Estimation and Simulation of a COGARCH(p,q) model in the YUIMA project

Stefano, M. Iacus[*1], Lorenzo Mercuri [†2], and Edit Rroji [‡3]

[1]*Department of Economics, Management and Quantitative Methods, University of Milan, CREST Japan Science and Technology Agency,*
[2]*Department of Economics Management and Quantitative Methods, University of Milan, CREST Japan Science and Technology Agency,*
[3]*Department of Statistics and Quantitative Methods, University of Milano-Bicocca,*

May 18, 2015

Abstract

In this paper we show how to simulate and estimate a COGARCH(p,q) model in the R package yuima. Several routines for simulation and estimation are available. Indeed for the generation of a COGARCH(p,q) trajectory, the user can choose between two alternative schemes. The first is based on the Euler discretization of the stochastic differential equations that identifies a COGARCH(p,q) model while the second one considers the explicit solution of the variance process.

Estimation is based on the matching of the empirical with the theoretical autocorrelation function. In this case three different approaches are implemented: minimization of the mean square error, minimization of the absolute mean error and the generalized method of moments where the weighting matrix is continuously updated.

Numerical examples are given in order to explain methods and classes used in the yuima package.

Contents

[*]Electronic address: stefano.iacus@unimi.it
[†]Electronic address: lorenzo.mercuri@unimi.it
[‡]Electronic address: e.rroji@unimib.it

1 Introduction

The Continuous-Time GARCH(1,1) process has been introduced in [18] as a continuous counterpart of the discrete-time GARCH(1,1) model proposed by [3].

The idea is to develop in continuous time a model that is able to capture some stylized facts observed in financial time series [10] exploiting only one source of randomness for returns and for variance dynamics. Indeed, in the Continuous-Time GARCH (COGARCH hereafter), the stochastic differential equation for variance is driven by the discrete part of the quadratic variation of the same Lévy process used for modeling returns. The continuous nature of the COGARCH makes it particularly appealing for discribing the behaviour of high frequency data [see [13] for an application of method of moments using intraday returns].

The generalization to higher order COGARCH(p,q) processes has been proposed in [5, 9]. Starting from the observation that the variance of a GARCH(p,q) is an ARMA(q, p-1), the Variance is modeled with a CARMA(q,p-1) process [see [7, 27, 6] and many others] driven by the discrete part of the quadratic variation of the Lévy process in the returns. Although this representation is different from the one used by [18] for the COGARCH(1,1) process, this last can be again retrieved as a special case.

Many authors recently have investigated the COGARCH(1,1) model from a theoretical and an empirical point of view [see [22, 16, 23, 2] and many others]. Some R codes for estimation and simulation of a COGARCH(1,1) driven by a Compound Poisson and Variance Gamma are available in [20]. For the general COGARCH(p,q), the main contribution remain the seminal works [5] and [9]. The aim of this paper is to describe the simulation and the estimation schemes in the `yuima` package [26] for a COGARCH(p,q) model driven by a general Lévy process. Based on our knowledge `yuima` is the first R package available on CRAN that allows the user to manage a higher order COGARCH(p,q) model. Moreover, the estimation algorithm gives the option to recover the increments of the underlying noise process and estimates the Lévy measure parameters. We recall that a similar procedure is available in `yuima` also for the CARMA model [**?**, see]for a complete discussion]IacusMercur2015. The `yuima` package is developed within the `YUIMA` project [8] whose aim is to provide an environment for simulation and estimation of stochastic differential equations.

The outline of the paper is as following. In Sect. 2 we discuss the main properties of the COGARCH(p,q) process. In particular we review the condition for existence of a strictly stationary variance process, its higher moments and the behaviour of the autocorrelation of the square increments of the COGARCH(p,q) model. In Sect. 3 we analyze two different simulation schemes. The first is based on the Euler discretization while the second one uses the solution of the state process in the CARMA(q,p-1) model. Sect. 4 is devoted to the estimation algorithm. In Sect. 5 we show the main classes and corresponding methods in `yuima` package and in the Sect. 6 we present some numerical examples about the simulation and the estimation of a COGARCH(p,q) model

2

2 COGARCH Models driven by a Lévy process

In this section we review the mathematical definition of a COGARCH(p,q) process and its properties. In particular we focus on the conditions for the existence of a strictly stationary COGARCH(p,q) process and compute the first four unconditional moments. The existence of higher order moments plays a central role for the computation of the autocorrelation function of the squared increments of the COGARCH(p,q) model and consequently the estimation procedure implemented in the `yuima` package.

The COGARCH(p,q) process, introduced in [5] as a generalization of the COGARCH(1,1) model, is defined through the following system of stochastic differential equations:

$$\begin{cases} \mathrm{d}G_t = \sqrt{V_t}\mathrm{d}L_t \\ V_t = a_0 + \mathbf{a}^\intercal Y_{t-} \\ \mathrm{d}Y_t = AY_{t-}\mathrm{d}t + \mathbf{e}\left(a_0 + \mathbf{a}^\intercal Y_{t-}\right)\mathrm{d}\left[L,L\right]_t^d \end{cases} \tag{1}$$

where q and p are integers such that $q \geq p \geq 1$. The state space process Y_t is a vector with q components:

$$Y_t = [Y_{1,t}, \ldots, Y_{q,t}]^\intercal.$$

The vector $\mathbf{a} \in \mathcal{R}^q$ is defined as:

$$\mathbf{a} = [a_1, \ldots, a_p, a_{p+1}, \ldots, a_q]^\intercal$$

with $a_{p+1} = \cdots = a_q = 0$. The companion $q \times q$ matrix A is

$$A = \begin{bmatrix} 0 & 1 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & 1 \\ -b_q & -b_{q-1} & \ldots & -b_1 \end{bmatrix}.$$

The vector $\mathbf{e} \in \mathcal{R}^q$ contains zero entries except the last component that is equal to one.

$[L,L]_t^d$ is the discrete part of the quadratic variation of the underlying Lévy process L_t and is defined as:

$$[L,L]_t^d := \sum_{0 \leq s \leq t} \left(\Delta L_s\right)^2. \tag{2}$$

Remark 1 *A COGARCH(p,q) model is constructed starting from the observation that in the GARCH(p,q) process, its discrete counterpart, the dynamics of the variance is a predictable ARMA(q, p-1) process driven by the squares of the past increments. In the COGARCH(p,q) case, the ARMA process leaves the place to a CARMA(q,p-1) model [see [7] for details about the CARMA(p,q) driven by a Lévy process] and the infinitesimal increments of the COGA-RCH(p,q) are defined through the differential of the driven Lévy L_t as done in* (1).

As observed above the COGARCH(p,q) model generalizes the COGARCH(1,1) process that has been introduced following different arguments from those for the (p,q) case. However choosing $q = 1$ and $p = 1$ in (1) the COGARCH(1,1) process developed in [18, 13] can be retrieved through straightforward manipulations and, for obtaining the same parametrization in Proposition 3.2 of [18], the following equalities are necessary:

$$\omega_0 = a_0 b_1, \ \omega_1 = a_1 e^{-b_1} \text{ and } \eta = b_1.$$

Before introducing the conditions for strict stationarity and the existence of unconditional higher moments, it is worth noting that the state space process Y_t can be seen as a Multivariate Stochastic Recurrence Equation and the corresponding theory can be applied to the

COGARCH(p,q) process [see [5, 9] more details] in order to derive its main features[1]. In the case of the Compound Poisson driven noise, the representation through the stochastic difference equations is direct in the sense that the random coefficients of the state process Y_t can be written explicitly while in the general case, it is always possible to identify a sequence of Compound Poisson processes that as limit to the choosen driven Lévy process.

In the following, we require that matrix A can be diagonalized, that means:

$$A = SDS^{-1}$$

with

$$S = \begin{bmatrix} 1 & \dots & 1 \\ \lambda_1 & \dots & \lambda_q \\ \vdots & & \vdots \\ \lambda_1^{q-1} & \dots & \lambda_q^{q-1} \end{bmatrix}, \ D = \begin{bmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_q \end{bmatrix} \tag{3}$$

where the $\lambda_1, \lambda_q, \dots, \lambda_q$ are the eigenvalues of matrix A and are ordered as follows:

$$\Re\{\lambda_1\} \geq \Re\{\lambda_2\} \geq \dots \geq \Re\{\lambda_q\}.$$

Applying the theory of stochastic recurrence equations, [5] provide A sufficient condition for the strict stationarity of a COGARCH(p,q) model. We review the result for the stationarity of the state process Y_t. Two fundamental assumptions are the fact that the eigenvalues $\lambda_1, \dots, \lambda_q$ are distinct and the underlying process L must have a non-trivial $\nu_L(l)$ measure. Then, the process Y_t converges in distribution to the random variable Y_∞ if exist some $r \in [1, +\infty]$ such that:

$$\int_{-\infty}^{+\infty} \ln\left(1 + \|S^{-1}\mathbf{ea}^\mathsf{T}S\|_r l^2\right) d\nu_L(l) \leq \Re\{\lambda_1\} \tag{4}$$

for some matrix S such that the matrix A is diagonalizable. If we choose as a starting condition $Y_0 \overset{d}{=} Y_\infty$ than the process Y_t is strictly stationary and consequently the variance V_t is also a strictly stationary process. Unlikely for the general case, the inequality in (4) gives only a sufficient condition about the strict stationarity, and it is difficult to verify in practice[2]. As shown in [18] and remarked in [5], the condition in (4) is also necessary for the COGARCH(1,1) case and can be simplified as:

$$\int_{-\infty}^{+\infty} \ln\left(1 + a_1 l^2\right) d\nu_L(l) \leq b_1. \tag{5}$$

Using again the SRE theory, it is also possible to determine the condition for existence of higher moments of the state process Y_t. In this way it is possible to determine the autocorrelations of the squared COGARCH increments that are used in the estimation procedure illustrated in Section 4. As reported in [5], the κ-th order moment of the process Y_t exists finite if, for some $r \geq 1$ and S such that the matrix A is diagonalizable, the following conditions hold:

$$E\left(L_1^{2\kappa}\right) < +\infty, \ \int_{-\infty}^{+\infty} \left[\left(1 + \|S^{-1}\mathbf{ea}^\mathsf{T}S\|_r l^2\right)^\kappa - 1\right] d\nu_L(l) < \Re\{\lambda_1\}\kappa. \tag{6}$$

---

[1]The Stochastic Recurrence Equations theory [4, 17] has been also used to prove the strictly and weakly stationarity for the GARCH(p,q) model [1]

[2]In the `yuima` package a diagnostic for condition (4) is available choosing matrix S as done in (3) and $r = 2$. We remark that the process Y_t is stationary if the diagnostic gives a positive answer otherwise we can conclude nothing about the stationarity of the process.

As special case of (6), the unconditional stationary mean $E\left(Y_\infty\right) = -a_0\mu\left(A + \mu\mathbf{e}\mathbf{a}^\mathsf{T}\right)^{-1}\mathbf{e}$ of the vector process Y_t exists if

$$E\left(L_1^2\right) < +\infty, \ \|S^{-1}\mathbf{e}\mathbf{a}^\mathsf{T}S\|_r\mu < \Re\left\{\lambda_1\right\}$$

where

$$\mu := \int_{-\infty}^{+\infty} l^2 \mathrm{d}\nu_L\left(l\right) \tag{7}$$

is the second moment of the Lévy measure $\nu_L\left(l\right)$. It is worth noting that the condition (2) ensures also the strict stationarity since, using $\ln(1+x) \leq x$, we have the following sequences of inequalities:

$$\int_{-\infty}^{+\infty} \ln\left(1 + \|S^{-1}\mathbf{e}\mathbf{a}^\mathsf{T}S\|_r l^2\right) \mathrm{d}\nu_L\left(l\right) \leq \|S^{-1}\mathbf{e}\mathbf{a}^\mathsf{T}S\|_r\mu \leq \Re\left\{\lambda_1\right\}.$$

For the stationary covariance matrix [9]

$$cov\left(Y_\infty\right) = \frac{a_0^2 b_q^2 \rho}{\left(b_q - \mu a_1\right)^2 \left(1 - m\right)} \int_0^\infty e^{\tilde{A}t}\mathbf{e}\mathbf{e}^\mathsf{T}e^{\tilde{A}^\mathsf{T}t}\mathrm{d}t, \tag{8}$$

the existence of the second moment of Y_∞ in (6) becomes:

$$E\left(L_1^4\right) < \infty, \ \|S^{-1}\mathbf{e}\mathbf{a}^\mathsf{T}S\|_r\rho < 2\left(-\Re\left\{\lambda_1\right\} - \|S^{-1}\mathbf{e}\mathbf{a}^\mathsf{T}S\|_r\mu\right)$$

where

$$\rho := \int_{-\infty}^{+\infty} l^4 \mathrm{d}\nu_L\left(l\right)$$

is the fourth moment of the Lévy measure $\nu_L\left(l\right)$ and

$$m := \int_0^{+\infty} \mathbf{a}^\mathsf{T}e^{\tilde{A}t}\mathbf{e}\mathbf{e}^\mathsf{T}e^{\tilde{A}^\mathsf{T}t}\mathbf{a}\mathrm{d}t.$$

Before introducing the higher moments and the autocorrelations, we recall the conditions for the nonegativity for a strictly stationary variance process in (1). Indeed, under the assumption that all the eigenvalues of matrix A are distinct and the relation in (4) holds, the variance process $V_t \geq a_0 > 0$ a.s. if:

$$\mathbf{a}^\mathsf{T}e^{At}\mathbf{e} \geq 0, \ \forall t \geq 0. \tag{9}$$

The condition in (9) is costly since we need to check it each time. Nevertheless some useful controls are available [28].

1. A necessary and sufficient condition to guarantee that $a^\mathsf{T}e^{At}\mathbf{e} \geq 0$ in the COGARCH(2,2) case is that the eigenvalues of A are real and $a_2 \geq 0$ and $a_1 \geq -a_2\lambda\left(A\right)$ where $\lambda\left(A\right)$ is the biggest eigenvalue.

2. Under condition $2 \leq p \leq q$, that all eigenvalues of A are negative and ordered in an increasing way $\lambda_1 \geq \lambda_2 \geq, \ldots, \geq \lambda_{p-1}$ and γ_j are the roots of $a\left(z\right) = 0$ ordered as $0 > \gamma_1 \geq \gamma_2 \geq \ldots \geq \gamma_{p-1}$. Then sufficient condition for (9) is

$$\sum_{i=1}^k \gamma_i \leq \sum_{i=1}^k \lambda_i \ \ \forall k \in \left\{1, \ldots, p-1\right\}.$$

3. For a COGARCH(1,q) model a sufficient condition that ensures (9) is that all eigenvalues must be real and negative.

Combining the requirement in (4) with that in (9) it is possible to derive the higher moments and the autocorrelations for a COGARCH(p,q) model. As a first step, we define the returns of a COGARCH(p,q) process on the interval $(t, t + r]$, $\forall t \geq 0$ as:

$$G_t^{(r)} := \int_t^{t+r} \sqrt{V_s} \mathrm{d}L_s. \tag{10}$$

Let L_t be a symmetric and centered Lévy process such that the fourth moment of the associated Lévy measure is finite, we define the matrix $\tilde{A}$ as:

$$\tilde{A} := A + \mu \mathbf{e}\mathbf{a}^\mathsf{T}.$$

It is worth noting that the structure of $\tilde{A}$ is the same of A except for the last row q where $\tilde{A}_q, = (-b_q + \mu a_1, \ldots, -b_1 + \mu a_q)$. For any $t \geq 0$ and for any $h \geq r > 0$, the first two moments of (10) are

$$E\left[\left(G_t^{(r)}\right)\right] = 0 \tag{11}$$

and

$$E\left[\left(G_t^{(r)}\right)^2\right] = \frac{\alpha_0 b_q r}{b_q - \mu a_1} E\left[L_1^2\right]. \tag{12}$$

The computation of the autocovariances and variance for squared returns (10) require the following quantities defined as:

$$\begin{aligned} P_0 &= 2\mu^2 \left[3\tilde{A}^{-1}\left(\tilde{A}^{-1}\left(e^{\tilde{A}r}I\right) - rI\right) - I\right] cov\left(\epsilon_\infty\right) \\ P_h &= \mu^2 e^{\tilde{A}h}\tilde{A}^{-1}\left(I - e^{\tilde{A}r}\right)\tilde{A}^{-1}\left(e^{\tilde{A}r} - I\right) cov\left(\epsilon_\infty\right) \end{aligned}, \tag{13}$$

$$\begin{aligned} Q_0 &= 6\mu \left[\left(rI - \tilde{A}^{-1}\left(e^{\tilde{A}r} - I\right)\right) cov\left(\epsilon_\infty\right) - \tilde{A}^{-1}\left(\tilde{A}^{-1}\left(e^{\tilde{A}r} - I\right) - rI\right) cov\left(\epsilon_\infty\right)\tilde{A}^\mathsf{T}\right]\mathbf{e} \\ Q_h &= \mu e^{\tilde{A}h}\tilde{A}^{-1}\left(I - e^{-\tilde{A}r}\right)\left[\left(I - e^{\tilde{A}r}\right) - \tilde{A}^{-1}\left(e^{\tilde{A}r} - I\right) cov\left(\epsilon_\infty\right)\tilde{A}^\mathsf{T}\right]\mathbf{e} \end{aligned} \tag{14}$$

and

$$R = 2r^2\mu^2 + \rho. \tag{15}$$

The terms P_h and P_0 are $q \times q$ matrices. Q_h and Q_0 are $q \times 1$ vectors and the term R is a scalar. The $q \times q$ matrix $cov\left(\epsilon_\infty\right)$ is defined as:

$$cov\left(\epsilon_\infty\right) = \rho \int_0^\infty e^{\tilde{A}t}\mathbf{e}\mathbf{e}^\mathsf{T}e^{\tilde{A}^\mathsf{T}t}\mathrm{d}t \tag{16}$$

where μ, ρ and m have been defined before.

Using the $q \times q$ matrix P_h and the $q \times 1$ vector Q_h the autocovariances of the squared returns (10) are defined as:

$$\gamma_r\left(h\right) := cov\left[(G_t^{(r)})^2, (G_{t+h}^{(r)})^2\right] = \frac{a_0^2 b_q^2\left(\mathbf{a}^\mathsf{T}P_h\mathbf{a} + \mathbf{a}Q_h\right)}{\left(1 - m\right)\left(b_q - \mu a_1\right)^2}, \quad h \geq r \tag{17}$$

while the variance of the process $\left(G_t^{(r)}\right)^2$ is

$$\gamma_r\left(0\right) := var\left(\left(G_t^{(r)}\right)^2\right) = \frac{a_0^2 b_q^2\left(\mathbf{a}^\mathsf{T}P_0\mathbf{a} + \mathbf{a}Q_0 + R\right)}{\left(1 - m\right)\left(b_q - \mu a_1\right)^2}. \tag{18}$$

Combining the autocovariances in (17) with (12) and (18) we obtain the autocorrelations:

$$\rho_r\left(h\right) := \frac{\gamma_r\left(h\right)}{\gamma_r\left(0\right)} = \frac{\left(\mathbf{a}^\mathsf{T} P_h \mathbf{a} + \mathbf{a} Q_h\right)}{\left(\mathbf{a}^\mathsf{T} P_0 \mathbf{a} + \mathbf{a} Q_0 + R\right)}. \tag{19}$$

We conclude this Section with the following remark. As done for the GARCH(p,q) model the autocovariances of the returns $G_t^{(r)}$ are all zeros.

3 Simulation of a COGARCH(P,Q) model

In this Section we illustrate the theory behind the simulation routines available in the `yuima` package when a COGARCH(p,q) model is considered. The corresponding sample paths are generated according two different schemes.

The first method, in the `Yuima` project [8], is based on the Euler-Maruyama [14] discretization of the system in (1). In this case the algorithm follows these steps:

1. We determine an equally spaced grid from 0 to T composed by N intervals with the same length Δt

$$0, \ldots, (n-1) \times \Delta t, n \times \Delta t, \ldots, N \times \Delta t.$$

2. We generate the trajectory of the underlying Lévy process L_t sampled on the grid defined in step 1.

3. Choosing a starting point for the state process $Y_0 = y_0$ and $G_0 = 0$, we have

$$Y_n = \left(I + A\Delta t\right) Y_{n-1} + \mathbf{e}\left(a_0 + \mathbf{a}^\mathsf{T} Y_{n-1}\right) \Delta\left[LL\right]_n^d. \tag{20}$$

The discrete quadratic variation of the L_t process is approximated by

$$\Delta\left[LL\right]_n^d = \left(L_n - L_{n-1}\right)^2. \tag{21}$$

4. Once the approximated state process Y_n is obtained we can generate the trajectory of Variance and the process G_n according the following equations:

$$V_n = a_0 + \mathbf{a}^\mathsf{T} Y_{n-1} \tag{22}$$

and

$$G_n = G_{n-1} + \sqrt{V_n}\left(L_n - L_{n-1}\right). \tag{23}$$

It is worth noting that, although the discretized version of the state process Y_n in (20) can be seen as a stochastic recurrence equation, the conditions for stationarity and non-negativity for the variance V_n process are not the same of ones analyzed in the Section 2. In particular it is possible to determine an example where the discretized variance process V_n assumes negative values while the true process is non negative with probability one.

In order to clarify deeper this issue we consider a COGARCH(1,1) model driven by a Variance Gamma Lévy process [see [21, 19] for more details about the VG model]. In this case, the condition for the non-negativity of the Variance in (9) is ensured if $a_0 > 0$ and $a_1 > 0$ while the strict stationarity condition in (5) for the COGARCH(1,1) is $E[L^2] = 1$ and $a_1 - b_1 < 0$. The last two requirements guarantee also the existence of the stationary unconditional mean for the process V_t. We define the model using the `yuima` function `setCogarch`. Its usage is completely explained in the Section 5. The following command line instructs `yuima` to build the COGARCH(1,1) model with VG noise:

```
> model1 <- setCogarch(p = 1, q = 1, work = FALSE,
+         measure=list("rngamma(z, 1, sqrt(2), 0, 0)"), measure.type = "code",
+         Cogarch.var = "G", V.var = "v", Latent.var="x", XinExpr=TRUE)
```
Choosing the following values for the parameters
```
> param1 <- list(a1 = 0.038, b1 = 301, a0 =0.01, x01 = 0)
```
the COGARCH(1,1) is stationary and the variance is strictly positive. Nevertheless, if we simulate the trajectory using the Euler discretization, the value of Δt can lead to negative values for the process as shown in the example:
```
> Terminal1=5
> n1=750
> samp1 <- setSampling(Terminal=Terminal1, n=n1)
> set.seed(123)
> sim1 <- simulate(model1, sampling = samp1, true.parameter = param1,
+         method="euler")
> plot(sim1, main="Sample Path of a VG COGARCH(1,1) model with Euler scheme")
```


Sample Path of a VG COGARCH(1,1) model with Euler scheme
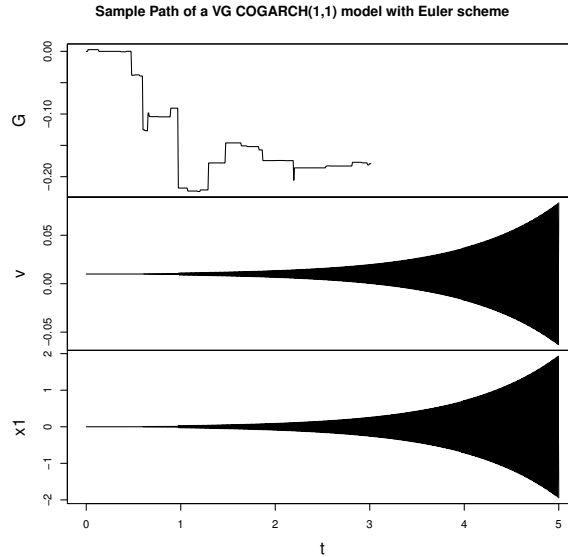
Looking to the Figure, we observe a divergent and oscillatory behaviour for the simulated state Y_n and the variance V_n processes while, from theoretical point of view, the conditions for nonnegativity and stationarity for the variance of a COGARCH(1,1) model are satisfied by the values used for the parameters.
To overcome this problem, we provide an alternative simulation scheme based on the solution of the process Y_n given the starting point Y_{n-1}.

Applying the Ito's Lemma for semimartingales [25] to the transformation $e^{-At}Y_t$, we have:

$$e^{-A\Delta t}Y_t = Y_{t-\Delta t} - \int_{t-\Delta t}^{t} Ae^{-Au}Y_{u-}\mathrm{d}u + \int_{t-\Delta t}^{t} e^{-Au}\mathrm{d}Y_u + \sum_{s\le t}\left[e^{-As}\left(Y_s - Y_{s-}\right) - e^{-As}\left(Y_s - Y_{s-}\right)\right].$$

We substitute the definition of Y_t in (1) and get

$$e^{-At}Y_t = Y_{t-\Delta t} - \int_{t-\Delta t}^{t} Ae^{-Au}Y_{u-}\mathrm{d}u + \int_{t-\Delta t}^{t} e^{-Au}AY_{u-}\mathrm{d}u + \int_{t-\Delta t}^{t} e^{-Au}e\left(a_0 + a'Y_{u-}\right)\mathrm{d}\left[LL\right]_u^d$$

8

Using the following property for an exponential matrix

$$
\begin{aligned}
Ae^{At} &= A\left(I + At + \frac{1}{2}A^2t^2 + \frac{1}{3!}A^3t^3 + ...\right)\\
&= \left(A + tA^2 + \frac{1}{2}t^2A^3 + t^3\frac{1}{3!}A^4 + ...\right)\\
&= \left(I + tA + \frac{1}{2}t^2A^2 + t^3\frac{1}{3!}A^3 + ...\right)A = e^{At}A,
\end{aligned}
$$

we get

$$
Y_t = e^{At}Y_{t-\Delta t} + \int_{t-\Delta t}^{t} e^{A(t-u)}e\left(a_0 + a'Y_{u-}\right)\mathrm{d}\left[LL\right]_u^d \tag{24}
$$

Except for the case where the noise is a Compound Poisson, the simulation scheme follows the same steps of the Euler-Maruyama discretization where the state space process Y_n on the sample grid is generated according the approximation of the relation in (24):

$$
Y_n = e^{A\Delta t}Y_{n-1} + e^{A(\Delta t)}e\left(a_0 + a'Y_{n-1}\right)\left(\left[LL\right]_n^d - \left[LL\right]_{n-1}^d\right) \tag{25}
$$

or equivalently:

$$
Y_n = a_0 e^{A(\Delta t)}e\Delta\left[LL\right]_n^d + e^{A\Delta t}\left(I + ea'\Delta\left[LL\right]_n^d\right)Y_{n-1}. \tag{26}
$$

where $\Delta\left[LL\right]_n^{(d)} := \left[LL\right]_n^d - \left[LL\right]_{n-1}^d$ is the increment of the discrete part of quadratic variation. In the previous example, the sample path is simulated according the recursion in (26) choosing `method = mixed` in the `simulate` function as done below:

```
> set.seed(123)
> sim2 <- simulate(model1, sampling = samp1, true.parameter = param1,
+         method="mixed")
> plot(sim2, main="Sample Path of a VG COGARCH(1,1) model with mixed scheme")
```

In the case of the COGARCH(p,q) driven by a Compound Poisson Lévy process, a trajectory can be simulated without any approximation of the solution in (24). Indeed it is possible to determine the time where the jumps occur[3] and then evaluate the corresponding quadratic variation in an exact way. Once the trajectory of a random time is obtained, the piecewise constant interpolation is used on the fixed grid, in order to mantain the càdlàg property of the sample path.

4 Estimation of a COGARCH(P,Q) model in the yuima package

In this Section we explain the estimation procedure that we propose in the yuima package for the COGARCH(p,q) model. As done for the CARMA(p,q) model driven by Lévy process [15], even in this case a three step estimation procedure is proposed that allows the user to obtain estimated values for the COGARCH(p,q) and the parameters of the Lévy measure.
This procedure is structured as follows:

---

[3]In a general Compound Poisson the jump time follow an exponential r.v. with rate λ

Sample Path of a VG COGARCH(1,1) model with mixed scheme
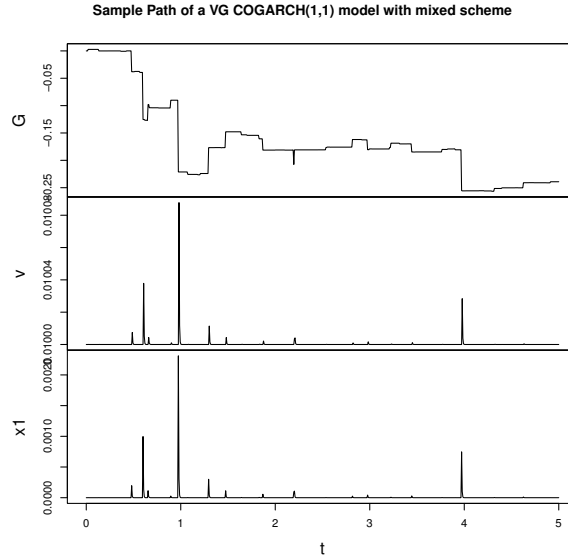
1. Using the moment matching procedure explained below, we estimate the COGARCH(p,q) parameters $\mathbf{a} := [a_1, \ldots, a_p]$, $\mathbf{b} := [b_1, \ldots, b_q]$ and the constant term a_0 in the Variance process V_t. In this phase, the estimation is obtained by minimizing some distances between the empirical and theoretical autocorrelation function.

2. Once the COGARCH(p,q) parameters are available, we recover the increments of the underlying Lévy process using the methodology describe below.

3. In the third step, we use the increments obtained in the second step and estimate the Lévy measure parameters by means of maximum likelihood estimation procedure.

Let $G_0, G_1, \ldots, G_n, \ldots, G_T$ be the observed values of the process subsampled at equally spaced instants $0, h, 2h, \ldots, Nh$ where the length of each interval is $h = \frac{T}{N}$. The time of the last observation is T and N is the number of observations of the process G_n.

In the following we assume that the underlying Lévy process is symmetric and centered in zero. Starting from the sample $\{G_n\}_{n=0}^N$ we define the COGARCH(p,q) increments of lag one as:

$$G_n^{(1)} := G_n - G_{n-1}$$

and the increments of lag r as:

$$G_n^{(r)} := G_n - G_{n-r} \tag{27}$$

where $r \geq 1$ is a natural number and for $r = 1$ the definition in (4) coincides with (27).

It is worth mentioning that the increments $G_n^{(r)}$ can be obtained as a time aggregation of increments of lag one as follows:

$$G_n^{(r)} = \sum_{h=n-r}^{n} G_n^{(1)} \tag{28}$$

The time aggregation in (28) can be useful during the optimization routine when the values of increments $G^{(1)}$ are very small in absolute value.

10

Using the sample $\left\{G_n^{(r)}\right\}_{n \geq r}$ we compute the empirical second moment

$$\hat{\mu}_r := \frac{1}{N-d-r+1} \sum_{n=r}^{N-d} \left(G_n^{(r)}\right)^2$$

and empirical autocovariance function $\hat{\gamma}(h)$ is defined as:

$$\hat{\gamma}_r(h) := \frac{1}{N-d-r+1} \sum_{n=r}^{N-d} \left(\left(G_{n+h}^{(r)}\right)^2 - \hat{\mu}_r\right)\left((G_n^r)^2 - \hat{\mu}_r\right) \quad h=0,1,\ldots,d$$

where d is the maximum lag considered.
The empirical autocorrelations are:

$$\hat{\rho}_r(h) = \frac{\hat{\gamma}_r(h)}{\hat{\gamma}_r(0)}. \tag{29}$$

We use the theoretical and empirical autocorrelations in order to define the moment conditions for the estimation procedure. By the introduction of the $q+p$ vector $\theta := (\mathbf{a}, \mathbf{b})$ we define the vector function $g(\theta) : \mathbb{R}^{q+p} \rightarrow \mathbb{R}^d$ as follows:

$$g(\theta) := E\left[f\left(G_n^{(r)}, \theta\right)\right] \tag{30}$$

where $f\left(G_n^{(r)}, \mathbf{a}, \mathbf{b}\right)$ is a d dimensional real function where the components are defined as:

$$f_h\left(\Delta G_n^{(r)}, \theta\right) = \rho_r(h) - \frac{\left(\left(G_{n+h}^{(r)}\right)^2 - \mu_r\right)\left(\left(G_n^{(r)}\right)^2 - \mu_r\right)}{\gamma_r(0)}, \quad h=1,\ldots,d. \tag{31}$$

In the estimation algorithm, we replace the expectation in (30) with the sample counterpart. The components of vector $\hat{g}(\theta) = [\hat{g}_1(\theta),\ldots,\hat{g}_d(\theta)]^\intercal$ are defined as:

$$
\begin{aligned}
\hat{g}_h(\theta) &= \frac{1}{N-d-r+1} \sum_{n=r}^{N-d} \left[\rho_r(h) - \frac{\left(\left(G_{n+h}^{(r)}\right)^2 - \hat{\mu}_r\right)\left(\left(G_n^{(r)}\right)^2 - \hat{\mu}_r\right)}{\hat{\gamma}_r(0)}\right] \\
&= \rho_r(h) - \hat{\rho}_r(h), \quad h=1,\ldots,d.
\end{aligned}
\tag{32}
$$

The vector $\theta = (\mathbf{a}, \mathbf{b})$ containing the COGARCH(p,q) parameters are obtained by minimizing some distances between empirical and theoretical autocorrelations. The optimization problem is:

$$\min_{\theta \in \mathbb{R}^{q+p}} d(\rho_r, \hat{\rho}_r)$$

where the components of vectors ρ_r and $\hat{\rho}_r$ are the theoretical and empirical autocorrelations defined in (19) and (29) respectively. Function $d(x,y)$ measures the distance between vectors x and y. In the yuima environment, three distances are available and listed below:

1. the L_1 norm

$$\|\hat{g}(\theta)\|_1 = \sum_{h=1}^d |\hat{g}_h(\theta)|. \tag{33}$$

11

2. the squared of L_2 norm

$$\|\hat{g}(\theta)\|_2^2 = \sum_{h=1}^{d} (\hat{g}_h(\theta))^2. \tag{34}$$

3. The quadratic form

$$\|\hat{g}(\theta)\|_{\mathbf{W}}^2 = \hat{g}(\theta)^\intercal \mathbf{W} \hat{g}(\theta) \tag{35}$$

where the positive definite weighting matrix $\mathbf{W}$ is choosen to obtain efficient estimators between those that belong to the class of asymptotically normal estimators.

It is worth noting that the objective function $\|\hat{g}(\theta)\|_2^2$ is a special case of the function $\|\hat{g}(\theta)\|_{\mathbf{W}}^2$ where the weighting matrix $\mathbf{W}$ coincides with the identity matrix. Both distances are related with the Generalized Method of Moments (GMM) introduced by [11]. Under some regularity conditions [24], the GMM estimators are consistent and, for any general positive definite matrix $\mathbf{W}$, their asymptotic variance-covariance matrix $\mathbf{V}$ are:

$$\mathbf{V} = \frac{1}{N - d - r + 1} (\mathbf{D}^\intercal \mathbf{W} \mathbf{D})^{-1} \mathbf{D}^\intercal \mathbf{W} \mathbf{S} \mathbf{W} \mathbf{D} (\mathbf{D}^\intercal \mathbf{W} \mathbf{D})^{-1},$$

The matrix $\mathbf{D}$ is defined as:

$$\mathbf{D} = E\left[\frac{\partial f\left(G_n^{(r)}, \theta\right)}{\partial \theta^\intercal}\right]. \tag{36}$$

While

$$\mathbf{S} = E\left[f\left(G_n^{(r)}, \theta\right) f\left(G_n^{(r)}, \theta\right)^\intercal\right]. \tag{37}$$

For the squared L_2 norm in (34) matrix $\mathbf{V}$ becomes:

$$\mathbf{V} = \frac{1}{N - d - r + 1} (\mathbf{D}^\intercal \mathbf{D})^{-1} \mathbf{D}^\intercal \mathbf{S} \mathbf{D} (\mathbf{D}^\intercal \mathbf{D})^{-1} \tag{38}$$

while for (35), as observed above, the choice of the matrix $\mathbf{W}$ is done in order to obtain efficient estimators in the class of all asymptotically normal estimators. To obtain this result we prefer to use the Continuously Updated GMM estimator [12]. In this case the matrix $\mathbf{W}$ is determinated simultaneously with the estimation of the vector parameters θ. Introducing the function $\|\hat{g}(\theta)\|_{\hat{\mathbf{W}}}^2$ as the sample counterpart of the quadratic form in (35), the minimization problem becomes:

$$\min_{\theta \in \mathbb{R}^{q+p}} \|g(\theta)\|_{\hat{\mathbf{W}}}^2 = \hat{g}(\theta)^\intercal \hat{\mathbf{W}}(\theta) \hat{g}(\theta)$$

where function $\hat{\mathbf{W}}(\theta)$ maps from $\mathcal{R}^{p+q}$ to $\mathcal{R}^{d \times d}$ and is defined as:

$$\hat{\mathbf{W}}(\theta) = \left(\frac{1}{N - r - d + 1} \sum_{n=r}^{N-r-d} f\left(G_n^{(r)}, \theta\right) f\left(G_n^{(r)}, \theta\right)^\intercal\right)^{-1}. \tag{39}$$

Observe that $\hat{\mathbf{W}}(\theta)$ is a consistent estimator of matrix $\mathbf{S}^{-1}$ that means:

$$\hat{\mathbf{W}}(\theta) \underset{N \to +\infty}{\overset{P}{\to}} \mathbf{S}^{-1} \tag{40}$$

consequently the asymptotic variance-covariance matrix $\mathbf{V}$ in (38) becomes:

$$\mathbf{V} = \frac{1}{N - r - d + 1} (\mathbf{D}^\intercal \mathbf{S}^{-1} \mathbf{D})^{-1} \tag{41}$$

Once the estimates of vector θ are obtained, the user is allowed to retrieve the increments of the underlying Lévy process according the following procedure. This stage is independent

on the nature of the Lévy measure but it is only based on the solution of the state process Y_t and on the approximation of the quadratic variation with the squared increments of the Lévy driven process.

Starting from the discrete time equally spaced observations $G_1, G_2, \ldots, G_{N\Delta t}$, we remark that the increment $\Delta G_t := G_t - G_{t-}$ can be approximated using the observations $\{G_{n\Delta t}\}_{n=0}^N$ as follows:

$$\Delta G_t \approx \Delta G_{n\Delta t} = G_{n\Delta t} - G_{(n-1)\Delta t} \tag{42}$$

Recalling that $G_t = \sum_{0 \le s \le t} \sqrt{V_s} \left(\Delta L_s\right)$, the approximation in (42) becomes:

$$\Delta G_t \approx \sqrt{V_{n\Delta t}} \left(\Delta L_{n\Delta t}\right) \tag{43}$$

where $V_{n\Delta t}$ is the value of the variance process at the observation time $t = n\frac{T}{N}$ and $\Delta L_{n\Delta t} = L_{n\Delta t} - L_{(n-1)\Delta t}$ is the discretized Lévy increment from $t - \Delta t$ to t.
Using the discretization scheme introduced in (25) the process $Y_{n\delta t}$ is written as:

$$Y_{n\delta t} \approx e^{A\Delta t} Y_{(n-1)\delta t} + e^{A(\Delta t)} e \left(a_0 + a' Y_{t-\Delta t}\right) \left([LL]_n^d - [LL]_{(n-1)}^d\right) \tag{44}$$

since the difference $[LL]_n^d - [LL]_{(n-1)}^d \approx (\Delta L_{n\Delta t})^2$ and using the result in (43), the difference equation (44) can be approximated in terms of the squared increments of COGARCH(p,q) process and we have:

$$\begin{aligned} Y_t &\approx e^{A\Delta t} Y_{t-\Delta t} + e^{A(\Delta t)} e \left(a_0 + a' Y_{t-\Delta t}\right) (\Delta L_t)^2 \\ &= e^{A\Delta t} Y_{t-\Delta t} + e^{A(\Delta t)} e \left(\Delta G_t\right)^2. \end{aligned} \tag{45}$$

Choosing Y_0 equal to the unconditional mean of the process Y_t, we are able to retrieve its sample path according to the recursive equation in (45). The only quantities that we need to compute are the squared increments of the COGARCH(p,q) process on the grid $\{0, \Delta t, 2\Delta t, \ldots, n\Delta t \ldots, N\Delta t\}$. The estimated state process in (45) is also useful for getting the estimated trajectory of the variance process. Finally note the Lévy increment at a general time $t = n\Delta t$ is obtained as:

$$\Delta L_{n\delta t} = \frac{\Delta G_{n\delta t}}{\sqrt{V_{n\delta t}}}. \tag{46}$$

Once the increments of the underlying Lévy are obtained, it is possible to obtain the estimates for the Lévy measure parameters through the Maximum Likelihood Estimation procedure [we refere to the yuima documentation [8, 26] for the available Lévy processes].

5 Package R

In this Section we illustrate the main classes and methods in yuima that allow the user to deal with a COGARCH(p,q) model. The routines implemented are based on the results considered in Section 3 for simulation and Section 4 for the estimation procedures.
In particular we classify these classes and methods in three groups. The first group contains the classes and functions that allow the user to define a specific COGARCH(p,q) model in the yuima framework. The second group is used for the simulation of the sample paths for the COGARCH(p,q) model and the third is related to the possibility of estimation using simulated or real data.

5.1 Classes and Methods for the definition of a COGARCH(P,Q) model

The main object for a COGARCH(p,q) process in the `yuima` environment is an object of `yuima.cogarch-class` that contains all the information about a specific COGARCH(p,q) process. This class extends the `yuima.model-class` and it has only one additional slot, called `@info`, that contains an object of `cogarch.info-class`. We refer to the `yuima` documentation for a complete description of the slots that constitute the objects of class `yuima.model`. In this paper we focus only on the object of class `cogarch.info`. In particular its structure is composed by the slots listed below:

- `@p` is an integer that is the number of moving average coefficients in the Variance process V_t.

- `@q` is an integer number that corresponds to the dimension of autoregressive coefficients in the variance process V_t.

- `@ar.par` contains a string that is the label for the autoregressive coefficients.

- `@ma.par` is the Label for the moving average coefficients.

- `@loc.par` indicates the name of the location coefficient in the process V_t.

- `@Cogarch.var` string that contains the name of the observed process G_t.

- `@V.var` is the Label of the V_t process.

- `@Latent.var` indicates the label of the state process Y_t.

- `@XinExpr` is a logical variable. If the value is `FALSE`, default value, the starting condition for the state process Y_t is a zero vector. Otherwise the user has to fix the starting condition as argument `true.parameter` in the method `simulate`.

- `@measure` identifies the Lévy measure of the underlying noise and consequently the discrete part of the quadratic variation that drives the state process.

- `@measure.type` says if the Lévy measure belongs to the family of Compound Poisson or is another type of Lévy

The user builds an object of class `yuima.cogarch` through to the constructor `setCogarch`:
```
    setCogarch(p, q, ar.par = "b", ma.par = "a", loc.par = "a0", Cogarch.var = "g",
V.var = "v", Latent.var = "y", jump.variable = "z", time.variable = "t", measure
= NULL, measure.type = NULL, XinExpr = FALSE, startCogarch = 0, work = FALSE, ...)
```

The arguments used in a call of the function `setCogarch` are illustrated in the following list:

- `p`: A non negative integer that is the number of the moving average coefficients in the variance process.

- `q`: A non negative integer that indicates the number of the autoregressive coefficients in the variance process.

- `ar.par`: A character-string that is the label of the autoregressive coefficients.

- `ma.par`: A character-string that is the label of the autoregressive coefficients.

- `loc.par`: A string that indicates the label of the location coefficient in the variance process.

- `Cogarch.var`: A character-string that is the label of the observed COGARCH process.

- `V.var`: A character-string that is the label of the latent variance process.

- `Latent.var`: A character-string that is the label of the latent process in the state space representation for the variance process.

- `jump.variable`: Label of the underlying Lévy process .

- `time.variable`: Label of the time variable.

- `measure`: Lévy measure of the underlying Lévy process.

- `measure.type`: Label that indicates whether the underlying noise is a Compound Poisson process or another Lévy without the diffusion component.

- `XinExpr`: A logical variable that identifies the starting condition. In particular, the default value `XinExpr = FALSE` implies that the starting condition for the state process is zero. Alternatively `XinExpr = TRUE` means that the user is allowed to specify as parameters the starting values for each component of the state variable.

- `startCogarch`: Initial value for the COGARCH process.

- `...`: Arguments to be passed to `setCogarch` such as the slots of the `yuima.model-class`.

5.2 Classes and Methods for the simulation of a COGARCH(P,Q) model

The `simulate` is a method for an object of class `yuima.model`. It is also available for an object of class `yuima.cogarch`. The function requires the following inputs:

```
simulate(object, nsim=1, seed=NULL, xinit, true.parameter, space.discretized
= FALSE, increment.W = NULL, increment.L = NULL, method = "euler", hurst, methodfGn
= "WoodChan", sampling=sampling, subsampling=subsampling, ...)
```

In this work we focus on the argument `method` that identifies the type of discretization scheme for the time when the `object` belongs to the class of `yuima.cogarch`. The default value `euler` means that the simulation of a sample path is based on the Euler-Maruyama discretization of the stochastic differential equations. This approach is available for all objects of class `yuima.model`. For the COGARCH(p,q) an alternative simulation scheme is available choosing `method = mixed`. In this case the generation of trajectories is based on the solution (24) for the state process. In particular if the underlying noise is a Compound Poisson Lévy process, the trajectory is built using a two step algorithm. First the jump time is simulated internally using the Exponential distribution with parameter λ and then the size of jump is simulated using the random variable specified in the slot `yuima.cogarch@model@measure`. For the other Lévy processes, the simulation scheme is based on the discretization of the state process solution (25) in Section 5.

5.3 Classes and Methods for the estimation of a COGARCH(P,Q) model

The `cogarch.gmm` class is a class of the `yuima` package that contains estimated parameters obtained by the `gmm` function.

- `@model` is an object of of `yuima.cogarch-class`.

- `@objFun` is an object of class `character` that indicates the objective function used in the minimization problem. L2 refers to the squared of L_2 norm in (34), `L2CUE` for the quadratic form (35) and the `L1` for the L_1 norm in (33)

- `@call` is an object of class `language`.

- @coef is an object of class `numeric` that contains the estimated parameters.
- @fullcoef is an object of class `numeric` that contains the estimated and fixed parameters from the user.
- @vcov is an object of class `matrix`.
- @min is an object of class `numeric`.
- @minuslogl is an object of class `function`.
- @method is an object of class `character`.

The `cogarch.gmm.incr` is a class of the `yuima` package that extends the `cogarch.gmm-class` and is filled from the function `gmm`.

- `Incr.Lev` is an object of class `zoo` that contains the estimated increments of the noise obtained using `cogarchNoise`.
- `model` is an object of `yuima.cogarch-class`.
- `logL.Incr` is an object of class `numeric` that contains the value of the log-likelihood for the estimated Levy increments.
- `objFun` is an object of class `character` that indicates the objective function used in the minimization problem. The values are the same for the slot `@objFun` in an object of class `cogarch.gmm`.
- `call` is an object of class `language`.
- `coef` is an object of class `numeric` that contains the estimated parameters.
- `fullcoef` is an object of class `numeric` that contains estimated and fixed parameters.
- `vcov` is an object of class `matrix`.
- `min` is an object of class `numeric`.
- `minuslogl` is an object of class `function`.
- `method` is an object of class `character`.

The function `gmm` returns the estimated parameters of a COGARCH(p,q) model. The parameters are obtained by matching the theoretical with the empirical autocorrelation function.

    gmm(yuima, data = NULL, start, method="BFGS", fixed = list(), lower, upper, lag.max
= NULL, equally.spaced = TRUE, Est.Incr = "NoIncr", objFun = "L2")

- `yuima` is a `yuima` object or an object of `yuima.cogarch-class`
- `data` is an object of class `yuima.data-class` contains the observations available at uniformly spaced instants of time. If `data=NULL`, the default, the function uses the data in an object of `yuima-class`.
- `start` is a `list` that contains the starting values for the optimization routine.
- `method` is a string that indicates one of the methods available in the function `optim`.
- `fixed` a list of fixed parameters in the optimization routine.
- `lower` a named list for specifying lower bounds for parameters.
- `upper` a named list for specifying upper bounds for parameters.
- `lag.max` maximum lag for which we calculate the theoretical and empirical acf. Default is $\sqrt{N}$ where `N` is the number of observations.

- `equally.spaced` Logical variable. If `equally.spaced = TRUE`, in each unitary interval we have the some number of observations. If `equally.spaced = FALSE`, each unitary interval is composed by different number of observations.

- `Est.Incr` a string variable. If `Est.Incr = "NoIncr"`, default value, `gmm` returns an object of class `cogarch.gmm-class` that contains the COGARCH parameters. If `Est.Incr = "Incr"` or `Est.Incr = "IncrPar"` the output is an object of class `cogarch.gmm.incr-class`. In the first case the object contains the increments of the underlying noise while in the second case it contains also the estimated parameters of the Lévy measure.

- `objFun` a string variable that indentifies the objective function in the optimization step. `objFun = "L2"`, default value, the objective function is a quadratic form where the weighting matrix is the identity one. `objFun = "L2CUE"` the weighting matrix is estimated using Continuously Updating GMM (L2CUE). `objFun = "L1"`, the objective function is the mean absolute error. In the last case standard errors for estimators are not available.

Function `gmm` uses function `cogarchNoise` for the estimation of the underlying Lévy in a COGARCH(p,q) model. This function assumes that the underlying Lévy process is symmetric and centered in zero.

```
cogarchNoise(yuima.cogarch, data=NULL, param, mu=1)
```
The arguments of the `cogarchNoise` are listed below

- `yuima.cogarch` is an object of `yuima-class` or an object of `yuima.cogarch-class` that contains all the information about the COGARCH(p,q) model.

- `data` is an object of class `yuima.data-class` that contains the observations available at uniformly spaced instants of time. If `data=NULL`, the default, the `cogarchNoise` uses the data in an object of `yuima.data-class`.

- `param` is a `list` of parameters for the COGARCH(p,q) model.

- `mu` is a `numeric` object that contains the value of the second moments of the Lévy measure.

6 Numerical results

In this section we show how to use the `yuima` package for the simulation and the estimation of a COGARCH(p,q) model driven by different symmetric Lévy processes. As a first step we focus on a COGARCH(1,1) model driven by different Lévy processes available on the package. In particular we consider the cases in which the driven noise are a Compound Poisson with jump size normally distributed and a Variance Gamma process. In the last part of this section, we show also that the estimation procedure implemented seems to be adequate even for higher order COGARCH models. In particular we simulate and then estimate different kind of COGARCH(p,q) models driven by a Compound Poisson process where the distribution of the jump size is a normal.

6.1 Simulation and Estimation of a COGARCH(1,1)

The first example is a COGARCH(1,1) model driven by a Compound Poisson process. As a first step, we choose the set of the model parameters:

```
> numSeed <- 200
> param.cp <- list(a1 = 0.038, b1 =  0.053, a0 = 0.04/0.053,
+             lambda = 1, eta=0, sig2=1, x01 = 50.33)
```

a_1, b_1 and a_0 are the parameters of the state process Y_t. λ is the intensity of the Compound Poisson process while η and σ^2 are the mean and the variance of the jump size. $x_{0,1}$ is the starting point of the process X_t, the choosen value is the stationary mean of the state process and it is used in the simulation algorithm.

In the following command line we define the model using the `setCogarch` function.

```
> mod.cp <- setCogarch(p = 1, q = 1, work = FALSE,
+           measure=list(intensity="lambda",df=list("dnorm(z,eta,sig2)")),measure.type = "CP",
+           Cogarch.var = "g", V.var = "v", Latent.var="x",
+           XinExpr=TRUE)
```

We simulate a sample path of the model using the Euler discretization. We fix $\Delta t = \frac{1}{15}$ and the command lines below are used to instruct `yuima` for the choice of the simulation scheme:

```
> Term <- 1600
> num <- 24000
> set.seed(numSeed)
> samp.cp <- setSampling(Terminal=Term, n=num)
> sim.cp <- simulate(mod.cp, true.parameter=param.cp,
+           sampling=samp.cp, method="euler")
```

In the following figure we show the behaviour of the simulated trajectories for the COGA-RCH(1,1) model G_t, the variance V_t and the state space X_t:

```
> plot(sim.cp, main = "simulated COGARCH(1,1) model driven by a Compound Poisson process")
```
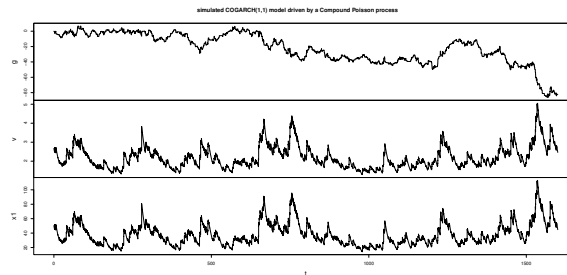

We use the two step algorithm developed in Section 4 for the estimation of the COG-ARCH(p,q) and the Lévy measure parameters. In the `yuima` function `gmm`, we fix `objFun = L2` meaning that the objective function used in the minimization is the mean squared error. Setting also `Est.Incr=IncrPar`, the function `gmm` returns the estimated increments of the underlying noise.

```
> res.cp <- gmm(sim.cp, start = param.cp, objFun = "L2", Est.Incr = "IncrPar")
```

The results can be displayed using the method `summary` and in the following figure we report the recovered increments of the underlying noice process.

```
> summary(res.cp)

Two Stages GMM estimation

Call:
gmm(yuima = sim.cp, start = param.cp, Est.Incr = "IncrPar", objFun = "L2")
```

```
Coefficients:
           Estimate Std. Error
b1     6.783324e-02 0.06862392
a1     3.403071e-02 0.02897625
a0     1.032014e+00         NA
lambda 1.073912e+00         NA
eta    6.818470e-09         NA
sig2   7.837838e-01         NA

 Log.objFun L2: -3.491179


Number of increments: 24000

Average of increments: -0.002114

Standard Dev. of increments: 0.256610


-2 log L of increments: 2851.529874

Summary statistics for increments:
     Min.    1st Qu.    Median       Mean    3rd Qu.       Max.
-2.840000   0.000000  0.000000  -0.002114   0.000000   3.686000

> plot(res.cp, main = "Compound Poisson Increment of a COGARCH(1,1) model")
```
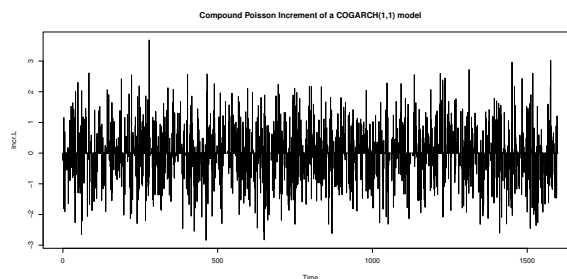

Compound Poisson Increment of a COGARCH(1,1) model

We are able also to generate the original process using the increments stored into the object `res.cp` using the `simulate` function.

```
> traj.cp<- simulate(res.cp)
> plot(traj.cp, main = "estimated COGARCH(1,1) driven by compound poisson process")
```

In the next example, we simulate and estimate a COGARCH(1,1) model driven by a Variance Gamma process. We set the values for the parameters and define the model using the following command lines:

```
> param.VG <- list(a1 = 0.038,  b1 =  0.053, a0 = 0.04/0.053,
+            lambda = 1, alpha = sqrt(2), beta = 0, mu = 0, x01 = 50.33)
> cog.VG <- setCogarch(p = 1, q = 1, work = FALSE,
+          measure=list("rngamma(z, lambda, alpha, beta, mu)"),
```

estimated COGARCH(1,1) driven by compound poisson process
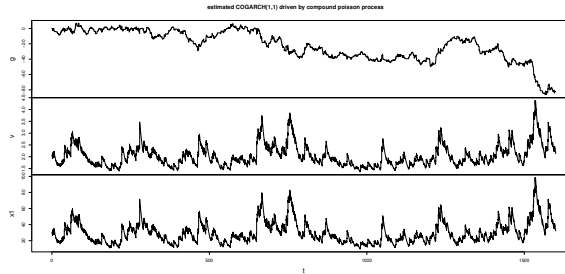
```
+               measure.type = "code", Cogarch.var = "y", V.var = "v",
+               Latent.var = "x", XinExpr = TRUE)
```

We obtain a trajectory for COGARCH(1,1) with Variance Gamma noise.

```
> set.seed(numSeed)
> samp.VG <- setSampling(Terminal = Term, n = num)
> sim.VG <- simulate(cog.VG, true.parameter = param.VG,
+               sampling = samp.VG, method = "euler")
> plot(sim.VG, main = "simulated COGARCH(1,1) model driven by a Variance Gamma process")
```


simulated COGARCH(1,1) model driven by a Variance Gamma process
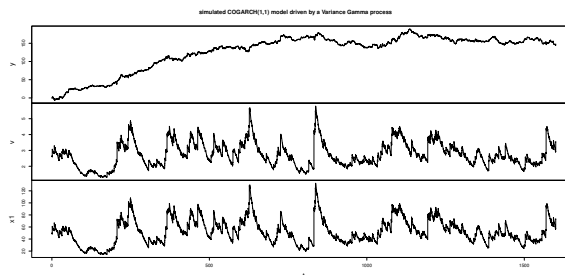
and then we estimate the model parameters:

```
> res.VG <- gmm(sim.VG, start = param.VG, Est.Incr = "IncrPar")
> summary(res.VG)

Two Stages GMM estimation

Call:
gmm(yuima = sim.VG, start = param.VG, Est.Incr = "IncrPar")

Coefficients:
          Estimate Std. Error
b1     0.051449188 0.04168365
a1     0.028791052 0.01810412
a0     1.248576654         NA
lambda 1.049274382 0.09432438
alpha  1.466220182 0.08769087
beta   0.051526860 0.03929050
mu     0.003357025 0.02935151
```

```
 Log.objFun L2: -3.755496


Number of increments: 24000

Average of increments: 0.003635

Standard Dev. of increments: 0.258127


-2 log L of increments: 4291.499302

Summary statistics for increments:
     Min.    1st Qu.    Median       Mean   3rd Qu.       Max.
-5.548000 -0.001729   0.000000   0.003635   0.002092   4.005000

> plot(res.VG, main = "Variance Gamma Increment of a COGARCH(2,1) model")
```
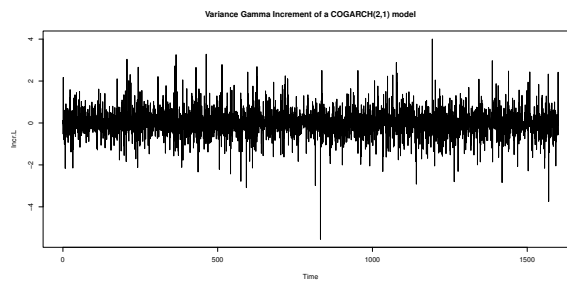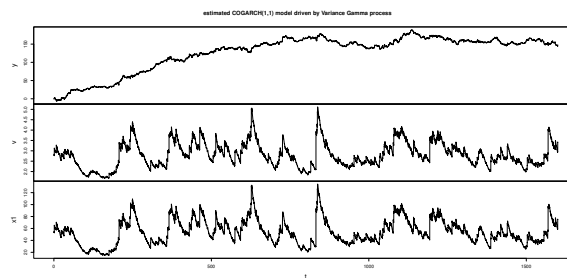

Even in this case we can obtain the COGARCH(1,1) trajectory using the estimated increments as follows:

```
> traj.VG <- simulate(res.VG)
> plot(traj.VG, main="estimated COGARCH(1,1) model driven by Variance Gamma process")
```

## 6.2 COGARCH(p,q) model driven by a Compound Poisson process

We conclude this Section by illustrating an example using COGARCH(p,q) process. In this way, we show the ability of the yuima package in managing in a complete way these models. For this reason in the following we consider a COGARCH(2,1) driven by Compound Poisson Processes where the jump size is normally distributed.

We define the COGARCH(2,1) model in the yuima using the command lines:

```
> param.cp2 <- list(a0 = 0.5, a1 = 0.1, b1 =1.5, b2 = 0.5,
+                lambda = 1, eta = 0, sig2 = 1, x01 = 2.5, x02 = 0)
> mod.cp2 <- setCogarch(p = 1, q = 2, work = FALSE,
+             measure = list(intensity = "lambda",df = list("dnorm(z,eta,sig2)")),
+             measure.type = "CP", Cogarch.var = "y", V.var = "v",
+             Latent.var = "x", XinExpr = TRUE)
```

We simulate a trajectory.

```
> samp.cp2 <- setSampling(Terminal = Term, n = num)
> set.seed(numSeed)
> sim.cp2 <- simulate(mod.cp2, sampling = samp.cp2,
+             true.parameter = param.cp2, method="euler")
> plot(sim.cp2, main = "simulated COGARCH(2,1) model driven by a Compound Poisson process")
```
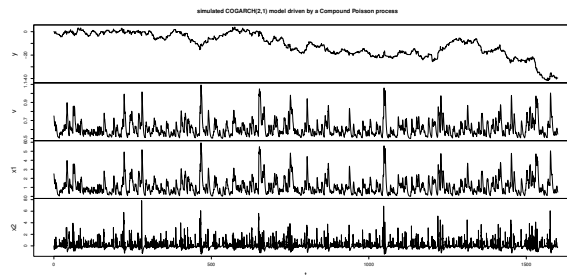

We estimate the model parameters and recover the underlying Lévy noise increments:

```
> res.cp2 <- gmm(yuima = sim.cp2, start = param.cp2, Est.Incr = "IncrPar")
> summary(res.cp2)

Two Stages GMM estimation

Call:
gmm(yuima = sim.cp2, start = param.cp2, Est.Incr = "IncrPar")

Coefficients:
          Estimate Std. Error
b2      0.0569630413 0.20054247
b1      0.9520642366 3.54500502
a1      0.0281299955 0.09775311
a0      0.2956658497         NA
lambda 1.0423762156         NA
```

```
eta    0.0002425553         NA
sig2   0.8154399532         NA

 Log.objFun L2: -3.323979


Number of increments: 24000

Average of increments: -0.001929

Standard Dev. of increments: 0.258830


-2 log L of increments: 2861.417140

Summary statistics for increments:
      Min.    1st Qu.    Median      Mean    3rd Qu.       Max.
-3.054000  0.000000  0.000000  -0.001929  0.000000  3.483000
> plot(res.cp2, main = "Compound Poisson Increment of a COGARCH(2,1) model")
```
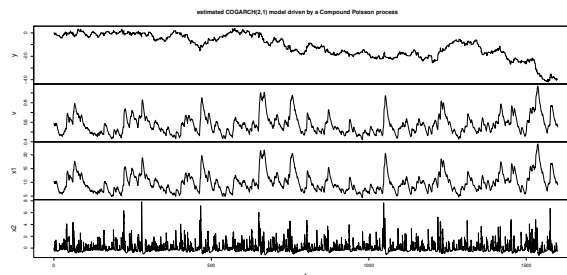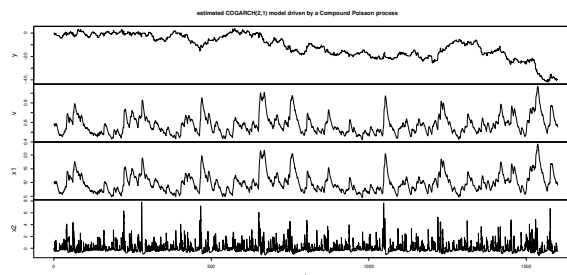

estimated COGARCH(2,1) model driven by a Compound Poisson process

The path of the COGARCH(2,1) driven by the estimated increments are reported below:

```
> traj.cp2 <- simulate(res.cp2)
> plot(traj.cp2, main = "estimated COGARCH(2,1) model driven by a Compound Poisson process")
```


estimated COGARCH(2,1) model driven by a Compound Poisson process

Acknowledgements

References

[1] B. Basrak, R. A. Davis, and T. Mikosch. A characterization of multivariate regular variation. *The Annals of Applied Probability*, 12(3):908–920, 2002.

[2] E. Bibbona and I. Negri. Higher moments and prediction-based estimation for the cogarch(1,1) model. *Scandinavian Journal of Statistics*, pages n/a–n/a, 2015.

[3] T. Bollerslev. Generalized autoregressive conditional heteroskedasticity. *Journal of Econometrics*, 31(3):307–327, April 1986.

[4] A. Brandt. The stochastic equation yn+1=anyn+bn with stationary coefficients. *Advances in Applied Probability*, 18(1):211–220, 1986.

[5] P. Brockwell, E. Chadraa, and A. Lindner. Continuous-time GARCH processes. *Annals of Applied Probability*, 16(2):790–826, 2006.

[6] P. J. Brockwell, R. A. Davis, and Y. Yang. Estimation for non-negative lévy-driven ornstein-uhlenbeck processes. *Journal of Applied Probability*, 44:987–989, 2007.

[7] P.J. Brockwell. Lévy-driven carma processes. *Annals of the Institute of Statistical Mathematics*, 53(1):113–124, 2001.

[8] A. Brouste, M. Fukasawa, H. Hino, S. M. Iacus, K. Kamatani, Y. Koike, H. Masuda, R. Nomura, T. Ogihara, Shimuzu Y., M. Uchida, and Yoshida N. The yuima project: A computational framework for simulation and inference of stochastic differential equations. *Journal of Statistical Software*, 57(4):1–51, 2014.

[9] E. Chadraa. *Statistical Modelling with COGARCH(P,Q) Processes.*, 2009. PhD Thesis.

[10] R. Cont. Empirical properties of asset returns: stylized facts and statistical issues. *Quantitative Finance*, 1:223–236, 2001.

[11] L. P. Hansen. Large sample properties of generalized method of moments estimators. *Econometrica*, 50(4):1029–1054, 1982.

[12] L. P. Hansen, J. Heaton, and A. Yaron. Finite-sample properties of some alternative gmm estimators. *Journal of Business & Economic Statistics*, 14(3):262–280, 1996.

[13] S. Haug, C. Klüppelberg, A. Lindner, and M. Zapp. Method of moment estimation in the cogarch(1,1) model. *Econometrics Journal*, 10(2):320–341, 2007.

[14] S. M. Iacus. *Simulation and Inference for Stochastic Differential Equations: With R Examples.* Springer, 2008.

[15] S. M. Iacus and L. Mercuri. Implementation of lvy carma model in yuima package. *Computational Statistics*, pages 1–31, 2015.

[16] J. Kallsen and B. Vesenmayer. {COGARCH} as a continuous-time limit of garch(1,1). *Stochastic Processes and their Applications*, 119(1):74 – 98, 2009.

[17] H. Kesten. Random difference equations and renewal theory for products of random matrices. *Acta Mathematica*, 131(1):207–248, 1973.

[18] C. Klüppelberg, A. Lindner, and R. Maller. A continuous-time garch process driven by a lévy process: Stationarity and second-order behaviour. *Journal of Applied Probability*, 41(3):601–622, 2004.

[19] A. Loregian, L. Mercuri, and E. Rroji. Approximation of the variance gamma model with a finite mixture of normals. *Statistics & Probability Letters*, 82(2):217 – 224, 2012.

[20] Granzer M. *Estimation of COGARCH Models with implementation in R.*, 2013. Master Thesis.

[21] D. B. Madan and E. Seneta. The variance gamma (v.g.) model for share market returns. *The Journal of Business*, 63(4):511–24, 1990.

[22] R. A. Maller, G. Müller, and A. Szimayer. Garch modelling in continuous time for irregularly spaced time series data. *Bernoulli*, 14(2):519–542, 05 2008.

[23] G. Müller. Mcmc estimation of the cogarch (1, 1) model. *Journal of Financial Econometrics*, 8(4):481–510, 2010.

[24] Whitney K Newey and Daniel McFadden. Large sample estimation and hypothesis testing. *Handbook of econometrics*, 4:2111–2245, 1994.

[25] P. Protter. *Stochastic integration and differential equations*. Springer, 1990.

[26] YUIMA Project Team. *yuima: The YUIMA Project package (stable version)*, 2013. R package version 1.0.2.

[27] H. Tomasson. Some computational aspects of gaussian carma modelling. *Statistics and Computing*, pages 1–13, 2013.

[28] H. Tsai and K. S. Chan. A note on non-negative continuous time processes. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(4):589–597, 2005.