Université Paris I, Panthéon - Sorbonne

Master 2 M.O. 2020 – 2021

Time Series Tutorial n^0 3 : Parameter estimation of ARMA and GARCH processes, model selection, goodness-of-fit and forecasting

The aims of this tutorial is the computation of QMLE estimators of ARMA and GARCH processes, to use the BIC criterion for selecting an "optimal" GARCH process, to test the goodness-of-fit of the selected model and use this for forecasting the future values of the time series.

Yule-Walker and QMLE estimation of the parameters of an AR[1] process

In the sequel, we define and compute a natural estimator of the parameters of a AR[1] process, which is the Yule-Walker estimator. Hence, if $X_{t+1} = \theta\, X_t + \xi_t$ for $t \in \mathbf{Z}$, with $|\theta| < 1$, then $\theta = \mathrm{cov}(X_0, X_1)/\mathrm{var}(X_0)$. Hence, the Yule-Walker estimator of θ and σ_ξ^2 could be obtained by replacing "theoretical" covariances by empirical covariances:

$$\widehat{\theta} = \frac{\frac{1}{n}\sum_{i=1}^{n-1} X_i X_{i+1}}{\frac{1}{n}\sum_{i=1}^{n} X_i^2}.$$

Use the previous program for exhibiting the asymptotic behavior of $\widehat{\theta}$. Consider the cases $n = 100$, $n = 500$ and $n = 1000$. How to check the $\sqrt{n}$ convergence rate of this estimator? Write the Gaussian conditional log-likelihood of (X_t) and deduce the QMLE-estimators. Which differences with the previous one?

QMLE estimation of the parameters of an ARCH[1] process

In GARCH framework, the Yule-Walker estimation could not be used since the auto-covariance of the process are vanished. As a consequence we use a QMLE estimator for estimating the parameters, as with the following commands:

```
n=1000
spec = garchSpec(model = list(omega = 1, alpha = c(0.6), beta = 0))
X=garchSim(spec,n)
plot(X)
logLik=function(the){sum(log(the[1]+the[2]*X[1:(n-1)]^2)+X[2:n]^2/(the[1]+the[2]*X[1:(n-1)]^2))}
QMLE=optim(c(0.1,0.1),logLik,method = "L-BFGS-B",lower = c(0.01,0.01), upper = c(100,0.999))
QMLE$par
```

Explain all the commands. By repeating these commands draw an histogram drawed by $\widehat{\omega}$ and $\widehat{a}_1$. Which distribution could be suspected? Do the same thing for $n = 100$ and $n = 10000$. What is the evolution of the standard error of these estimators with n? Write a similar program for estimating the parameters of a GARCH[1,1] process, with $b_1 = 0.3$.

A numerically more accurate way for estimating these parameters can also be to use the following procedure:

```
QMLE2=garchFit(~ garch(1,0), data = X, trace = FALSE)
coef(QMLE2)
```

Do you find the same results? Note that this algorithm also provides an estimation of the standard deviation for each estimated parameter. Give it for all the parameters. By choosing different values of n remark how these standard deviations decrease with n.

Application to financial time series

Download the daily values of SP500 from November 28 2011 (see Yahoo Finance website for instance or JMB webpage). A `.csv` file is downloaded and you can read it with R software from the following command (change the directory!):

```
SP500=read.csv("C:/Users/.../TP/SP500.csv")
```

Then using the following commands, we consider the log-returns of the SP500 closing values and we estimate the parameters from a GARCH[1,1] model:

```
names(SP500)
X=SP500$Close
ts.plot(X)
n=length(X)
Y=log(X[2:n]/X[1:(n-1)])
ts.plot(Y)
QMLE3=garchFit(~ garch(1,1), data = Y, trace = FALSE)
coef(QMLE3)
```

What is the result? But there is still a question: why use a GARCH[1,1] instead of other GARCH[p,q]? More generally, how to chose a model and test its goodness-of-fit?

How to fit an optimal model of time series?

We are going back first to the commands `garchSim` and `garchFit` already used in TP4:

```
n=1000
spec = garchSpec(model = list(omega = 1, alpha = c(0.6), beta = c(0.2)))
X=garchSim(spec,n)
FitX=garchFit(~ garch(1,1), data = X, trace = FALSE)
summary(FitX)
```

A lot of informations are included in the fGarch object FitX. They can notably be adressed using the following commands:

```
FitX@fit$coef
FitX@fit$ics
BIC1=FitX@fit$ics[2]; BIC1
volatility(FitX)
```

A lot of other results could also be obtained... The BIC is computed, which may be compared to another possible model:

```
FitX2=garchFit(~ garch(2,1), data = X, trace = FALSE)
BIC2=FitX2@fit$ics[2]; BIC2
FitX3=garchFit(~ garch(1,0), data = X, trace = FALSE)
BIC3=FitX3@fit$ics[2]; BIC3
```

What is done? Which conclusions could you obtain? Could you generalize this method for determinating an optimal model for the BIC criterion (remark that GARCH(p, q) with $p = 0$ can not be consider as well as generic GARCH(p, q): the package should be imporved...)? Portemanteau tests are also computed:

```
summary(FitX)
```

Several Ljung-Box tests on residuals and squared residuals are aivalable, for 3 different values of lags. However those goodness-of-fit tests are not theoretically valid!! Use instead the test that is proposed to you from R programs.

Consider now the log-returns of the SP500 closing values. Find an optimal GARCH model (I found a GARCH$(2, 1)$...) for these data and test the goodness-of-fit of this model. Note also that ARMA-GARCH or APARCH or ARMA-APARCH processes could be used for fitting these model. Try with ARMA$(1, 1)$-GARCH$(2, 1)$. Is the BIC criterion smaller for this model than for the previous optimal model?

Forecast with a chosen model of time series

The package `fGarch` also allows to forecast future values for a time series once a model is chosen. For instance:

```
FitX=garchFit(~ garch(1,1), data = X, trace = FALSE)
predict(FitX,n.ahead=5)
FitX1=garchFit(~ arma(1,1)+garch(2,1), data = X, trace = FALSE)
predict(FitX1,n.ahead=5)
```

Observe and explain the differences of both this forecastings.
Consider the log-returns of the SP500 closing values and memorize the 10 last estimations of the volatility. Then drop these 10 last values, adjust an optimal model and forecast the 10 future values. Compare with the estimations.