The Model Confidence Set package for R

Mauro Bernardi[a], Leopoldo Catania[b,*]

[a] *Department of Statistical Sciences, University of Padova, Padova, Italy.*
[b] *Department of Economics and Finance, University of Rome, Tor Vergata, Rome, Italy*

Abstract

This paper presents the R package `MCS` which implements the Model Confidence Set (MCS) procedure for model comparison. The MCS procedure consists on a sequence of tests which permits to build a set of "superior" models, where the null hypothesis of Equal Predictive Ability (EPA) is not rejected at a certain confidence level. The EPA statistic test is calculated for an arbitrary loss function, meaning that we could test models on various aspects, such as for example, punctual forecasts and density evaluation. The relevance of the package is shown using an example which aims at illustrating in details the use of the provided functions. The example compares the ability of different models belonging to the GARCH family to predict large financial losses. Codes for reproducibility purposes are also reported.

Keywords: MCS, model choice, R, VaR.

1. Introduction

The large amount of available econometric models to address the same empirical question open the issues of model choice, model comparison and model combination. Although several alternative models are usually available, it is not always the case that a particular model clearly outperforms all the available competitors. The goal of model choice can then be reformulated in terms of reducing the number of available models in a way that the "real" best performing one is not excluded with a given probability. Ideally, the number of remaining models should be determined by the level of uncertainty across the different performances, such as if there is no evidence of over performance of a model versus a particular alternative according to a defined loss measure, both models should be considered. The Model Confidence Set (MCS) procedure, recently developed by Hansen et al. (2011), casts this kind of reasonings into a statistical framework.

Our contribution in this paper is to develop and introduce the `MCS` package for R to provide an integrated environment for the comparison of alternative models using the Hansen et al. (2011) procedure. We believe the main feature lies in the tools that the package provides for specifying different loss functions and the

[*] Department of Economics and Finance, University of Rome "Tor Vergata" , Via Columbia 2, 00135, Rome, Italy, `leopoldo.catania@uniroma2.it`.

iterative model selection.

The MCS procedure starts from an initial set of models M^0 of dimension m encompassing all the available model specifications, and delivers, for a given confidence level $1-\alpha$, a smaller set $\hat{\mathrm{M}}^*_{1-\alpha}$ named "Superior Set of Models" (SSM) of dimension $m^* \leq m$, where the null hypothesis of equal predictive ability (EPA) is not rejected at a certain confidence level. The EPA statistic test is calculated for an arbitrary loss function that satisfies general weak stationarity conditions. Formally, let $Y_t \in K \subseteq \mathbb{R}^d$ be the observation at time t and $\hat{Y}_{i,t} \in K \subseteq \mathbb{R}^d$ be the output of model i at time t for $i = 1, \ldots, m$ and $t = 1, \ldots, n$. Define also the loss function $\ell : K \times K \to S \subseteq \mathbb{R}$ such as

$$\ell_{i,t} = \ell\left(Y_t, \hat{Y}_{i,t}\right), \tag{1}$$

is the loss of model i at time t. Essentially, apart from some technical condition detailed in Hansen et al. (2011), the only requirement of the loss function is that the ordering across the m available models is complete at each point in time t. Define also $d_{ij,t}$ as the loss differential between models i and j:

$$d_{ij,t} = \ell_{i,t} - \ell_{j,t}, \quad i,j = 1, \ldots, m, \quad t = 1, \ldots, n, \tag{2}$$

and let

$$d_{i\cdot,t} = (m-1)^{-1} \sum_{j \in \mathrm{M}} d_{ij,t} \quad i = 1, \ldots, m, \tag{3}$$

be the simple loss of model i relative to any other model j at time t. The EPA hypothesis for a given set of models M can be formulated in two alternative ways:

$$\begin{aligned} \mathrm{H}_{0,\mathrm{M}} &: \quad c_{ij} = 0, \qquad \text{for all} \quad i,j = 1, 2, \ldots, m \\ \mathrm{H}_{\mathrm{A},\mathrm{M}} &: \quad c_{ij} \neq 0, \qquad \text{for some} \quad i,j = 1, \ldots, m, \end{aligned} \tag{4}$$

or

$$\begin{aligned} \mathrm{H}_{0,\mathrm{M}} &: \quad c_{i\cdot} = 0, \qquad \text{for all} \quad i = 1, 2, \ldots, m \\ \mathrm{H}_{\mathrm{A},\mathrm{M}} &: \quad c_{i\cdot} \neq 0, \qquad \text{for some} \quad i = 1, \ldots, m, \end{aligned} \tag{5}$$

where $c_{ij} = \mathbb{E}\left(d_{ij}\right)$ and $c_{i\cdot} = \mathbb{E}\left(d_{i\cdot}\right)$ are assumed to be finite and not time dependent. According to Hansen et al. (2011), in order to test the two hypothesis above, the following two statistics are constructed:

$$t_{ij} = \frac{\bar{d}_{ij}}{\sqrt{\widehat{\mathrm{var}}\left(\bar{d}_{ij}\right)}} \qquad \text{and} \quad t_{i\cdot} = \frac{\bar{d}_{i,\cdot}}{\sqrt{\widehat{\mathrm{var}}\left(\bar{d}_{i,\cdot}\right)}} \quad \text{for} \quad i,j \in \mathrm{M}, \tag{6}$$

where $\bar{d}_{i,\cdot} = (m-1)^{-1} \sum_{j \in \mathrm{M}} \bar{d}_{ij}$ is the simple loss of the i–th model relative to the averages losses across models in the set M, and $\bar{d}_{ij} = n^{-1} \sum_{t=1}^{n} d_{ij,t}$ measures the relative sample loss between the i–th and j–th models, while $\widehat{\mathrm{var}}\left(\bar{d}_{i,\cdot}\right)$ and $\widehat{\mathrm{var}}\left(\bar{d}_{ij}\right)$ are bootstrapped estimates of $\mathrm{var}\left(\bar{d}_{i,\cdot}\right)$ and $\mathrm{var}\left(\bar{d}_{ij}\right)$, respectively. The first t–statistic t_{ij} is used in the well know test for comparing two forecasts; see e.g. Diebold and Mariano (2002) and West (1996), while the second one is used in Hansen et al. (2003); Hansen (2005); Hansen et al. (2011). As discussed in Hansen et al. (2011) the two EPA null hypothesis presented in equations (4) and (5) map naturally into the two test statistics:

$$\mathrm{T}_{\mathrm{R},\mathrm{M}} = \max_{i,j\in\mathrm{M}} \mid t_{ij} \mid \quad \text{and} \quad \mathrm{T}_{\max,\mathrm{M}} = \max_{i\in\mathrm{M}} t_{i\cdot}, \tag{7}$$

where t_{ij} and $t_{i\cdot}$ are defined in equation (6). The test statistics defined in equation (7) can be used in order to test the two hypothesis (4) and (5), respectively. Since the asymptotic distributions of the two tests statistics is nonstandard, the relevant distributions under the null hypothesis is estimated using a bootstrap procedure similar to that used to estimate var $\left(\bar{d}_{i,\cdot}\right)$ and var $\left(\bar{d}_{ij}\right)$, see for details White (2000); Hansen et al. (2011); Kilian (1999); Clark and McCracken (2001).

Briefly, the MCS procedure to obtain the SSM, consists of the following steps:

1. set $\mathrm{M} = \mathrm{M}_0$
2. test for EPA–hypothesis: if EPA is accepted terminate the algorithm and set $\mathrm{M}^*_{1-\alpha} = \mathrm{M}$, otherwise determinate the worst model at the current iteration
3. remove the worst model, and go to step 2.

The choice of the worst model to be eliminated is made using an elimination rule that is coherent with the statistic tests defined in equation (6). For the $\mathrm{T}_{\mathrm{R},\mathrm{M}}$ and $\mathrm{T}_{\max,\mathrm{M}}$ statistic tests the elimination rules are

$$e_{\mathrm{R},\mathrm{M}} = \arg\max_{i} \left\{ \sup_{j\in\mathrm{M}} \frac{\bar{d}_{ij}}{\sqrt{\widehat{\mathrm{var}}\left(\bar{d}_{ij}\right)}} \right\}, \qquad e_{\max,\mathrm{M}} = \arg\max_{i\in\mathrm{M}} \frac{\bar{d}_{i,\cdot}}{\widehat{\mathrm{var}}\left(\bar{d}_{i,\cdot}\right)}, \tag{8}$$

respectively.

The rest of the paper is organised as follow. Section 2 illustrates the issue of models proliferation within the famous class of Generalised Autoregressive Conditional Heteroscedasticity (GARCH) for volatility modelling. The models detailed in this section will be used throughout the paper for exemplification purposes. Although popular for modelling time–varying volatility, the application of the MCS procedure cannot be confined within the range of GARCH models. The MCS procedure has been effectively applied to several alternative model specifications, such as the Generalised autoregressive Score models of Creal et al. (2013) and Harvey (2013), and the Dynamic quantile models of Engle and Manganelli (2004), see, e.g., Bernardi and Catania (2016). In Section 3, we discuss how the `MCS` package is used to determine the optimal superior set of models (SSM). Section 4 reports an empirical application which aims at illustrating how the procedure can be practically implemented. Section 5 concludes.

2. Forecasting volatiliy: the case of the GARCH specifications

In the context of volatility modeling, the class of GARCH models introduced by Engle (1982) and Bollerslev (1986) counts hundreds of different specifications, see e.g. Bollerslev (2008) and Bauwens et al. (2006) for a review. Moreover, if the research interest in not only on volatility modeling, but also in estimating the conditional distribution of a stochastic process at each point in time, the econometrician also faces the problem of determine the most adequate parametric assumption for the conditional return distribution. The number of different possible specifications may also increases due to different assumptions on the conditional mean of the process. Here we report a list of different specifications for the first two conditional moments as well as for the distributional parametric assumption for a time series of financial log–returns. The specifications reported below can be easily estimated by Maximum Likelihood in R using the popular `rugarch` package of Ghalanos (2014).

Formally, let y_t be the logarithmic return at time t, we consider the following general AR(1)–GARCH–in mean specification originally proposed by Engle et al. (1987)

$$\begin{aligned} y_t &= \mu + \lambda\sigma_t^2 + \phi y_{t-1} + \varepsilon_t, \qquad \varepsilon_t = \sigma_t\zeta_t, \qquad \zeta_t \sim \mathcal{D}\left(0,1\right) \\ \sigma_t^2 &= h\left(\sigma_{t-1}, \varepsilon_{t-1}, \boldsymbol{\theta}_\sigma \mid \mathcal{F}_{t-1}\right), \end{aligned}$$

where $\mathcal{F}_t$ is the information set up to time t, ζ_t is a sequence of independently and identically distributed random variables with general standardized distribution $\mathcal{D}$, σ_t is the conditional standard deviation of y_t and ϕ is the autoregressive parameter assumed to be $|\phi| < 1$ to preserve stationarity. The risk–premium parameter λ is set equal to zero if the "in mean" specification is omitted, otherwise it is jointly estimated with the other parameters. Finally, the function $h\left(\cdot\right)$ refers to one of the GARCH–type dynamics reported below, where the vector $\boldsymbol{\theta}_\sigma$ contains all the conditional variance dynamic parameters.

Among the most popular distributions that econometricians usually choose to model the error term ζ_t we consider: the Gaussian $\mathcal{N}\left(0,1\right)$, the Student–t $\mathcal{T}_\nu\left(0,\frac{\nu-2}{\nu}\right)$, with ν degrees of freedom, the Generalised Error distribution $\mathcal{GED}\left(0,1,\kappa\right)$, where κ is the shape parameter, and their asymmetric counterparts, the Skew–Normal, the Skew Student–t and the Skew–GED, obtained by applying the skewing mechanism of Fernández and Steel (1998). Furhermore, we consider the Johnson's reparametrised SU distribution of Rigby and Stasinopoulos (2005) and Generalized Hyperbolic distribution introduced by Barndorff-Nielsen (1977).

Concerning the assumption for the conditional volatility dynamics, we compare the popular GARCH(p,q) specification introduced by Bollerslev (1986) whose dynamics is given by

$$\sigma_t^2 = \omega + \sum_{i=1}^{p} \alpha\varepsilon_{t-i-1}^2 + \sum_{j=1}^{q} \beta\sigma_{t-j-1}^2, \tag{9}$$

where $P \equiv \sum_{i=1}^{p}\alpha + \sum_{j=1}^{q}\beta < 1$ denotes the persistence parameter, and $\omega > 0$ and $0 \leq \alpha_i < 1, \forall i = 1, 2, \ldots, p$ and $0 \leq \beta_j < 1, \forall j = 1, 2, \ldots, q$, with several well known alternative specifications, such as, the

EGARCH(p,q) volatility dynamics of Nelson (1991)

$$\log\left(\sigma_t^2\right) = \omega + \sum_{i=1}^{p} \left[\alpha_i \zeta_{t-i} + \gamma_i \left(|\zeta_{t-i}| - \mathbb{E}|\zeta_{t-i}|\right)\right] + \sum_{j=1}^{q} \beta_j \log\left(\sigma_{t-j}^2\right), \tag{10}$$

with persistence $P = \sum_{j=1}^{q} \beta_j < 1$, where the parameters γ_i modulates the asymmetric response of volatility to positive and negative shocks. Another widely used asymmetric GARCH model is the GJR–GARCH(p,q) specification of Glosten et al. (1993), which accounts for the leverage effect

$$\sigma_t^2 = \omega + \sum_{i=1}^{p} \left(\alpha_i + \gamma_i \mathbb{1}_{(-\infty,0)}\left(\varepsilon_{t-i}\right)\right) \varepsilon_{t-i}^2 + \sum_{j=1}^{q} \beta_j \sigma_{t-j}^2, \tag{11}$$

where the indicator function $\mathbb{1}_{(-\infty,0)}\left(\varepsilon_{t-i}\right)$ assumes value one if $\varepsilon_{t-i} < 0$ for $i = 1, 2, \dots, p$ and zero otherwise. Because of the presence of the indicator function, the persistence of the GJR–GARCH specification crucially depends on the asymmetry of the conditional distribution used to model the error term ε_t

$$P = \sum_{i=1}^{p} \alpha_i + \sum_{j=1}^{q} \beta_j + \sum_{i=1}^{p} \gamma_i \mathbb{P}\left(\varepsilon_{t-i} \leq 0\right), \tag{12}$$

where $\mathbb{P}\left(\varepsilon_{t-i} \leq 0\right)$ denotes the probability of observing negative shocks and $\omega > 0$, $\alpha_i \geq 0$, for $i = 1, 2, \dots, p$, $\beta_j \geq 0$, for $j = 1, 2, \dots, q$, and the additional constraint $\alpha_i + \gamma_i \geq 0$ for $i = 1, 2 \dots, p$ is imposed to preserve the positiveness of the conditional variance. Finally, the Asymmetric–Power–ARCH(p,q) (APARCH, henceforth) of Ding et al. (1993) imposes the following dynamic to the conditional variance

$$\sigma_t^{(\delta)} = \omega + \sum_{i=1}^{p} \alpha_i \left(|\varepsilon_{t-i}| - \gamma_i \varepsilon_{t-i}\right)^{\delta} + \sum_{j=1}^{q} \beta_j \sigma_{t-j}^{(\delta)}, \tag{13}$$

where $x^{(\delta)} = \frac{x^{\delta}-1}{\delta}$ is the Box–Cox transformation of Box and Cox (1964). The parameters restrictions to ensure the positiveness of the conditional variance are $\omega > 0$, $\delta > 0$, $-1 \leq \gamma_i \leq 1$ for $i = 1, \dots, p$ and the usual condition $\alpha_i \geq 0$, and $\beta_j \geq 0$, for $i, j = 1, 2, \dots, \max\{p, q\}$. As for the GJR–GARCH specification the persistence strongly depends to the probability density function chosen for the innovation term ζ_t, and it is given by

$$P = \sum_{i=1}^{p} \alpha_i \kappa_i + \sum_{j=1}^{q} \beta_j, \tag{14}$$

where $\kappa_i = \mathbb{E}\left[|\zeta| - \gamma_i \zeta\right]^{\delta}$, for $i = 1, \dots, q$. The APARCH specification results in a very flexible model that nests several of the most popular univariate GARCH parameterisations, such as

- the GARCH(p,q) of Bollerslev (1986) for $\delta = 0$ and $\gamma_i = 0$, for $i = 1, 2, \dots, p$;

- the Absolute–Value–GARCH (AVARCH, henceforth) specification for $\delta = 1$ and $\gamma_i = 0$ for $i = 1, 2, \dots, p$, proposed by Taylor (1986) and Schwert (1990) to mitigates the influence of large, in an absolute sense, shocks with respect to the traditional GARCH specification;

- the GJR–GARCH model of Glosten et al. (1993) for $\delta = 2$ and $0 \leq \gamma_i \leq 1$ for $i = 1, 2, \ldots, p$;
- the Threshold GARCH (TGARCH, henceforth) of Zakoian (1994) for $\delta = 1$, which allows different reactions of the volatility to different signs of the lagged errors;
- the Nonlinear GARCH (NGARCH, henceforth) of Higgins and Bera (1992) for $\gamma_i = 0$ for $i = 1, 2, \ldots, p$ and $\beta_j = 0$ for $j = 1, 2, \ldots, q$.

Another relevant volatility specification we consider is the Component–GARCH(p,q) (CGARCH, henceforth) of Engle and Lee (1993) which decomposes the conditional variance into a permanent and transitory component in a straightforward way

$$\begin{aligned}\sigma_t^2 &= \xi_t + \sum_{i=1}^{p} \alpha_i \left(\varepsilon_{t-i}^2 - \xi_{t-i}\right) + \sum_{j=1}^{q} \beta_j \left(\sigma_{t-j}^2 - \xi_{t-j}\right) \\ \xi_t &= \omega + \rho \xi_{t-1} + \eta \left(\varepsilon_{t-1}^2 - \sigma_{t-1}^2\right), \end{aligned} \tag{15}$$

where to ensure the stationarity of the process at the usual $\sum_{i=1}^{p} \alpha_i + \sum_{j=1}^{q} \beta_j < 1$ condition the $\rho < 1$ constrain must be added. Further parameters restrictions for the positiveness of the conditional variance are given in Engle and Lee (1993). This solution is usually employed because it permits to investigate the long and short–run movements of volatility. To estimate model parameters the Maximum Likelihood estimator can be easily adopted, see e.g. Francq and Zakoian (2011).

2.1. GARCH model estimation and forecast in R

Within the R environment a wide range of statistical packages are available in order to deal with GARCH models estimation and forecast such as the `fGarch` package of Wuertz et al. (2013). Here, in order to estimate the competing GARCH models we use some functions belonging to the library `rugarch` developed by Ghalanos (2014). Of course, the following treatment is only useful for illustrative purposes on the use of the `MCS` package, and readers are free to choose a different package to estimate models as well as their handwritten functions. Before starting the MCS procedure, it is necessary to define the set of competing GARCH models M_0. This can be done using the the `ugarchspec()` function that permits to specify a variety of GARCH models such as those previously described. For example, the following portion of code

```
R> library(rugarch)
R> spec <- ugarchspec(mean.model = list(armaOrder = c(1, 0)),
                      variance.model = list(model = "sGARCH",
                      garchOrder = c(1, 1)),
                      distribution.model = "norm")
```

creates an uGARCHspec object “spec” which defines a AR(1) - GARCH(1,1) model with Gaussian innovations. The object spec can be subsequently used into the ugarchfit() function in order to estimate the model on a given financial time series such as one of those included in the MCS package. For example, the portion of code

```
R> library(MCS)
R> data(STOXXIndexesRet)
R> ret <- STOXXIndexesRet[,"SXA1E"]
R> fit <- ugarchfit(spec = spec, data = ret)
```

creates an uGARCHfit object “fit” containing parameter estimates of the STOXX North America 600 index (“SXA1E”) as well as several additional informations such as the Information Criteria, tests on standardised residuals, among others. Prediction using GARCH models can be easily performed using the ugarchforecast() function which takes as an argument the output of the fitting procedure uGARCHfit or alternatively the uGARCHspec object. The one step ahead forecast can be easily obtained using the following routine

```
R> OneStepForc <- ugarchforecast(fitORspec = fit, n.ahead = 1)
```

which reports an uGARCHforecast, object “OneStepForc”. The rugarch package also includes the ugarchroll() function which permits to construct a series of one step ahead rolling forecasts allowing also the user to define a “refit window” and the length of the forecast series. For example, a rolling forecast series of length 2000, using a refit window of 5 observations can be computed using

```
R> roll <- ugarchroll(spec = spec, data = ret, forecast.length = 2000,
                      refit.every = 5)
```

which reports an uGARCHroll object roll. Finally, a variety of methods are present in the rugarch package in order to deal with “uGARCHspec” , “uGARCHfit” , “uGARCHforecast” and “uGARCHroll” objects. For more informations see Ghalanos (2014) or consult the help() in R.

3. The MCS package

As previously detailed, the MCS procedure can be used to compare different models under an user defined loss function. The loss function measures the “performance” of the competing models at each time point $t = 1, \dots, n$ in the evaluating period. Suppose now to compare m alternative models over the evaluating period of length n, then the outputs of the loss function defined in equation 1 can be stored in a loss matrix of dimension $(m \times n)$ containing, for each time $t = 1, \dots .n$, the losses associated to each competing model. The R function MCSprocedure() can then be used to build the set of superior models. In this section we firstly report how it is possible to build the loss matrix associated to the one step ahead Value–at–Risk (VaR) forecasts delivered by the previously reported GARCH specifications using the `rugarch` package. Secondly, we detail several loss functions that are available in the `MCS` package, finally, we show how to build the SSM using the `MCSprocedure()` routine.

3.1. Comparing GARCH models using MCS

Suppose we are interested in comparing the VaR forecasts delivered by different GARCH models. As previously explained, the `ugarchroll()` function can be used to obtain the one step ahead rolling forecast series of a specified GARCH model. Furthermore, the `as.data.frame()` method permits to extract from the `uGARCHroll` object the VaR forecasts series at both the 1% and 5% confidence levels. For example, if we want to compare five different GARCH specifications, such as

- the GARCH(1,1) of Bollerslev (1986), "`sGARCH`" in the `rugarch` package;
- EGARCH(1,1) of Nelson (1991), "`eGARCH`" in the `rugarch` package;
- GJRGARCH(1,1) of Glosten et al. (1993), "`gjrGARCH`" in the `rugarch` package;
- APARCH(1,1) of Ding et al. (1993), "`apARCH`" in the `rugarch` package;
- CGARCH(1,1) of Engle and Lee (1993), "`csGARCH`" in the `rugarch` package;

we can simply define the five GARCH specifications and combine them with the distributional assumptions presented in the previous section using the `ugarchspec()` function as follows

```
R> models <- c("sGARCH", "eGARCH", "gjrGARCH", "apARCH", "csGARCH")
R> distributions <- c("norm", "std", "ged", "snorm", "sstd", "sged", "jsu", "ghyp")
R> spec.comp <- list()
R> for( m in models ) {
     for( d in distributions ) {
         spec.comp[[paste( m, d, sep = "-" )]] <-
                    ugarchspec(mean.model = list(armaOrder = c(0, 0)),
                               variance.model = list(model = m, garchOrder = c(1, 1)),
                               distribution.model=d)
                                }
                        }
R> specifications <- names( spec.comp )
```

In this way we have defined a list containing the 40 combinations of GARCH specifications and innovation term distributions. Now we perform a rolling forecast using the last 2000 observations using a "refit window" of length 200

```
R> roll.comp <- list()
R> for( s in specifications ){
   roll.comp[[s]] <-  ugarchroll(spec = spec.comp[[s]], data = ret,
                                 forecast.length = 2000, refit.every = 200)
                      }
```

Finally, the VaR forecasts at the confidence level $\tau = 1\%$ are stored in a new list `VaR.comp` exploiting the `as.data.frame()` method,

```
R> VaR.comp=list()
R> for( s in specifications ) {
       VaR.comp[[s]] <-  as.data.frame(roll.comp[[s]], which = "VaR")[, 1]
       }
```

Now it is possible to evaluate the loss associated to each model at each time. Here, we consider the asymmetric VaR loss function of González-Rivera et al. (2004) and considered also by Bernardi et al. (2016) and Bernardi and Catania (2016), which is implemented in the `MCS` package through the `LossVaR()` function, more details of the available loss functions are reported in the next Subsection,

```
R> Loss <- do.call(cbind,lapply(specifications,
                   function(s) LossVaR(tau=0.01, realized=tail(ret, 2000)/100,
                                       evaluated=VaR.comp[[s]]/100)))
R> colnames(Loss) <- specifications
```

The object “`Loss`” is a matrix of dimension (2000×40) which contains the VaR losses associated to the different models specifications. The object “`Loss`” is also included in the `MCS` packages for descriptive purposes and can be easily loaded using `data(Loss)`. Nevertheless, in the next Subsection we describe some alternative loss functions which are implemented in the `MCS` package.

3.2. Loss functions

As previously discussed the MCS procedure is able to discriminate models under an user defined loss function. The choice of the loss function is somewhat arbitrary, and crucially depends on the nature of the competing models and the scope of their usage. For more considerations about the choice of the loss function for model comparison purposes we refer to Hansen and Lunde (2005), Bollerslev et al. (1994), Diebold and Lopez (1996) and Lopez (2001). In what follows, we report the loss functions available within the `MCS` package. However, in the `MCS` package the user is free to define and use its own loss function. Three different loss functions are available within the `MCS` package:

1. the `LossVaR()` that can be used to check the performances associated to VaR (or more generally quantile) forecasts;
2. the `LossVol()` for volatility forecasts assessment;
3. the `LossLevel()` that can be used for level forecasts, as the punctual mean forecast of a regression model.

These loss functions accept three common arguments. The first two arguments are `realized` that consists of a vector of realised observations (i.e. the ones that a model hopes to accurately forecast or describe), and `evaluated` which is a vector or a matrix of models output. Note that we call the second argument of those functions "`evaluated`" instead of "`forecasted`" since the MCS procedure is more general than simply a procedure for forecasts evaluation. In fact, as reported by Hansen et al. (2011), the MCS procedure also adapts to in sample studies such as goodness of fit for regression analysis. The `which` argument instead is function dependent. The available choices and other function specific arguments are reported below.

- For `LossVaR()` only `which = "asymmetricLoss"` is available. This coincides with the asymmetric VaR loss function of González-Rivera et al. (2004) which is particularly suited to assess quantile risk measures, such as the VaR, since it penalises more heavily observations below the τ–th quantile level, i.e. $y_t < \mathrm{VaR}_t^\tau$. The asymmetric loss function is defined as

  $$\ell\left(y_t, \mathrm{VaR}_t^\tau\right) = \left(\tau - d_t^\tau\right)\left(y_t - \mathrm{VaR}_t^\tau\right), \tag{16}$$

  where $d_t^\tau = \mathbb{1}\left(y_t < \mathrm{VaR}_t^\tau\right)$ is the τ–quantile loss function. Further arguments are `tau`, which represents the VaR confidence level and `type` with possible choices `"normal"` and `"differentiable"`. The `type` argument permits to discriminate between the normal and the differentiable versions of the loss function: `"normal"` permits the specification of the loss function of González-Rivera et al. (2004) defined in equation (16) while `"differentiable"` considers the following loss function

  $$\ell\left(r_t, \mathrm{VaR}_t^\tau\right) = \left(\tau - m_\delta\left(r_t, \mathrm{VaR}_t^\tau\right)\right)\left(r_t - \mathrm{VaR}_t^\tau\right), \tag{17}$$

  where $m_\delta\left(a, b\right) = \left[1 + \exp\left\{\delta\left(a - b\right)\right\}\right]^{-1}$. Note that the δ parameter, controlling the function smoothness, can be chosen by the `delta` argument in the `LossVaR()` function and it is set equal to 25 by default.

- For `LossVol()`, the six loss functions reported in Hansen and Lunde (2005) are implemented. Note that for this kind of loss functions the `realized` and the `evaluated` arguments should be some realised volatility measures $\tilde{\sigma}_{t+1}$ and the punctual volatility forecasts $\hat{\sigma}_{t+1}$. In this context, we use the term volatility as for the conditional standard deviation. The available loss functions are:

1. $\text{SE}_{1,t+1} = (\tilde{\sigma}_{t+1} - \hat{\sigma}_{t+1})^2$, by setting `which = "SE1"`,

2. $\text{SE}_{2,t+1} = \left(\tilde{\sigma}^2_{t+1} - \hat{\sigma}^2_{t+1}\right)^2$, by setting `which = "SE2"`,

3. $\text{QLIKE}_{t+1} = \log\left(\hat{\sigma}^2_{t+1}\right) + \tilde{\sigma}^2_{t+1}\hat{\sigma}^{-2}_{t+1}$, by setting `which = "QLIKE"`,

4. $\text{R}^2\text{LOG}_{t+1} = \left[\log\left(\tilde{\sigma}^2_{t+1}\hat{\sigma}^{-2}_{t+1}\right)\right]^2$, by setting `which = "R2LOG"`,

5. $\text{AE}_{1,t+1} = |\tilde{\sigma}_{t+1} - \hat{\sigma}_{t+1}|$, by setting `which = "AE1"`,

6. $\text{AE}_{2,t+1} = |\tilde{\sigma}^2_{t+1} - \hat{\sigma}^2_{t+1}|$, by setting `which = "AE2"`.

- For `LossLevel()`, the `which` argument accepts values: `"SE"` and `"AE"` and coincide with the squared error and the absolute error, respectively.

3.3. Building the Superior Set of Models using the MCSprocedure() function

The function `MCSprocedure()` employs the MCS procedure previously detailed. It returns a S4 object of the class "`SSM`", which has several arguments we now briefly describe here. The main inputs of the function `MCSprocedure()` are

- `Loss`, which must be a `matrix` object or something coercible to that (using the as.matrix() function) which contains the loss series for each model to compare.

- `alpha`, that must be a positive scalar in $(0, 1)$ indicating the confidence level of the MCS tests.

- `B`, which is an integer indicating the number of bootstrapped samples used to construct the statistic test.

- `cluster`, that coincides with a cluster object created by calling makeCluster from the parallel package. By default this is set to NULL but if an appropriate cluster object is submitted, then this will be used for parallel processing.

- `statistic`, which is the statistic that should be used to test the EPA at each step of the iteration. Possible choices are "Tmax" and "TR", which coincide with the $\text{T}_{\text{max},M}$ and the $\text{T}_{\text{R,M}}$ reported in equation (6).

For sample purposes, in the MCS package the "Loss" dataset is included. This coincide with the `Loss` matrix obtained in the previous Subsection and can be loaded using `data(Loss)`. To build the SSM one simply need to run the following portion of code:

```
R> library(MCS)
R> data(Loss)
R> SSM <- MCSprocedure(Loss = Loss, alpha = 0.2, B = 5000, statistic = "Tmax")
```

```
R> SSM
------------------------------------------------------------------------------
-                            Superior Set of Models                          -
------------------------------------------------------------------------------
              Rank_M          v_M  MCS_M Rank_R         v_R  MCS_R       Loss
sGARCH-ged         27  0.996797601 0.4486     27  1.46532329 0.2352 0.0003986329
sGARCH-snorm       25  0.896954417 0.5920     26  1.38772567 0.2998 0.0003982803
sGARCH-sstd        26  0.938712179 0.5248     25  1.37978372 0.3166 0.0003977886
sGARCH-sged        20  0.521486029 0.9732     20  1.20336841 0.5346 0.0003956815
sGARCH-jsu         23  0.820277282 0.7052     24  1.35541006 0.3166 0.0003971334
sGARCH-ghyp        22  0.685040892 0.8628     22  1.27425551 0.4478 0.0003964821
eGARCH-ged         30  1.136085890 0.2480     31  1.55570183 0.1574 0.0003994099
eGARCH-sstd        17 -0.004284375 1.0000     17  0.99146387 0.7916 0.0003933537
eGARCH-sged        12 -0.538418981 1.0000     12  0.68337534 0.9802 0.0003910679
eGARCH-jsu         15 -0.132370421 1.0000     15  0.92181102 0.8454 0.0003928127
eGARCH-ghyp        16 -0.107676711 1.0000     16  0.94757871 0.8260 0.0003929190
gjrGARCH-norm      18  0.213742841 1.0000     18  1.10231033 0.6710 0.0003943638
gjrGARCH-std       13 -0.427196927 1.0000     13  0.75538350 0.9588 0.0003916026
gjrGARCH-ged        6 -0.954946367 1.0000      8  0.37895234 1.0000 0.0003891435
gjrGARCH-snorm      1 -1.295629237 1.0000      2  0.04458467 1.0000 0.0003870702
gjrGARCH-sstd       5 -0.962068788 1.0000      5  0.33202562 1.0000 0.0003887004
gjrGARCH-sged       2 -1.252823785 1.0000      1 -0.04527337 1.0000 0.0003867926
gjrGARCH-jsu        7 -0.942033667 1.0000      6  0.33760463 1.0000 0.0003887359
gjrGARCH-ghyp       8 -0.929362792 1.0000      4  0.32763998 1.0000 0.0003886976
apARCH-norm        21  0.535927772 0.9668     21  1.25059782 0.4788 0.0003959991
apARCH-std         19  0.367561226 0.9980     19  1.19805785 0.5490 0.0003949103
apARCH-ged         14 -0.317682324 1.0000     14  0.80503488 0.9304 0.0003920526
apARCH-snorm        4 -0.977717453 1.0000      7  0.33884203 1.0000 0.0003889586
apARCH-sstd         9 -0.628176498 1.0000      9  0.63183157 0.9904 0.0003905724
apARCH-sged         3 -1.022889502 1.0000      3  0.29488276 1.0000 0.0003884919
apARCH-jsu         10 -0.588780764 1.0000     10  0.63897617 0.9892 0.0003906327
apARCH-ghyp        11 -0.582730804 1.0000     11  0.64705812 0.9878 0.0003906581
csGARCH-sstd       31  1.165077099 0.2170     30  1.53265345 0.1782 0.0003992507
csGARCH-sged       24  0.839136228 0.6792     23  1.34145394 0.3166 0.0003972841
csGARCH-jsu        29  1.063487090 0.3454     28  1.47330668 0.2280 0.0003985447
```

```
csGARCH-ghyp       28  1.046751336 0.3706     29  1.47854362 0.2244 0.0003984870
------------------------------------------------------------------------------
Details
Number of eliminated models : 9
Statistic : Tmax
Elapsed Time : Time difference of 6.771243 mins
```

4. Application to financial returns

In this section we aim to demonstrate the relevance of the MCS procedure in the context of VaR forecast comparison. To this end, we collect a panel of four major worldwide stock markets indices at a daily frequency. The indices includes the Asia/Pacific 600 (SXP1E), the North America 600 (SXA1E) and the Europe 600 (SXXP) as well as the Global 1800 (SXW1E). The data are freely available and can be download from the STOXX web site. The considered period spans from 31 December 1991 to 24 July 2014, for a total of 5874 observations per each series. For each market, the returns are calculated as the logarithmic difference of the daily price index and multiplied by 100, i.e.

$$y_t = (\log(p_t) - \log(p_{t-1})) \times 100,$$

where p_t is the closing index value on day t. To examine the performance of the models to predict VaR levels, the complete dataset of daily returns is divided into two samples: an in–sample period from 1 January 1992 to 06 October 2006, for a total of 3814 observations, and a forecast period, containing the remaining 2000 observations: from 09 October 2006 to 24 July 2014. A rolling window approach is used to produce 1–day ahead forecasts of the 1% VaR thresholds $\text{VaR}_{t+1}^{0.01}$, for $t = 1, 2, \ldots, 2000$ in the forecast samples. To apply the MCS procedure we consider 160 GARCH–type specifications obtained by combining the models reported in the previous part of the paper. More precisely, the 160 model specifications have been obtained by considering all the possible combinations of the 10 GARCH dynamics, the 8 conditional distributions and the *in–mean/non in–mean* options as detailed in the previous part of the paper. VaR estimates are performed by inverting the conditional cumulative density function of the corresponding estimated model. Then, the MCS procedure is applied to obtain the set of models with superior predictive ability in term of the VaR forecast at the 1% confidence level.

Table 1 reports the compositions of the Superior Set of Models discriminating by the conditional volatility specification ("model"), the distributional assumption for the error term ("distribution") and for the presence or not of the *in–mean* specification for the conditional mean of the process ("in–mean"). The different entries in each column represent the number of models that belong to the SSM at the end of the MCS procedure discriminated by model, distribution and *in–mean* options. From Table 1 we can observe that for the

SXA1E and SXP1E indexes the SSM is quite homogeneous with respect to the dynamics and innovation assumption. In these cases, the one step ahead 1% VaR forecasting performances of the competing models are quite similar, suggesting that for those series the use of complicated nonlinear volatility specifications is not entirely justified. Moreover, it is interestingly to note that the EGARCH specification is the most frequently eliminated. Concerning the error term distributional assumption, we observe that the MCS procedure confirms the common finding of inadequacy for the Gaussian distribution to describe the financial returns series behaviour. Conversely, looking at the third and the fourth columns, it is possible to asses that for the SXW1E and SXXP indexes the MCS procedure reports more discriminant results. For the SXW1E index, all the specifications that include the GARCH, IGARCH, C–GARCH and N–GARCH dynamics were eliminated while also the GJR–GARCH dynamic is not present in the SSM for the SXXP index. The exclusion of those models suggests that for the SXW1E and SXXP series, more complicated dynamics are necessary to describe the shape of the conditional returns density function.

5. Conclusion

In this paper we have illustrated the main features of the R package `MCS` which implements the Model Confidence Set procedure introduced by Hansen et al. (2011). The Hansen *et.al*'s technique is useful when more models are available and it is not obvious which one is the best. The MCS sequence of tests delivers the Superior Set of Models (SSM) having Equal Predictive Ability (EPA) in terms of an user supplied loss function. The `MCS` package is very flexible in the types of model and loss functions that can be specified by the researcher. This freedom allows the user to concentrate on substantive issues, such as the construction of the initial set of model specifications M^0, without being limited by the constraints imposed by the software. An empirical example shows the relevance of the package by illustrating in details the use of the provided functions. In particular, the example compares the ability of different models belonging to the GARCH family to predict the VaR.

	Asset			
	SXA1E	SXP1E	SXW1E	SXXP
Models				
eGARCH	7	8	7	5
sGARCH	16	16	0	0
gjrGARCH	16	16	16	0
iGARCH	12	9	0	0
apARCH	16	16	16	10
csGARCH	12	16	0	0
TGARCH	13	13	14	14
AVGARCH	15	15	7	11
NGARCH	15	16	0	0
NAGARCH	16	16	13	13
Distributions				
norm	12	19	5	0
snorm	16	18	9	8
std	16	18	10	5
sstd	19	17	11	9
ged	18	18	8	5
sged	19	18	11	9
ghyp	19	16	9	9
jsu	19	17	10	8
In–Mean Specification				
Not in mean	72	76	43	29
in mean	66	65	30	24
Total Number	138	141	73	53

Table 1: Composition of remaining models in the Superior Set for each index.

References

Barndorff-Nielsen, O. (1977). Exponentially decreasing distributions for the logarithm of particle size. *Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences*, 353(1674):401–419.

Bauwens, L., Laurent, S., and Rombouts, J. V. (2006). Multivariate garch models: a survey. *Journal of applied econometrics*, 21(1):79–109.

Bernardi, M. and Catania, L. (2016). Comparison of value-at-risk models using the mcs approach. *Computational Statistics*, pages 1–30.

Bernardi, M., Catania, L., and Petrella, L. (2016). Are news important to predict the value-at-risk? *The European Journal of Finance*, 0(0):1–38.

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

Bollerslev, T. (2008). Glossary to ARCH (GARCH). CREATES Research Papers 2008-49, School of Economics and Management, University of Aarhus.

Bollerslev, T., Engle, R. F., and Nelson, D. B. (1994). Arch models. *Handbook of econometrics*, 4:2959–3038.

Box, G. E. and Cox, D. R. (1964). An analysis of transformations. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 211–252.

Clark, T. E. and McCracken, M. W. (2001). Tests of equal forecast accuracy and encompassing for nested models. *Journal of econometrics*, 105(1):85–110.

Creal, D., Koopman, S. J., and Lucas, A. (2013). Generalized autoregressive score models with applications. *Journal of Applied Econometrics*, 28(5):777–795.

Diebold, F. X. and Lopez, J. A. (1996). Forecast evaluation and combination. In Maddala, G. and Rao, C., editors, *Handbook of Statistics*, volume 14, pages 241–268. North-Holland, Amsterdam.

Diebold, F. X. and Mariano, R. S. (2002). Comparing predictive accuracy. *Journal of Business & economic statistics*, 20(1).

Ding, Z., Granger, C. W., and Engle, R. F. (1993). A long memory property of stock market returns and a new model. *Journal of empirical finance*, 1(1):83–106.

Engle, R. F. (1982). Autoregressive conditional heteroscedasticity with estimates of the variance of united kingdom inflation. *Econometrica: Journal of the Econometric Society*, pages 987–1007.

Engle, R. F. and Lee, G. G. (1993). A permanent and transitory component model of stock return volatility. *University of California at San Diego, Economics Working Paper Series.*

Engle, R. F., Lilien, D. M., and Robins, R. P. (1987). Estimating time varying risk premia in the term structure: The arch-m model. *Econometrica: Journal of the Econometric Society*, pages 391–407.

Engle, R. F. and Manganelli, S. (2004). Caviar. *Journal of Business & Economic Statistics*, 22(4):367–381.

Fernández, C. and Steel, M. F. (1998). On bayesian modeling of fat tails and skewness. *Journal of the American Statistical Association*, 93(441):359–371.

Francq, C. and Zakoian, J.-M. (2011). *GARCH models: structure, statistical inference and financial applications*. John Wiley & Sons.

Ghalanos, A. (2014). *rugarch: Univariate GARCH models.* R package version 1.3-1.

Glosten, L. R., Jagannathan, R., and Runkle, D. E. (1993). On the relation between the expected value and the volatility of the nominal excess return on stocks. *The journal of finance*, 48(5):1779–1801.

González-Rivera, G., Lee, T.-H., and Mishra, S. (2004). Forecasting volatility: A reality check based on option pricing, utility function, value-at-risk, and predictive likelihood. *International Journal of Forecasting*, 20(4):629–645.

Hansen, P. R. (2005). A test for superior predictive ability. *Journal of Business & Economic Statistics*, 23(4).

Hansen, P. R. and Lunde, A. (2005). A forecast comparison of volatility models: does anything beat a garch (1, 1)? *Journal of applied econometrics*, 20(7):873–889.

Hansen, P. R., Lunde, A., and Nason, J. M. (2003). Choosing the best volatility models: The model confidence set approach. *Oxford Bulletin of Economics and Statistics*, 65(s1):839–861.

Hansen, P. R., Lunde, A., and Nason, J. M. (2011). The model confidence set. *Econometrica*, 79(2):453–497.

Harvey, A. C. (2013). *Dynamic Models for Volatility and Heavy Tails: With Applications to Financial and Economic Time Series*. Number 52. Cambridge University Press.

Higgins, M. L. and Bera, A. K. (1992). A class of nonlinear arch models. *International Economic Review*, pages 137–158.

Kilian, L. (1999). Exchange rates and monetary fundamentals: What do we learn from long-horizon regressions? *Journal of Applied Econometrics*, 14(5):491–510.

Lopez, J. A. (2001). Evaluating the predictive accuracy of volatility models. *Journal of Forecasting*, 20(2):87–109.

Nelson, D. B. (1991). Conditional Heteroskedasticity in Asset Returns: A New Approach. *Econometrica*, 59(2):347–70.

Rigby, R. and Stasinopoulos, D. (2005). Generalized additive models for location, scale and shape. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 54(3):507–554.

Schwert, G. (1990). Stock volatility and the crash of '87. *Review of Financial Studies*, 3(1):77–102.

Taylor, S. J. (1986). *Modelling Financial Times Series*. Wiley.

West, K. D. (1996). Asymptotic inference about predictive ability. *Econometrica: Journal of the Econometric Society*, pages 1067–1084.

White, H. (2000). A reality check for data snooping. *Econometrica*, 68(5):1097–1126.

Wuertz, D., with contribution from Michal Miklovic, Y. C., Boudt, C., Chausse, P., and others (2013). *fGarch: Rmetrics - Autoregressive Conditional Heteroskedastic Modelling*. R package version 3010.82.

Zakoian, J.-M. (1994). Threshold heteroskedastic models. *Journal of Economic Dynamics and control*, 18(5):931–955.