Feasible Multivariate GARCH Models

Alexios Galanos

2026-11-02

Contents

1 Introduction **1**

2 Covariance Decomposition Models (`dcc` and `cgarch`) **2**
2.1 Constant Correlation . . . 2
2.2 Dynamic Conditional Correlation . . . 3
2.3 Estimation . . . 4
2.4 Standard Errors . . . 4
2.5 Distributions . . . 5
2.6 Generalized Dynamics . . . 6
2.7 Forecasting and Simulation . . . 7
2.8 Weighted Distribution . . . 7
2.9 News Impact Surface . . . 8
2.10 Critique . . . 8

3 Generalized Orthogonal GARCH Model (`gogarch`) **8**
3.1 Estimation . . . 9
3.2 Co-Moments . . . 10
3.3 Weighted Distribution . . . 12
3.4 Factor News Impact Surface . . . 13

4 Methods and Classes **14**

5 Appendix **16**
5.1 Correlation and Kendall's τ . . . 16
5.2 Notes on Selected Methods . . . 17

References **18**

1 Introduction

The `tsmarch` package represents a re-write and re-think of the models in rmgarch. It is written using simpler S3 methods and classes, has a cleaner code base, extensive documentation and unit tests, provides speed gains by making use of parallelization in both R (via the `future` package) and in the C++ code (via RcppParallel package), and works with the new univariate GARCH package tsgarch.

For the conditional mean dynamics, the DCC and Copula GARCH models offer two mutually exclusive workflows. The first is to jointly estimate ARMA-GARCH dynamics in the first stage by passing an `arma` order to `garch_modelspec` (from `tsgarch`), in which case the resulting conditional mean is automatically propagated through the multivariate model's specification, filtering, prediction and simulation methods. The second is to pre-filter the series for conditional mean dynamics outside the system (since the distributions are location/scale invariant) and pass the conditional mean (`cond_mean`) as an argument in the specification,

filtering, prediction and simulation methods so that it is incorporated into the output of various methods at each step of the way; this requires all first stage models to use `arma = c(0,0)`. The two workflows cannot be mixed: `cond_mean` is rejected whenever any first stage model includes ARMA dynamics. Alternatively, the user can location shift (re-center) the simulated predictive distribution matrix by a pre-calculated conditional mean forecast, or pass the zero mean correlated predictive distribution innovations as an input to the conditional mean prediction simulation dynamics. For the GOGARCH model, only the external pre-filtering approach is available since the first stage models are estimated on latent independent components rather than the original series.

Separately from the conditional mean workflows above, the first stage models can also include variance equation regressors via the `vreg` argument of `garch_modelspec` (with `multiplicative` controlling whether they enter the variance intercept additively or in exponential form). These propagate through the same specification, filtering, prediction and simulation surfaces via the `newvreg` and `vreg` arguments described for each method. Since they say nothing about the conditional mean, they are independent of the two workflows above and can be combined freely with `cond_mean`. Note also that whilst missing mean regressors are replaced with zeros and a warning, a series specified with `vreg` whose future regressors are not supplied raises an error naming the affected series.

The nature of the models implemented allows for separable dynamics in estimation which means that individual series have univariate dynamics which can be estimated in parallel. However, the drawback to this is that for standard errors a partitioned Hessian must be calculated which may be very time consuming. While the package defaults to the use of OPG (outer of product of gradients) for standard errors which is fast, there is always the option to calculate the full partitioned Hessian for other types of standard error estimators.

The next sections provide a detailed overview of each of the 3 models available, namely the Dynamic Correlation GARCH model of Engle (2002) (**DCC**), the Copula-DCC model (see Patton (2009) and Ausin and Lopes (2010)) (**CGARCH**) and the Generalized Orthogonal GARCH model with multivariate affine Generalized Hyperbolic Distribution (see Chen et al. (2010) and Broda and Paolella (2009)) (**GOGARCH**).

Code snippets are provided in each section to highlight the code implementation, whilst Section 4 provides diagrams for the methods in the package.

2 Covariance Decomposition Models (`dcc` and `cgarch`)

The covariance decomposition type models decompose the dynamics of the conditional covariance into the conditional standard deviations and correlations, so that they may be expressed in such a way that the univariate and multivariate dynamics may be separated, thus simplifying the estimation process. This decomposition allows a great deal of flexibility by defining individual univariate dynamics for each marginal series, and separate dynamics for the conditional correlation, using a 2-stage estimation process.

2.1 Constant Correlation

In the Constant Conditional Correlation (**CCC**) model of Bollerslev (1990) the equation for the conditional covariance can be represented as:

$$H_t = D_t R D_t = \rho_{ij}\sqrt{h_{iit}h_{jjt}}, \tag{1}$$

where $D_t = diag(\sqrt{h_{11,t}}, ..., \sqrt{h_{nn,t}})$, and R is the positive definite constant correlation matrix.

The conditional variances, $h_{ii,t}$, which can be estimated separately, usually admit some type of GARCH(p,q) model, but can be generalized to any admissible model for the variance dynamics.

The conditions for the positivity of the covariance matrix H_t are that R is positive definite and the diagonal elements of D_t, $h_{ii,t}$, are positive.

Formally, the model may be represented as follow:

$$
\begin{aligned}
y_t &= \mu_t + \varepsilon_t, \varepsilon_t \sim N\left(0, \Sigma_t\right) \\
\Sigma_t &= D_t R D_t \\
z_t &= D_t^{-1} \varepsilon_t \\
LL_t\left(\theta\right) &= \frac{1}{2}\left(n \log\left(2\pi\right) + 2\log\left|D_t\right| + \log\left|R\right| + z_t' R^{-1} z_t\right)
\end{aligned}
\tag{2}
$$

where y_t is a vector representing the series values at time t, μ_t a vector of the conditional mean at time t for each series, ε_t a vector of zero mean residuals which have a multivariate Normal distribution with conditional covariance Σ_t, and $LL_t\left(\theta\right)$ is the log-likelihood at time t given a parameter vector θ of univariate GARCH (or any other model) parameters. The constant correlation matrix R is usually approximated by its sample counterpart as

$$
R = \frac{1}{T}\sum_{t=1}^{T} z_t z_t' \tag{3}
$$

where z_t is the vector of GARCH standardized residuals at time t.

Code Snippet

```
dcc_modelspec(..., dynamics = 'constant')
```

2.2 Dynamic Conditional Correlation

Allowing the correlation to be time varying, whilst retaining the 2-stage separation of dynamics, creates an additional challenge to the positive definiteness constraint at each point in time for the conditional correlation. Engle (2002) proposed the following proxy process :

$$
\begin{aligned}
Q_t &= \bar{Q} + \sum_{j=1}^{q} a_j\left(z_{t-j} z_{t-j}' - \bar{Q}\right) + \sum_{j=1}^{p} b_j\left(Q_{t-j} - \bar{Q}\right) \\
&= \left(1 - \sum_{j=1}^{q} a_j - \sum_{j=1}^{p} b_j\right)\bar{Q} + \sum_{j=1}^{q} a_j\left(z_{t-j} z_{t-j}'\right) + \sum_{j=1}^{p} b_j Q_{t-j}
\end{aligned}
\tag{4}
$$

where a_j and b_j are non-negative coefficients with the constraint that they sum to less than 1 for stationarity and positive definiteness, and $\bar{Q}$ is the unconditional matrix of the standardized residuals z_t. To recover R_t, the following re-scaling operation is performed:

$$
R_t = \operatorname{diag}\left(Q_t\right)^{-1/2} Q_t \operatorname{diag}\left(Q_t\right)^{-1/2} \tag{5}
$$

Under the assumption that $\varepsilon_t \sim N\left(0, \Sigma_t\right)$, the log-likelihood is:

$$
\begin{aligned}
LL_t\left(\theta\right) &= -\frac{1}{2}\left(n\log\left(2\pi\right) + 2\log\left|D_t\right| + z_t z_t'\right) - \frac{1}{2}\left(z_t z_t' + \log\left|R_t\right| + z_t' R_t^{-1} z_t\right) \\
&= LL_{V,t}(\phi) + LL_{R,t}\left(\psi|\phi\right)
\end{aligned}
\tag{6}
$$

where $LL_{V,t}(\phi)$ is the log-likelihood of the conditional variances given the GARCH parameters ϕ, and $LL_{R,t}\left(\psi|\phi\right)$ is the log-likelihood of the conditional correlation given the DCC parameters $\psi|\phi$ (conditioned on the GARCH parameters).

Code Snippet

```
dcc_modelspec(..., dynamics = 'dcc')
```

2.3 Estimation

Estimation of the CCC and DCC models is done in 2 steps. The first step is to estimate the conditional variances for each series using a univariate GARCH model. The univariate models do not need to conform to the any single type of dynamics, but can be any admissible model. The conditional variances are then used to standardize the residuals which are then used to estimate the conditional correlation matrix. Since the first step is separable, it can be done in parallel for each series. After considerable experimentation, it was chosen to avoid the use of TMB for automatic differentiation and instead use numerical derivatives for the second stage estimation due to speed considerations, and observing that there was little gain in terms of accuracy.

Code Snippet

```
dcc_modelspec(...) |> estimate(...)
```

2.4 Standard Errors

Since estimation is performed in 2 stages, the log-likelihood is the sum of these 2 stages. More specifically, the second stage log-likelihood is estimated conditioning on the parameters of the first stage. Details of the assumptions behind this and the arguments for the asymptotic distribution of the estimates are put forth in Engle and Sheppard (2001), who posit that the asymptotic distribution of the 2-stage DCC model estimates with parameters $\theta = (\phi, \psi)$ is:

$$\sqrt{T}\left(\hat{\theta}_T - \hat{\theta}_0\right) \overset{\text{A}}{\sim} N\left(0, B_0^{-1} M_0 B'^{-1}_0\right) \tag{7}$$

where:

$$B_0 = \begin{bmatrix} \nabla_{\phi\phi} \ln f_1(\phi_0) & 0 \\ \nabla_{\phi\psi} \ln f_2(\phi_0) & \nabla_{\psi\psi} \ln f_2(\theta_0) \end{bmatrix} = \begin{bmatrix} B_{11} & 0 \\ B_{12} & B_{22} \end{bmatrix} \tag{8}$$

and

$$M_0 = \text{var}\left[\sum_{t=1}^{T} \left\{ T^{-1/2} \nabla'_\phi \ln f_1(r_t, \phi_0), T^{-1/2} \nabla'_\psi \ln f_2(r_t, \phi_0, \psi_0) \right\}\right] = \begin{bmatrix} M_{11} & M_{12} \\ M_{12} & M_{22} \end{bmatrix} \tag{9}$$

where B_0^{-1} and M_0 are the Hessian and the covariance matrix of the score vector respectively, also called the bread and meat of the sandwich estimator. In the case of this 2 stage estimation procedure, the Hessian is partitioned into 4 blocks, B_{11}, B_{12}, B_{22} and M_{11}, M_{12}, M_{22}, where B_{11} and M_{11} are the Hessian and covariance matrix of the first stage log-likelihood, and B_{22} and M_{22} are the Hessian and covariance matrix of the second stage log-likelihood. B_{12} and M_{12} are the cross derivatives of the first and second stage log-likelihoods.

Because the Hessian is partitioned, the calculation of the standard errors is more computationally intensive than the standard OPG (outer product of gradients) method. The package defaults to the OPG method, but the user can choose to calculate the full partitioned Hessian if they so wish, for which parallel functionality has been implemented to provide some speed gains. The package uses numerical derivatives for the second stage estimation, as well as the calculation of the standard errors. When the partitioned Hessian is present, a number of additional estimators become available including the those of the QML and HAC (using the bandwidth of Newey and West (1994)) type.

Code Snippet

```
dcc_modelspec(...) |> estimate(...) |> vcov(...)
```

2.5 Distributions

The original DCC model was based on the assumption that the residuals are Normally distributed, as shown in equations (2) and (6).

However, this is not always the case in practice. There are 2 possible approaches to this problem. The first is to use a different multivariate distribution for the DCC model and the second is to use a copula approach.

2.5.1 Multivariate Student Distribution (`mvt`)

Introducing a multivariate distribution which requires estimation of higher moment parameters implies that the univariate margins must also have the same distribution and usually the same higher moment parameters. This in turn implies the need for joint estimation of all dynamics in the model with specific constraints on the joint parameters. This is computationally intensive and may not be feasible for anything but a few series. However, it may be possible to estimate the higher moment parameters using QML (quasi-maximum likelihood) for the first stage (essentially use a Normal distribution for the residuals), and then use these estimates in the second stage estimation. This is the approach taken in the `tsmarch` package when using the multivariate Student distribution with log-likelihood equal to:

$$\begin{aligned} LL_t\left(\theta,\nu|\phi\right) = &\log\Gamma\left(\frac{\nu+n}{2}\right) - \log\Gamma\left(\frac{\nu}{2}\right) - \frac{n}{2}\log\left(\pi\left(\nu-2\right)\right) \\ &- \frac{1}{2}\log\left|R_t\right| - \log\left|D_t\right| - \frac{\nu+2}{2}\log\left(1+\frac{1}{\nu-2}z_t' R_t^{-1} z_t\right) \end{aligned} \tag{10}$$

where ν is the degrees of freedom parameter of the Student distribution, and n is the number of series. The assumption post-estimation is that the marginal distribution of each series is Student with the same degrees of freedom parameter ν.

Code Snippet

```
dcc_modelspec(distribution = 'mvt')
```

2.5.2 Copula Distribution (`cgarch`)

The copula approach is a more general approach which allows for the use of different marginal distributions for each series. In simple terms, the copula is a function which links the marginal distributions to the joint distribution by applying a transformation to the marginal residuals to make them uniformly distributed. The joint distribution is then obtained by applying the inverse distribution of the copula to the uniform margins.

Specifically:

$$\begin{aligned} u_t &= C\left(F_1\left(z_{1,t}\right),\ldots,F_n\left(z_{n,t}\right)\right) \\ \epsilon_t &= F^{-1}\left(u_t\right) \end{aligned} \tag{11}$$

where F is the marginal distribution function, F^{-1} is the inverse of the multivariate distribution function, C is the copula function, and u_t is the vector of uniform margins. The transformation of the margins is called the Probability Integral Transform (PIT). There are a number of ways to transform the margins to uniform distributions, including a parametric transform (also called the Inference-Functions-For-Margins by Joe (1997)), using the empirical CDF (also called pseudo-likelihood and investigated by Genest et al. (1995)) and the semi-parametric approach (see Davison and Smith (1990)). All three approaches are implemented in the `tsmarch` package. The package implements both a Normal and Student copula, with either constant or dynamic correlation. For the Student Copula, the log-likelihood is given by:

$$
\begin{aligned}
LL_t(\theta) &= \sum_{i=1}^{n}\left(-\frac{1}{2}\log h_{i,t} + \log f_{i,t}(z_{i,t})\right) + \log c_t\left(u_{i,t},\ldots,u_{n,t}\right) \\
&= LL_{V,t}(\phi) + LL_{R,t}(\psi,\eta|\phi)
\end{aligned} \tag{12}
$$

where

$$
\log c_t\left(u_{1,t},\ldots,u_{n,t}\right) = \log f_t\left(F_i^{-1}\left(u_{1,t}|\eta\right),\ldots,F_i^{-1}\left(u_{n,t}|\eta\right)|R_t,\eta\right) - \sum_{i=1}^{n}\log f_i\left(F_i^{-1}(u_{i,t})|\eta\right) \tag{13}
$$

with $u_{i,t} = F_{i,t}\left(\varepsilon_{i,t}|\phi_i\right)$ is the PIT transformation of each series by it's conditional distribution estimated in the first stage GARCH process by any choice of dynamic and distributions, $F_i^{-1}\left(u_{i,t}|\eta\right)$ represents the quantile transformation of the uniform margins subject to a common shape parameter of the multivariate Student Copula, and $f_t\left(F_i^{-1}\left(u_{1,t}|\eta\right),\ldots,F_i^{-1}\left(u_{n,t}|\eta\right)|R_t,\eta\right)$ is the joint density of the Student Copula with shape parameter η and correlation matrix R_t.

Therefore, the log-likelihood is composed of essentially 3 parts: the first stage univariate models, the univariate PIT transformation and inversion given the Copula distribution and the second stage multivariate model.

The correlation can be modeled as either constant or dynamic in the copula model. The dynamic correlation is modeled in the same way as the DCC model, but the residuals are transformed to uniform margins before estimating the correlation. Appendix 5.1 provides a more detailed overview for the constant correlation case.

Essentially, the difference between the standard DCC type model and the one with a Copula distribution involves the PIT transformation step, which is reflected in the inference (standard errors), forecasting and simulation methods for this model.

Code Snippet

```
cgarch_modelspec(...)
```

2.6 Generalized Dynamics

A number of extensions to the DCC model have been proposed in the literature. These include the use of asymmetric and generalized dynamics. Cappiello et al. (2006) generalize the DCC model with the introduction of the Asymmetric Generalized DCC (AGDCC) where the dynamics of Q_t now become:

$$
Q_t = \left(\bar{Q} - A'\bar{Q}A - B'\bar{Q}B - G'\bar{N}G\right) + \sum_{j=1}^{q}A_j'z_{t-j}z_{t-j}'A_j + \sum_{j=1}^{q}G_j'\bar{z}_{t-j}\bar{z}_{t-j}'G_j + \sum_{j=1}^{p}B_j'QB_j \tag{14}
$$

where A, B and G are the $N\times N$ parameter matrices, $\bar{z}_t$ are the zero-threshold standardized residuals (equal to z_t when less than zero else zero otherwise), $\bar{Q}$ and $\bar{N}$ are the unconditional matrices of z_t and $\bar{z}_t$ respectively. The generalization allows for cross dynamics between the standardized residuals, and cross-asymmetric dynamics. However, this comes at the cost of increased computational complexity making it less feasible for large scale systems. Additionally, covariance targeting in such a high dimensional model where the parameters are no longer scalars creates difficulties in imposing positive definiteness constraints. The `tsmarch` package does not implement this generalized model, but instead a scalar version to capture asymmetric dynamics (`adcc`):

$$
Q_t = \left(1 - \sum_{j=1}^{q}a_j - \sum_{j=1}^{p}b_j\right)\bar{Q} - \sum_{j=1}^{q}g_j\bar{N} + \sum_{j=1}^{q}a_j\left(z_{t-k}z_{t-j}'\right) + \sum_{j=1}^{q}g_j\left(\bar{z}_{t-k}\bar{z}_{t-j}'\right) + \sum_{j=1}^{p}b_jQ_{t-j} \tag{15}
$$

Code Snippet

```
cgarch_modelspec(..., dynamics = 'adcc')
```

2.7 Forecasting and Simulation

Because of the non linearity of the DCC evolution process, the multi-step ahead forecast of the correlation cannot be directly solved. Engle and Sheppard (2001) provide a couple of suggestions which form a reasonable approximation. In the `tsmarch` package we instead use a simulation approach, similar to the other packages in the `tsmodels` universe, and in doing so generate a simulated distribution of the h-step ahead correlation and covariance. This also generalizes to the copula model which does not have a closed form solution. Formally, for a draw s at time t, the simulated correlation is given by:

$$\begin{aligned} Q_t^s &= \bar{Q} + \sum_{j=1}^{q} a_j \left(z_{t-j}^s z^{s\prime}{}_{t-j} - \bar{Q} \right) + \sum_{j=1}^{p} b_j \left(Q^s{}_{t-j} - \bar{Q} \right) \\ R_t^s &= \text{diag}\left(Q^s{}_t \right)^{-1/2} Q^s{}_t \text{diag}\left(Q^s \right)^{-1/2} \\ z_t^s &= E \Lambda^{1/2} \epsilon^s{}_t \end{aligned} \tag{16}$$

where $\epsilon^s{}_t$ is a vector of standard Normal random variables, Λ is the diagonal matrix of the eigenvalues of $R^s{}_t$ and E is the matrix of eigenvectors of $R^s{}_t$, such that:

$$R^s{}_t = E \Lambda E' \tag{17}$$

For the multivariate Student distribution, the draws for z_t^s are generated as :

$$z_t^s = E \Lambda^{1/2} \epsilon^s{}_t \sqrt{\frac{\nu}{W^s}} \tag{18}$$

where W^s is a random draw from a chi-squared distribution with ν degrees of freedom. For $\epsilon^s{}_t$, it is also possible to instead use the empirical standardized and de-correlated residuals, an option which is offered in the package.

Code Snippet

```
dcc_modelspec(...) |> estimate(...) |> predict(...)
dcc_modelspec(...) |> estimate(...) |> simulate(...)
```

2.8 Weighted Distribution

Since all predictions are generated through simulation, the weighted predictive distribution $\bar{y_t}^s$ can be obtained as:

$$\bar{y_t}^s | w = \sum_{i=1}^{n} w_{i,t} y_{i,t}{}^s \tag{19}$$

where s is a sample draw from the simulated predictive distribution of the n series, and $w_{i,t}$ the weight on series i at time t.

Code Snippet

```
dcc_modelspec(...) |> estimate(...) |> tsaggregate(...)
```

2.9 News Impact Surface

Engle and Ng (1993) introduced the news impact curve as a tool for evaluating the effects of news on the conditional variances, and further extended by Kroner and Ng (1998) to the multivariate GARCH case. The news impact surface shows the effect of shocks from a pair of assets on their conditional covariance (or correlation). Formally, the correlation $R_{i,j}$ given a set of shock (z_i, z_j) is given by:

$$R_{i,j}\left(z_i, z_j\right) = \frac{\left(1-a-b\right)\bar{Q}_{i,j} - g\bar{N}_{i,j} + az_iz_j + g\bar{z}_i\bar{z}_j + b\bar{Q}_{i,j}}{\sqrt{\left[\left(1-a-b\right)\bar{Q}_{i,i} - g\bar{N}_{i,i} + az_i^2 + g\bar{z}_i^2 + b\bar{Q}_{i,i}\right]\cdot\left[\left(1-a-b\right)\bar{Q}_{j,j} - g\bar{N}_{i,j} + az_j^2 + g\bar{z}_j^2 + b\bar{Q}_{j,j}\right]}} \tag{20}$$

where $\bar{z}$, $\bar{Q}$ and $\bar{N}$ are defined as in equation (14).

Code Snippet

```
dcc_modelspec(...) |> estimate(...) |> newsimpact(...) |> plot(...)
```

2.10 Critique

The DCC model has come under a considerable amount of criticism in the literature, particularly by Caporin and McAleer (2008) who argue that consistency and asymptotic Normality of the estimators has not been fully established, and proposes instead the indirect DCC model (essentially a scalar BEKK) which models directly the conditional covariance matrix and forgoes univariate GARCH dynamics.

Aielli (2013) also points out that the estimation of Q_t as the empirical counterpart of the correlation matrix of z_t is inconsistent since $E\left[z_t z_t'\right] = E\left[R_t\right] \neq E\left[Q_t\right]$. They propose a DCC (cDCC) model which includes a corrective step which eliminates this inconsistency, albeit at the cost of targeting which is not allowed.

3 Generalized Orthogonal GARCH Model (`gogarch`)

The GOGARCH model in the `tsmarch` package is based on the approach described in Chen et al. (2010), Broda and Paolella (2009) and Ghalanos et al. (2015). Unlike the covariance decomposition models, the Generalized Orthogonal GARCH model is akin to a full distributional decomposition model rather than just the covariance, since it makes use of higher moments to achieve independent margins from which univariate models can then be used to estimate the conditional dynamics (see Schmidt et al. (2006) for details on the multivariate affine Generalized Hyperbolic distribution).

Formally, the stationary return series y_t follows a linear dynamic model:

$$\begin{aligned} y_t &= \mu_t + \varepsilon_t \\ \varepsilon_t &= Af_t \end{aligned} \tag{21}$$

where A is an invertible and constant mixing matrix, which transforms the independent components f_t back into the residuals ε_t. This can be further decomposed into de-whitening and rotation matrices, $\Sigma^{1/2}$ and U respectively:

$$A = \Sigma^{1/2}U \tag{22}$$

where Σ is the unconditional covariance matrix of the residuals and performs the role of de-correlating (whitening) the residuals prior to rotation (independence). It is at this juncture where dimensionality reduction may also be performed, by selecting the first $m\,(\leq n)$ principal components of the residuals. In that case $\varepsilon_t \approx \hat{\varepsilon}_t = Af_t$ since the dimensionality reduced system only approximates the original residuals.

The rotation matrix U is calculated through the use of Independent Component Analysis (ICA) which is a method separating multivariate mixed signals into additive statistically independent and non-Gaussian components[1] (see Hyvärinen and Oja (2000)). The algorithm used in the `tsmarch` package for ICA is based on the RADICAL algorithm of Learned-Miller and Fisher III (2003) which uses an efficient entropy estimator (due to Vasicek (1976)) which is robust to outliers and requires no strong characterization assumptions about the data generating process.[2]

The independent components f_t can be further decomposed as:

$$f_t = H_t^{1/2} z_t \tag{23}$$

where H_t is the conditional covariance matrix of the independent components with $H_t = E[f_t f_t' | \mathfrak{F}_{t-1}]$, being a diagonal matrix with elements $(h_{1,t}, \dots, h_{n,t})$ which are the conditional variances of the factors, and $z_t = (z_{1,t}, \dots, z_{n,t})'$. The random variable $z_{i,t}$ is independent of z_{jt-s} $\forall j \neq i$ and $\forall s$, with $E[z_{i,t}|\mathfrak{F}_{t-1}] = 0$ and $E[z_{i,t}^2] = 1$, implying that $E[f_t|\mathfrak{F}_{t-1}] = \mathbf{0}$ and $E[\varepsilon_t|\mathfrak{F}_{t-1}] = \mathbf{0}$. The factor conditional variances, $h_{i,t}$ can be modeled as a GARCH-type process. The unconditional distribution of the factors is characterized by:

$$E[f_t] = \mathbf{0} \quad E[f_t f_t'] = I_n \tag{24}$$

which, in turn, implies that:

$$E[\varepsilon_t] = \mathbf{0} \quad E[\varepsilon_t \varepsilon_t'] = AA'. \tag{25}$$

It then follows that the return series can be expressed as:

$$y_t = \mu_t + A H_t^{1/2} z_t. \tag{26}$$

The conditional covariance matrix, $\Sigma_t \equiv E[(y_t - \mu_t)(y_t - \mu_t)'|\mathfrak{F}_{t-1}]$ is given by $\Sigma_t = A H_t A'$.

The Orthogonal Factor model of Alexander (2002)[3] which uses only information in the covariance matrix, leads to uncorrelated components but not necessarily independent unless assuming a multivariate Normal distribution. However, while whitening is not sufficient for independence, it is nevertheless an important step in the pre-processing of the data in the search for independent factors, since by exhausting the second order information contained in the covariance matrix it makes it easier to infer higher order information, reducing the problem to one of rotation (orthogonalization).

Code Snippet

```
gogarch_modelspec(...)
```

3.1 Estimation

The estimation of the GOGARCH model is performed in 3 steps.

1. Filter the data for conditional mean dynamics ($\mu_{i,t}$). Similar to the DCC model, this is performed outside the system.
2. Perform Independent Component Analysis (ICA) on the residuals to separate out the additive and statistically independent components.

[1] Technically, up to 1 component may be Gaussian.

[2] To speed estimation, parts of the ICA algorithm are parallelized in C++ using 'RcppParallel' and hence benefits from multi-threading.

[3] When U is restricted to be an identity matrix, the model reduces to the Orthogonal Factor model.

3. Estimate the conditional distribution of each independent component using an admissible univariate model. In the `tsmarch` package we use the `tsgarch` package for estimating the GARCH model for each factor with either a Generalized Hyperbolic (`gh`) or Normal Inverse Gaussian (`nig`) distribution[4]

The log-likelihood of the GOGARCH model is composed of the individual log-likelihoods of the univariate independent factor GARCH models and a term for the mixing matrix:

$$LL_t(\theta, A) = \log\left|A^{-1}\right| + \sum_{i=1}^{n} \log\left(GH_{\lambda_i}(\theta_i)\right) \tag{27}$$

where GH_{λ_i} is the Generalized Hyperbolic distribution with shape parameter λ_i for the i-th factor. Since the inverse of the mixing matrix is the un-mixing matrix $A^{-1} = W = K'U$, and since the determinant of U is 1 ($|U| = 1$), the the log-likelihood simplifies to:

$$LL_t(\theta, A) = \log|K| + \sum_{i=1}^{n} \log\left(GH_{\lambda_i}(\theta_i)\right) \tag{28}$$

In the presence of dimensionality reduction, K is no longer square and we cannot directly compute its determinant. Instead, we use the determinant of KK' , to reflect the combined effect of whitening and dimensionality reduction:

$$LL_t(\theta, A) = \log|KK'| + \sum_{i=1}^{m} \log\left(GH_{\lambda_i}(\theta_i)\right) \tag{29}$$

where m are the number of factors. A key advantage of this approach is the ability to estimate multivariate dynamics using only univariate methods, providing for feasible estimation of large scale systems. The speed advantage of this approach does not come for free, as there is a corresponding need for memory to handle the parallelization offered, particularly when generating higher co-moment matrices.

Code Snippet

```
gogarch_modelspec(...) |> estimate(...)
```

3.2 Co-Moments

The linear affine representation of the GOGARCH model provides for a closed-form expression for the conditional co-moments. The first 3 co-moments are given by:

$$\begin{aligned} M_t^2 &= AM_{f,t}^2 A' \\ M_t^3 &= AM_{f,t}^3 (A' \otimes A') \\ M_t^4 &= AM_{f,t}^4 (A' \otimes A' \otimes A') \end{aligned} \tag{30}$$

where $M_{f,t}^2$, $M_{f,t}^3$ and $M_{f,t}^4$ are the $(N \times N)$ conditional covariance, $(N \times N^2)$ the conditional third co-moment matrix and the $(N \times N^3)$ conditional fourth co-moment matrix of the factors, respectively, represented as unfolded tensors[5]

These are calculated from the univariate factor dynamics, and defined as:

[4]Technically, the 'nig' distribution is a special case of the 'gh' distribution with $\lambda = -0.5$, but since is possesses some nice properties, it is included as a separate distribution for speed purposes.

[5]The package allows to extract these tensors in both folded and unfolded form.

$$
\begin{aligned}
M^3_{f,t} &= \left[M^3_{1,f,t}, M^3_{2,f,t}, \ldots, M^3_{n,f,t}\right] && (31)\\
M^4_{f,t} &= \left[M^4_{1,1,f,t}, M^4_{1,2,f,t}, \ldots, M^4_{1,n,f,t} | \ldots | M^4_{n,1,f,t}, M^4_{n,2,f,t}, \ldots, M^4_{n,n,f,t}\right] && (32)
\end{aligned}
$$

where $M^3_{k,f,t}, k = 1, \ldots, n$ and $M^4_{k,l,f,t}, k, l = 1, \ldots, n$ are the $N \times N$ sub-matrices of $M^3_{f,t}$ and $M^4_{f,t}$, respectively, with elements

$$
\begin{aligned}
m^3_{i,j,k,f,t} = &\quad E[f_{i,t} f_{j,t} f_{k,t} | \mathfrak{F}_{t-1}] \\
m^4_{i,j,k,l,f,t} = &\quad E[f_{i,t} f_{j,t} f_{k,t} f_{l,t} | \mathfrak{F}_{t-1}].
\end{aligned} \tag{33}
$$

Since the factors $f_{i,t}$ can be decomposed as $z_{i,t}\sqrt{h_{i,t}}$, and given the assumptions on $z_{i,t}$ then $E[f_{i,t} f_{j,t} f_{k,t} | \mathfrak{F}_{t-1}] = 0$. It is also true that for $i \neq j \neq k \neq l$ $E[f_{i,t} f_{j,t} f_{k,t} f_{l,t} | \mathfrak{F}_{t-1}] = 0$ and when $i = j$ and $k = l$,

$$
E[f_{i,t} f_{j,t} f_{k,t} f_{l,t} | \mathfrak{F}_{t-1}] = h^2_{i,t} h^2_{k,t}. \tag{34}
$$

Thus, under the assumption of mutual independence, all elements in the conditional co-moments matrices with at least 3 different indices are zero. Finally, standardizing the conditional co-moments one obtains conditional co-skewness and co-kurtosis of y_t,

$$
\begin{aligned}
S_{i,j,k,t} &= \frac{m^3_{i,j,k,t}}{(\sigma_{i,t}\sigma_{j,t}\sigma_{k,t})}, \\
K_{i,j,k,l,t} &= \frac{m^4_{i,j,k,l,t}}{(\sigma_{i,t}\sigma_{j,t}\sigma_{k,t}\sigma_{l,t})}
\end{aligned} \tag{35}
$$

where $S_{i,j,k,t}$ represents the series co-skewness between elements i, j, k of y_t, $\sigma_{i,t}$ the standard deviation of $y_{i,t}$, and in the case of $i = j = k$ represents the skewness of series i at time t, and similarly for the co-kurtosis tensor $K_{i,j,k,l,t}$. Two natural applications of return co-moments matrices are Taylor type utility expansions in portfolio allocation and higher moment news impact surfaces.

Code Snippet

```
tscov(...)
tscor(...)
tscoskew(...)
tscoskurt(...)
```

The weighted moments are calculated as follows:

$$
\begin{aligned}
\mu_{1,t} &= w_t \text{`} M^1_t \\
\mu_{2,t} &= w'_t M^2_t w_t \\
\mu_{3,t} &= w'_t M^3_t \left(w_t \otimes w_t\right) \\
\mu_{4,t} &= w'_t M^4_t \left(w_t \otimes w_t \otimes w_t\right)
\end{aligned} \tag{36}
$$

where w_t is the vector of weights at time t. Using the vectorization operator vec the weighted moments can be re-written compactly as:

$$
\mu_k = \text{vec}\left(M_k\right)' \underbrace{\left(w_t \otimes \ldots \otimes w_t\right)}_{k \text{ times}} \tag{37}
$$

One application of this is in the calculation of the Constant Absolute Risk Aversion utility function, using a Taylor series expansion as in Ghalanos et al. (2015):

$$E_{t-1}\left[U\left(W_t\right)\right] = E_{t-1}\left[-\exp\left(-\eta\bar{W}_t\right)\right] \approx -\exp\left(-\eta\bar{W}_t\right)\left[1+\frac{\eta}{2}\mu_{2,t}-\frac{\eta^3}{3!}\mu_{3,t}+\frac{\eta^4}{4!}\mu_{4,t}\right] \tag{38}$$

where η represents the investor's constant absolute risk aversion, with higher (lower) values representing higher (lower) aversion, and $\bar{W}_t = w_t' E_{t-1}\left[y_t\right]$ is the expected portfolio return at time t. Note that this approximation includes two moments more than the mean-variance criterion, with the skewness and kurtosis are directly related to investor preferences (dislike) for odd (even) moments, under certain mild assumptions given in Scott and Horvath (1980).

Code Snippet

```
gogarch_modelspec(...) |> estimate(...) |> predict(...) |> tsaggregate(...)
```

3.3 Weighted Distribution

The weighted conditional density of the model may be obtained via the inversion of the Generalized Hyperbolic characteristic function through the fast fourier transform (FFT) method as in Chen et al. (2010) or by simulation as in Section 2.8. The `tsmarch` package generates both simulated predictive distributions as well as the weighted p^*, d^* and q^* functions based on FFT.

It should be noted that while the n-dimensional NIG distribution is closed under convolution, when the distributional higher moment parameters are allowed to be different across assets, as is the case here, this property no longer holds.

Provided that ϵ_t is a n-dimensional vector of innovations, marginally distributed as 1-dimensional standardized GH, the density of weighted asset returns, $w_{i,t}y_{i,t}$ is:

$$w_{i,t}y_{i,t} - w_{i,t}\mu_{i,t} = \bar{w}_{i,t}\epsilon_{i,t} \sim \text{maGH}_{\lambda_i}\left(\bar{w}_{i,t}\bar{\mu}_{i,t}, \left|\bar{w}_{i,t}\delta_{i,t}\right|, \frac{\alpha_{i,t}}{\left|\bar{w}_{i,t}\right|}, \frac{\beta_{i,t}}{\left|\bar{w}_{i,t}\right|}\right) \tag{39}$$

where $\bar{w}_t = w_t' A H_t^{1/2}$ and hence $\bar{w}_{i,t}$ is the i-th element of this vector, $\mu_{i,t}$ is the conditional mean of series i at time t, and the parameter vector $(\bar{\mu}_{i,t}, \delta_{i,t}, \beta_{i,t}, \alpha_{i,t}, \lambda_i)$ represents the generalized hyperbolic distributional parameters which have been converted from their (ρ, ζ) parameterization used in the `tsgarch` package estimation routine. In order to obtain the weighted density we need to sum the individual log densities of $\epsilon_{i,t}$:

$$\varphi(u) = \prod_{i=1}^{n}\varphi_{\bar{w}\epsilon_i}(u) = \exp\left(\mathrm{i}u\sum_{j=1}^{d}\bar{\mu}_j + \sum_{j=1}^{d}\left(\frac{\lambda_j}{2}\log\left(\frac{\gamma}{\upsilon}\right) + \log\left(\frac{K_{\lambda_j}(\bar{\delta}_j\sqrt{\upsilon})}{K_{\lambda_j}(\bar{\delta}_j\sqrt{\gamma})}\right)\right)\right) \tag{40}$$

where, $\gamma = \bar{\alpha}_j^2 - \bar{\beta}_j^2$, $\upsilon = \bar{\alpha}_j^2 - (\bar{\beta}_j + \mathrm{i}u)^2$, and $(\bar{\alpha}_j, \bar{\beta}_j, \bar{\delta}_j, \bar{\mu}_j)$ are the scaled versions of the parameters $(\alpha_i, \beta_i, \delta_i, \mu_i)$ as shown in equation (39). The density may be accurately approximated by FFT as follows:

$$f_y(r) = \frac{1}{2\pi}\int_{-\infty}^{+\infty} e^{(-\mathrm{i}ur)}\varphi_y(u)\mathrm{d}u \approx \frac{1}{2\pi}\int_{-s}^{s} e^{(-\mathrm{i}ur)}\varphi_y(u)\mathrm{d}u. \tag{41}$$

Once the density is obtained, we use smooth approximations to obtain functions for the density (d^*), quantile (q^*) and distribution (p^*), with random number generation provided by the use of the inverse transform method using the quantile function.

Code Snippet

```
gogarch_modelspec(...) |> estimate(...) |> predict(...) |> tsconvolve(...) |> dfft(...)
```

3.4 Factor News Impact Surface

Consider the equation for the construction of the conditional covariance matrix Σ_t:

$$\Sigma_t = AH_tA' \tag{42}$$

where H_t is the diagonal conditional covariance matrix of the independent components. The factor news impact covariance surface is a measure of the effect of shocks from the independent factors on the conditional covariance of the returns of assets i and j at time $t-1$. In order to construct this we first generate the news impact curves for all the factors using a common grid of shock values, and then calculate the effect of these on the conditional covariance of the returns of assets i and j. Since the factors are independent, we simply loop through all pairwise combinations of these shocks[6] to calculate the covariance surface, keeping the value of the other factors constant:

$$\Sigma_{i,j} = a_{k,i} \cdot h_{k,k}|\epsilon_k \cdot a_{k,j} + a_{l,i} \cdot h_{l,l}|\epsilon_l \cdot a_{l,j} + \sum_{\substack{m=1 \\ m\neq k,l}}^{n} a_{m,i} \cdot h_{m,m}|\epsilon_m = 0 \cdot a_{m,j} \tag{43}$$

where :

1. $a_{k,i}$ is the loading of factor k on asset i,
2. $h_{k,k}|\epsilon_k$ is the conditional variance of factor k from the news impact curve given the shock ϵ_k,
3. $a_{k,i} \cdot h_{k,k}|\epsilon_k \cdot a_{k,j}$ captures the contribution of factor k to the covariance between assets i and j, based on the shock ϵ_k
   applied to factor k,
4. $a_{l,i} \cdot h_{l,l}|\epsilon_l \cdot a_{l,j}$ captures the contribution of factor l to the covariance between assets i and j, based on the shock ϵ_l applied to factor l and
5. $\sum_{\substack{m=1 \\ m\neq k,l}}^{n} a_{m,i} \cdot s_{m,m}(0) \cdot a_{m,j}$ captures the contributions of all other factors (except k and l), where their variances are held constant at their unconditional levels (i.e., with no shocks).

This means that the covariance will be affected by both the relative sign and magnitude of the loadings and the relative magnitude of the variance of the factors.

The news impact surface is not restricted to the covariance matrix, but can be extended to the conditional co-skewness and co-kurtosis, but that is only truly meaningful when the higher moment distributional parameters are allowed to vary as in Ghalanos et al. (2015).

Code Snippet

```
gogarch_modelspec(...) |> estimate(...) |> newsimpact(...) |> plot(...)
```

[6] We loop through all possible combinations of shocks for factors k and l which were previously calculated from their respective news impact curves using a common grid of shock values $\{\epsilon_1, \ldots, e_z\}$

4 Methods and Classes

Figures 1 and 2 provide a visual summary of the main methods in the package for the DCC and GOGARCH type models. Blue boxes denote a method which produces a custom class which can be further manipulated with other methods[7], whilst Appendix 5.2 provides a summary of some of the methods and output classes in the `tsmarch` package and the type of parallelization they use.

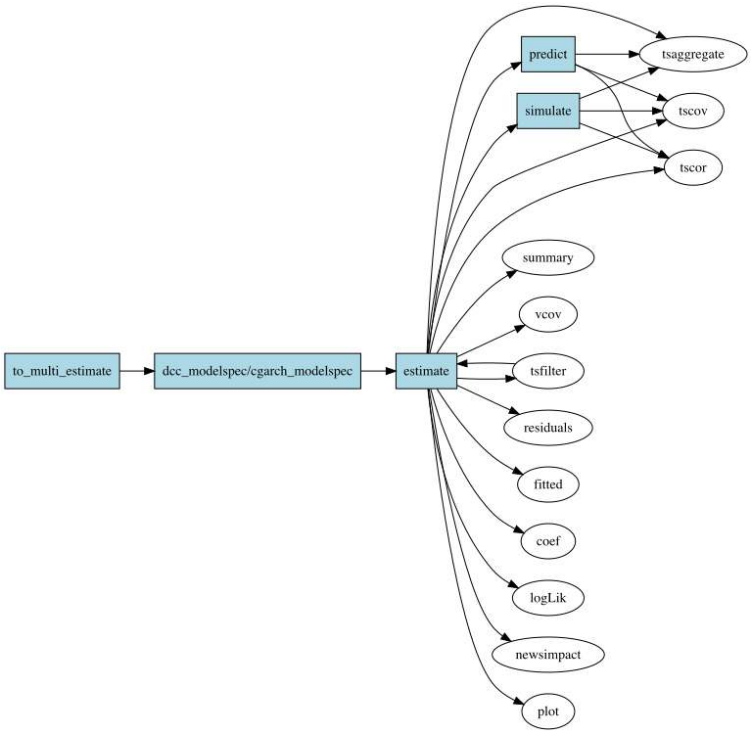


Figure 1: DCC/CGARCH diagram representing function/method relationships in the package.

[7]The tsfilter method returns an updated object of the same class as the 'estimate' method hence has an arrow going back to that method to denote this.

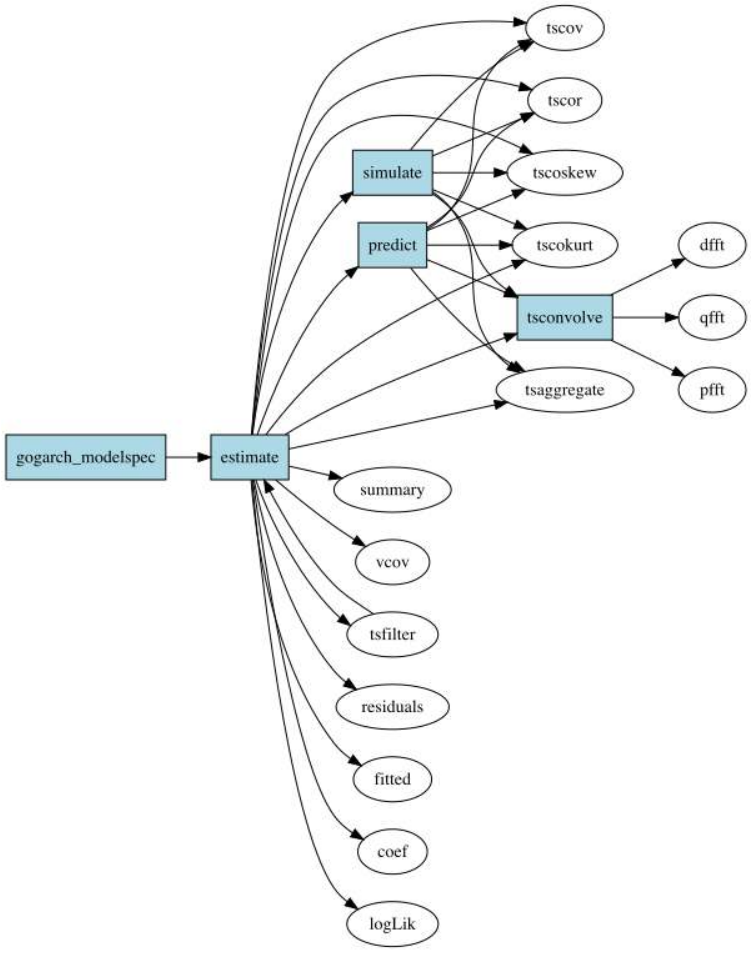


Figure 2: GOGARCH diagram representing function/method relationships in the package.

5 Appendix

5.1 Correlation and Kendall's τ

Pearson's product moment correlation R totally characterizes the dependence structure in the multivariate Normal case, where zero correlation also implies independence, but can only characterize the ellipses of equal density when the distribution belongs to the elliptical class. In the latter case for instance, with a distribution such as the multivariate Student, the correlation cannot capture tail dependence determined by the shape parameter. Furthermore, it is not invariant under monotone transformations of original variables making it inadequate in many cases. An alternative measure which does not suffer from this is Kendall's τ (see Kruskal (1958)), based on rank correlations which makes no assumption about the marginal distributions but depends only on the copula C. It is a pairwise measure of concordance calculated as:

$$\tau(x_i, x_j) = 4 \int_0^1 \int_0^1 C\left(u_i, u_j\right) \, dC\left(u_i, u_j\right) - 1 \tag{44}$$

For elliptical distributions, Lindskog et al. (2003), proved that there is a one-to-one relationship between this measure and Pearson's correlation coefficient ρ given by:

$$\tau(x_i, x_j) = \left(1 - \sum_{x \in \mathbb{R}} \left(\mathbb{P}\{X_i = x\}^2\right)\right) \frac{2}{\pi} \arcsin \rho_{ij} \tag{45}$$

which under certain assumptions (e.g. in the multivariate Normal case) simplifies to

$$\frac{2}{\pi} \arcsin \rho_{ij}$$

. Kendall's τ is invariant under monotone transformations making it more suitable when working with non-elliptical distributions. A useful application arises in the case of the multivariate Student distribution where the maximum likelihood approach for the estimation of the correlation matrix R become unfeasible for large dimensions. Instead, estimating the sample counterpart of Kendall's τ from the transformed margins and then translating that into a correlation matrix as detailed in equation (45) provides for a method of moments type estimator. The shape parameter ν is then estimated keeping the correlation matrix constant, with little loss in efficiency.

5.2 Notes on Selected Methods

The table below provides a summary of selected methods, the classes they belong to, the classes they output, whether they are parallelized and the dimensions of the objects involved. For example, the tsaggrate method on predicted and simulated objects will return a list of length 4, with each slot having an object of class `tsmodel.distribution` of size $sim \times h$ representing the predictive simulatated distribution of each of the weighted moments generated by the model.[8] The output for the `tsconvolve` method which is `list(h):list(sim)` means that the output is a list of length h where each slot contains a list of length sim.

Method	class (in)	class (out)	future	RcppParallel	dimensions
`tscov`	(1,4,7)	`matrix`	no	no	[n, n, T]
`tscov`	(2,3,5,6,8,9)	`matrix`	no	no	[n, n, h, sim]
`tscoskew`	(1)	`matrix`	no	yes	[n, n^2, T]
`tscoskew`	(2,3)	`matrix`	no	yes	[n, n^2, h, sim]
`tscokurt`	(1)	`matrix`	no	yes	[n, n^3]
`tscokurt`	(2,3)	`matrix`	no	yes	[n, n^3, h, sim]
`tsconvolve`	(1)	`gogarch.fft`	yes	no	list(T)
`tsconvolve`	(2,3)	`gogarch.fftsim`	yes	no	list(h):list(sim)
`tsaggregate`	(1,4,7)	`list`	no	yes	4:n
`tsaggregate`	(2,3,5,6,8,9)	`list`	no	yes	4:(sim x h)

Table 1: Selected Methods

Key for class (in):

- **1**: `gogarch.estimate`
- **2**: `gogarch.predict`
- **3**: `gogarch.simulate`
- **4**: `dcc.estimate`
- **5**: `dcc.predict`
- **6**: `dcc.simulate`
- **7**: `cgarch.estimate`
- **8**: `cgarch.predict`
- **9**: `cgarch.simulate`

[8] For the DCC and Copula GARCH models only the first 2 moments are time varying and hence contain a simulated distribution.

References

Aielli, Gian Piero. 2013. "Dynamic Conditional Correlation: On Properties and Estimation." *Journal of Business & Economic Statistics* 31 (3): 282–99.

Alexander, Carol. 2002. "Principal Component Models for Generating Large GARCH Covariance Matrices." *Economic Notes* 31 (2): 337–59.

Ausin, M. C., and H. F. Lopes. 2010. "Time-Varying Joint Distribution Through Copulas." *Computational Statistics and Data Analysis* 54 (11): 2383–99.

Bollerslev, T. 1990. "Modelling the Coherence in Short-Run Nominal Exchange Rates: A Multivariate Generalized ARCH Model." *Review of Economics and Statistics* 72 (3): 498–505.

Broda, S. A., and M. S. Paolella. 2009. "CHICAGO: A Fast and Accurate Method for Portfolio Risk Calculation." *Journal of Financial Econometrics* 7 (4): 412.

Caporin, Massimiliano, and Michael McAleer. 2008. "Scalar BEKK and Indirect DCC." *Journal of Forecasting* 27 (6): 537–49.

Cappiello, L., R. F. Engle, and K. Sheppard. 2006. "Asymmetric Correlations in the Dynamics of Global Equity and Bond Returns." *Journal of Financial Econometrics* 4 (4): 537–72.

Chen, Y., W. Härdle, and V. Spokoiny. 2010. "GHICA–Risk Analysis with GH Distributions and Independent Components." *Journal of Empirical Finance* 17 (2): 255–69.

Davison, A. C., and R. L. Smith. 1990. "Models for Exceedances over High Thresholds." *Journal of the Royal Statistical Society: Series B (Methodological)* 52 (3): 393–442.

Engle, R. F. 2002. "Dynamic Conditional Correlation." *Journal of Business and Economic Statistics* 20 (3): 339–50.

Engle, R. F., and V. K. Ng. 1993. "Measuring and Testing the Impact of News on Volatility." *The Journal of Finance* 48 (5): 1749–78.

Engle, R. F., and K. Sheppard. 2001. *Theoretical and Empirical Properties of Dynamic Conditional Correlation Multivariate GARCH*. UCSD Working Paper NO.

Genest, C., K. Ghoudi, and L. P. Rivest. 1995. "A Semiparametric Estimation Procedure of Dependence Parameters in Multivariate Families of Distributions." *Biometrika* 82 (3): 543–52.

Ghalanos, A., E. Rossi, and G. Urga. 2015. "Independent Factor Autoregressive Conditional Density Model." *Econometric Reviews* 34 (5): 594–616.

Hyvärinen, Aapo, and Erkki Oja. 2000. "Independent Component Analysis: Algorithms and Applications." *Neural Networks* 13 (4-5): 411–30.

Joe, H. 1997. *Multivariate Models and Dependence Concepts*. Vol. 73. Chapman & Hall/CRC.

Kroner, K. F., and V. K. Ng. 1998. "Modeling Asymmetric Comovements of Asset Returns." *The Review of Financial Studies* 11 (4): 817–44.

Kruskal, W. H. 1958. "Ordinal Measures of Association." *Journal of the American Statistical Association* 53 (284): 814–61.

Learned-Miller, Erik G., and John W. Fisher III. 2003. "ICA Using Spacings Estimates of Entropy." *The Journal of Machine Learning Research* 4: 1271–95.

Lindskog, F., A. Mcneil, and U. Schmock. 2003. "Kendall's Tau for Elliptical Distributions." *Credit Risk: Measurement, Evaluation and Management*.

Newey, Whitney K, and Kenneth D West. 1994. "Automatic Lag Selection in Covariance Matrix Estimation." *The Review of Economic Studies* 61 (4): 631–53.

Patton, A. J. 2009. "Copula–Based Models for Financial Time Series." In *Handbook of Financial Time Series*. Springer.

Schmidt, Rafael, Tomas Hrycej, and Eric Stützle. 2006. "Multivariate Distribution Models with Generalized Hyperbolic Margins." *Computational Statistics & Data Analysis* 50 (8): 2065–96.

Scott, R. C., and P. A. Horvath. 1980. "On the Direction of Preference for Moments of Higher Order Than the Variance." *Journal of Finance* 35 (4): 915–19.

Vasicek, Oldrich. 1976. "A Test for Normality Based on Sample Entropy." *Journal of the Royal Statistical Society Series B: Statistical Methodology* 38 (1): 54–59.