Differential Evolution ([Price et al. 2006](#)) as implemented in the **DEoptim** package ([Ardia et al. 2015](#)).

The list of control parameters `ctr` can also ensure that **DEoptim** is used as a first step to define good starting value for the optimization. The control parameter `do.enhance.theta0` performs volatility estimation over a rolling window and sets some parameters of the conditional variance in each state such that they match empirical quantiles of the volatility



distribution. This ad-hoc procedure has proved to be a good way to set the starting values. We refer the reader to the documentation for details; see `help("fit.mle")`.

```
R> data("sp500")
R> spec = MSGARCH::create.spec(model = c("sGARCH", "sGARCH"),
+   distribution = c("std", "std"),
+   do.skew = c(FALSE, FALSE), do.mix = FALSE, do.shape.ind = FALSE)
R> out.mle = MSGARCH::fit.mle(spec = spec, y = sp500)
R> summary(out.mle)
```

```
[1] "Specification Type: Markov-Switching"
[1] "Specification Name: sGARCH_student_sym sGARCH_student_sym"
[1] "Number of parameters in each variance model: 3 3"
[1] "Number of parameters in each distribution: 1 1"
[1] "Default parameters:"
alpha0_1 alpha1_1 beta_1 nu_1 alpha0_2 alpha1_2 beta_2 nu_2 P P
[1,] 0.1 0.1 0.8 10 0.1 0.1 0.8 10 0.5 0.5
[1] "DEoptim initialization: FALSE"
[1] "Fitted Parameters:"
alpha0_1 alpha1_1 beta_1 nu_1 alpha0_2 alpha1_2 beta_2 nu_2 P
[1,] 0.0003284 0.0719 0.928 2.374 0.02023 0.09678 0.898 20.45 0.02864
P
[1,] 0.237
[1] "Transition matrix:"
t = 1 t = 2
t + 1 = 1 0.02864 0.237
t + 1 = 2 0.97136 0.763
[1] "Stable probabilities:"
Stable probabilities
State 1 0.1961
State 2 0.8039
[1] "Unconditional volatility:"
State 1 State 2
[1,] 1.812 1.962
Log-kernel: -6506
AIC: 13063
BIC: 13128
```

The optimization was run with default parameters. They can be modified along the following input parameters:

- `do.init` indicates if there is a pre-optimization with the R package **DEoptim**. Please refer to the **DEoptim** documentation for more details; see `help("DEoptim")`. (Default: `do.init = FALSE`)
- `NP` sets the number of vector of parameters in the population. (Default: `NP = 200`)

- `itermax` sets the maximum number of iterations (number of populations generated). (Default: `maxit = 200`)
- `theta0` is the starting value for the chain (if empty the specification default value is used).
- `do.enhance.theta0` uses the volatilities of rolling windows of `y` and adjusts the default parameters so that the unconditional volatility of each regime is set to different quantiles of the volatilities obtained with rolling windows on `y`. (Default: `do.enhance.theta0 = TRUE`)

### Bayesian estimation

To perform Bayesian estimation we use the adaptive Metropolis–Hastings sampler described in [Vihola \(2012\)](#) and available in the R package `adaptMCMC` ([Andreas 2012](#)). Various input parameters can be used to design the MCMC experiment via the list of control parameters `ctr`.

```
R> ctr.bay = list(N.burn = 5000, N.mcmc = 10000, N.thin = 10)
R> set.seed(123)
R> out.bay = MSGARCH::fit.bayes(spec = spec, y = sp500, ctr = ctr.bay)
R> summary(out.bay)
```

```
[1] "Specification Type: Markov-Switching"
[1] "Specification Name: sGARCH_student_sym sGARCH_student_sym"
[1] "Number of parameters in each variance model: 3 3"
[1] "Number of parameters in each distribution: 1 1"
[1] "Default parameters:"
alpha0_1 alpha1_1 beta_1 nu_1 alpha0_2 alpha1_2 beta_2 nu_2 P P
[1,]      0.1      0.1   0.8  10      0.1      0.1   0.8  10 0.5 0.5
[1] "Bayesian posterior mean:"
alpha0_1 alpha1_1 beta_1 nu_1 alpha0_2 alpha1_2 beta_2 nu_2
0.01146 0.08429 0.90644 9.86376 0.55834 0.38338 0.49062 10.11514
P P
0.97443 0.67645
[1] "Posterior variance-covariance matrix"
alpha0_1 alpha1_1 beta_1 nu_1 alpha0_2 alpha1_2
alpha0_1 1.306e-05 2.052e-05 -2.920e-05 -5.745e-05 7.725e-05 6.534e-05
alpha1_1 2.052e-05 1.097e-04 -1.100e-04 -5.996e-05 1.389e-04 1.365e-04
beta_1 -2.920e-05 -1.100e-04 1.233e-04 6.688e-05 -1.002e-04 -1.218e-04
nu_1 -5.745e-05 -5.996e-05 6.688e-05 1.716e-02 -3.352e-03 -6.777e-03
alpha0_2 7.725e-05 1.389e-04 -1.002e-04 -3.352e-03 2.247e-02 1.131e-02
alpha1_2 6.534e-05 1.365e-04 -1.218e-04 -6.777e-03 1.131e-02 7.335e-03
beta_2 -9.290e-05 -3.238e-04 2.584e-04 1.597e-02 -1.948e-02 -1.400e-02
nu_2 -2.290e-05 -1.215e-04 1.008e-04 -4.591e-03 -1.058e-02 -3.305e-03
beta_2 nu_2 P P
alpha0_1 -9.290e-05 -0.0000229 3.352e-05 6.665e-05
```

```

alpha1_1 -3.238e-04 -0.0001215  5.504e-05  2.033e-04
beta_1    2.584e-04  0.0001008 -6.261e-05 -1.899e-04
nu_1      1.597e-02 -0.0045914  3.114e-04 -1.810e-02
alpha0_2 -1.948e-02 -0.0105786  7.371e-04  3.351e-03
alpha1_2 -1.400e-02 -0.0033052  2.850e-04  7.197e-03
beta_2    2.818e-02  0.0042056 -7.035e-05 -1.725e-02
nu_2      4.206e-03  0.0078102 -3.933e-04  4.995e-03
[ reached getOption("max.print") -- omitted 2 rows ]
[1] "Posterior mean transition matrix:"
t = 1  t = 2
t + 1 = 1 0.97443 0.6764
t + 1 = 2 0.02557 0.3236
[1] "Posterior mean stable probabilities:"
Stable probabilities
State 1          0.96357
State 2          0.03643
[1] "Posterior mean unconditional volatility:"
State 1 State 2
[1,]    1.112    2.105
Acceptance rate:  0.986
AIC:  13007
BIC:  13071
DIC:  13003

```

The function `fit.bayes` outputs summary results of the MCMC estimation, such as the posterior mean and posterior covariance matrix, and various information criteria. We refer the reader to the documentation for details; see `help("fit.bayes")`. The posterior sample is also an element of the output and exported as an `mcmc` object ([Plummer \*et al.\* 2006](#)):

```
R> out.bay$theta
```

Markov Chain Monte Carlo (MCMC) output:

```

Start = 1
End = 1000
Thinning interval = 1
alpha0_1 alpha1_1 beta_1  nu_1 alpha0_2 alpha1_2 beta_2  nu_2      P
[1,] 0.011845  0.08601 0.9054  9.892   0.4773   0.3331 0.60078 10.137 0.9896
[2,] 0.010333  0.09109 0.9036  9.956   0.4416   0.2993 0.66062 10.128 0.9926
[3,] 0.011945  0.10301 0.8912  9.918   0.4850   0.3368 0.57464 10.110 0.9938
[4,] 0.013571  0.09164 0.8965  9.960   0.4287   0.2919 0.67781 10.124 0.9783
[5,] 0.007648  0.08841 0.9069  9.888   0.4704   0.3310 0.55750 10.128 0.9586
[6,] 0.012823  0.08313 0.9040  9.866   0.4695   0.3341 0.56646 10.133 0.9469
[7,] 0.010310  0.08402 0.9102  9.710   0.5924   0.4396 0.34054 10.137 0.9592
[8,] 0.010870  0.09859 0.8948  9.795   0.5547   0.4035 0.41248 10.114 0.9666
P
[1,] 0.6385
[2,] 0.5827

```

```
[3,] 0.6398
[4,] 0.5735
[5,] 0.6618
[6,] 0.6664
[7,] 0.8388
[8,] 0.7690
[ reached getOption("max.print") -- omitted 992 rows ]
```

The function `fit.bayes` takes up to five controls arguments in `ctr`:

- `N.burn`: number of discarded draws. (Default: `N.burn = 5000`)
- `N.mcmc`: number of draws to keep. (Default: `N.mcmc = 10000`)
- `N.thin`: thinning factor. (Default: `N.thin = 10`)
- `theta0`: starting value for the chain (if empty the specification default value is used).
- `do.enhance.theta0`: boolean indicating whether to use a custom starting parameters or not. (Default: `do.enhance.theta0 = TRUE`)

The main purpose of `N.thin` is to diminish the auto-correlation in the MCMC chain. The argument `N.burn` also serves as pre-optimization step; this is why it is set to a large value in the example. One alternative is to use a custom starting parameter `theta0` in the `ctr` argument or to set `do.enhance.theta0 = TRUE`. For example, we could set `theta0` as the ML estimator obtained with `fit.mle`. The total length of the chain is: `N.mcmc / N.thin`. The chain is converted to a **coda** object meaning that all functions for MCMC analysis available in the R package **coda** (Plummer *et al.* 2006) are available. We refer the reader to the documentation for details; see `help("fit.bayes")`.

## 4. Other functionalities

Many functionalities are available in the R package **MSGARCH**, which allow the user to filter (functions `ht` and `Pstate`), to simulate (functions `sim` and `simahead`), to compute the predictive density (function `pred`) and the probability integral transform (function `pit`), or to compute risk measures such as the Value-at-Risk (VaR) or Expected-shortfall (ES) (function `risk`). We refer the reader to the documentation manual for details; see `help("MSGARCH")`.

In all cases, the object from the ML or Bayesian fit can be used as an input. In the case of the MCMC estimation, the functions return the aggregated value over MCMC draws, hence the *true* predictive distribution, and the VaR or ES which integrate the parameter uncertainty.

Finally, to perform in-sample model selection, information criteria such as the Aikaike (AIC) criterion (Akaike 1974), the Bayesian Information criterion (BIC) (Schwarz *et al.* 1978), and the Deviance Information criterion (DIC) (Gelman *et al.* 2014) are available. These are all measures of the relative quality of statistical models for a given set of data, where lower values are preferred.

## 5. Empirical illustration

We illustrate the package's usage on daily log-returns of the Swiss market index for a period ranging from November 12, 1990, to October 20, 2000. The data set is also used in [Mullen \*et al.\* \(2011\)](#) in the case of a MSGARCH model estimated by ML. In our empirical study, we consider a single-regime GJR model with a skewed Student- $t$  distribution and a two-state Markov-switching GJR model with skewed Student- $t$  distributions in each regime. Figure 1 displays the time series of log-returns.

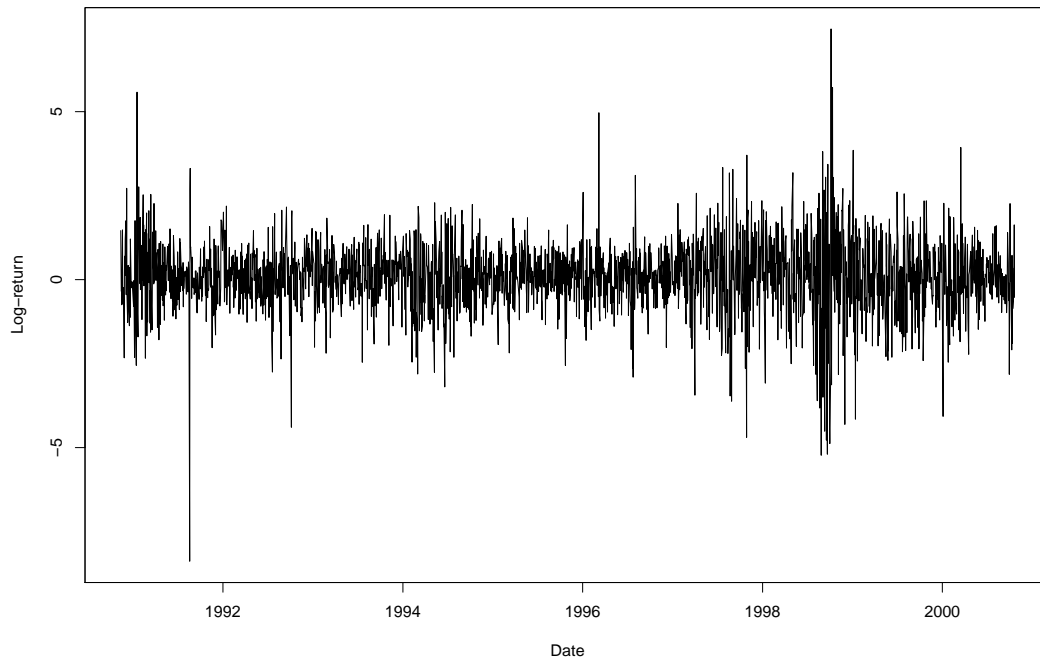


Figure 1: Log-returns of the Swiss Market Index. Data range from November 12, 1990, to October 20, 2000.

The dataset is available in the R package **DEoptim**. The plot is generated with the following code:

```
R> rm(list = ls())
R> require("MSGARCH")
R> require("coda")
R> require("DEoptim") # used for SMI data
R> data("SMI")
R> plot(y, xlab = "Date", ylab = "Log-return")
R> SMI = as.matrix(y)
R> date = as.Date(rownames(SMI))
R> date = c(date, date[length(date)] + 1)
```

We first estimate both models by ML with the pre-optimization argument `do.init = TRUE`:<sup>4</sup>

<sup>4</sup>Note that the estimation with **DEoptim** initialization is computationally demanding.

```
R> spec.1 = create.spec(model = "gjrGARCH",
+   distribution = "std", do.skew = TRUE,
+   do.mix = FALSE, do.shape.ind = FALSE)
R> out.mle.1 = MSGARCH::fit.mle(spec = spec.1, y = SMI,
+   ctr = list(do.init = TRUE))
R> summary(out.mle.1)
```

```
[1] "Specification Type: Single-Regime"
[1] "Specification Name: gjrGARCH_student_skew"
[1] "Number of parameters in variance model: 4"
[1] "Number of parameters in distribution: 2"
[1] "Default parameters:"
      alpha0_1 alpha1_1 alpha2_1 beta_1 nu_1 xi_1
[1,]      0.1      0.05      0.1      0.8  10      1
[1] "DEoptim initialization: TRUE"
[1] "Fitted Parameters:"
      alpha0_1 alpha1_1 alpha2_1 beta_1 nu_1 xi_1
[1,] 0.03934 0.04298 0.1143 0.8702 8.138 0.8555
[1] "Unconditional volatility:"
      State 1
[1,] 1.285
Log-kernel: -3376
AIC: 6743
BIC: 6778
```

The results indicate a high level of volatility persistence in the conditional variance process together with skewness and fat tails in the conditional distribution. A plot of the conditional variance process can be generated using the following code:

```
R> ht = sqrt(250) * MSGARCH::ht(out.mle.1)
R> plot(ht, date = date) # annual vol
```

Results are displayed in Figure 2.

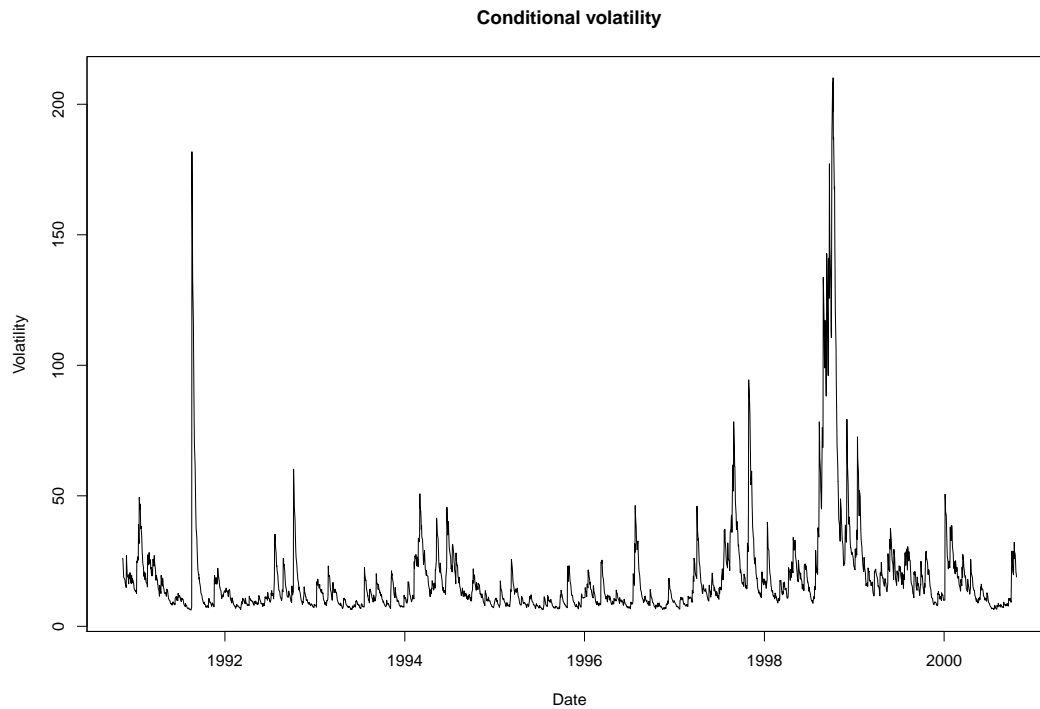


Figure 2: Conditional volatility of the single-regime GJR model with skewed Student- $t$  innovations estimated by ML.

Let us now perform the ML estimation of the Markov-switching model. This is achieved with the following code:

```
R> spec.2 = MSGARCH::create.spec(model = c("gjrGARCH", "gjrGARCH"),
+   distribution = c("std", "std"),
+   do.skew = c(TRUE, TRUE), do.mix = FALSE, do.shape.ind = FALSE)
R> set.seed(123)
R> out.mle.2 = MSGARCH::fit.mle(spec = spec.2, y = SMI,
+   ctr = list(do.init = TRUE))
R> summary(out.mle.2)
```

```
[[1]] "Specification Type: Markov-Switching"
[1] "Specification Name: gjrGARCH_student_skew gjrGARCH_student_skew"
[1] "Number of parameters in each variance model: 4 4"
[1] "Number of parameters in each distribution: 2 2"
[1] "Default parameters:"
      alpha0_1 alpha1_1 alpha2_1 beta_1 nu_1 xi_1 alpha0_2 alpha1_2 alpha2_2
[1,]      0.1      0.05      0.1    0.8  10    1      0.1      0.05      0.1
      beta_2 nu_2 xi_2    P    P
[1,]      0.8  10    1 0.5 0.5
[1] "DEoptim initialization: TRUE"
[1] "Fitted Parameters:"
```

```

      alpha0_1 alpha1_1 alpha2_1 beta_1  nu_1   xi_1 alpha0_2 alpha1_2 alpha2_2
[1,]    0.2227 0.001063   0.2131 0.5401 5.944 0.8521  0.08294 0.006224   0.1393
      beta_2 nu_2   xi_2      P      P
[1,] 0.8774   20 0.8582 0.9981 0.003125
[1] "Transition matrix:"
      t = 1    t = 2
t + 1 = 1 0.998056 0.003125
t + 1 = 2 0.001944 0.996875
[1] "Stable probabilities:"
      Stable probabilities
State 1                0.5464
State 2                0.4536
[1] "Unconditional volatility:"
      State 1 State 2
[1,]  0.8102   1.423
Log-kernel: -3365
AIC:  6688
BIC:  6769

```

From the results, we first note that the first regime of the MSGARCH model exhibits less persistence in the conditional variance. We also observe that parameter `alpha2_1` is larger in the first regime, implying a larger leverage effect in the less persistent state. The estimated degrees of freedom suggests that the first regime is more fat-tailed than the second regime, but the unconditional volatility of the first regime is much lower than that of the second regime. Both conditional distributions are negatively skewed. The transition matrix indicates that the regime does not switch very often. This can be observed by computing the filtered probabilities:

```

R> state = MSGARCH::Pstate(out.mle.2)
R> plot(state, date = date)

```

Result is displayed in Figure 3.



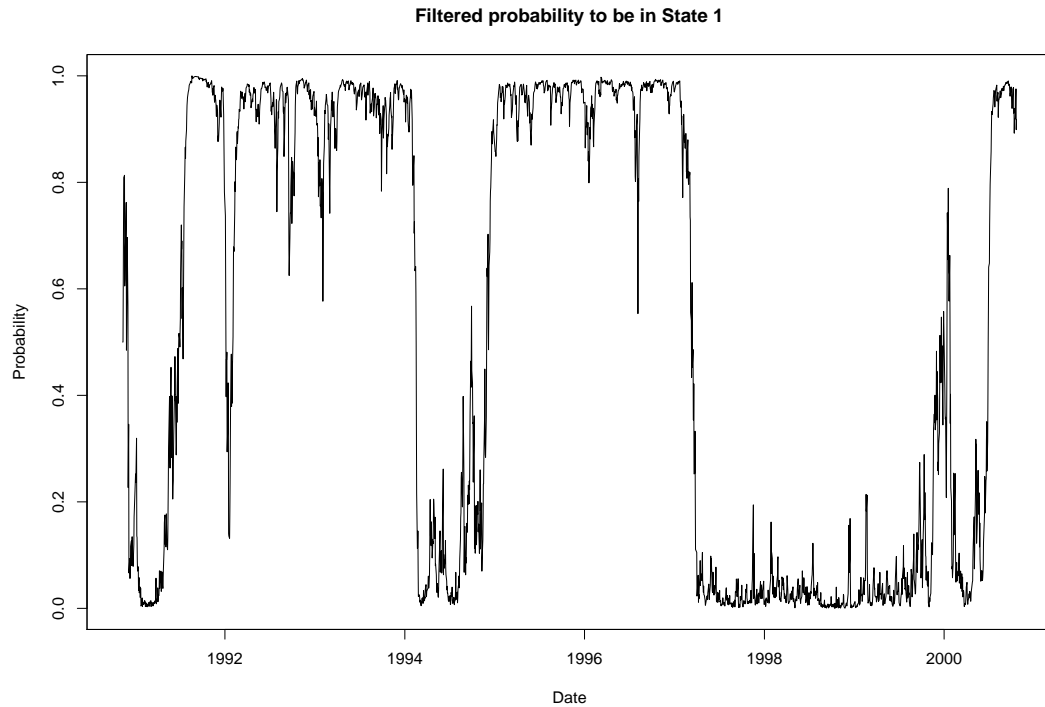


Figure 3: Filtered probabilities of the first regime obtained by ML for the two-state Markov-switching GJR model with skewed Student- $t$  innovations.

Bayesian estimation of the MSGARCH model can also be easily performed. We use here the ML estimator as the starting values:

```
R> ctr.bay.2 = list(N.burn = 5000, N.mcmc = 10000, N.thin = 10,
  theta0 = out.mle.2$theta)
R> set.seed(123)
R> out.bay.2 = MSGARCH::fit.bayes(spec = spec.2, y = SMI, ctr = ctr.bay.2)
R> summary(out.bay.2)
```

```
[1] "Specification Type: Markov-Switching"
[1] "Specification Name: gjrGARCH_student_skew gjrGARCH_student_skew"
[1] "Number of parameters in each variance model: 4 4"
[1] "Number of parameters in each distribution: 2 2"
[1] "Default parameters:"
alpha0_1 alpha1_1 alpha2_1 beta_1 nu_1 xi_1 alpha0_2 alpha1_2 alpha2_2
[1,]      0.1      0.05      0.1      0.8      10      1      0.1      0.05      0.1
beta_2 nu_2 xi_2 P P
[1,]      0.8      10      1 0.5 0.5
[1] "Bayesian posterior mean:"
alpha0_1 alpha1_1 alpha2_1 beta_1 nu_1 xi_1 alpha0_2 alpha1_2
0.224409 0.007664 0.233680 0.535233 5.950086 0.848320 0.084607 0.011201
alpha2_2 beta_2 nu_2 xi_2 P P
```

```

0.155427 0.867021 19.991778 0.863425 0.996790 0.004452
[1] "Posterior variance-covariance matrix"
alpha0_1 alpha1_1 alpha2_1 beta_1 nu_1 xi_1
alpha0_1 8.727e-05 -3.024e-06 -9.409e-05 2.609e-05 -2.215e-04 1.133e-04
alpha1_1 -3.024e-06 3.411e-05 9.404e-05 -7.543e-06 3.972e-05 -2.976e-05
alpha2_1 -9.409e-05 9.404e-05 3.875e-04 -6.516e-05 3.615e-04 -2.119e-04
beta_1 2.609e-05 -7.543e-06 -6.516e-05 3.206e-05 -9.236e-05 3.639e-05
nu_1 -2.215e-04 3.972e-05 3.615e-04 -9.236e-05 6.307e-04 -3.300e-04
alpha0_2 alpha1_2 alpha2_2 beta_2 nu_2 xi_2
alpha0_1 -6.033e-05 3.034e-06 -1.131e-04 7.951e-05 -2.835e-06 5.743e-05
alpha1_1 3.455e-05 3.057e-06 7.774e-05 -4.958e-05 -6.947e-05 2.314e-05
alpha2_1 1.736e-04 3.919e-06 3.519e-04 -2.350e-04 -2.089e-04 1.060e-05
beta_1 -1.936e-06 -2.524e-05 -6.069e-05 4.029e-05 -2.112e-06 2.621e-05
nu_1 1.848e-04 8.092e-07 3.864e-04 -2.707e-04 -6.808e-05 -1.265e-04
P
P
alpha0_1 -1.805e-06 4.064e-07
alpha1_1 1.026e-06 -4.006e-07
alpha2_1 -8.846e-06 4.874e-06
beta_1 6.101e-06 -1.704e-06
nu_1 -6.753e-06 1.168e-06
[ reached getOption("max.print") -- omitted 9 rows ]
[1] "Posterior mean transition matrix:"
t = 1 t = 2
t + 1 = 1 0.99679 0.004452
t + 1 = 2 0.00321 0.995548
[1] "Posterior mean stable probabilities:"
Stable probabilities
State 1 0.5435
State 2 0.4565
[1] "Posterior mean unconditional volatility:"
State 1 State 2
[1,] 0.8301 1.493
Acceptance rate: 0.992
AIC: 6689
BIC: 6771
DIC: 6672

```

We can test the mixing properties of the chains as follows:

```

R> require()
R> coda::traceplot(out.bay.2$theta[,1:6])

```

Results are displayed in Figure 4. We observe that the chain is mixing well, *i.e.*, that the MCMC algorithm converges to the stationary distribution.

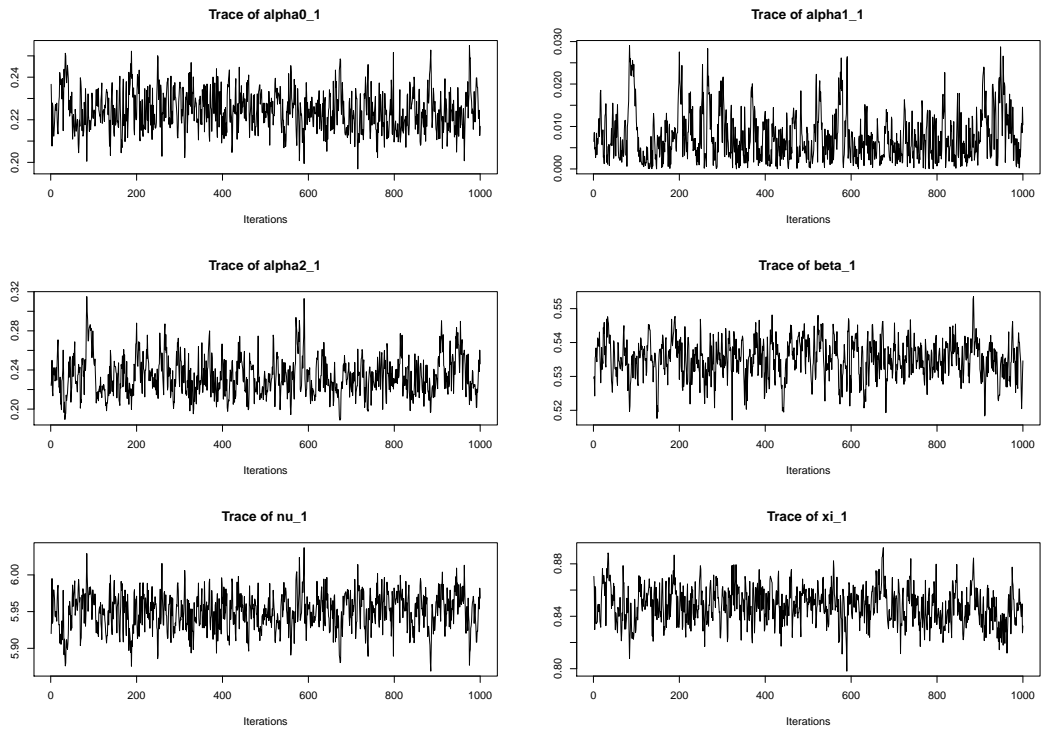


Figure 4: Trace of MCMC samples for the parameters of the first state in the two-state Markov-switching GJR model with skewed Student- $t$  innovations.

We can also display the pairs plot of MCMC draws with the following code:

```
R> pairs(x = as.matrix(out.bay.2$theta[,c(1,3,4,7,9,10)]), pch = 20, cex = 0.8)
```

Results are displayed in Figure 4. We can observe in Figure 5 that there is a high positive correlation between `alpha2_1` and `alpha2_2`, a high negative correlation between `alpha2_1` and `beta_2`, and a high negative correlation between `alpha2_2` and `beta_2`.

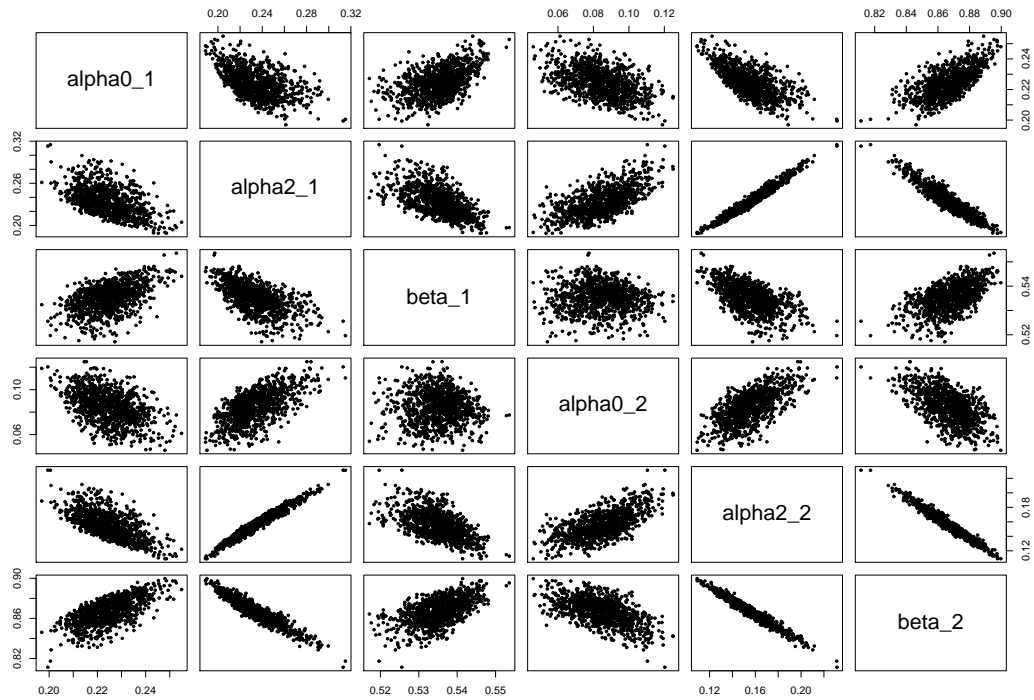


Figure 5: Pairs plot of the MCMC draws for the two-state Markov-switching GJR model with skewed Student- $t$  innovations.

Figure 6 displays the filtered probabilities in the Bayesian case, reported as a fan plot.

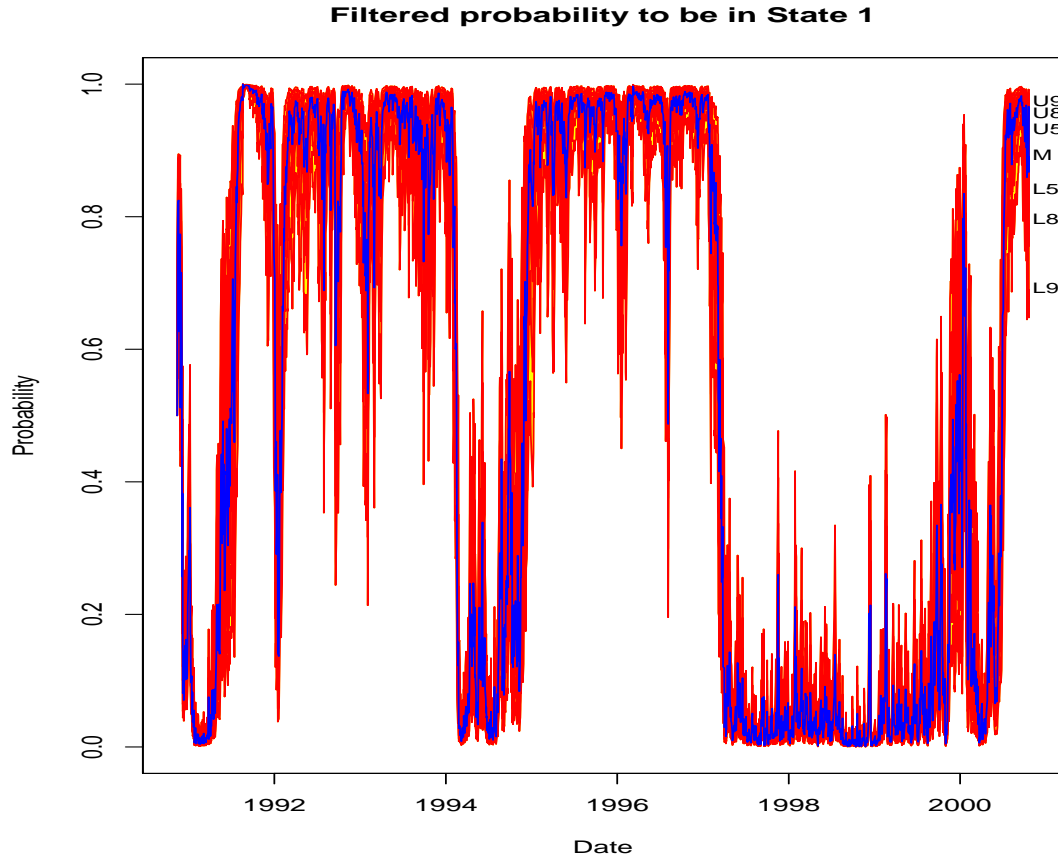


Figure 6: Filtered probabilities of the first regime obtained by MCMC for the two-state Markov-switching GJR model with skewed Student- $t$  innovations. Blue line indicates the median.

Finally, we can compute and compare the ML and Bayesian VaR at the 95% risk level for the two model specifications<sup>5</sup>:

```
R> risk.mle.1 = MSGARCH::risk(out.mle.1, level = c(0.95), ES = FALSE,
  do.its = TRUE)
R> risk.mle.2 = MSGARCH::risk(out.mle.2, level = c(0.95), ES = FALSE,
  do.its = TRUE)
R> risk.bay.1 = MSGARCH::risk(out.bay.1, level = c(0.95), ES = FALSE,
  do.its = TRUE)
R> risk.bay.2 = MSGARCH::risk(out.bay.2, level = c(0.95), ES = FALSE,
  do.its = TRUE)
```

The Value-at-Risk at 5% for both ML estimation of each model can be seen in Figure 7. They look similar except that the MSGARCH model often shows bigger spikes than the single-regime model when there is a large shift in volatility.

<sup>5</sup>Note that the current implementation for the in-sample VaR estimation is sub-optimal (time consuming). This will be improved in a next package release.

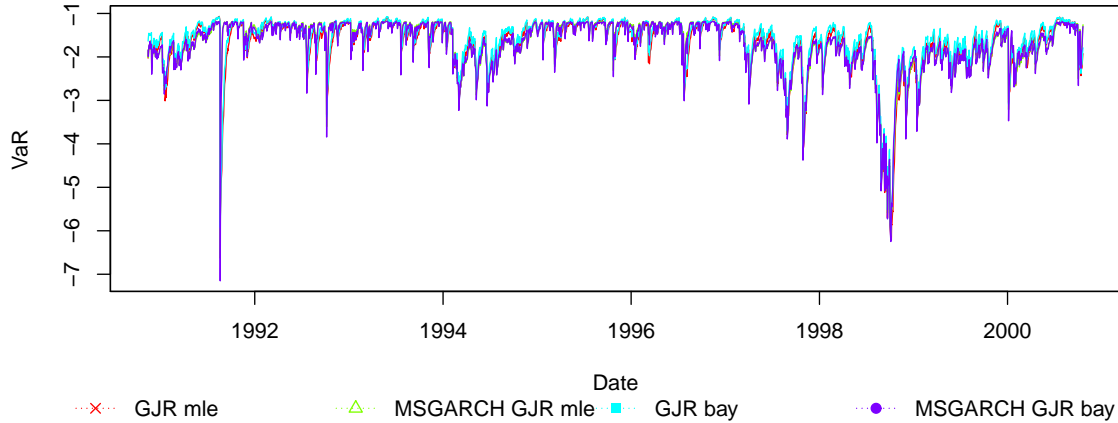


Figure 7: In-sample ML and Bayesian Value-at-Risk at the 95% risk level for the single-regime and regime-switching models.

## 6. Conclusion

This vignette introduced the R package **MSGARCH** which allows us to estimate, simulate and forecast Markov-switching GARCH models in the R statistical software. We detailed how to create various single-regime and regime-switching specifications with various scedastic functions and conditional distributions. We documented how to perform Maximum Likelihood and Bayesian estimation of these models. In an empirical illustration with real financial data, we showed how to fit and compare the in-sample performance of two complicated single-regime and Markov-switching GARCH specifications.

The R language has become an important vector for knowledge transfer in quantitative finance over the last years. We hope the R package **MSGARCH** will provide risk managers and regulators with new methodologies for improving risk forecasts of their portfolios.

Finally, if you use R or **MSGARCH**, please cite the software in publications.

Use `citation(package = "MSGARCH")`.

## Computational details

The results in this paper were obtained using R 3.2.3 (R Core Team 2016) with the packages: **MSGARCH** (Ardia *et al.* 2016), **adaptMCMC** (Andreas 2012), **DEoptim** (Ardia *et al.* 2015), **dfoptim** (Varadhan and Borchers 2016), **nloptr** (Johnson 2014), **Rcpp** (Eddelbuettel *et al.* 2016a; Eddelbuettel and François 2011), **RcppArmadillo** (Eddelbuettel *et al.* 2016b; Eddelbuettel and Sanderson 2014) and **xts** (Ryan and Ulrich 2015). R itself and all packages

used are available from CRAN at <http://CRAN.R-project.org/>. The package **MSGARCH** is under development in GitHub at <https://github.com/keblu/MSGARCH>.

## Acknowledgments

The authors acknowledge Google for financial support via the Google Summer of Code 2016 project "MSGARCH"; see <https://summerofcode.withgoogle.com/projects/?sp-search=KBlutea#6497774455488512>, the International Institute of Forecasters (IIF, <https://forecasters.org>) and Industrielle–Alliance.

## References

- Akaike H (1974). "A New Look at the Statistical Model Identification." *Automatic Control, IEEE Transactions on*, **19**(6), 716–723. doi:10.1007/978-1-4612-1694-0\_16.
- Andreas S (2012). *adaptMCMC: Implementation of a Generic Adaptive Monte Carlo Markov Chain Sampler*. URL <https://cran.r-project.org/package=adaptMCMC>.
- Ardia D (2008). *Financial Risk Management with Bayesian Estimation of GARCH Models: Theory and Applications*, volume 612 of *Lecture Notes in Economics and Mathematical Systems*. Springer-Verlag, Berlin. doi:10.1007/978-3-540-78657-3. ISBN 978-3-540-78656-6.
- Ardia D, Bluteau K, Boudt K, Peterson B, Trottier DA (2016). *MSGARCH: Markov-Switching GARCH Models in R*. R package version 0.17.7, URL <https://cran.r-project.org/package=MSGARCH>.
- Ardia D, Mullen KM, Peterson BG, Ulrich J (2015). *DEoptim: Differential Evolution in R*. URL <https://cran.r-project.org/package=DEoptim>.
- Bauwens L, Backer B, Dufays A (2014). "A Bayesian Method of Change-Point Estimation With Recurrent Regimes: Application to GARCH Models." *Journal of Empirical Finance*, **29**, 207–229. doi:10.1016/j.jempfin.2014.06.008.
- Bauwens L, Preminger A, Rombouts J (2010). "Theory and Inference for a Markov Switching GARCH Model." *Econometrics Journal*, **13**(2), 218–244. doi:10.1111/j.1368-423X.2009.00307.x.
- Black F (1976). "Studies of Stock Price Volatility Changes." In *Proceedings of the 1976 Meetings of the Business and Economics Statistics Section*, pp. 177–181. American Statistical Association.
- Bollerslev T (1986). "Generalized Autoregressive Conditional Heteroskedasticity." *Journal of Econometrics*, **31**(3), 307–327. doi:10.1016/0304-4076(86)90063-1.
- Christie AA (1982). "The Stochastic Behavior of Common Stock Variances: Value, Leverage and Interest Rate Effects." *Journal of Financial Economics*, **10**(4), 407–432. doi:10.1016/0304-405x(82)90018-6.

- Creal D, Koopman SJ, Lucas A (2013). “Generalized Autoregressive Score Models with Applications.” *Journal of Applied Econometrics*, **28**(5), 777–795. doi:10.1002/jae.1279.
- Dueker MJ (1997). “Markov Switching in GARCH Processes and Mean-Reverting Stock-Market Volatility.” *Journal of Business & Economic Statistics*, **15**(1), 26–34. doi:10.2307/1392070.
- Eddelbuettel D, François R (2011). “**Rcpp**: Seamless R and C++ Integration.” *Journal of Statistical Software*, **40**(8), 1–18. doi:10.18637/jss.v040.i08.
- Eddelbuettel D, François R, Allaire J, Ushey K, Kou Q, Bates D, Chambers J (2016a). **Rcpp**: Seamless R and C++ Integration. R package version 0.12.5, URL <https://cran.r-project.org/package=Rcpp>.
- Eddelbuettel D, François R, Bates D (2016b). **RcppArmadillo**: Rcpp Integration for the **Armadillo** Templated Linear Algebra Library. R package version 0.7.100.3.1, URL <https://cran.r-project.org/package=RcppArmadillo>.
- Eddelbuettel D, Sanderson C (2014). “**RcppArmadillo**: Accelerating R with High-Performance C++ Linear Algebra.” *Computational Statistics & Data Analysis*, **71**, 1054–1063. doi:10.1016/j.csda.2013.02.005.
- Fernández C, Steel MF (1998). “On Bayesian Modeling of Fat Tails and Skewness.” *Journal of the American Statistical Association*, **93**(441), 359–371. doi:10.1080/01621459.1998.10474117.
- Francq C, Zakoian JM (2011). *GARCH models: Structure, statistical inference and financial applications*. John Wiley & Sons. doi:10.1002/9780470670057.
- Gelman A, Carlin JB, Stern HS, Rubin DB (2014). *Bayesian Data Analysis*, volume 2. Chapman & Hall/CRC Boca Raton, FL, USA. doi:10.2307/2965436.
- Glosten LR, Jagannathan R, Runkle DE (1993). “On the Relation Between the Expected Value and the Volatility of the Nominal Excess Return on Stocks.” *The Journal of Finance*, **48**(5), 1779–1801. doi:10.1111/j.1540-6261.1993.tb05128.x.
- Gray SF (1996). “Modeling the Conditional Distribution of Interest Rates as a Regime-Switching Process.” *Journal of Financial Economics*, **42**(1), 27–62. doi:10.1016/0304-405x(96)00875-6.
- Haas M, Mittnik S, Paolella MS (2004a). “A New Approach to Markov-Switching GARCH Models.” *Journal of Financial Econometrics*, **2**(4), 493–530. doi:10.1093/jjfinec/nbh020.
- Haas M, Mittnik S, Paolella MS (2004b). “Mixed Normal Conditional Heteroskedasticity.” *Journal of Financial Econometrics*, **2**(2), 211–250. doi:10.1093/jjfinec/nbh009.
- Hamilton JD, Susmel R (1994). “Autoregressive Conditional Heteroskedasticity and Changes in Regime.” *Journal of Econometrics*, **64**(1), 307–333. doi:10.1016/0304-4076(94)90067-1.



- Johnson SG (2014). *The **NLopt** Nonlinear-Optimization Package*. URL <https://cran.r-project.org/web/packages/nloptr/>.
- Klaassen F (2002). “Improving GARCH Volatility Forecasts with Regime-Switching GARCH.” In *Advances in Markov-Switching Models*, pp. 223–254. Springer-Verlag. doi:10.1007/978-3-642-51182-0\_10.
- Mullen KM, Ardia D, Gil DL, Windover D, Cline J, *et al.* (2011). “**DEoptim**: An R Package for Global Optimization by Differential Evolution.” *Journal of Statistical Software*, **40**(6), 1–26. doi:10.18637/jss.v040.i06.
- Nelson DB (1991). “Conditional Heteroskedasticity in Asset Returns: A New Approach.” *Econometrica*, pp. 347–370. doi:10.2307/2938260.
- Plummer M, Best N, Cowles K, Vines K (2006). “CODA: Convergence Diagnosis and Output Analysis for MCMC.” *R News*, **6**(1), 7–11. URL <https://cran.r-project.org/package=coda>.
- Price K, Storn RM, Lampinen JA (2006). *Differential Evolution: A Practical Approach to Global Optimization*. Springer Science & Business Media.
- R Core Team (2016). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. R version 3.2.3, URL <https://www.R-project.org/>.
- Ryan JA, Ulrich JM (2015). *xts: Extensible Time Series*. R package version 0.9-7, URL <https://CRAN.R-project.org/package=xts>.
- Schwarz G, *et al.* (1978). “Estimating the Dimension of a Model.” *The Annals of Statistics*, **6**(2), 461–464. doi:10.1214/aos/1176344136.
- Trottier DA, Ardia D (2016). “Moments of Standardized Fernandez-Steel Skewed Distributions: Applications to the Estimation of GARCH-Type Models.” *Finance Research Letters*, **18**, 311–316. doi:10.1016/j.frl.2016.05.006.
- Varadhan R, Borchers HW (2016). *dfoptim: Derivative-Free Optimization*. R package version 2016.7-1, URL <https://CRAN.R-project.org/package=dfoptim>.
- Vihola M (2012). “Robust Adaptive Metropolis Algorithm with Coerced Acceptance Rate.” *Statistics and Computing*, **22**(5), 997–1008. doi:10.1007/s11222-011-9269-5.
- Zakoian JM (1994). “Threshold Heteroskedastic Models.” *Journal of Economic Dynamics & Control*, **18**(5), 931–955. doi:10.1016/0165-1889(94)90039-6.

### Affiliation:

David Ardia  
 Institute of Financial Analysis  
 University of Neuchâtel, Switzerland  
 &

Department of Finance, Insurance and Real Estate  
Laval University, Canada  
E-mail: [david.ardia@unine.ch](mailto:david.ardia@unine.ch)

Keven Bluteau  
Institute of Financial Analysis  
Neuchatel University, Neuchatel, Switzerland  
E-mail: [keven.bluteau@unine.ch](mailto:keven.bluteau@unine.ch)

Kris Boudt  
Vrije Universiteit Brussel, Belgium  
&  
Vrije Universiteit Amsterdam, The Netherlands  
E-mail: [kris.boudt@vub.ac.be](mailto:kris.boudt@vub.ac.be)

Denis–Alexandre Trottier  
Laval University, Canada  
E-mail: [denis-alexandre.trottier.1@ulaval.ca](mailto:denis-alexandre.trottier.1@ulaval.ca)

# **GARCH combination paper**

[https://papers.ssrn.com/sol3/papers.cfm?abstract\\_id=2688633](https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2688633)

# The impact of parameter and model uncertainty on market risk predictions from GARCH-type models<sup>☆</sup>

David Ardia<sup>a,c</sup>, Jeremy Kolly<sup>b,c,\*</sup>, Denis-Alexandre Trottier<sup>c</sup>

<sup>a</sup>*Institute of Financial Analysis, University of Neuchâtel, Neuchâtel, Switzerland*

<sup>b</sup>*Department of Management, University of Fribourg, Fribourg, Switzerland*

<sup>c</sup>*Finance, Insurance and Real Estate Department, Laval University, Quebec, Canada*

---

## Abstract

We study the effect of parameter and model uncertainty on the left-tail of predictive densities and in particular on VaR forecasts. To this end, we evaluate the predictive performance of several GARCH-type models estimated via Bayesian and maximum likelihood techniques. In addition to individual models, several combination methods are considered such as Bayesian model averaging and (censored) optimal pooling for linear, log or beta linear pools. Daily returns for a set of stock market indexes are predicted over about 13 years from the early 2000s. We find that Bayesian predictive densities improve the VaR backtest at the 1% risk level for single models and for linear and log pools. We also find that the robust VaR backtest exhibited by linear and log pools is better than the backtest of single models at the 5% risk level. Finally, the equally-weighted linear pool of Bayesian predictives tends to be the best VaR forecaster in a set of 42 forecasting techniques.

**Keywords:** GARCH models, Bayesian and frequentist estimation, predictive density combination, beta linear pool, censored optimal pooling, backtesting

**JEL classification:** C53, C58, G17, G32

---

---

<sup>☆</sup>This paper was previously titled “Predicting market risk with combinations of GARCH-type models”. The authors thank Kris Boudt, William Doehler, Lennart Hoogerheide as well as conference and seminar participants at the European Meeting of the Econometric Society 2016 (Geneva, Switzerland), at the International Symposium on Forecasting 2015 (Riverside, California) and at Laval University (Quebec, Canada) for their valuable comments. They also thank Keven Bluteau and Yoann Racine for research assistance. David Ardia acknowledges FRQSC, IFM2 Montreal and IIF for financial support. Jeremy Kolly is grateful to the Swiss National Science Foundation for financial support (grant #158754) and to the Finance, Insurance and Real Estate Department at Laval University for its hospitality. Denis-Alexandre Trottier is thankful to the FRQSC for financial support. The authors are solely responsible for the views expressed in this paper and for any remaining errors or shortcomings.

\*Corresponding author at: Department of Management, University of Fribourg, Boulevard de Pérolles 90, CH-1700 Fribourg, Switzerland

Email addresses: david.ardia@unine.ch (David Ardia), jeremy.kolly@unifr.ch (Jeremy Kolly), denis-alexandre.trottier.1@ulaval.ca (Denis-Alexandre Trottier)

## 1. Introduction

Asset returns demonstrate volatility clustering and an abnormal amount of extreme values. The autoregressive conditionally heteroscedastic (ARCH) model introduced in Engle (1982) is able to seize these empirical regularities. A more flexible specification, the generalized ARCH (GARCH) model, was later proposed in Bollerslev (1986). These models define the conditional volatility as a deterministic function of past innovations. However, they do not consider the leverage effect, that is, the asymmetric relation between news and volatility (Black, 1976). As a consequence, many asymmetric specifications for conditional volatility appeared around the 1990s (see among others Nelson, 1991; Glosten et al., 1993; Zakoian, 1994). Furthermore, GARCH specifications were initially coupled with the normal conditional distribution. However, this appears insufficient to fully account for asset return leptokurticity and skewness that can be empirically observed. Other distributions with fatter tails were proposed such as the standardized Student- $t$  distribution (Bollerslev, 1987) or the generalized error distribution (Nelson, 1991) as well as methods to introduce skewness in these distributions (see e.g. Fernández and Steel, 1998). Recently, GARCH-type models with complex updating mechanisms appeared such as those obtained from the generalized autoregressive score modeling framework (Creal et al., 2013) with skewed and leptokurtic conditional distributions.

A predictive density fully depicts the uncertainty related to a prediction. GARCH-type models can typically be used to generate predictive densities for future returns of financial assets (e.g. indexes or stocks). In financial risk management, precise estimation of the left-tail of asset returns' predictive densities is crucial to reliably depict downside risk (Tay and Wallis, 2000). There are several kinds of predictive densities possessing different properties. Among them, Bayesian predictive densities are of particular interest since they account for *parameter uncertainty* in a small-sample framework (Geweke and Amisano, 2010). Such predictive densities can improve out-of-sample left-tail predictive performance over those that do not integrate parameter uncertainty (Hoogerheide et al., 2012a) and the Bayesian approach is an appropriate way to account for

parameter uncertainty when the purpose is to produce Value-at-Risk (VaR) estimates (Aussenegg and Miazghynskaia, 2006). However, it has never been shown in the literature that integrating parameter uncertainty can improve VaR forecasts; we aim at filling this gap. Figure 1 illustrates why integrating parameter uncertainty can be useful for left-tail prediction. The Bayesian predictive density (in blue) is a particular averaging of the predictives that can be formed with individual posterior draws. It is generally more conservative than the predictive density with plugged maximum likelihood (ML) estimates (in red) and offers additional flexibility by accounting for all likely scenarios within the model structure. Nevertheless, it is also interesting to go beyond this structure by aggregating predictive densities originating from different models (see among others Genest and Zidek, 1986; Hall and Mitchell, 2007; Gneiting and Ranjan, 2013; Moral-Benito, 2015, Sec. 5). This extra step allows us to account for *model uncertainty* and delivers further flexibility for downside risk prediction.<sup>1</sup>

[Insert Figure 1 here.]

In this research, we assess the impact of these two forms of uncertainty on the left-tail of predictive densities and in particular on VaR forecasts obtained from these densities. Our investigations are performed in the universe of GARCH-type models. Besides being studied for decades in financial econometrics, these models are extensively used in the financial industry. The effect of parameter uncertainty is studied using Bayesian and ML estimation of GARCH-type models (Ardia, 2008). The effect of model uncertainty is investigated using the linear and log pools as well as the recent beta linear pool. Weights and parameters of the different pools are computed from past data. Methods for weight computation include Bayesian model averaging with predictive likelihoods (Eklund and Karlsson, 2007), as well as optimal pooling or OP (Geweke and

---

<sup>1</sup>Some studies already rely on model combination for VaR forecasting (see Pesaran et al., 2009; Massacci, 2015; Opschoor et al., 2015). However, they confine themselves to the linear pool and do not simultaneously account for parameter and model uncertainty.

Amisano, 2011, 2012). Broadly speaking, the former method averages measures of past predictive performance to form the weights while the latter looks for the weights that maximize past predictive performance. We also use a censoring-based version of OP, referred to as COP, that allows us to focus on the left-tail (Opschoor et al., 2015). We contribute to the literature by applying this method to all of the previously mentioned pools, including the beta linear pool, and by comparing it to other combination methods such as for instance Bayesian model averaging. We investigate whether COP improves VaR forecasts for combinations of GARCH-type models.

Large forecasting experiments are carried out with several non-nested GARCH-type volatility specifications using skewed and heavy-tailed conditional distributions. We predict daily returns of a set of indexes over a window of about 13 years from the early 2000s. For each index, different predictive densities are produced and aggregated. Then, we evaluate VaR estimates obtained from individual and combined predictives. We also assess the quality of densities in the left-tail using probability integral transforms. We find that Bayesian predictive densities improve VaR estimates at the 1% risk level for individual models as well as for linear and log pools. We also find that the VaR backtest is more robust when linear or log pools are used and that VaR estimates from these methods are globally better than those of single models at the 5% risk level. Finally, the equally-weighted linear pool of Bayesian predictives tends to be the best method for VaR prediction in a set of 42 forecasting techniques.

The outline of this paper is the following. Section 2 presents GARCH-type models. Section 3 describes model estimation and the different types of predictive densities. Section 4 compares single model predictions in a first application to stock market indexes. Section 5 discusses the combination of predictive densities. Section 6 compares single and combined forecasts in a second application to stock market indexes. Section 7 concludes.

## 2. GARCH-type models

Let  $y_t$  be a return series with a negligible conditional mean such that we can write  $y_t = \sigma_t \epsilon_t$  where the innovations  $\epsilon_t$  are iid (independent and identically distributed) with zero mean and unit variance. In the generalized autoregressive conditionally heteroscedastic (GARCH) model, the conditional variance is given by:<sup>2</sup>

$$\sigma_t^2 \equiv \alpha_0 + \alpha_1 y_{t-1}^2 + \beta \sigma_{t-1}^2, \quad (1)$$

where  $\alpha_0 > 0, \alpha_1, \beta \geq 0$  to guarantee that (1) is positive and where  $\alpha_1 + \beta < 1$  to ensure stationarity. Although widely used in practice, the GARCH model does not account for the leverage effect first evidenced by Black (1976). This effect is the fact that negative returns tend to increase more volatility than positive ones. Around 1990, many specifications for conditional volatility appeared to capture this effect. We consider here three of them that are popular and non-nested. The first is the exponential GARCH (EGARCH) model (Nelson, 1991):

$$\ln \sigma_t^2 \equiv \alpha_0 + \alpha_1 (|\epsilon_{t-1}| - E|\epsilon_{t-1}|) + \gamma \epsilon_{t-1} + \beta \ln \sigma_{t-1}^2, \quad (2)$$

where  $|\beta| < 1$  is required for stationarity. The second is the GJR model (Glosten et al., 1993):

$$\sigma_t^2 \equiv \alpha_0 + \alpha_1 y_{t-1}^2 + \gamma y_{t-1}^2 I\{y_{t-1} < 0\} + \beta \sigma_{t-1}^2, \quad (3)$$

where  $\alpha_0 > 0, \alpha_1, \beta \geq 0$  and  $\alpha_1 + \gamma \geq 0$  and where  $I\{\bullet\}$  is an indicator function equal to one when the condition in brackets holds and zero otherwise. Stationarity is ensured when  $\alpha_1 + \gamma E[\epsilon_t^2 I\{\epsilon_t <$

---

<sup>2</sup>We consider GARCH-type models in their  $(1, 1)$  form. In this model class, best performances are often obtained from the most parsimonious specifications.



0}] + \beta < 1. The third is the threshold GARCH (TGARCH) model (Zakoian, 1994):

$$\sigma_t \equiv \alpha_0 + \alpha_1^+ y_{t-1}^+ - \alpha_1^- y_{t-1}^- + \beta \sigma_{t-1}, \quad (4)$$

where  $\alpha_0 > 0, \alpha_1^+, \alpha_1^-, \beta \geq 0$  and where  $y_t^+ \equiv \max\{y_t, 0\}$  and  $y_t^- \equiv \min\{y_t, 0\}$ . The stationarity condition for this model can be found in Francq and Zakoian (2010). In (2)–(3), the leverage effect is captured by  $\gamma$ . In (4), it stems from the difference between  $\alpha_1^+$  and  $\alpha_1^-$ . Note that these three asymmetric volatility models exhibit different news impact curves. It is also noteworthy that the GJR and TGARCH models are nested in the asymmetric power ARCH model of Ding et al. (1993). However, we prefer the GJR and TGARCH models as they are more parsimonious and provide a more stable estimation.

Recently, Creal et al. (2013) suggest the generalized autoregressive score (GAS) modeling framework for latent variables which uses the derivative of the log predictive likelihood as an updating mechanism. When applied to the conditional variance, it gives rise to a class of GARCH-type models. The GAS model is given by:

$$\sigma_t^2 \equiv \alpha_0 + \alpha_1 s_{t-1} + \beta \sigma_{t-1}^2, \quad s_t \equiv S_t \nabla_t, \quad \nabla_t \equiv \frac{\partial}{\partial \sigma_t^2} \ln p(y_t | \sigma_t^2),$$

where  $\alpha_0 > 0, \alpha_1, \beta \geq 0$  and where  $p(\bullet)$  denotes a density function. The scaling factor  $S_t$  is defined as the inverse of the information matrix. We require  $\beta < 1$  for stationarity and we ensure positive volatility. When the innovations are iid standard normal, this model is equivalent to (1). However, more advanced distributions lead to complex updating mechanisms.

To properly account for the leptokurticity exhibited by the empirical distribution of asset returns, we model innovations with the standardized Student- $t$  distribution with  $\nu > 2$  degrees of

freedom as proposed by Bollerslev (1987):

$$p(\epsilon_t|\nu) = \frac{\Gamma\left(\frac{\nu+1}{2}\right)}{\Gamma\left(\frac{\nu}{2}\right)\sqrt{\pi(\nu-2)}} \left(1 + \frac{\epsilon_t^2}{\nu-2}\right)^{-\frac{\nu+1}{2}},$$

where  $\Gamma(\bullet)$  is the gamma function. As a non-nested alternative, we also consider the generalized error distribution (GED) with zero mean and unit variance (see Nelson, 1991):

$$p(\epsilon_t|\lambda) = \frac{\lambda \exp\left(-\frac{1}{2}|\epsilon_t/\varphi_\lambda|^\lambda\right)}{\varphi_\lambda 2^{\frac{\lambda+1}{\lambda}} \Gamma\left(\frac{1}{\lambda}\right)}, \quad \varphi_\lambda \equiv \left[\frac{\Gamma\left(\frac{1}{\lambda}\right)}{2^{\frac{2}{\lambda}} \Gamma\left(\frac{3}{\lambda}\right)}\right]^{\frac{1}{2}}.$$

The parameter  $\lambda > 0$  controls tail-thickness. The GED reduces to the standard normal distribution when  $\lambda = 2$ , while the Laplace distribution appears when  $\lambda = 1$ . Empirical distribution of asset returns is also typically skewed. We introduce skewness in the above distributions using the approach by Fernández and Steel (1998). A standardized skew distribution is given by:

$$p^*(\epsilon_t|\xi, \vartheta) = \frac{2\sigma_\xi}{\xi + \frac{1}{\xi}} p \left[ \frac{\sigma_\xi \epsilon_t + \mu_\xi}{\xi} I\{\epsilon_t \geq -\mu_\xi/\sigma_\xi\} + \xi(\sigma_\xi \epsilon_t + \mu_\xi) I\{\epsilon_t < -\mu_\xi/\sigma_\xi\} | \vartheta \right],$$

where  $\xi > 0$  is the skewness parameter,  $\vartheta$  denotes the parameters of the initial distribution and:

$$\mu_\xi \equiv m_1 \left( \xi - \frac{1}{\xi} \right), \quad \sigma_\xi^2 \equiv (1 - m_1^2) \left( \xi^2 + \frac{1}{\xi^2} \right) + 2m_1^2 - 1, \quad m_1 \equiv 2 \int_0^\infty u p(u|\vartheta) du.$$

The previous expressions can be used to obtain a skew version of any symmetric unimodal density function which has zero mean and unit variance. We use formulas in Trottier and Ardia (2015) for computing moments of skew distributions.

### 3. Model estimation and predictive densities

Both Bayesian and maximum likelihood (ML) estimation methodologies are considered. To perform Bayesian estimation of a GARCH-type model  $M_k$ , we use an independence chain

Metropolis-Hastings algorithm (Tierney, 1994) that simulates the posterior density  $p(\theta_k|y, M_k)$  of the parameter vector  $\theta_k \in \Theta_k$  given the sample of data  $y$ . The proposal distribution is constructed with the MitISEM method proposed in Hoogerheide et al. (2012b). Furthermore, we use diffuse proper priors as in Ardia (2008) and confirmed with sensitivity analyses that they have a negligible influence on posterior results. To carry out ML estimation of  $M_k$ , we look for the vector of parameter estimates  $\hat{\theta}_k \in \Theta_k$  that maximizes the log likelihood function of the sample  $y$  using numerical methods.

GARCH-type models can be used to generate predictive densities for future asset returns. Let  $\mathcal{M} \equiv \{M_1, \dots, M_K\}$  be a set of such models. The Bayesian predictive density provided by  $M_k$  for  $y_t$  given the sample  $Y_{t-1}^r \equiv (y_{t-r}, \dots, y_{t-1})'$  of  $r$  past observations can be written as:<sup>3</sup>

$$p(y_t|Y_{t-1}^r, M_k) = \int_{\Theta_k} p(y_t|Y_{t-1}^r, \theta_k, M_k)p(\theta_k|Y_{t-1}^r, M_k)d\theta_k, \quad (5)$$

where  $p(y_t|Y_{t-1}^r, \theta_k, M_k)$  is known analytically in GARCH-type models. The density in (5) can easily be evaluated from a posterior sample. Furthermore, we see that it accounts for parameter uncertainty. This feature has already proven useful for improving GARCH predictive performance in terms of log score (Geweke and Amisano, 2010) and censored log score (Hoogerheide et al., 2012a). Another predictive density that also accounts for parameter uncertainty could be built for instance with the asymptotic sampling distribution of ML parameter estimates. However, as explained in Geweke and Amisano (2010), it is difficult to interpret besides being a large sample approximation. Furthermore, Aussenegg and Miazhyńska (2006) show that the Bayesian approach is advantageous over other methods that account for parameter uncertainty when the purpose is to produce Value-at-Risk estimates.

Of course, we can also produce predictive densities that condition on particular parameter values instead of integrating parameter uncertainty. Posterior means  $\bar{\theta}_k$  computed from  $Y_{t-1}^r$  can

---

<sup>3</sup>Here we consider one-step ahead predictions, but extension to larger forecast horizons is straightforward.

be used to form the density  $p(y_t|Y_{t-1}^r, \bar{\theta}_k, M_k)$  which we call the PM predictive density. It is also common to consider the ML predictive density  $p(y_t|Y_{t-1}^r, \hat{\theta}_k, M_k)$  where ML estimates  $\hat{\theta}_k$  obtained over  $Y_{t-1}^r$  are plugged into the density. In large samples, Bayesian, PM and ML predictives are similar because the posterior density is very concentrated around the mode of the likelihood function. However, in practice, they exhibit important differences as highlighted in Sections 4 and 6 for the left-tail.

Finally, the predictive densities introduced above can be used to compute a fundamental risk measure that is extensively used in the industry known as the Value-at-Risk or VaR (Duffie and Pan, 1997; Jorion, 2007). It can merely be calculated as the  $100\delta\%$  quantile of these predictive densities. In general, the risk level  $\delta \in (0, 1)$  is fixed to 0.01 or 0.05 to consider left-tail risk. The above predictives can also be used to obtain the probability integral transform (PIT), that is, the predicted probability to have an outcome smaller than or equal to the actual realization. For a given sample, PITs from one-step ahead predictives will be iid uniform if the predictives correspond to the data generating process (DGP) densities (Diebold et al., 1998). Therefore, studying PIT independence and distribution allows us to evaluate one-step ahead predictive densities. In what follows, we will assess the quality of our left-tail forecasts by considering only the PITs associated to left-tail outcomes.

#### 4. Application with stock market indexes I

We consider daily returns (in percentage points) of 8 major stock market indexes (CAC40, DAX, DJIA, FTSE, Nikkei, NASDAQ, SMI and S&P500) provided by the Oxford-Man Institute. These samples start from January 3, 2000 and are made of 3265 observations (about 13 years). Note that the series are demeaned and that a first-order autoregressive filter is applied on each of them to focus on volatility and higher conditional moments.

We work here with the EGARCH, GJR, TGARCH and GAS models with skew Student- $t$  and skew GED innovations. These non-nested GARCH-type models are used to produce one-

day ahead Bayesian, PM and ML predictive densities for each index. Rolling windows of 750 past observations are used for posterior and ML estimations. Such windows allow us to take into account potential parameter instability over time. Note that we set aside the 500 first predictives to be consistent with the application of Section 6 where they are used to form combination weights. For each series, we use predictives to compute 1% and 5% VaR estimates as well as PITs below 5%.

#### 4.1. Backtest methodologies

To backtest our VaR estimates, we first consider the standard unconditional coverage (UC) test (Kupiec, 1995) which is simply a likelihood ratio (LR) test for the correct proportion of VaR exceedances, or hits, and the conditional coverage (CC) test of Christoffersen (1998) where the alternative hypothesis is that the hits follow a first-order Markov chain. We do not perform the CC test when the 1% risk level is involved as it is often invalid due to a lack of consecutive hits at this risk level. To gain power, we complement these tests with the Monte Carlo test of unconditional coverage (MCS) proposed in Ziggel et al. (2014). It is based on the statistic:

$$\text{MCS} \equiv \sum_{t=1}^T I_t + \epsilon ,$$

where  $I_t \equiv I\{y_t < \text{VaR}_t^\delta\}$  and  $\epsilon \sim \text{iid}N(0, 0.001^2)$ . Critical values for this test are obtained via Monte Carlo simulation. As our VaR estimates are generally not sufficiently conservative, we perform upper-tailed MCS tests. We also consider the CAViaR test (Engle and Manganelli, 2004) as implemented in Berkowitz et al. (2011). We estimate the following logit model:

$$\Pr(I_t) \equiv \frac{\exp(\phi_0 + \phi_1 I_{t-1} + \phi_2 \text{VaR}_t^\delta)}{1 + \exp(\phi_0 + \phi_1 I_{t-1} + \phi_2 \text{VaR}_t^\delta)} ,$$

and perform a LR test of the joint hypothesis ( $\phi_0 = \ln[\delta/(1 - \delta)]$ ,  $\phi_1 = 0$ ,  $\phi_2 = 0$ ). Finally, we calculate the (tick) asymmetric linear losses induced by our VaR forecasts:

$$L(y_t, \text{VaR}_t^\delta) \equiv (\delta - I_t)(y_t - \text{VaR}_t^\delta),$$

and test significance of differences between mean tick losses with the Diebold-Mariano (DM) test (Diebold and Mariano, 1995) using a heteroscedasticity and autocorrelation consistent variance estimate.

To analyze PITs below 5%, we begin by rescaling and normalizing them (see Christoffersen and Pelletier, 2004). Then, we carry out the ARCH, JB and LR tests suggested in Deschamps (2012). The ARCH test is a  $F$ -test of the nullity of autoregression coefficients (intercept excluded) in a 6-order autoregressive process of the squared PITs. We use a heteroscedasticity consistent covariance matrix estimate for this test. The JB test is the Jarque-Bera test for normality and the LR test is simply a test of  $N(0, 1)$  against an unconstrained normal alternative.

#### 4.2. Backtest results

Table 1 reports the numbers of rejections at the 5% significance level over the set of 8 indexes for the backtest of 2015 VaR estimates at the 1% and 5% risk levels. These quantities are obtained from Bayesian, PM and ML predictives generated by our GARCH-type models. We see that, at the 1% risk level, Bayesian predictives show globally fewer rejections than PM or ML predictives. At the 5% risk level, evidence is less clear. Performances of Bayesian and PM predictives are similar. Moreover, they both tend to provide better results in terms of unconditional coverage than ML predictives while their CAViaR tests show more rejections than for ML predictives. Regarding models' performance, the skew GED tends to improve performance, especially at the 5% risk level. Note the perfect backtest of the GJR and GAS models with skew GED innovations at the 5% risk level. Globally, the EGARCH model provides the worst outcomes.

[Insert Table 1 here.]

Table 2 presents the numbers of rejections over the 8 indexes at the 5% significance level for the tests used to analyze rescaled PITs below 5%. PITs are computed from 2015 Bayesian, PM and ML predictives generated by our GARCH-type models. We observe few rejections for the ARCH test whatever the predictive used. Regarding normality, the JB and LR tests favor Bayesian predictives over PM or ML predictives. Note however that the JB test for the skew Student- $t$  TGARCH model is rejected 5 times over 8 with Bayesian predictives.

[Insert Table 2 here.]

In summary, the backtest in this application tends to favor 1% VaR estimates and left-tail forecasts generated from Bayesian predictives. For single models, it is therefore better to forecast the left-tail and compute 1% VaR estimates from predictives that integrate parameter uncertainty than from predictives that do not, such as the PM or ML ones. Now it remains to investigate if accounting for model uncertainty also improves our forecasts and under which circumstances. For this purpose, we consider model combination.

## 5. Combination of predictive densities

Predictive densities generated from models in  $\mathcal{M}$  can be aggregated together. The older and probably the more intuitive formula used for this purpose is the linear pool (Stone, 1961):<sup>4</sup>

$$p_{lin}(y_t|Y_{t-1}^r, w_{t-1}) \equiv \sum_{k=1}^K w_{t-1,k} p(y_t|Y_{t-1}^r, M_k), \quad (6)$$

---

<sup>4</sup>For notational simplicity, we present aggregation formulas with Bayesian predictive densities. However, they also apply to other types of predictives.

where  $w_t \equiv (w_{t,1}, \dots, w_{t,K})'$  is a weight vector depending on data up to time  $t$  and satisfying the conditions  $\sum_{k=1}^K w_{t,k} = 1$  and  $w_{t,1}, \dots, w_{t,K} \geq 0$  in order for (6) to be a valid density. The linear pool may be multimodal and tends to be overdispersed. Nevertheless, it performs well in many applications. Gneiting and Ranjan (2013) argue that it is because individual densities are often underdispersed relative to the true density. On the other hand, Krüger (2014) shows that important scoring rules for the linear pool including the log scoring rule satisfy a lower bound. For those scoring rules, the linear pool score cannot be lower than a weighted average of individual densities' scores. He explains that the linear pool should thus outperform its components on average over time because, unlike its components' score, the linear pool score will exceed the lower bound at each time period.

Alternative non-linear pools appeared in the literature. A popular one is the log pool:

$$p_{\log}(y_t | Y_{t-1}^r, w_{t-1}) \equiv \frac{\prod_{k=1}^K p(y_t | Y_{t-1}^r, M_k)^{w_{t-1,k}}}{\int_{-\infty}^{\infty} \prod_{k=1}^K p(u | Y_{t-1}^r, M_k)^{w_{t-1,k}} du},$$

where the weights meet the same constraints than those of the linear pool for convenience. Its log kernel corresponds to a weighted average of log densities of individual models. The log pool is generally unimodal and less dispersed than the linear pool (Genest and Zidek, 1986). Moreover, its log score has a lower bound (Kascha and Ravazzolo, 2010; Krüger, 2014). In the area of inflation forecasting, Krüger (2014) finds some empirical evidence in favor of the logarithmic pool against the linear one, while discrimination is more difficult in Kascha and Ravazzolo (2010).

As discussed among others in Gneiting and Ranjan (2011), it is important for a predictive density to be well calibrated, that is, to be statistically consistent with the DGP. Inadequate calibration can be diagnosed by the PIT distribution. The beta linear pool devised by Ranjan and Gneiting (2010) and Gneiting and Ranjan (2013) comes along with an improved calibration. Its density is obtained by applying a beta distribution function to the distribution function of the linear pool and



by taking the derivative. We write it as follows:

$$p_{blin}(y_t|Y_{t-1}^r, w_{t-1}) \equiv \beta_{a,b} \left[ \sum_{k=1}^K w_{t-1,k} \int_{-\infty}^{y_t} p(u|Y_{t-1}^r, M_k) du \right] \sum_{k=1}^K w_{t-1,k} p(y_t|Y_{t-1}^r, M_k),$$

where  $\beta_{a,b}(\bullet)$  is the beta density with shape parameters  $a, b > 0$  and where the non-negative weights add to one. The linear pool results when  $a = b = 1$ . Note that the beta linear pool is a special case of the generalized linear pool introduced by Kapetanios et al. (2015) that lets the weights depend on  $y_t$ . Recent studies (Bassetti et al., 2015, Casarin et al., 2016) also consider a mixture of beta calibration functions for pooling schemes, however this will not be used in this research.

Besides the aggregation formula, the determination of the weights is also crucial. A basic solution to this problem is to use equal weights. This approach is successful for combining point forecasts (Clemen, 1989; Stock and Watson, 2004; Smith and Wallis, 2009). When the weights of a linear pool of Bayesian predictive densities are defined as posterior model probabilities, we obtain the Bayesian model averaging (BMA) method. This approach stems naturally from probability rules and formally accounts for model uncertainty (Leamer, 1978, Ch. 4). Of course, we can also heuristically account for model uncertainty by using posterior model probabilities with other aggregation formulas and other kinds of predictive densities. In the BMA method, the posterior model probability of a single model in  $\mathcal{M}$  will be equal to one asymptotically even if this model is false. This situation is illustrated in Ardia and Kolly (2016). As this is questionable, we prefer to consider an implementation of BMA where marginal likelihoods are replaced by predictive likelihoods (Eklund and Karlsson, 2007). In this framework, BMA weights can be written as:

$$w_{t-1,k}^{BMA} \equiv \frac{p(Y_{t-1}^s|Y_{t-s-1}^r, M_k)}{\sum_{l=1}^K p(Y_{t-1}^s|Y_{t-s-1}^r, M_l)},$$

where we assume equal prior model probabilities and where  $s$  corresponds to the size of the weight

estimation window. This implementation of BMA implies slower convergence to a single model and allows us to work with diffuse priors. Note that the BMA weights do not depend on the aggregation formula and that BMA does not provide a way to estimate  $a$  and  $b$  in the beta linear pool.

An alternative to BMA is the optimal pooling (OP) approach introduced in Hall and Mitchell (2007) and subsequently deepened in Geweke and Amisano (2011, 2012). For a given pool  $p_c(y_t|Y_{t-1}^r, w_{t-1})$ , the OP weights are given by:

$$w_{t-1}^{OP-c} \equiv \arg \max_{w_{t-1}} \sum_{\tau=t-s}^{t-1} \ln p_c(y_\tau|Y_{\tau-1}^r, w_{t-1}), \quad (7)$$

where optimal weights must be found in the unit simplex. The objective function in (7) adds pool log scores over the sample  $Y_{t-1}^s$  and its maximization corresponds to the minimization of the Kullback-Leibler (KLIC) distance from the DGP to the pool (Hall and Mitchell, 2007). Furthermore, when the DGP is a model or a linear or log pool of models in  $\mathcal{M}$ , the OP method asymptotically recovers true weights (Krüger, 2014). However, we observed in our own experiments that uncertainty around true weights can remain substantial even with large estimation windows. On the other hand, when the DGP cannot be obtained from models in  $\mathcal{M}$ , several OP limiting weights are typically positive.

In risk management, pool predictive densities can be used to obtain VaR estimates (see e.g. Ardia and Kolly, 2016). Therefore, it is possible to improve these estimates by using weights that give more importance to left-tail outcomes. Opschoor et al. (2015) propose to replace the log score in (7) by the censored scoring rule introduced in Diks et al. (2011).<sup>5</sup> This scoring rule is equal to the log predictive likelihood when the outcome falls below a threshold and to the log probability mass above the threshold otherwise. It thus neglects the shape of the predictive above

---

<sup>5</sup>Note that another approach using censoring in a BMA framework to generate weights that give more importance to tail events is proposed in Gatarek et al. (2014).

the threshold. The censored OP (COP) weights can be obtained as follows:

$$w_{t-1}^{COP-c} \equiv \arg \max_{w_{t-1}} \sum_{\tau=t-s}^{t-1} I\{y_\tau < \hat{q}_\psi\} \ln p_c(y_\tau | Y_{\tau-1}^r, w_{t-1}) \\ + I\{y_\tau \geq \hat{q}_\psi\} \ln \int_{\hat{q}_\psi}^{\infty} p_c(u | Y_{\tau-1}^r, w_{t-1}) du, \quad (8)$$

subject to the usual weight restrictions. We define the censoring bound  $\hat{q}_\psi$  as the empirical  $100\psi\%$  quantile obtained from the sample  $Y_{t-1}^s$ . The choice of  $\psi$  thus determines the percentage of uncensored observations used to compute the weights.

Specific expressions are very useful for implementing the COP method. One is provided in Opschoor et al. (2015) in the linear pool case while another is given in Ardia and Kolly (2016) in the log pool case. For the beta linear pool, it can be shown that (8) reduces to:

$$w_{t-1}^{COP-blin} \equiv \arg \max_{w_{t-1}, a, b} \sum_{\tau=t-s}^{t-1} I\{y_\tau < \hat{q}_\psi\} \ln p_{blin}(y_\tau | Y_{\tau-1}^r, w_{t-1}) \\ + I\{y_\tau \geq \hat{q}_\psi\} \ln \left( 1 - B_{a,b} \left[ \sum_{k=1}^K w_{t-1,k} \int_{-\infty}^{\hat{q}_\psi} p(u | Y_{\tau-1}^r, M_k) du \right] \right),$$

where  $B_{a,b}(\bullet)$  is the beta distribution function. Let us now illustrate the relevance of using COP for the beta linear pool when we are interested by the calibration of the left-tail. We simulate 1000 observations from the mixture DGP  $0.6N(-2, 1) + 0.4N(2, 1)$  to form an estimation period and 1000 other observations from the same DGP to constitute a forecasting period. We consider the misspecified models  $N(-1, 1)$  and  $N(1, 1)$  and combine them with linear and beta linear pools whose weights and parameters are estimated with the OP and COP methods on the estimation period. In the COP method, the censoring bound is fixed to  $-2$  such that about 30% of the lowest observations are uncensored. Then, we compute PITs below 30% on the forecasting period and rescale them so that they lie between 0 and 1. Figure 2 presents the empirical distribution functions of these tail PITs with straight lines passing through 0 and 1 representing the ideal behavior. We

see that the tail PITs provided by the beta linear pool are closer to the straight line. This was already highlighted in Bassetti et al. (2015) and Casarin et al. (2016). Furthermore, we observe that the calibration is even better when COP is used to estimate the beta linear pool. We have this because COP focuses on the left-tail.

[Insert Figure 2 here.]

## 6. Application with stock market indexes II

Due to the computational cost implied by model combination, and by analogy with Opschoor et al. (2015), we consider here a subset of 4 indexes (DJIA, FTSE, Nikkei and S&P500) among those described in Section 4. We work again with the EGARCH, GJR, TGARCH and GAS models with skew Student- $t$  and skew GED innovations. We produce one-day ahead Bayesian and ML predictive densities for each index using rolling estimation windows of 750 past observations. Predictives are then aggregated using linear and log pools with equal, BMA, OP and COP weights and the beta linear pool with OP and COP weights. Rolling windows of 500 past observations are used for weight estimation.<sup>6</sup> They allow us to account for potential variation in model prevalence over time. In the COP method, we define censoring bounds such that 15% and 25% of the lowest observations of these windows are uncensored. This way, we keep enough uncensored observations to form model weights based on their left-tail performance. For each index, 1% and 5% VaR estimates as well as PITs below 5% are computed.

### 6.1. Backtest results

The methodologies used to backtest our forecasts are described in Subsection 4.1. Tables 3 and 4 present the backtest of 2015 VaR estimates at the 1% and 5% risk levels, respectively, obtained from Bayesian and ML predictives of our models and model combinations. We only

---

<sup>6</sup>Parameters  $a$  and  $b$  in the beta linear pool are also estimated on these windows.

report statistical significance results at 1%, 5% and 10% to ease analysis. Tables with test statistics are available upon request from the authors. We start by studying results at the 1% risk level in Table 3. We observe that Bayesian predictives globally provide better unconditional coverage for single models and under linear and log pooling. A notable exception is the log pool with COP weights and 25% censoring for the S&P500 index where ML predictives are preferable. Regarding individual models, the skew Student TGARCH model delivers the best outcomes while the performance of the EGARCH and GAS models is poor. Interestingly, the results among linear or log pools are more similar than among single models given a particular index. Note the very good VaR backtest of the equally-weighted (EW) linear pool. On the other hand, the beta linear pool shows puzzling outcomes. It can give correct unconditional coverage, especially when COP is used for estimation. However, its CAViaR test is systematically rejected, indicating that the calibration provided by the beta linear pool is detrimental to the independence of VaR violations. Moreover, Bayesian predictives seem to be harmful to the beta linear pool.

[Insert Table 3 here.]

We now turn to the 5% risk level in Table 4. In this case, discrimination among Bayesian and ML predictives is more difficult. Bayesian predictives are favored for the S&P500 index while they are not for the FTSE index. About single models, the EGARCH model and the skew Student GAS models give bad results again. Note also the poor VaR backtest exhibited by the TGARCH model for the FTSE index. Given a particular index, results among combination methods are anew more similar. They are also globally better than those of single models. It is noteworthy that the EW linear pool is the only method providing a perfect backtest for the FTSE index and that the puzzling behavior of the beta linear pool can only be observed for the FTSE index.

[Insert Table 4 here.]

Figures 3 and 4 present DM test statistics for significance of differences between mean (tick) asymmetric linear losses computed for 2015 VaR estimates at the 1% and 5% risk levels. Figure 3 compares the skew GED GJR model and the EW linear pool with Bayesian predictives to all other forecasting techniques, while Figure 4 compares these two methods with ML predictives to all other techniques. In both Figures, we observe at the 1% risk level that – relative to our two benchmarks – forecasting techniques using ML predictives are less accurate than those using Bayesian predictives. This phenomenon is not observable at the 5% risk level. In overall, the skew GED GJR model and the EW linear pool exhibit a close performance. However, a larger number of significant differences favor the EW linear pool at the 5% risk level. Finally, note that the beta linear pool performs very badly against both benchmarks at the 1% risk level while there are few significant differences at the 5% risk level.

[Insert Figures 3 and 4 here.]

Table 5 presents the analysis of rescaled PITs below 5% derived from 2015 Bayesian and ML predictives produced by our models and model combinations. Again, we only report statistical significance results at 1%, 5% and 10% to facilitate analysis and can provide Tables with test statistics upon request. Regarding the ARCH test, it shows few rejections. There is however an issue with normality for the Nikkei index and also for the S&P500 index to a lesser extent. We found that it is due to very large negative returns that occur for these indexes and that are not captured by predictive densities. Note also that it is harder for ML predictives to achieve normality for the S&P500 index and that tail PITs of the beta linear pool are often not normally distributed.

[Insert Table 5 here.]

We reported in Subsection 4.2 that Bayesian predictives integrating parameter uncertainty improve 1% VaR forecasts for single models. We found here that 1% VaR forecasts from Bayesian

predictives are also better under linear and log pooling. Besides this, we observed that linear and log pools are preferred to single models at the 5% risk level. We also highlighted that they provide a more homogeneous VaR backtest than single models. This suggests that these methods are robust against their worst performing components. Among all forecasting techniques, the better VaR forecaster tend to be the EW linear pool of Bayesian predictives. This result can be explained by the superiority of VaR forecasts derived from Bayesian predictives and by the success of the EW combination in the literature. We also remarked in our own experiments that the optimal weight estimates exhibit considerable uncertainty. Finally, focusing on the left-tail with COP does not provide the expected results.

## 7. Conclusion

In this study, we use a set of GARCH-type models to assess the influence of parameter and model uncertainty on the left-tail of predictive densities and especially on VaR estimates. Our main findings can be summarized as follows. First, accounting for parameter uncertainty within the Bayesian framework improves the VaR backtest at the 1% risk level for single models as well as linear and log pools. It tends also to improve left-tail forecasts as indicated by tail PIT analyses. Second, accounting for model uncertainty via linear or log pooling produces robust VaR backtests. Moreover, these pooling methods present better VaR estimates than single models at the 5% risk level. Third, the EW linear pool of Bayesian predictive densities tends to be the best VaR forecaster among 42 forecasting techniques.

Regarding the relative performance of single models, the worst VaR estimates are undoubtedly provided by the EGARCH model while the best ones tend to be produced by the GJR model with a preference for skew GED innovations. We also demonstrate that GAS models with skew Student and GED innovations are competitive GARCH-type models. Lastly, we propose a novel combination scheme – the beta linear pool with COP weights – that simultaneously calibrates and focuses on the left-tail. This method gives puzzling results in our applications. Its VaR forecasts

can show correct unconditional coverage while generating dependent VaR violations. In further research, it would be interesting to refine this method and to use it in other applications. Another interesting extension would be to see how priors that amplify or moderate time-varying volatility affect left-tail forecasts incorporating parameter and/or model uncertainty.

## References

- Ardia, D., 2008. Financial risk management with Bayesian estimation of GARCH models: Theory and applications. Springer, Berlin.
- Ardia, D., Kolly, J., 2016. Predicting market risk with density combination: An introduction. *Wilmott Journal* 81, 52–57.
- Aussenegg, W., Miazhyńska, T., 2006. Uncertainty in Value-at-Risk estimates under parametric and nonparametric modeling. *Financial Markets and Portfolio Management* 20 (3), 243–264.
- Bassetti, F., Casarin, R., Ravazzolo, F., 2015. Bayesian nonparametric calibration and combination of predictive distributions. Working Paper.
- Berkowitz, J., Christoffersen, P. F., Pelletier, D., 2011. Evaluating Value-at-Risk models with desk-level data. *Management Science* 57 (12), 2213–2227.
- Black, F., 1976. Studies of stock price volatility changes. In: *Proceedings of the 1976 Business Meeting of the Business and Economic Statistics Section*. American Statistical Association, pp. 177–181.
- Bollerslev, T., 1986. Generalized autoregressive conditional heteroskedasticity. *Journal of Econometrics* 31 (3), 307–327.
- Bollerslev, T., 1987. A conditionally heteroskedastic time series model for speculative prices and rates of return. *Review of Economics and Statistics* 69 (3), 542–547.
- Casarin, R., Mantoan, G., Ravazzolo, F., 2016. Bayesian calibration of generalized pools of predictive distributions. *Econometrics* 4 (1), 17.
- Christoffersen, P. F., 1998. Evaluating interval forecasts. *International Economic Review* 39 (4), 841–862.
- Christoffersen, P. F., Pelletier, D., 2004. Backtesting Value-at-Risk: A duration-based approach. *Journal of Financial Econometrics* 2 (1), 84–108.
- Clemen, R. T., 1989. Combining forecasts: A review and annotated bibliography. *International Journal of Forecasting* 5 (4), 559–583.



- Creal, D., Koopman, S. J., Lucas, A., 2013. Generalized autoregressive score models with applications. *Journal of Applied Econometrics* 28 (5), 777–795.
- Deschamps, P. J., 2012. Bayesian estimation of generalized hyperbolic skewed student GARCH models. *Computational Statistics and Data Analysis* 56 (11), 3035–3054.
- Diebold, F. X., Gunther, T. A., Tay, A. S., 1998. Evaluating density forecasts with applications to financial risk management. *International Economic Review* 39 (4), 863–883.
- Diebold, F. X., Mariano, R. S., 1995. Comparing predictive accuracy. *Journal of Business and Economic Statistics* 13 (3), 253–263.
- Diks, C., Panchenko, V., van Dijk, D., 2011. Likelihood-based scoring rules for comparing density forecasts in tails. *Journal of Econometrics* 163 (2), 215–230.
- Ding, Z., Granger, C. W. J., Engle, R. F., 1993. A long memory property of stock market returns and a new model. *Journal of Empirical Finance* 1 (1), 83–106.
- Duffie, D., Pan, J., 1997. An overview of Value at Risk. *Journal of Derivatives* 4 (3), 7–49.
- Eklund, J., Karlsson, S., 2007. Forecast combination and model averaging using predictive measures. *Econometric Reviews* 26 (2-4), 329–363.
- Engle, R. F., 1982. Autoregressive conditional heteroscedasticity with estimates of the variance of United Kingdom inflation. *Econometrica* 50 (4), 987–1007.
- Engle, R. F., Manganelli, S., 2004. CAViaR: Conditional autoregressive Value at Risk by regression quantiles. *Journal of Business and Economic Statistics* 22 (4), 367–381.
- Fernández, C., Steel, M. F. J., 1998. On Bayesian modeling of fat tails and skewness. *Journal of the American Statistical Association* 93 (441), 359–371.
- Francq, C., Zakoian, J.-M., 2010. GARCH models: Structure, statistical inference and financial applications. Wiley, Chichester.
- Gatarek, L. T., Hoogerheide, L. F., Hooning, K., van Dijk, H. K., 2014. Censored posterior and predictive likelihood in Bayesian left-tail prediction for accurate Value at Risk estimation. Tinbergen Institute Discussion Paper.
- Genest, C., Zidek, J. V., 1986. Combining probability distributions: A critique and an annotated bibliography. *Statistical Science* 1 (1), 114–135.
- Geweke, J., Amisano, G., 2010. Comparing and evaluating Bayesian predictive distributions of asset returns. *International Journal of Forecasting* 26 (2), 216–230.
- Geweke, J., Amisano, G., 2011. Optimal prediction pools. *Journal of Econometrics* 164 (1), 130–141.
- Geweke, J., Amisano, G., 2012. Prediction with misspecified models. *American Economic Review* 102 (3), 482–486.

- Glosten, L. R., Jagannathan, R., Runkle, D. E., 1993. On the relation between the expected value and the volatility of the nominal excess return on stocks. *Journal of Finance* 48 (5), 1779–1801.
- Gneiting, T., Ranjan, R., 2011. Comparing density forecasts using threshold- and quantile-weighted scoring rules. *Journal of Business and Economic Statistics* 29 (3), 411–422.
- Gneiting, T., Ranjan, R., 2013. Combining predictive distributions. *Electronic Journal of Statistics* 7, 1747–1782.
- Hall, S. G., Mitchell, J., 2007. Combining density forecasts. *International Journal of Forecasting* 23 (1), 1–13.
- Hoogerheide, L. F., Ardia, D., Corré, N., 2012a. Density prediction of stock index returns using GARCH models: Frequentist or Bayesian estimation? *Economics Letters* 116 (3), 322–325.
- Hoogerheide, L. F., Opschoor, A., van Dijk, H. K., 2012b. A class of adaptive importance sampling weighted EM algorithms for efficient and robust posterior and predictive simulation. *Journal of Econometrics* 171 (2), 101–120.
- Jorion, P., 2007. *Value at Risk: The new benchmark for managing financial risk*, 3rd Edition. McGraw-Hill, New York.
- Kapetanios, G., Mitchell, J., Price, S., Fawcett, N., 2015. Generalised density forecast combinations. *Journal of Econometrics* 188 (1), 150–165.
- Kascha, C., Ravazzolo, F., 2010. Combining inflation density forecasts. *Journal of Forecasting* 29 (1-2), 231–250.
- Krüger, F., 2014. Combining density forecasts under various scoring rules: An analysis of UK inflation. Working Paper (version of March 3, 2014).
- Kupiec, P. H., 1995. Techniques for verifying the accuracy of risk measurement models. *The Journal of Derivatives* 3 (2), 73–84.
- Leamer, E. E., 1978. *Specification searches: Ad hoc inference with nonexperimental data*. Wiley, New York.
- Massacci, D., 2015. Predicting the distribution of stock returns: Model formulation, statistical evaluation, VaR analysis and economic significance. *Journal of Forecasting* 34 (3), 191–208.
- Moral-Benito, E., 2015. Model averaging in economics: An overview. *Journal of Economic Surveys* 29 (1), 46–75.
- Nelson, D. B., 1991. Conditional heteroskedasticity in asset returns: A new approach. *Econometrica* 59 (2), 347–370.
- Opschoor, A., van Dijk, D., van der Wel, M., 2015. Combining density forecasts using censored likelihood scoring rules. Tinbergen Institute Discussion Paper.
- Pesaran, M. H., Schleicher, C., Zaffaroni, P., 2009. Model averaging in risk management with an application to futures markets. *Journal of Empirical Finance* 16 (2), 280–305.
- Ranjan, R., Gneiting, T., 2010. Combining probability forecasts. *Journal of the Royal Statistical Society: Series B* 72 (1), 71–91.
- Smith, J., Wallis, K. F., 2009. A simple explanation of the forecast combination puzzle. *Oxford Bulletin of Economics*

- and Statistics 71 (3), 331–355.
- Stock, J. H., Watson, M. W., 2004. Combination forecasts of output growth in a seven-country data set. *Journal of Forecasting* 23 (6), 405–430.
- Stone, M., 1961. The opinion pool. *The Annals of Mathematical Statistics* 32 (4), 1339–1342.
- Tay, A. S., Wallis, K. F., 2000. Density forecasting: A survey. *Journal of Forecasting* 19 (4), 235–254.
- Tierney, L., 1994. Markov chains for exploring posterior distributions. *The Annals of Statistics* 22 (4), 1701–1728.
- Trottier, D.-A., Ardia, D., 2015. Moments of standardized Fernández-Steel skewed distributions: Applications to the estimation of GARCH-type models. Working Paper.
- Zakoian, J.-M., 1994. Threshold heteroskedastic models. *Journal of Economic Dynamics and Control* 18 (5), 931–955.
- Ziggel, D., Berens, T., Weiss, G. N. F., Wied, D., 2014. A new set of improved Value-at-Risk backtests. *Journal of Banking and Finance* 48, 29–41.

Table 1: Numbers of rejections at the 5% significance level over 8 indexes (CAC40, DAX, DJIA, FTSE, Nikkei, NASDAQ, SMI and S&P500) for the backtest of our GARCH-type models using 2015 VaR estimates obtained from one-day ahead Bayesian, PM and ML predictives. The UC, MCS and CAViaR tests are presented in Subsection 4.1. Definitions of model acronyms can be obtained in Table A.1.

	Bayesian predictives			PM predictives			ML predictives		
	UC	MCS	CAViaR	UC	MCS	CAViaR	UC	MCS	CAViaR
<b>1% risk level</b>									
egarch.st	3	3	2	3	4	3	3	3	2
egarch.sged	3	3	1	3	3	1	3	4	2
gjr.st	0	1	0	0	1	0	0	1	1
gjr.sged	0	0	0	0	1	0	1	1	1
tgarch.st	2	1	0	2	1	0	1	1	0
tgarch.sged	1	1	0	1	1	0	1	1	0
gas.st	1	2	1	2	4	1	2	4	1
gas.sged	0	0	0	0	1	1	2	4	1
<b>5% risk level</b>									
egarch.st	1	2	2	1	3	2	2	2	1
egarch.sged	0	0	2	0	1	2	1	2	2
gjr.st	0	0	1	0	0	1	0	2	0
gjr.sged	0	0	0	0	0	0	0	0	0
tgarch.st	0	1	2	0	1	2	1	2	1
tgarch.sged	0	0	2	0	0	2	1	1	1
gas.st	1	3	0	1	2	0	1	2	0
gas.sged	0	0	0	0	0	0	0	0	0

Table 2: Numbers of rejections at the 5% significance level over 8 indexes (CAC40, DAX, DJIA, FTSE, Nikkei, NASDAQ, SMI and S&P500) for the analysis of rescaled PITs below 5%. These PITs are obtained from 2015 one-day ahead Bayesian, PM and ML predictives produced by our GARCH-type models. The ARCH, JB and LR tests are presented in Subsection 4.1. Definitions of model acronyms can be obtained in Table A.1.

	Bayesian predictives			PM predictives			ML predictives		
	ARCH	JB	LR	ARCH	JB	LR	ARCH	JB	LR
egarch.st	1	1	2	0	1	4	2	1	2
egarch.sged	2	3	2	0	3	3	0	4	5
gjr.st	0	2	1	1	3	3	0	2	1
gjr.sged	0	4	2	1	4	3	0	4	3
tgarch.st	1	5	1	0	1	3	0	1	1
tgarch.sged	0	4	2	0	4	1	0	4	2
gas.st	0	2	1	1	3	3	1	2	1
gas.sged	0	2	3	0	3	5	1	3	4

Table 3: Statistical significance results for the backtest of one-day ahead VaR forecasts at the 1% risk level obtained from Bayesian and ML predictive densities produced by our GARCH-type models and from combinations of these densities. VaR forecasts are generated for 2015 daily returns of the DJIA, FTSE, Nikkei and S&P500 indexes. The symbols \*, \*\* and \*\*\* indicate significance at 10%, 5% and 1%, respectively. The UC, MCS and CAViaR tests are presented in Subsection 4.1. Definitions of model acronyms can be obtained in Table A.1.

	Bayesian predictives			ML predictives		
	UC	MCS	CAViaR	UC	MCS	CAViaR
<b>DJIA</b>						
egarch.st	**	***	***	**	***	***
egarch.sged	**	**	***	***	***	***
gjr.st					*	**
gjr.sged						*
tgarch.st						
tgarch.sged					*	
gas.st		*			*	**
gas.sged			*	*	*	***
lin.ew						
lin.bma						
lin.op						*
lin.cop-0.15			*		*	
lin.cop-0.25			*			
log.ew					*	
log.bma		*	*			
log.op			*			
log.cop-0.15			**			*
log.cop-0.25			**		*	
beta.op			***			**
beta.cop-0.15			***			***
beta.cop-0.25			***			***
<b>FTSE</b>						
egarch.st						
egarch.sged					*	
gjr.st						
gjr.sged		*				
tgarch.st						
tgarch.sged		*	*		*	
gas.st	*	*		**	**	*
gas.sged		*		**	**	*
lin.ew				*	**	
lin.bma		*		*	**	
lin.op		*		**	**	
lin.cop-0.15				*	**	
lin.cop-0.25				*	**	
log.ew				**	**	
log.bma		*		*	**	
log.op				*	**	
log.cop-0.15				*	*	
log.cop-0.25				*	**	
beta.op	*	**	***	**	**	***
beta.cop-0.15			**			*
beta.cop-0.25			**			**

Table 3: Continued from previous page

	Bayesian predictives			ML predictives		
	UC	MCS	CAViaR	UC	MCS	CAViaR
<b>Nikkei</b>						
egarch.st					*	
egarch.sged				*	*	
gjr.st						
gjr.sged						
tgarch.st						
tgarch.sged						
gas.st						
gas.sged						
lin.ew						
lin.bma						
lin.op						
lin.cop-0.15						
lin.cop-0.25						
log.ew						
log.bma						
log.op						
log.cop-0.15						
log.cop-0.25						
beta.op	**	** *	** *	*	**	*
beta.cop-0.15		*				*
beta.cop-0.25	**	**	**	**	**	*
<b>S&amp;P500</b>						
egarch.st	**	** *	*	**	** *	*
egarch.sged	**	**		**	**	
gjr.st					*	
gjr.sged						
tgarch.st						
tgarch.sged						
gas.st		*		*	**	*
gas.sged					*	
lin.ew						
lin.bma				*	*	
lin.op		*			*	
lin.cop-0.15		*			*	
lin.cop-0.25		*				
log.ew					*	
log.bma				*	**	*
log.op	*	**	*	*	**	*
log.cop-0.15		*			*	
log.cop-0.25	*	**				
beta.op	**	**	** *		*	**
beta.cop-0.15			**			*
beta.cop-0.25			*			*

Table 4: Statistical significance results for the backtest of one-day ahead VaR forecasts at the 5% risk level obtained from Bayesian and ML predictive densities produced by our GARCH-type models and from combinations of these densities. VaR forecasts are generated for 2015 daily returns of the DJIA, FTSE, Nikkei and S&P500 indexes. The symbols \*, \*\* and \*\*\* indicate significance at 10%, 5% and 1%, respectively. The UC, CC, MCS and CAViaR tests are presented in Subsection 4.1. Definitions of model acronyms can be obtained in Table A.1.

	Bayesian predictives				ML predictives			
	UC	CC	MCS	CAViaR	UC	CC	MCS	CAViaR
<b>DJIA</b>								
egarch.st			*				*	
egarch.sged								
gjr.st			*				*	
gjr.sged								
tgarch.st								
tgarch.sged								
gas.st	**	*	**		*		**	
gas.sged			*				*	
lin.ew								
lin.bma								
lin.op								
lin.cop-0.15								
lin.cop-0.25								
log.ew								
log.bma								
log.op								
log.cop-0.15							*	
log.cop-0.25								
beta.op				*				**
beta.cop-0.15								
beta.cop-0.25								
<b>FTSE</b>								
egarch.st	*	**	**	**				*
egarch.sged		**		**		*		**
gjr.st			*				*	
gjr.sged			*					
tgarch.st		**	*	**	*	*	**	*
tgarch.sged		** *	*	**		*	*	*
gas.st	*		**				*	
gas.sged							*	
lin.ew								
lin.bma		*	*				*	
lin.op		*	**				*	
lin.cop-0.15		*	*				*	
lin.cop-0.25			*					
log.ew			*					
log.bma		*	*				*	
log.op		*	*				*	
log.cop-0.15		*	*				*	
log.cop-0.25		*	**				*	
beta.op		**	*	** *				** *
beta.cop-0.15				** *				** *
beta.cop-0.25				** *				** *



Table 4: Continued from previous page

	Bayesian predictives				ML predictives			
	UC	CC	MCS	CAViaR	UC	CC	MCS	CAViaR
<b>Nikkei</b>								
egarch.st								
egarch.sged								
gjr.st								
gjr.sged								
tgarch.st								
tgarch.sged								
gas.st								
gas.sged								
lin.ew								
lin.bma								
lin.op								
lin.cop-0.15								
lin.cop-0.25								
log.ew								
log.bma								
log.op								
log.cop-0.15								
log.cop-0.25								
beta.op								
beta.cop-0.15								
beta.cop-0.25								
<b>S&amp;P500</b>								
egarch.st	**	*	**		**	*	** *	
egarch.sged			*		**		**	
gjr.st			*		*		**	
gjr.sged								
tgarch.st								
tgarch.sged								
gas.st	*		**		**	*	**	
gas.sged							*	
lin.ew								
lin.bma								
lin.op							*	
lin.cop-0.15								
lin.cop-0.25								
log.ew								
log.bma								
log.op							*	
log.cop-0.15								
log.cop-0.25							*	
beta.op								
beta.cop-0.15								
beta.cop-0.25								

Table 5: Statistical significance results for the analysis of rescaled PITs for observations below one-day ahead VaR forecasts at the 5% risk level obtained from Bayesian and ML predictive densities and combinations of such densities. VaR forecasts are generated for 2015 daily returns of the DJIA, FTSE, Nikkei and S&P500 indexes. The symbols \*, \*\* and \*\*\* indicate significance at 10%, 5% and 1%, respectively. The ARCH, JB and LR tests are presented in Subsection 4.1. Definitions of model acronyms can be obtained in Table A.1.

	Bayesian predictives			ML predictives		
	ARCH	JB	LR	ARCH	JB	LR
<b>DJIA</b>						
egarch.st						
egarch.sged			*			**
gjr.st						
gjr.sged		** *			** *	
tgarch.st		**				
tgarch.sged		*			** *	
gas.st						
gas.sged					*	
lin.ew						
lin.bma					*	
lin.op						
lin.cop-0.15		*				
lin.cop-0.25		** *				
log.ew		*				
log.bma					** *	
log.op					**	
log.cop-0.15		**				
log.cop-0.25		**			*	
beta.op					** *	
beta.cop-0.15		** *			** *	
beta.cop-0.25		** *			** *	
<b>FTSE</b>						
egarch.st				**		
egarch.sged						**
gjr.st				*		
gjr.sged			**			**
tgarch.st	*					
tgarch.sged	**					*
gas.st	*					*
gas.sged	*		*			**
lin.ew	**					
lin.bma						
lin.op						
lin.cop-0.15						
lin.cop-0.25						
log.ew						*
log.bma						
log.op						
log.cop-0.15						
log.cop-0.25						
beta.op						
beta.cop-0.15						
beta.cop-0.25						

Table 5: Continued from previous page

	Bayesian predictives			ML predictives		
	ARCH	JB	LR	ARCH	JB	LR
<b>Nikkei</b>						
egarch.st			***			**
egarch.sged		***	***		***	***
gjr.st		**	*		***	**
gjr.sged		***	**		***	***
tgarch.st			**			**
tgarch.sged		***	***		***	***
gas.st		***	***		***	***
gas.sged		***	***		***	***
lin.ew		***	***		**	**
lin.bma		***	**		***	***
lin.op		**	**		***	**
lin.cop-0.15		***	**		***	**
lin.cop-0.25		***	**		***	***
log.ew		***	***		***	***
log.bma		***	***		***	***
log.op		***	**		***	***
log.cop-0.15	**	***	***		***	***
log.cop-0.25		***	***		***	***
beta.op			**			**
beta.cop-0.15		**			***	
beta.cop-0.25		*	**		***	**
<b>S&amp;P500</b>						
egarch.st						
egarch.sged					***	**
gjr.st						
gjr.sged		**			***	
tgarch.st						
tgarch.sged		***	*		***	
gas.st						
gas.sged		**			**	
lin.ew			*		**	
lin.bma					**	*
lin.op						
lin.cop-0.15					***	
lin.cop-0.25					***	
log.ew					***	
log.bma					***	
log.op					***	
log.cop-0.15					***	
log.cop-0.25					***	
beta.op		*	*		***	**
beta.cop-0.15		***	*		***	
beta.cop-0.25		***			***	

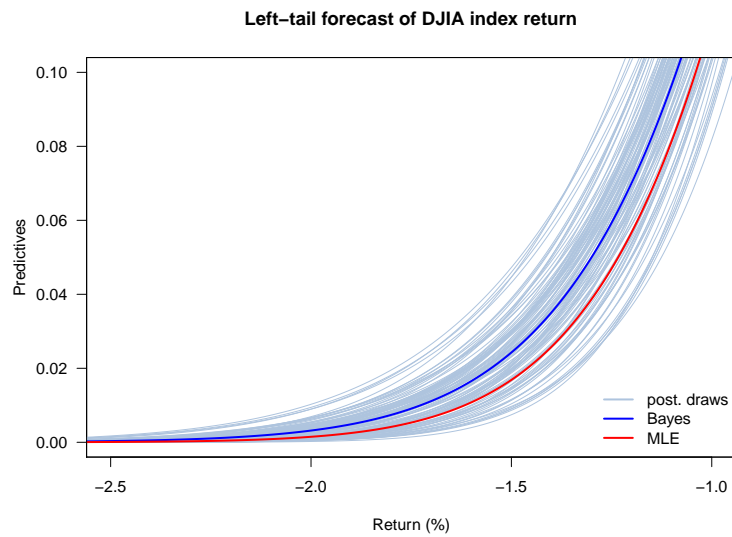


Figure 1: Typical left-tail of predictive densities generated by a GARCH-type model when we use particular posterior draws (light blue), when parameter uncertainty is integrated using the Bayesian approach (blue) and when we plug ML estimates in the predictive (red). The Bayesian predictive is more conservative than the ML one and accounts for more likely scenarios.

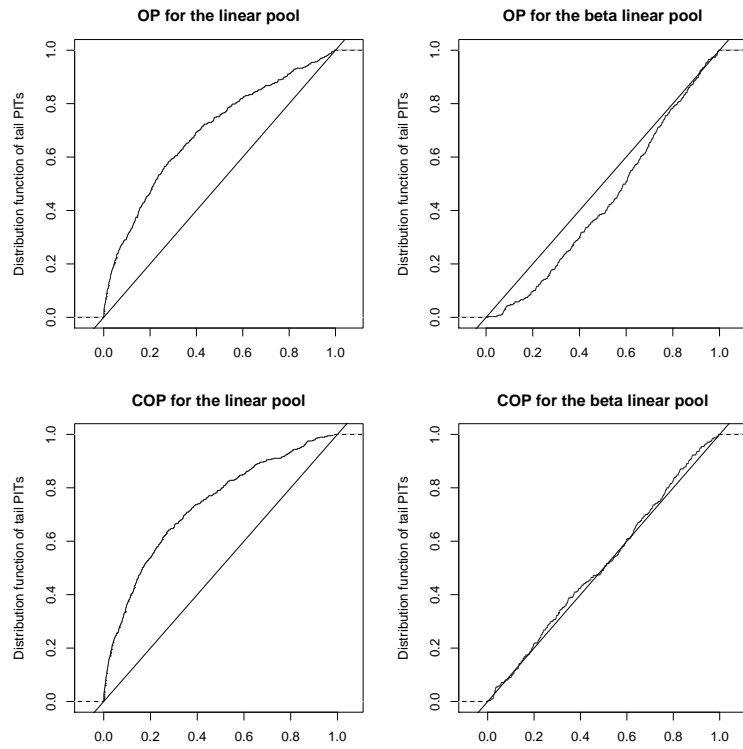


Figure 2: Empirical distribution functions of tail PITs rescaled to lie between 0 and 1 with ideal straight lines passing through 0 and 1. Tail PITs are produced by linear and beta linear pools of misspecified models over a forecasting period. Weights are estimated with the OP and COP methods from observations of an estimation period.

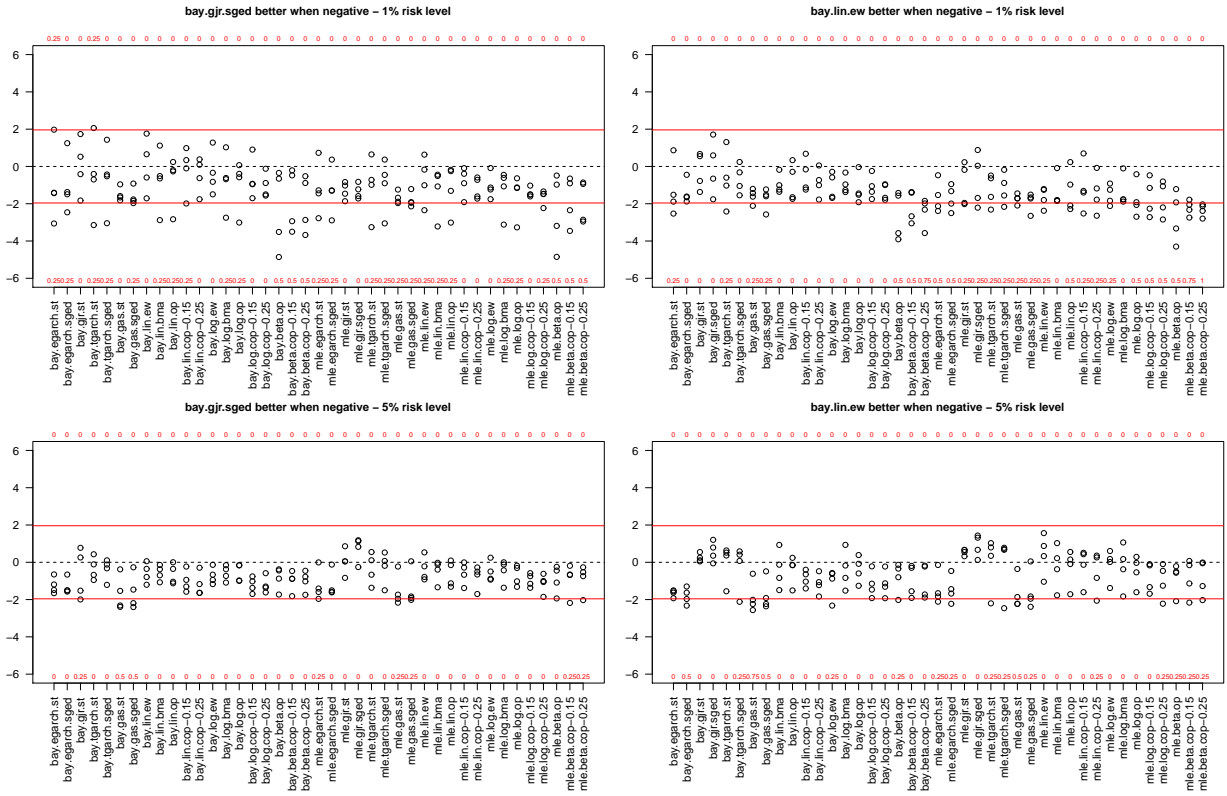


Figure 3: DM test statistics for significance of differences between mean (tick) asymmetric linear losses computed for 2015 daily returns of the DJIA, FTSE, Nikkei and S&P500 indexes. Left panels compare the skew GED GJR model with Bayesian predictives to all other models. Right panels compare the EW linear pool of Bayesian predictives to all other models. Top panels use 1% VaR forecasts while bottom panels consider 5% VaR forecasts. For each model pair, each circle corresponds to a different index. Red lines indicate asymptotic critical values at the 5% level. More details on the testing methodology can be found in Subsection 4.1. Definitions of model acronyms can be obtained in Table A.1.

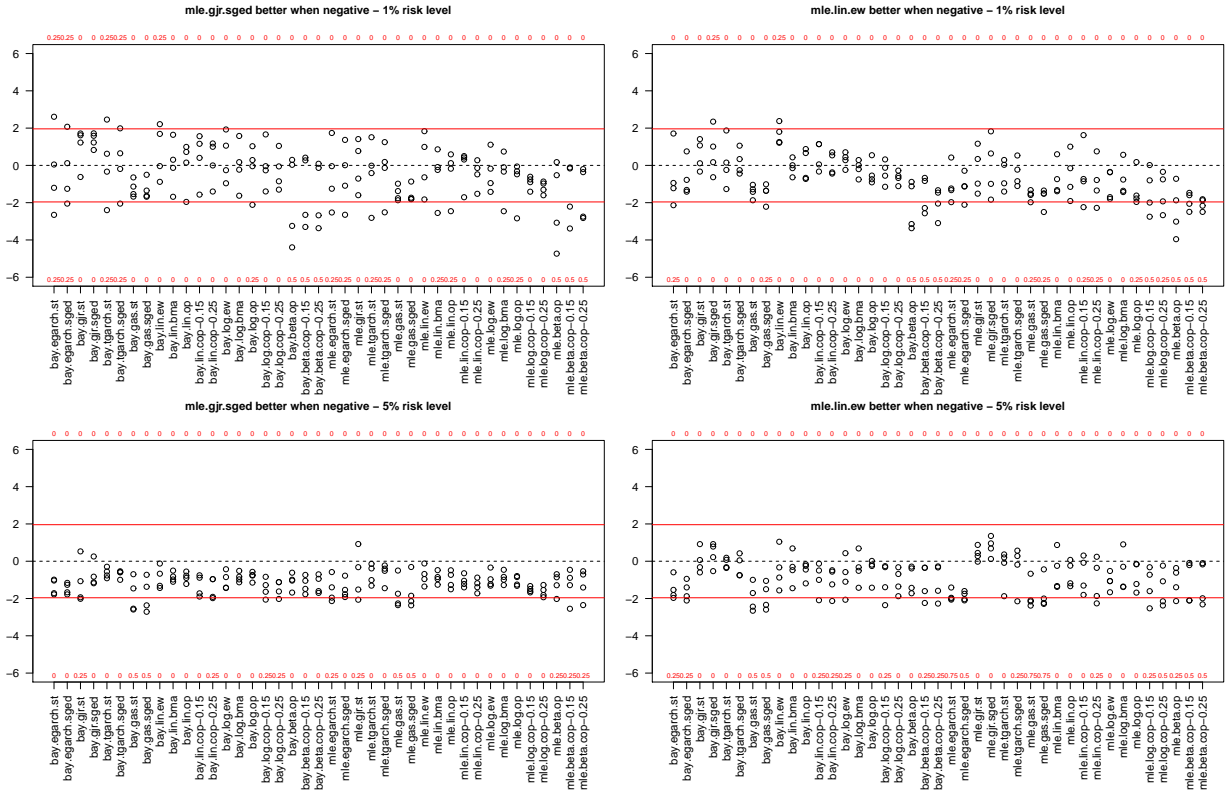


Figure 4: DM test statistics for significance of differences between mean (tick) asymmetric linear losses computed for 2015 daily returns of the DJIA, FTSE, Nikkei and S&P500 indexes. Left panels compare the skew GED GJR model with ML predictives to all other models. Right panels compare the EW linear pool of ML predictives to all other models. Top panels use 1% VaR forecasts while bottom panels consider 5% VaR forecasts. For each model pair, each circle corresponds to a different index. Red lines indicate asymptotic critical values at the 5% level. More details on the testing methodology can be found in Subsection 4.1. Definitions of model acronyms can be obtained in Table A.1.

## Appendix

### A. Some acronyms

Table A.1: Definitions of model acronyms that are used in Tables and Figures for Bayesian predictives. Model acronyms for ML predictives are similar but start with “mle”. Sometimes the type of predictives used is not specified in the acronym when it is clear from the context.

---

bay.egarch.st	Skew Student EGARCH model (Bayesian predictives)
bay.egarch.sged	Skew GED EGARCH model (Bayesian predictives)
bay.gjr.st	Skew Student GJR model (Bayesian predictives)
bay.gjr.sged	Skew GED GJR model (Bayesian predictives)
bay.tgarch.st	Skew Student TGARCH model (Bayesian predictives)
bay.tgarch.sged	Skew GED TGARCH model (Bayesian predictives)
bay.gas.st	Skew Student GAS model (Bayesian predictives)
bay.gas.sged	Skew GED GAS model (Bayesian predictives)
bay.lin.ew	EW linear pool of Bayesian predictives
bay.lin.bma	Linear pool of Bayesian predictives with BMA weights
bay.lin.op	Linear pool of Bayesian predictives with OP weights
bay.lin.cop-0.15	Linear pool of Bayesian predictives with COP weights (15% censoring)
bay.lin.cop-0.25	Linear pool of Bayesian predictives with COP weights (25% censoring)
bay.log.ew	EW log pool of Bayesian predictives
bay.log.bma	Log pool of Bayesian predictives with BMA weights
bay.log.op	Log pool of Bayesian predictives with OP weights
bay.log.cop-0.15	Log pool of Bayesian predictives with COP weights (15% censoring)
bay.log.cop-0.25	Log pool of Bayesian predictives with COP weights (25% censoring)
bay.beta.op	Beta linear pool of Bayesian predictives with OP weights
bay.beta.cop-0.15	Beta linear pool of Bayesian predictives with COP weights (15% censoring)
bay.beta.cop-0.25	Beta linear pool of Bayesian predictives with COP weights (25% censoring)

---



# MSGARCH package manual

<https://CRAN.R-project.org/package=MSGARCH>

# Package ‘MSGARCH’

January 19, 2017

**Type** Package

**Title** Markov-Switching GARCH Models

**Version** 0.17.8

**Date** 2016-01-19

**Author** David Ardia [aut],  
Keven Bluteau [aut, cre],  
Kris Boudt [ctb],  
Brian Peterson [ctb],  
Denis-Alexandre Trottier [aut]

**Maintainer** Keven Bluteau <Keven.Bluteau@unine.ch>

**Description** The MSGARCH package offers methods to fit (by Maximum Likelihood or Bayesian), simulate, and forecast various Markov-Switching GARCH processes.

**License** GPL (>= 2)

**Imports** Rcpp, adaptMCMC, nloptr, DEoptim, methods, stringr, ggplot2, reshape2, zoo, expm, fanplot, dfoptim

**LinkingTo** Rcpp, RcppArmadillo

**RoxygenNote** 5.0.1

**NeedsCompilation** yes

## R topics documented:

MSGARCH-package	2
AIC	3
AMZN	4
BIC	4
cdf	5
create.spec	6
crps	9
DIC	10
fit.bayes	11
fit.mle	13
ht	15
kernel	16
pdf	17
pit	18
pred	20

Pstate . . . . .	21
risk . . . . .	22
sim . . . . .	24
simahead . . . . .	25
sp500 . . . . .	26
transmat . . . . .	26
unc.vol . . . . .	27

<b>Index</b>	<b>29</b>
--------------	-----------

---

MSGARCH-package	<i>The R package MSGARCH</i>
-----------------	------------------------------

---

## Description

The R package MSGARCH aims to provide a comprehensive set of functionalities for Markov-switching GARCH processes (Haas et al. 2004), including fitting, filtering, forecasting, and simulating. Other functions related to Value-at-Risk, Expected-Shortfall, and conditional distributions are also available. The main functions of the package are coded in C++ with Rcpp (Eddelbuettel and Francois, 2011) and RcppArmadillo (Eddelbuettel and Sanderson, 2014). In the R package MSGARCH there is no equation for the mean as in the R package rugarch (Ghalanos, 2015). This means that we assume that before modeling, the user has filtered the mean from their time series.

We provide a variety of single-regime GARCH processes and regime-switching allowing for many conditional distributions. This allows for a rich modeling environment for Markov-switching GARCH models. Each single-regime process is a one-lag process (e.g., GARCH(1,1)). Allowing for only one-lag has proved to be sufficient in many cases and it reduces models complexity which can become a problem during the optimization procedure. When optimization is taking place, we ensure that each regime is covariance-stationary and strictly positive (see details in [kernel](#) for more information) which makes the entire process also covariance-stationary and strictly positive. We also set a condition that each unique single-regime models type in a multiple-regime framework are in order of unconditional volatility. This means that is if a three regimes specification with two sGARCH regimes and one gjrGARCH regime is constructed with [create.spec](#), the first sGARCH regime will have a lower unconditional volatility than the second sGARCH regime while the gjrGARCH regime can have any unconditional volatility since it is the only regime with this model. For a full demonstration of the package please read Markov-Switching GARCH Models in R: The MSGARCH Package (see <https://ssrn.com/abstract=2845809>). The authors acknowledge Google for financial support via the Google Summer of Code 2016 project "MSGARCH"; see <https://summerofcode.withgoogle.com/projects/#6497774455488512>, the International Institute of Forecasting and Industrielle-Alliance.

## References

- Ardia, D. and Bluteau, K. and Boudt, K. and Trottier, D.-A. (2016). Markov-Switching GARCH Models in R: The MSGARCH Package. <https://ssrn.com/abstract=2845809>
- Eddelbuettel, D. Francois, R. (2011). Rcpp: Seamless R and C++ Integration. *Journal of Statistical Software*, 40, pp. 1-18, <http://www.jstatsoft.org/v40/i08/>.
- Eddelbuettel, D. Sanderson, C. (2014). RcppArmadillo: Accelerating R with High-Performance C++ Linear Algebra. *Computational Statistics & Data Analysis*, 71, pp. 1054-1063, <http://dx.doi.org/10.1016/j.csda.2013.02.005>.
- Haas, M. Mittnik, S. Paoletta, MS. (2004). A New Approach to Markov-Switching GARCH Models. *Journal of Financial Econometrics*, 2, pp. 493-530, <http://doi.org/10.1093/jffinec/nbh020>

Ghalanos, A. (2015). rugarch: Univariate GARCH Models. <https://cran.r-project.org/package=rugarch>.

---

AIC	<i>Compute Akaike information criterion (AIC).</i>
-----	--

---

## Description

Compute Akaike information criterion (AIC).

## Usage

```
AIC(fit)
```

## Arguments

fit	Fit object of type MSGARCH_MLE_FIT created with <code>fit.mle</code> or MSGARCH_BAY_FIT created with <code>fit.bayes</code> .
-----	---

## Details

Compute Akaike information criterion (AIC) based on the work of Akaike (Akaike, 1974). If a matrix of MCMC posterior draws estimates is given, the AIC on the posterior mean is calculated.

## Value

AIC value.

## References

Akaike, H. (1974). A New Look at the Statistical Model Identification. *IEEE Transactions on Automatic Control*, 19, pp. 716-723.

## Examples

```
require("MSGARCH")
# load data
data("sp500")
sp500 = sp500[1:1000]

# create model specification
spec = MSGARCH::create.spec()

# fit the model by MLE
fit = MSGARCH::fit.mle(spec = spec, y = sp500, ctr = list(do.init = FALSE))

# compute AIC
AIC = MSGARCH::AIC(fit)
```

---

AMZN	<i>Log return of Amazon inc. closing Value</i>
------	--

---

**Description**

The Amazon inc. closing value log return from 1998-01-01 to 2015-12-31 from Yahoo Finance <https://finance.yahoo.com/>.

**Usage**

```
data("AMZN")
```

**Format**

Matrix containing 4,529 observations.

**Source**

Yahoo Finance <https://finance.yahoo.com/>

---

BIC	<i>Compute Bayesian information criterion (BIC).</i>
-----	--

---

**Description**

Compute Bayesian information criterion (BIC).

**Usage**

```
BIC(fit)
```

**Arguments**

fit	Fit object of type MSGARCH_MLE_FIT created with <a href="#">fit.mle</a> or MSGARCH_BAY_FIT created with <a href="#">fit.bayes</a>
-----	---

**Details**

Compute Bayesian information criterion (BIC) based on the work of Schwarz (Schwarz, 1978). If a matrix of MCMC posterior draws estimates is given, the BIC on the posterior mean is calculated.

**Value**

BIC value.

**References**

Schwarz, G. (1978). Estimating the dimension of a model. *Annals of Statistics*, 6, pp. 461-464.

## Examples

```
require("MSGARCH")
# load data
data("sp500")
sp500 = sp500[1:1000]

# create model specification
spec = MSGARCH::create.spec()

# fit the model by MLE
fit = MSGARCH::fit.mle(spec = spec, y = sp500, ctr = list(do.init = FALSE))

# compute BIC
BIC = MSGARCH::BIC(fit)
```

---

cdf	<i>Cumulative distribution function.</i>
-----	--

---

## Description

Method returning the cumulative distribution function in-sample or of a vector of points consider as one step ahead draws ( $t = T + 1$ ).

## Usage

```
cdf(object, x, theta, y, log = FALSE, do.its = FALSE)
```

## Arguments

object	Model specification of class MSGARCH_SPEC created with <a href="#">create.spec</a> or fit object of type MSGARCH_MLE_FIT created with <a href="#">fit.mle</a> or MSGARCH_BAY_FIT created with <a href="#">fit.bayes</a> .
x	Vector (of size N) of points evaluated at $t = T + 1$ (used when <code>do.its = FALSE</code> ).
theta	Vector (of size d) or matrix (of size M x d) of parameter estimates (not required when using a fit object) where d must have the same length as the default parameters of the specification.
y	Vector (of size T) of observations (not required when using a fit object).
log	Boolean indicating if the log cumulative is returned. (Default: <code>log = FALSE</code> )
do.its	Boolean indicating if the in-sample cdf is returned. (Default: <code>do.its = FALSE</code> )

## Details

If a matrix of parameter estimates is given, each parameter estimate (each row) is evaluated individually. If `do.its = FALSE`, the points x are evaluated as  $t = T + 1$  realizations and the method uses the variance estimate at  $t = T + 1$ . If `do.its = TRUE`, y is evaluated using their respective variance estimate at each time t.

## Value

A list of class MSGARCH\_CDF containing three components:

- `cdf`:  
If `do.its = FALSE`: (Log-)Cumulative of the points  $x$  at  $t = T + 1$  (vector of size  $N$  or matrix of size  $M \times N$ ).  
If `do.its = TRUE`: In-sample (Log-)Cumulative of  $y$  (vector of size  $T$  or matrix of size  $M \times T$ ).
- `x`:  
If `do.its = FALSE`: Vector (of size  $N$ ) of points evaluated at  $t = T + 1$ .  
If `do.its = TRUE`: Vector (of size  $T$ ) of observations.
- `do.its`: Orinigal user inputed `do.its` for reference.

The class MSGARCH\_CDF contains the plot method only if `do.its = FALSE`.

## Examples

```
require("MSGARCH")
# load data
data("sp500")
sp500 = sp500[1:1000]

# create model specification
spec = MSGARCH::create.spec()

# fit the model on the data with ML estimation
set.seed(123)
fit = MSGARCH::fit.mle(spec = spec, y = sp500, ctr = list(do.init = FALSE))

# run pdf method in-sample
cdf.its = MSGARCH::cdf(object = fit, log = FALSE, do.its = TRUE)

# create mesh
x = seq(-3, 3, 0.01)

# run cdf method on mesh at T + 1
cdf = MSGARCH::cdf(object = fit, x = x, log = FALSE, do.its = FALSE)

plot(cdf)
```

---

create.spec

*Model specification*

---

## Description

Function for creating a model specification before fitting and using the R package MSGARCH functionalities.

## Usage

```
create.spec(model = c("sGARCH", "sGARCH"), distribution = c("norm", "norm"),
  do.skew = c(FALSE, FALSE), do.mix = FALSE, do.shape.ind = FALSE)
```

## Arguments

<code>model</code>	Vector (of size $K$ ) containing the variance model specifications. Valid models are "sGARCH", "eGARCH", "gjrGARCH", "tGARCH", and "GAS". (Default: <code>model = c("sGARCH", "sGARCH")</code> )
<code>distribution</code>	Vector (of size $K$ ) of conditional densities. Valid distributions are "norm", "std", and "ged". The vector must be of the same length as the models' vector. (Default: <code>distribution = c("norm", "norm")</code> )
<code>do.skew</code>	Vector (of size $K$ ) of boolean indicating if the conditional density is skewed. The vector must be of the same length as the distributions' vector. (Default: <code>do.skew = c(FALSE, FALSE)</code> )
<code>do.mix</code>	Boolean indicating if the specification is a mixture type. If the argument is TRUE, a Mixture of GARCH is created, while if the argument is FALSE, a Markov-Switching GARCH is created (see details). (Default: <code>do.mix = FALSE</code> )
<code>do.shape.ind</code>	Boolean indicating if the distribution are Regime-Independent. If the argument is TRUE, all distributions are the same and the distribution parameters do not depend on the regime to which the distribution is attributed. If the argument is TRUE, all distributions in the distribution argument and all skew arguments must be the same. (Default: <code>do.shape.ind = FALSE</code> )

## Details

The Markov-Switching specification created is based on the Haas et al. (2004a) MSGARCH specification. It is a MSGARCH model that is separated in  $K$  single-regimes specifications which are updated in parallel. Under the Haas et al. (2004a) specification, the conditional variance is a function of the past data and the current state. The Mixture of GARCH option is based on Haas et al. (2004b). A Mixture of GARCH is a mixture of distributions where the variance process of each distribution is a single-regime process. Every single-regime specification is a one-lag process (e.g., GARCH(1,1)) since it has proved to be sufficient in financial econometrics. This simplification of the processes also reduces models' complexity which can become a problem during the optimization procedure.

## Value

A list of class MSGARCH\_SPEC containing variables related to the created specification.  
The list contains:

- `theta0` : Vector (of size  $d$ ) of default parameters.
- `is.mix` : Boolean indicating if the specification is a mixture.
- `is.shape.ind` : Boolean indicating if the distributions' parameters are regime-independent.
- `K` : Number of regimes.
- `sigma0` : Default variance-covariance matrix (of size  $K \times K$ ) used for the Bayesian estimation.
- `lower` : Vector (of size  $d$ ) of lower parameters' bounds.
- `upper` : Vector (of size  $d$ ) of upper parameters' bounds.
- `ineqlb` : Vector (of size  $d$ ) of lower inequality bounds.
- `inequb` : Vector (of size  $d$ ) of upper inequality bounds.
- `n.params` : Vector (of size  $K$ ) of the total number of parameters by regime including distributions' parameters.



- `n.params.vol` : Vector (of size K) of the total number of parameters by regime excluding distributions' parameters.
- `do.init` : Boolean indicating the default `do.init` argument.
- `label` : Vector (of size d) of parameters' labels.
- `name` : Vector (of size K) of model specifications' names.
- `func` : List of internally used R functions.
- `rcpp.func` : List of internally used Rcpp functions.

The MSGARCH\_SPEC class possesses these methods:

- `sim` : Simulation method.
- `simahead` : Step ahead simulation method.
- `ht` : Conditional volatility in each regime.
- `kernel` : Kernel method.
- `unc.vol` : Unconditional volatility in each regime.
- `pred` : Predictive method.
- `pit` : Probability Integral Transform.
- `risk` : Value-at-Risk And Expected-Shortfall methods.
- `pdf` : Probability density function.
- `cdf` : Cumulative distribution function.
- `Pstate` : State probabilities filtering method.
- `fit.mle` : Maximum Likelihood estimation.
- `fit.bayes` : Bayesian estimation.
- `print and summary` : Summary of the created specification.

## References

- Bollerslev, T. (1986). Generalized Autoregressive Conditional Heteroskedasticity. *Journal of Econometrics*, 31, pp. 307-327.
- Creal, D. Koopman, S. J. & Lucas, A. (2013). Generalized Autoregressive Score Models with Applications. *Journal of Applied Econometrics*, 28, pp. 777-795.
- Fernandez, C. & Steel, M. F. (1998). On Bayesian Modeling of Fat Tails and Skewness. *Journal of the American Statistical Association*, 93, pp. 359-371.
- Glosten, L. R. Jagannathan, R. & Runkle, D. E. (1993). On the Relation Between the Expected Value and the Volatility of the Nominal Excess Return on Stocks. *Journal of Finance*, 48, pp. 1779-1801.
- Haas, M. Mittnik, S. & Paolella, M. S. (2004a). A New Approach to Markov-Switching GARCH Models. *Journal of Financial Econometrics*, 2, pp. 493-530.
- Haas, M. Mittnik, S. & Paolella, M. S. (2004b). Mixed Normal Conditional Heteroskedasticity. *Journal of Financial Econometrics*, 2, pp. 211-250.
- Nelson, D. B. (1991). Conditional Heteroskedasticity in Asset Returns: A New Approach. *Econometrica*, 59, pp. 347-370.
- Zakoian, J.-M. (1994). Threshold Heteroskedastic Models. *Journal of Economic Dynamics and Control*, 18, pp. 931-955.

## Examples

```
# create model specification
spec = MSGARCH::create.spec(model = c("sGARCH", "gjrGARCH"), distribution = c("norm", "std"),
                             do.skew = c(TRUE, FALSE), do.mix = FALSE, do.shape.ind = FALSE)
print(spec)
```

---

crps

*CRPS (continuous ranked probability score) measure.*


---

## Description

Method returning the CRPS at  $t = T + 1$ .

## Usage

```
crps(object, yn, ctr = list(lower = -20, upper = 20, n.mesh = 500, a = 0, b = 1))
```

## Arguments

object	Model specification of class MSGARCH_SPEC created with <a href="#">create.spec</a> or fit object of type MSGARCH_MLE_FIT created with <a href="#">fit.mle</a> or MSGARCH_BAY_FIT created with <a href="#">fit.bayes</a> .
yn	Scalar value to be evaluated at $t = T + 1$ .
ctr	Control list parameters.

## Details

If a matrix of MCMC posterior draws estimates is given, the Bayesian CRPS is calculated.

## Value

A vector with five crps measures

## Examples

```
require("MSGARCH")
# load data
data("sp500")
sp500 = sp500[1:1000]

# create model specification
spec = MSGARCH::create.spec()

# fit the model on the data with ML estimation
set.seed(123)
fit = MSGARCH::fit.mle(spec = spec, y = sp500, ctr = list(do.init = FALSE))

# run at T + 1 from model
crps = MSGARCH::crps(object = fit, yn = 0.6)
```

DIC

*Compute Deviance Information Criterion (DIC).***Description**

Compute Deviance Information Criterion (DIC).

**Usage**

DIC(fit)

**Arguments**

fit                      Fit object of type MSGARCH\_BAY\_FIT created with [fit.bayes](#).

**Details**

Compute the Deviance information criterion of Spiegelhalter, David J., et al. (2002). We define the deviance as:

$$D(\theta) = -2LLH(\mathbf{y}|\theta),$$

where  $\mathbf{y}$  are the data,  $\theta$  are the parameters, and  $LLH()$  is the log-likelihood function. The expectation

$$\bar{D} = \mathbf{E}^\theta[D(\theta)],$$

where  $\mathbf{E}^\theta$  is the expectation over all theta in a MCMC chain, is a measure of how well the model fits the data. The larger this expectation is, the worse is the fit. The effective number of parameters of the model can be defined as

$$p_V = \frac{1}{2} \widehat{var}(D(\theta)),$$

where  $\widehat{var}$  is the the population variance estimator. The larger the effective number of parameters is, the easier it is for the model to fit the data, and so the deviance needs to be penalized. Finally DIC is defined as:

$$DIC = p_V + \bar{D}.$$

**Value**

A list containing four variables:

- DIC : Deviance Information Criterion.
- IC : Bayesian Predictive Information Criterion (IC = 2 \* pV + D.bar).
- pV : Effective number of parameters (pV = var(D)/2)
- D.bar: Expected value of the deviance over the posterior

**References**

Spiegelhalter, David J., et al. (2002). Bayesian measures of model complexity and fit. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*

## Examples

```
require("MSGARCH")
# load data
data("sp500")
sp500 = sp500[1:1000]

# create model specification
spec = MSGARCH::create.spec()

# fit the model by Bayesian estimation
set.seed(123)
fit = MSGARCH::fit.bayes(spec = spec, y = sp500)

# compute DIC
DIC = MSGARCH::DIC(fit)
```

---

fit.bayes

*Bayesian estimation.*


---

## Description

Method that performs Bayesian estimation of a MSGARCH\_SPEC object on a set of observations.

## Usage

```
fit.bayes(spec, y, ctr = list())
```

## Arguments

spec	Model specification of class MSGARCH_SPEC created with <a href="#">create.spec</a> .
y	Vector (of size T) of observations.
ctr	A list of control parameters. The control parameters have three components: <ul style="list-style-type: none"> <li>• N.burn (integer <math>\geq 0</math>): Number of discarded draws. (Default: N.burn = 5000)</li> <li>• N.mcmc (integer <math>&gt; 0</math>): Number of draws. (Default: N.mcmc = 10000)</li> <li>• N.thin (integer <math>&gt; 0</math>): Thinning factor (every N.thin draws are kept). (Default: N.thin = 10)</li> <li>• theta0 : Starting value for the chain (if empty the specification default values are used).</li> <li>• do.enhance.theta0 : Boolean indicating if the default parameters values are enhanced using y variance. (Default: do.enhance.theta0 = FALSE)</li> </ul>

## Details

The total number of draws is equal to  $N.mcmc / N.thin$ . The Bayesian estimation uses the R package adaptMCMC (Andreas, 2012) which implements the adaptive sampler of Vihola (2012). The starting parameters are the specification default parameters. The argument do.enhance.theta0 uses the volatilities of rolling windows of y and adjusts the default parameter of the specification so that the unconditional volatility of each regime is set to different quantiles of the volatilities of the rolling windows of y.

## Value

A list of class MSGARCH\_BAY\_FIT containing four components:

- `theta` : The MCMC chain (matrix from the R package coda (Plummer et al., 2006) of size `N.mcmc / N.thin`).
- `accept` : Acceptation rate of the sampler.
- `y` : Vector (of size T) of observations.
- `spec` : Model specification of class MSGARCH\_SPEC created with `create.spec`.

The MSGARCH\_BAY\_FIT contains these methods:

- `AIC` : Compute Akaike information criterion (AIC).
- `BIC` : Compute Bayesian information criterion (BIC).
- `DIC` : Compute Deviance Information Criterion (DIC).
- `ht` : Conditional volatility in each regime.
- `kernel` : Kernel method.
- `unc.vol` : Unconditional volatility in each regime.
- `pred` : Predictive method.
- `pit` : Probability Integral Transform.
- `risk` : Value-at-Risk And Expected-Shortfall methods.
- `simahead` : Step ahead simulation method.
- `sim` : Simulation method.
- `pdf` : Probability density function.
- `cdf` : Cumulative distribution function.
- `Pstate` : State probabilities filtering method.
- `summary` : Summary of the fit.

## References

- Andreas, S. (2012). adaptMCMC: Implementation of a Generic Adaptive Monte Carlo Markov Chain Sampler. <https://cran.r-project.org/package=adaptMCMC>.
- Plummer, M. Best, N. Cowles, K. & Vines, K. (2006). CODA: Convergence Diagnosis and Output Analysis for MCMC. *R News*, 6, pp.7-11. <https://cran.r-project.org/package=coda>.
- Vihola, M. (2012). Robust Adaptive Metropolis Algorithm with Coerced Acceptance Rate. *Statistics and Computing*, 22, pp. 997-1008.

## Examples

```
require("MSGARCH")
# load data
data("sp500")
sp500 = sp500[1:1000]

# create model specification
spec = MSGARCH::create.spec()

# fit the model on the data with Bayesian estimation
set.seed(123)
```

```
fit = MSGARCH::fit.bayes(spec = spec, y = sp500,
                        ctr = list(N.burn = 500, N.mcmc = 1000, N.thin = 1))

summary(fit)
```

fit.mle

*ML estimation.*

## Description

Method that performs Maximum Likelihood estimation of a MSGARCH\_SPEC object on a set of observations.

## Usage

```
fit.mle(spec, y, ctr = list())
```

## Arguments

spec	Model specification created with <a href="#">create.spec</a> .
y	Vector (of size T) of observations.
ctr	List of control parameters. The control parameters have two components to it: <ul style="list-style-type: none"> <li>• <code>do.init</code> : Boolean indicating if there is a pre-optimization with the R package DEoptim (Ardia et al., 2011). (Default: <code>do.init = FALSE</code>)</li> <li>• <code>NP</code> : Number of parameter vectors in the population in DEoptim optimization. (Default: <code>NP = 200</code>)</li> <li>• <code>itermax</code> : Maximum iteration (population generation) allowed in DEoptim optimization. (Default: <code>maxit = 200</code>)</li> <li>• <code>theta0</code> : Starting value for the chain (if empty the specification default value are used).</li> <li>• <code>do.enhance.theta0</code> : Boolean indicating if the default parameters value are enhance using y variance. (Default: <code>do.enhance.theta0 = TRUE</code>)</li> </ul>

## Details

The Maximum likelihood estimation uses the R package `dfoptim` (Varadhan and Borchers, 2016) for main optimizer and `nloptr` (Johnson, 2014) in case of non-convergence while it uses the R package `DEoptim` when `do.init = TRUE` as an initialization for `dfoptim` and `nloptr`. The starting parameters are the specification default parameters. The argument `do.enhance.theta0` uses the volatilities of rolling windows of y and adjust the starting parameters of the specification so that the unconditional volatility of each regime is set to different quantiles of the volatilities of the rolling windows of y.

## Value

A list of class MSGARCH\_MLE\_FIT containing five components:

- `theta` : Optimal parameters (vector of size d).
- `log_kernel` : log-kernel of y given the optimal parameters.
- `spec` : Model specification of class MSGARCH\_SPEC created with [create.spec](#).

- `is.init` : Indicating if estimation was made with `do.init` option.
- `y` : Vector (of size T) of observations.

The `MSGARCH_MLE_FIT` contains these methods:

- `AIC` : Compute Akaike information criterion (AIC).
- `BIC` : Compute Bayesian information criterion (BIC).
- `ht` : Conditional volatility in each regime.
- `kernel` : Kernel method.
- `unc.vol` : Unconditional volatility in each regime.
- `pred` : Predictive method.
- `pit` : Probability Integral Transform.
- `risk` : Value-at-Risk And Expected-Shortfall methods.
- `simahead` : Step ahead simulation method.
- `sim` : Simulation method.
- `pdf` : Probability density function.
- `cdf` : Cumulative distribution function.
- `Pstate` : State probabilities filtering method.
- `summary` : Summary of the fit.

## References

- Ardia, D. Boudt, K. Carl, P. Mullen, K. M. & Peterson, B. G. (2011). Differential Evolution with DEoptim. *R Journal*, 3, pp. 27-34
- Ardia, D. Mullen, K. M. Peterson, B. G. & Ulrich, J. (2015). DEoptim: Differential Evolution in R. <https://cran.r-project.org/package=DEoptim>
- Mullen, K. M. Ardia, D. Gil, D. L. Windover, D. Cline, J. (2011) DEoptim: An R Package for Global Optimization by Differential Evolution. *Journal of Statistical Software*, 40, pp. 1-26, DOI: <http://dx.doi.org/10.18637/jss.v040.i06>
- Johnson, S. G. (2014). The NLOpt Nonlinear-Optimization. <https://cran.r-project.org/package=nloptr>.
- Varadhan, R., Borchers H. W. (2016). dfoptim: Derivative-Free Optimization. <https://cran.r-project.org/package=dfoptim>.

## Examples

```
require("MSGARCH")
# load data
data("sp500")
sp500 = sp500[1:1000]

# create model specification
spec = MSGARCH::create.spec()

# fit the model on the data with ML estimation using DEoptim initialization
set.seed(123)
fit = MSGARCH::fit.mle(spec = spec, y = sp500)
summary(fit)
```

---

ht	<i>Conditional variance in each regime.</i>
----	---

---

## Description

Method returning the conditional variance of each regime.

## Usage

```
ht(object, theta, y)
```

## Arguments

object	Model specification of class MSGARCH_SPEC created with <a href="#">create.spec</a> or fit object of type MSGARCH_MLE_FIT created with <a href="#">fit.mle</a> or MSGARCH_BAY_FIT created with <a href="#">fit.bayes</a> .
theta	Vector (of size d) or matrix (of size M x d) of parameter estimates (not required when using a fit object) where d must have the same length as the default parameters of the specification.
y	Vector (of size T) of observations (not required when using a fit object).

## Details

If a matrix of parameter estimates is given, each parameter estimate (each row) is evaluated individually.

## Value

Conditional variance (array of size (T + 1) x M x K) in each regime.

## Examples

```
require("MSGARCH")
# load data
data("sp500")
sp500 = sp500[1:1000]

# create model specification
spec = MSGARCH::create.spec()

# fit the model on the data with ML estimation
set.seed(123)
fit = MSGARCH::fit.mle(spec = spec, y = sp500)

# Compute the conditional variance
ht = MSGARCH::ht(object = fit)

plot(ht)
```



---

kernel	<i>Kernel function.</i>
--------	-------------------------

---

## Description

Method returning the kernel value of a vector of observations given a model specification.

## Usage

```
kernel(object, theta, y, log = TRUE)
```

## Arguments

object	Model specification of class MSGARCH_SPEC created with <a href="#">create.spec</a> or fit object of type MSGARCH_MLE_FIT created with <a href="#">fit.mle</a> or MSGARCH_BAY_FIT created with <a href="#">fit.bayes</a> .
theta	Vector (of size d) or matrix (of size M x d) of parameter estimates (not required when using a fit object) where d must have the same length as the default parameters of the specification.
y	Vector (of size T) of observations (not require when using a fit object).
log	Boolean indicating if the log kernel is returned. (Default: log = TRUE)

## Details

If a matrix of parameter estimates is given, each parameter estimate (each row) is evaluated individually. The kernel is a combination of the prior and the likelihood function. The kernel is equal to  $\text{prior}(\theta) + L(y|\theta)$  where L is the likelihood of y given the parameter  $\theta$ . When doing optimization, the goal is to minimize the negative log-kernel.

- Details on the prior  
The prior is different for each specification. It ensures that the  $\theta$  makes the conditional variance process stationary, positive, and that it respects that the sums of the probabilities in the case of a multiple-regime model are all equal to 1. If any of these three conditions is not respected the prior returns  $-1e10$ , meaning that the optimizer or the sampler will know that  $\theta$  is not a good candidate.

## Value

(Log-)Kernel value (scalar or vector of size M) of the vector of observations.

## References

Hamilton, J. D. (1989) A New Approach to the Economic Analysis of Nonstationary Time Series and the Business Cycle. *Econometrica*, 57, pp.357-38

## Examples

```
require("MSGARCH")
# load data
data("sp500")
sp500 = sp500[1:1000]

# create model specification
spec = MSGARCH::create.spec()

# fit the model on the data with ML estimation
set.seed(123)
fit = MSGARCH::fit.mle(spec = spec, y = sp500, ctr = list(do.init = FALSE))

# compute the kernel
kernel = MSGARCH::kernel(fit, log = TRUE)
```

---

pdf	<i>Probability density function.</i>
-----	--------------------------------------

---

## Description

Method returning the probability density in-sample or of a vector of points consider as one step ahead draws ( $t = T + 1$ ).

## Usage

```
pdf(object, x, theta, y, log = FALSE, do.its = FALSE)
```

## Arguments

object	Model specification of class MSGARCH_SPEC created with <a href="#">create.spec</a> or fit object of type MSGARCH_MLE_FIT created with <a href="#">fit.mle</a> or MSGARCH_BAY_FIT created with <a href="#">fit.bayes</a> .
x	Vector (of size N) of points evaluated at $t = T + 1$ (used when <code>do.its = FALSE</code> ).
theta	Vector (of size d) or matrix (of size M x d) of parameter estimates (not required when using a fit object) where d must have the same length as the default parameters of the specification.
y	Vector (of size T) of observations (not required when using a fit object).
log	Boolean indicating if the log-density is returned. (Default: <code>log = FALSE</code> )
do.its	Boolean indicating if the in-sample pdf is returned. (Default: <code>do.its = FALSE</code> )

## Details

If a matrix of parameter estimates is given, each parameter estimate (each row) is evaluated individually. If `do.its = FALSE`, the points x are evaluated as  $t = T + 1$  realization and the method uses the variance estimate at  $t = T + 1$ . If `do.its = TRUE`, y is evaluated using their respective variance estimate at each time t.

**Value**

A list of class MSGARCH\_PDF containing three components:

- pdf:  
If `do.its = FALSE`: (Log-)Probability density of the points  $x$  at  $t = T + 1$  (vector of size  $N$  or matrix of size  $M \times N$ )  
If `do.its = TRUE`: In-sample (Log-)Probability density of  $y$  (vector of size  $T$  or matrix of size  $M \times T$ ).
- $x$ :  
If `do.its = FALSE`: Vector (of size  $N$ ) of points evaluated at  $t = T + 1$ .  
If `do.its = TRUE`: Vector (of size  $T$ ) of observations.
- `do.its`: Original user inputted `do.its` for reference.

The class MSGARCH\_PDF contains the `plot` method only if `do.its = FALSE`.

**Examples**

```
require("MSGARCH")
# load data
data("sp500")
sp500 = sp500[1:1000]

# create model specification
spec = MSGARCH::create.spec()

# fit the model on the data with ML estimation using DEoptim initialization
set.seed(123)
fit = MSGARCH::fit.mle(spec = spec, y = sp500, ctr = list(do.init = FALSE))

# run pdf method in-sample
pdf.its = MSGARCH::pdf(object = fit, log = FALSE, do.its = TRUE)

sum(pdf.its$pdf, na.rm = TRUE)
# create mesh

x = seq(-3, 3, 0.01)

# run pdf method on mesh at  $T + 1$ 
pdf = MSGARCH::pdf(object = fit, x = x, log = FALSE, do.its = FALSE)

plot(pdf)
```

---

pit

*Probability Integral Transform.*

---

**Description**

Method returning the predictive probability integral transform (PIT) in-sample or of a vector of points consider as one step ahead draws ( $t = T + 1$ ).

**Usage**

```
pit(object, x, theta, y, do.norm = FALSE, do.its = FALSE)
```

## Arguments

object	Model specification of class MSGARCH_SPEC created with <code>create.spec</code> or fit object of type MSGARCH_MLE_FIT created with <code>fit.mle</code> or MSGARCH_BAY_FIT created with <code>fit.bayes</code> .
x	Vector (of size N) of pointsevaluated at $t = T + 1$ (used when <code>do.its = FALSE</code> ).
theta	Vector (of size d) or matrix (of size $M \times d$ ) of parameter estimates (not required when using a fit object) where d must have the same length as the default parameters of the specification.
y	Vector (of size T) of observations (not required when using a fit object).
do.norm	Boolean indicating if the PIT values are transformed into standard Normal variate. (Default: <code>do.norm = FALSE</code> ).
do.its	Boolean indicating if the in-sample pit is returned. (Default: <code>do.its = FALSE</code> )

## Details

If a matrix of MCMC posterior draws estimates is given, the Bayesian probability integral transform is calculated. If `do.its = FALSE`, the points x are evaluated as  $t = T + 1$  realizations and the method uses the variance estimate at  $t = T + 1$ . If `do.its = TRUE`, y is evaluated using their respective variance estimate at each time t. The `do.norm` argument transforms the PIT value into Normal variates so that normality test can be done.

## Value

A list of class MSGARCH\_PIT containing three components:

- `pit`:  
If `do.its = FALSE`: probability integral transform of the points x at  $t = T + 1$  or Normal variate derived from the probability integral transform of x (vector of size N).  
If `do.its = TRUE`: In-sample probability integral transform or Normal variate derived from the probability integral transform of y (vector of size T or matrix of size  $M \times T$ ).
- `x`:  
If `do.its = FALSE`: Vector (of size N) of at points evaluated at  $t = T + 1$ .  
If `do.its = TRUE`: Vector (of size T) of observations.
- `do.its`: Orinigal user inputed `do.its` for reference.

The class MSGARCH\_PIT contains the `plot` method only if `do.its = FALSE`.

## Examples

```
require("MSGARCH")
# load data
data("sp500")
sp500 = sp500[1:1000]

# create model specification
spec = MSGARCH::create.spec()

# fit the model on the data with ML estimation
set.seed(123)
fit = MSGARCH::fit.mle(spec = spec, y = sp500, ctr = list(do.init = FALSE))

# run pit method in-sample
```

```

pit.its = MSGARCH::pit(object = fit, do.norm = FALSE, do.its = TRUE)

plot(pit.its)

# generate random draws at T + 1 from model
set.seed(123)
sim.ahead = MSGARCH::simahead(object = fit, n = 1, m = 100)

x = sim.ahead$draws

# run pit method on random draws at T + 1 from model
pit = MSGARCH::pit(object = fit, x = x, do.norm = FALSE)

plot(pit)

```

---

pred

---

*Predictive function.*


---

## Description

Method returning the predictive probability density in-sample or of a vector of points consider as one step ahead draws ( $t = T + 1$ ).

## Usage

```
pred(object, x, theta, y, log = FALSE, do.its = FALSE)
```

## Arguments

object	Model specification of class MSGARCH_SPEC created with <a href="#">create.spec</a> or fit object of type MSGARCH_MLE_FIT created with <a href="#">fit.mle</a> or MSGARCH_BAY_FIT created with <a href="#">fit.bayes</a> .
x	Vector (of size N) of points evaluated at $t = T + 1$ (used when <code>do.its = FALSE</code> ).
theta	Vector (of size d) or matrix (of size M x d) of parameter estimates (not required when using a fit object) where d must have the same length as the default parameters of the specification.
y	Vector (of size T) of observations (not required when using a fit object).
log	Boolean indicating if the log-density is returned. (Default: <code>log = FALSE</code> )
do.its	Boolean indicating if the in-sample predictive is returned. (Default: <code>do.its = FALSE</code> )

## Details

If a matrix of MCMC posterior draws estimates is given, the Bayesian Probability integral transform is calculated. If `do.its = FALSE`, the points x are evaluated as  $t = T + 1$  realizations and the method uses the variance estimate at  $t = T + 1$ . If `do.its = TRUE`, y is evaluated using their respective variance estimate at each time t.

**Value**

A list of class MSGARCH\_PRED containing three components:

- `pred`:  
If `do.its = FALSE`: (Log-)Predictive of the points  $x$  at  $t = T + 1$  (vector of size  $N$ ).  
If `do.its = TRUE`: In-sample Predictive of  $y$  (vector of size  $T$  or matrix of size  $M \times T$ ).
- `x`:  
If `do.its = FALSE`: Vector (of size  $N$ ) of points evaluated at  $t = T + 1$ .  
If `do.its = TRUE`: Vector (of size  $T$ ) of observations.
- `do.its`: Original user inputted `do.its` for reference.

The class MSGARCH\_PRED contains the `plot` method only if `do.its = FALSE`.

**Examples**

```
require("MSGARCH")
# load data
data("sp500")
sp500 = sp500[1:1000]

# create model specification
spec = MSGARCH::create.spec()

# fit the model on the data with ML estimation using DEoptim initialization
set.seed(123)
fit = MSGARCH::fit.mle(spec = spec, y = sp500, ctr = list(do.init = FALSE))

# run pred method in-sample
pred.its = MSGARCH::pred(object = fit, log = TRUE, do.its = TRUE)

sum(pred.its$pred, na.rm = TRUE)

# create mesh
x = seq(-3,3,0.01)

# run pred method on mesh at T + 1
pred = MSGARCH::pred(object = fit, x = x, log = TRUE, do.its = FALSE)

plot(pred)
```

---

Pstate

*Filtered state probabilities.*


---

**Description**

Method returning the filtered probabilities of the states.

**Usage**

```
Pstate(object, theta, y)
```

### Arguments

object	Model specification of class MSGARCH_SPEC created with <code>create.spec</code> or fit object of type MSGARCH_MLE_FIT created with <code>fit.mle</code> or MSGARCH_BAY_FIT created with <code>fit.bayes</code> .
theta	Vector (of size d) or matrix (of size M x d) of parameter estimates (not required when using a fit object) where d must have the same length as the default parameters of the specification.
y	Vector (of size T) of observations (not required when using a fit object).

### Details

If a matrix of parameter estimates is given, each parameter estimate (each row) is evaluated individually.

### Value

Filtered state probabilities of class MSGARCH\_PSTATE (array of size  $(T + 1) \times M \times K$ ). The class MSGARCH\_PSTATE contains the plot method.

### Examples

```
require("MSGARCH")
# load data
data("sp500")
sp500 = sp500[1:1000]

# create model specification
spec = MSGARCH::create.spec()

# fit the model on the data with ML estimation
set.seed(123)
fit = MSGARCH::fit.mle(spec = spec, y = sp500, ctr = list(do.init = FALSE))

# compute the filtered state probabilities
Pstate = MSGARCH::Pstate(object = fit)

plot(Pstate)
```

---

risk

*Value-at-Risk And Expected-shortfall.*

---

### Description

Method returning the Value-at-Risk and Expected-shortfall in-sample or at  $t = T + 1$  based on the predictive density.

### Usage

```
risk(object, theta, y, level = c(0.95, 0.99), ES = TRUE, do.its = FALSE,
      ctr = list(n.mesh = 500, tol = 1e-04, itermax = 5))
```

## Arguments

object	Model specification of class MSGARCH_SPEC created with <code>create.spec</code> or fit object of type MSGARCH_MLE_FIT created with <code>fit.mle</code> or MSGARCH_BAY_FIT created with <code>fit.bayes</code> .
theta	Vector (of size d) or matrix (of size M x d) of parameter estimates (not required when using a fit object) where d must have the same length as the default parameters of the specification.
y	Vector (of size T) of observations (not required when using a fit object).
level	Vector (of size R) of Value-at-risk and Expected-shortfall levels. (Default: <code>level = c(0.95, 0.99)</code> )
ES	Boolean indicating if Expected-shortfall is also calculated. (Default: <code>ES = TRUE</code> )
do.its	Boolean indicating if the in-sample risk estimators are returned. (Default: <code>do.its = FALSE</code> )
ctr	List of control parameters for risk evaluation.

## Details

If a matrix of MCMC posterior draws estimates is given, the Bayesian Value-at-Risk and Expected-shortfall are calculated. If `do.its = FALSE`, the risk estimator at  $t = T + 1$ , the method uses the variance estimated at  $t = T + 1$ . If `do.its = TRUE`, the in-sample risk estimator are calculated.

## Value

A list of class MSGARCH\_RISK containing two or three components:

- `VaR` :  
If `do.its = FALSE`: Value-at-Risk at  $t = T + 1$  at the chosen levels (vector of size R).  
If `do.its = TRUE`: In-sample Value-at-Risk at the chosen levels (Matrix of size T x R).
- `ES` :  
If `do.its = FALSE`: Expected-shortfall at  $t = T + 1$  at the chosen levels (vector of size R).  
If `do.its = TRUE`: In-sample Expected-shortfall at the chosen levels (Matrix of size T x R).
- `y` : Vector (of size T) of observations.

The MSGARCH\_RISK contains the plot method. The Bayesian risk estimator can take long time to calculate depending on the size of the MCMC chain.

## Examples

```
require("MSGARCH")
# load data
data("sp500")
sp500 = sp500[1:1000]

# create model specification
spec = MSGARCH::create.spec()

# fit the model on the data with ML estimation
set.seed(123)
fit = MSGARCH::fit.mle(spec = spec, y = sp500, ctr = list(do.init = FALSE))

# compute the Value-at-Risk and Expected-shortfall
# Risk estimation in-sample
risk.its = MSGARCH::risk(object = fit, level = 0.95, ES = FALSE, do.its = TRUE)
```



```

plot(risk.its)

# Risk estimation at T + 1
risk = MSGARCH::risk(object = fit, level = 0.95, ES = FALSE, do.its = FALSE)

```

---

sim	<i>Process simulation method.</i>
-----	-----------------------------------

---

## Description

Method simulating a MSGARCH process.

## Usage

```
sim(object, n, m, theta, burnin = 500)
```

## Arguments

object	Model specification of class MSGARCH_SPEC created with <a href="#">create.spec</a> .
n	Simulation length. (Default: n = 1000)
m	Number of simulations. (Default: m = 1)
theta	Vector (of size d) or matrix (of size M x d) of parameter estimates (not required when using a fit object) where d must have the same length as the default parameters of the specification.
burnin	Burnin period discarded (first simulation draws). (Default: burnin = 500)

## Details

If a matrix of parameters estimates is given, each parameter estimates is evaluated individually and  $m = M$ . The difference between [sim](#) and [simahead](#) is that [sim](#) starts the simulation at  $t = 0$  creating an entire new process while [simahead](#) starts the simulation at  $t = T + 1$  taking in consideration all the information available in the original time series  $y$ .

## Value

A list of class MSGARCH\_SIM containing two components.

- draws: Matrix (of size M x n) of simulated draws.
- state: Matrix (of size M x n) of simulated states.

The MSGARCH\_SIM class contains the `plot` method.

## Examples

```

## Not run:
require("MSGARCH")
# create model specification
spec = MSGARCH::create.spec()

# generate process
set.seed(123)

```

```

sim = MSGARCH::sim(object = spec, n = 1000, m = 1, theta = spec$theta0, burnin = 500)

plot(sim)

## End(Not run)

```

simahead

*Step ahead simulation method.*

## Description

Method returning step ahead simulation up to time  $n$ .

## Usage

```
simahead(object, n, m, theta, y)
```

## Arguments

object	Model specification of class MSGARCH_SPEC created with <a href="#">create.spec</a> or fit object of type MSGARCH_MLE_FIT created with <a href="#">fit.mle</a> or MSGARCH_BAY_FIT created with <a href="#">fit.bayes</a> .
n	Number of steps ahead time steps. (Default: $n = 1$ )
m	Number of simulations. (Default: $m = 1$ )
theta	Vector (of size $d$ ) or matrix (of size $M \times d$ ) of parameter estimates (not required when using a fit object) where $d$ must have the same length as the default parameters of the specification.
y	Vector (of size $T$ ) of observations (not required when using a fit object).

## Details

If a matrix of parameters estimates is given, each parameter estimates is evaluated individually and  $m = M$ . The MSGARCH\_SIM class contains the `plot` method. The difference between [sim](#) and [simahead](#) is that [sim](#) starts the simulation at  $t = 0$  creating an entire new process while [simahead](#) starts the simulation at  $t = T + 1$  taking in consideration all the information available in the original time series  $y$ .

## Value

A list of class MSGARCH\_SIM containing two components:

- `draws`: Matrix (of size  $m \times n$ ) of step ahead simulated draws.
- `state`: Matrix (of size  $m \times n$ ) of step ahead simulated states.

The MSGARCH\_SIM class contains the `plot` method.

**Examples**

```

require("MSGARCH")
# load data
data("sp500")
sp500 = sp500[1:1000]

# create model specification
spec = MSGARCH::create.spec()

# fit the model on the data with ML estimation
set.seed(123)
fit = MSGARCH::fit.mle(spec = spec, y = sp500, ctr = list(do.init = FALSE))

# generate random draws
set.seed(123)
simahead = MSGARCH::simahead(object = fit, n = 30, m = 100)

plot(simahead)

```

sp500

*Log return of the S&P 500 index closing Value***Description**

The S&P 500 index closing value log return from 1998-01-01 to 2015-12-31 from Yahoo Finance <https://finance.yahoo.com/>.

**Usage**

```
data("sp500")
```

**Format**

Matrix containing 4,529 observations.

**Source**

Yahoo Finance <https://finance.yahoo.com/>

transmat

*Transition Matrix.***Description**

Method returning the transition matrix.

**Usage**

```
transmat(object, theta, n)
```

**Arguments**

object	Model specification of class MSGARCH_SPEC created with <a href="#">create.spec</a> or fit object of type MSGARCH_MLE_FIT created with <a href="#">fit.mle</a> .
theta	Vector (of size d) or matrix (of size M x d) of parameter estimates (not required when using a fit object) where d must have the same length as the default parameters of the specification.
n	Number of steps ahead. (Default: n = 1)

**Value**

A matrix (of size K x K) in the case of a Markov-Switching model or a vector (of size K) in the case of a Mixture model. The columns indicates the starting states while the rows indicates the transition states.

**Examples**

```
require("MSGARCH")
# load data
data("sp500")
sp500 = sp500[1:1000]

# create model specification
spec = MSGARCH::create.spec()

# fit the model on the data with ML estimation
set.seed(123)
fit = MSGARCH::fit.mle(spec = spec, y = sp500, ctr = list(do.init = FALSE))

# Extract the transition matrix 10 steps ahead
transmat.mle = MSGARCH::transmat(fit, n = 10)

print(transmat.mle)
```

unc.vol

*Unconditional volatility of each regime.***Description**

Method returning the unconditional volatility of the process in each state.

**Usage**

```
unc.vol(object, theta)
```

**Arguments**

object	Model specification of class MSGARCH_SPEC created with <a href="#">create.spec</a> or fit object of type MSGARCH_MLE_FIT created with <a href="#">fit.mle</a> or MSGARCH_BAY_FIT created with <a href="#">fit.bayes</a> .
theta	Vector (of size d) or matrix (of size M x d) of parameter estimates (not required when using a fit object) where d must have the same length as the default parameters of the specification.

**Details**

If a matrix of parameter estimates (each row) is given, each parameter estimates is evaluated individually.

**Value**

Unconditional volatility (vector of size K or matrix of size M x K) of each regime.

**Examples**

```
require("MSGARCH")
# create model specification
spec = MSGARCH::create.spec()

# compute the unconditional volatility in each regime
unc.vol = MSGARCH::unc.vol(object = spec, theta = spec$theta0)
```

# Index

## \*Topic **datasets**

AMZN, [4](#)

sp500, [26](#)

AIC, [3](#), [12](#), [14](#)

AMZN, [4](#)

BIC, [4](#), [12](#), [14](#)

cdf, [5](#), [8](#), [12](#), [14](#)

create.spec, [2](#), [5](#), [6](#), [9](#), [11–13](#), [15–17](#), [19](#), [20](#),  
[22–25](#), [27](#)

crps, [9](#)

DIC, [10](#), [12](#)

fit.bayes, [3–5](#), [8–10](#), [11](#), [15–17](#), [19](#), [20](#), [22](#),  
[23](#), [25](#), [27](#)

fit.mle, [3–5](#), [8](#), [9](#), [13](#), [15–17](#), [19](#), [20](#), [22](#), [23](#),  
[25](#), [27](#)

ht, [8](#), [12](#), [14](#), [15](#)

kernel, [2](#), [8](#), [12](#), [14](#), [16](#)

MSGARCH (MSGARCH-package), [2](#)

MSGARCH-package, [2](#)

pdf, [8](#), [12](#), [14](#), [17](#)

pit, [8](#), [12](#), [14](#), [18](#)

pred, [8](#), [12](#), [14](#), [20](#)

Pstate, [8](#), [12](#), [14](#), [21](#)

risk, [8](#), [12](#), [14](#), [22](#)

sim, [8](#), [12](#), [14](#), [24](#), [24](#), [25](#)

simahead, [8](#), [12](#), [14](#), [24](#), [25](#), [25](#)

sp500, [26](#)

transmat, [26](#)

unc.vol, [8](#), [12](#), [14](#), [27](#)