Lecture Note of Bus 41202, Spring 2017: More Volatility Models. Mr. Ruey Tsay

Package Note: We use \texttt{fGarch} to estimate most volatility models, but will discuss the package \texttt{rugarch} later, which can be used to estimate GRACH-M, IGARCH, and EGARCH models.

The GARCH-M model

\[ r_t = \mu + c \sigma_t^2 + a_t, \quad a_t = \sigma_t \epsilon_t, \quad \sigma_t^2 = \alpha_0 + \alpha_1 a_{t-1}^2 + \beta_1 \sigma_{t-1}^2 \]

where \( c \) is referred to as the risk premium, which is expected to be positive.

Example: A GARCH(1,1)-M model for the monthly excess returns of S&P 500 index from January 1926 to December 1991. The fitted model is

\[ r_t = 4.22 \times 10^{-3} + 0.561 \sigma_t^2 + a_t, \quad \sigma_t^2 = 0.814 \times 10^{-5} + 0.122 a_{t-1}^2 + 0.854 \sigma_{t-1}^2. \]

Standard error of the estimated risk premium is 0.896 so that the estimate is not statistically significant at the usual 5% level.

\textbf{R demonstration}

\begin{verbatim}
> source("garchM.R")
> sp5=scan(file="sp500.txt")
> m1=garchM(sp5)
> maximizing log-likelihood: 1269.053

Coefficient(s):

| Parameter | Estimate   | Std. Error | t value | Pr(>|t|) |
|-----------|------------|------------|---------|----------|
| mu        | 0.00422469 | 0.0024067 | 1.75539 | 0.0791929 |
| gamma     | 0.5612977  | 0.896194  | 0.62631 | 0.5311105 |
| omega     | 0.00813623 | 0.0029209 | 2.78548 | 0.0053449 ** |
| alpha     | 0.121976   | 0.212373   | 5.50995 | 3.5893e-08 *** |
| beta      | 0.0954361  | 0.222612   | 38.43945 | < 2.22e-16 *** |
\end{verbatim}
Remarks: The R script `garchM` is relatively slow. The intensive computation is due to using a recursive loop to evaluate the likelihood function.

The EGARCH model

The idea (concept) of EGARCH model is useful. In practice, it is easier to use the TGARCH model.

Asymmetry in responses to past positive and negative returns:

\[ g(\epsilon_t) = \theta \epsilon_t + \gamma [\epsilon_t - E(|\epsilon_t|)], \]

with \( E[g(\epsilon_t)] = 0 \).

To see asymmetry of \( g(\epsilon_t) \), rewrite it as

\[
\begin{cases} 
(\theta + \gamma)\epsilon_t - \gamma E(|\epsilon_t|) & \text{if } \epsilon_t \geq 0, \\
(\theta - \gamma)\epsilon_t - \gamma E(|\epsilon_t|) & \text{if } \epsilon_t < 0.
\end{cases}
\]

An EGARCH\((m, s)\) model:

\[ a_t = \sigma_t \epsilon_t, \quad \ln(\sigma_t^2) = \alpha_0 + \frac{1 + \beta_1 B + \cdots + \beta_{s-1} B^{s-1}}{1 - \alpha_1 B - \cdots - \alpha_m B^m} g(\epsilon_{t-1}). \]

Some features of EGARCH models:

- uses log trans. to relax the positiveness constraint
- asymmetric responses

Consider an EGARCH\((1,1)\) model

\[ a_t = \sigma_t \epsilon_t, \quad (1 - \alpha B) \ln(\sigma_t^2) = (1 - \alpha)\alpha_0 + g(\epsilon_{t-1}), \]
Under normality, \( E(|\epsilon_t|) = \sqrt{2/\pi} \) and the model becomes

\[
(1 - \alpha B) \ln(\sigma_t^2) = \begin{cases} 
\alpha_* + (\theta + \gamma)\epsilon_{t-1} & \text{if } \epsilon_{t-1} \geq 0, \\
\alpha_* + (\theta - \gamma)\epsilon_{t-1} & \text{if } \epsilon_{t-1} < 0 
\end{cases}
\]

where \( \alpha_* = (1 - \alpha)\alpha_0 - \frac{\sqrt{2}}{\pi}\gamma \).

This is a nonlinear function similar to that of the threshold AR model of Tong (1978, 1990).

Specifically, we have

\[
\sigma_t^2 = \sigma_{t-1}^{2\alpha} \exp(\alpha_*) \begin{cases} 
\exp[(\theta + \gamma)\frac{a_{t-1}}{\sqrt{\sigma_{t-1}^2}}] & \text{if } a_{t-1} \geq 0, \\
\exp[(\theta - \gamma)\frac{a_{t-1}}{\sqrt{\sigma_{t-1}^2}}] & \text{if } a_{t-1} < 0. 
\end{cases}
\]

The coefs \( (\theta + \gamma) \) & \( (\theta - \gamma) \) show the asymmetry in response to positive and negative \( a_{t-1} \). The model is, therefore, nonlinear if \( \theta \neq 0 \). Thus, \( \theta \) is referred to as the leverage parameter.

Focus on the function \( g(\epsilon_{t-1}) \). The leverage parameter \( \theta \) shows the effect of the sign of \( a_{t-1} \) whereas \( \gamma \) denotes the magnitude effect.

See Nelson (1991) for an example of EGARCH model.

**Another example**: Monthly log returns of IBM stock from January 1926 to December 1997 for 864 observations.

For textbook, an AR(1)-EGARCH(1,1) is obtained (RATS program):

\[
\begin{align*}
    r_t &= 0.0105 + 0.092r_{t-1} + a_t, \quad a_t = \sigma_t\epsilon_t \\
    \ln(\sigma_t^2) &= -5.496 + \frac{g(\epsilon_{t-1})}{1 - .856B}, \\
    g(\epsilon_{t-1}) &= -.0795\epsilon_{t-1} + .2647[|\epsilon_{t-1}| - \sqrt{2/\pi}],
\end{align*}
\]
Model checking:
For $\tilde{a}_t$: $Q(10) = 6.31(0.71)$ and $Q(20) = 21.4(0.32)$
For $\tilde{a}_t^2$: $Q(10) = 4.13(0.90)$ and $Q(20) = 15.93(0.66)$

Discussion:
Using $\sqrt{2/\pi} \approx 0.7979 \approx 0.8$, we obtain
$$\ln(\sigma_t^2) = -1.0 + 0.856 \ln(\sigma_{t-1}^2) + \begin{cases} 
0.1852 \epsilon_{t-1} & \text{if } \epsilon_{t-1} \geq 0 \\
-0.3442 \epsilon_{t-1} & \text{if } \epsilon_{t-1} < 0.
\end{cases}$$

Taking anti-log transformation, we have
$$\sigma_t^2 = \sigma_{t-1}^{2 \times 0.856} e^{-1.001 \times} \begin{cases} 
e^{0.1852 \epsilon_{t-1}} & \text{if } \epsilon_{t-1} \geq 0 \\
-0.3442 \epsilon_{t-1} & \text{if } \epsilon_{t-1} < 0.
\end{cases}$$

For a standardized shock with magnitude 2, (i.e. two standard deviations), we have
$$\frac{\sigma_t^2(\epsilon_{t-1} = -2)}{\sigma_t^2(\epsilon_{t-1} = 2)} = \frac{\exp[-0.3442 \times (-2)]}{\exp(0.1852 \times 2)} = e^{0.318} = 1.374.$$ 

Therefore, the impact of a negative shock of size two-standard deviations is about 37.4% higher than that of a positive shock of the same size.

Forecasting: some recursive formula available

**Another parameterization** of EGARCH models
$$\ln(\sigma_t^2) = \alpha_0 + \alpha_1 \frac{|a_{t-1}| + \gamma_1 a_{t-1}}{\sigma_{t-1}} + \beta_1 \ln(\sigma_{t-1}^2),$$
where $\gamma_1$ denotes the leverage effect.

Below, I re-analyze the IBM log returns by extending the data to December 2009. The sample size is 1008.
The fitted model is
\[ r_t = 0.012 + a_t, \quad a_t = \sigma_t \epsilon_t \]
\[ \ln(\sigma_t^2) = -0.611 + \frac{0.231|a_{t-1}| - 0.250a_{t-1}}{\sigma_{t-1}} + 0.92 \ln(\sigma_{t-1}^2). \]

Since EGARCH and TGARCH (below) share similar objective and the latter is easier to estimate. We shall use the TGARCH model.

**The Threshold GARCH (TGARCH) or GJR Model**

A TGARCH\((s, m)\) or GJR\((s, m)\) model is defined as
\[ r_t = \mu_t + a_t, \quad a_t = \sigma_t \epsilon_t, \quad \epsilon_t \sim D(0, 1) \]
\[ \sigma_t^\delta = \omega + \sum_{i=1}^{s} \alpha_i (|a_{t-i}| - \gamma_i a_{t-i})^\delta + \sum_{j=1}^{m} \beta_j \sigma_{t-j}^\delta, \]
where \(N_{t-i}\) is an indicator variable such that
\[ N_{t-i} = \begin{cases} 1 & \text{if } a_{t-i} < 0, \\ 0 & \text{otherwise}. \end{cases} \]

One expects \(\gamma_i\) to be positive so that prior negative returns have higher impact on the volatility.

**The Asymmetric Power ARCH (APARCH) Model.**

This model was introduced by Ding, Engle and Granger (1993) as a general class of volatility models. The basic form is
\[ r_t = \mu_t + a_t, \quad a_t = \sigma_t \epsilon_t, \quad \epsilon_t \sim D(0, 1) \]
\[ \sigma_t^\delta = \omega + \sum_{i=1}^{s} \alpha_i (|a_{t-i}| - \gamma_i a_{t-i})^\delta + \sum_{j=1}^{m} \beta_j \sigma_{t-j}^\delta, \]
where \(\delta\) is a non-negative real number. In particular, \(\delta = 2\) gives rise to the TGARCH model and \(\delta = 0\) corresponds to using \(\log(\sigma_t)\).
Theoretically, one can use any power $\delta$ to obtain a model. In practice, two things deserve further consideration. First, $\delta$ will also affect the specification of the mean equation, i.e., model for $\mu_t$. Second, it is hard to interpret $\delta$, except for some special values such as 0, 1, 2.

In R, one can fix the value of $\delta$ a priori using the subcommand `include.delta=F, delta = 2`.

Here I pre-fix $\delta = 2$. Thus, we can use APARCH model to estimate TGARCH model. Consider the percentage log returns of monthly IBM stock from 1926 to 2009.

**R demonstration**

```r
> da=read.table("m-ibm2609.txt",header=T)
> head(da)
   date  ibm
1 19260130 -0.010381
  ....
6 19260630  0.068493
> ibm=log(da$ibm+1)*100
> m1=garchFit(~aparch(1,1),data=ibm,trace=F,delta=2,include.delta=F)
> summary(m1)
Title:               GARCH Modelling
Call:              garchFit(formula = ~aparch(1, 1), data = ibm, delta = 2, include.delta = F,
                           trace = F)

Mean and Variance Equation:
   data ~ aparch(1, 1)
   [data = ibm]

Conditional Distribution:  norm

Coefficient(s):
   mu  omega   alpha1   gamma1   beta1
1.18659  4.33663  0.10767  0.22732  0.79468

Std. Errors: based on Hessian
```
Error Analysis:

|        | Estimate | Std. Error | t value | Pr(>|t|) |
|--------|----------|------------|---------|----------|
| mu     | 1.18659  | 0.20019    | 5.927   | 3.08e-09 *** |
| omega  | 4.33663  | 1.34161    | 3.232   | 0.00123 **  |
| alpha1 | 0.10767  | 0.02548    | 4.225   | 2.39e-05 *** |
| gamma1 | 0.22732  | 0.10018    | 2.269   | 0.02326 *    |
| beta1  | 0.79468  | 0.04554    | 17.449  | < 2e-16 ***  |

---

Log Likelihood:
-3329.177 normalized: -3.302755

Standardised Residuals Tests:

<table>
<thead>
<tr>
<th>Test</th>
<th>Statistic</th>
<th>p-Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jarque-Bera Test</td>
<td>R Chi^2</td>
<td>67.07416</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2.775558e-15</td>
</tr>
<tr>
<td>Shapiro-Wilk Test</td>
<td>R W</td>
<td>0.9870142</td>
</tr>
<tr>
<td></td>
<td></td>
<td>8.597234e-08</td>
</tr>
<tr>
<td>Ljung-Box Test R Q(10)</td>
<td>16.90603</td>
<td>0.07646942</td>
</tr>
<tr>
<td>Ljung-Box Test R Q(15)</td>
<td>24.19033</td>
<td>0.06193099</td>
</tr>
<tr>
<td>Ljung-Box Test R Q(20)</td>
<td>31.89097</td>
<td>0.04447407</td>
</tr>
<tr>
<td>Ljung-Box Test R^2 Q(10)</td>
<td>4.591691</td>
<td>0.9167342</td>
</tr>
<tr>
<td>Ljung-Box Test R^2 Q(15)</td>
<td>11.98464</td>
<td>0.6801912</td>
</tr>
<tr>
<td>Ljung-Box Test R^2 Q(20)</td>
<td>14.79531</td>
<td>0.7879979</td>
</tr>
<tr>
<td>LM Arch Test R TR^2</td>
<td>7.162971</td>
<td>0.8466584</td>
</tr>
</tbody>
</table>

Information Criterion Statistics:

<table>
<thead>
<tr>
<th>Criteria</th>
<th>AIC</th>
<th>BIC</th>
<th>SIC</th>
<th>HQIC</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>6.615430</td>
<td>6.639814</td>
<td>6.615381</td>
<td>6.624694</td>
</tr>
</tbody>
</table>

> plot(m1) <= shows normal distribution is not a good fit. 

> m1=garchFit(~aparch(1,1),data=ibm,trace=F,delta=2,include.delta=F,cond.dist="std")
> summary(m1)

Title:
GARCH Modelling
Call:
garchFit(formula = ~aparch(1, 1), data = ibm, delta = 2, cond.dist = "std",
include.delta = F, trace = F)

Mean and Variance Equation:
data ~ aparch(1, 1)

Conditional Distribution: std

Coefficient(s):

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>mu</td>
<td>1.20476</td>
<td>omega</td>
<td>3.98975</td>
<td>alpha1</td>
<td>0.10468</td>
</tr>
<tr>
<td></td>
<td>0.82732</td>
<td></td>
<td>0.22366</td>
<td></td>
<td>0.80711</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>6.67329</td>
</tr>
</tbody>
</table>
Std. Errors: based on Hessian

Error Analysis:

|    | Estimate | Std. Error | t value | Pr(>|t|)   |
|----|----------|------------|---------|-----------|
| mu | 1.20476  | 0.18715    | 6.437   | 1.22e-10  *** |
| omega | 3.98975 | 1.45331    | 2.745   | 0.006046  ** |
| alpha1 | 0.10468 | 0.02793    | 3.747   | 0.000179  *** |
| gamma1 | 0.22366 | 0.11595    | 1.929   | 0.053738  . |
| beta1 | 0.80711  | 0.04825    | 16.727  | < 2e-16   *** |
| shape | 6.67329  | 1.32779    | 5.026   | 5.01e-07  *** |

---

Log Likelihood:
-3310.21 normalized: -3.283938

Standardised Residuals Tests:

<table>
<thead>
<tr>
<th>Test</th>
<th>Statistic</th>
<th>p-Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jarque-Bera Test</td>
<td>R Chi^2</td>
<td>67.82336</td>
</tr>
<tr>
<td>Shapiro-Wilk Test</td>
<td>R W</td>
<td>0.9869698</td>
</tr>
<tr>
<td>Ljung-Box Test</td>
<td>R Q(10)</td>
<td>16.91352</td>
</tr>
<tr>
<td></td>
<td>R Q(15)</td>
<td>24.08691</td>
</tr>
<tr>
<td></td>
<td>R Q(20)</td>
<td>31.75305</td>
</tr>
<tr>
<td></td>
<td>R^2 Q(10)</td>
<td>4.553248</td>
</tr>
<tr>
<td></td>
<td>R^2 Q(15)</td>
<td>11.66891</td>
</tr>
<tr>
<td></td>
<td>R^2 Q(20)</td>
<td>14.18533</td>
</tr>
<tr>
<td>LM Arch Test</td>
<td>R TR^2</td>
<td>6.771675</td>
</tr>
</tbody>
</table>

Information Criterion Statistics:

<table>
<thead>
<tr>
<th>Statistic</th>
<th>AIC</th>
<th>BIC</th>
<th>SIC</th>
<th>HQIC</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>6.579782</td>
<td>6.609042</td>
<td>6.579711</td>
<td>6.590898</td>
</tr>
</tbody>
</table>

> plot(m1)

Make a plot selection (or 0 to exit):

1: Time Series
2: Conditional SD
3: Series with 2 Conditional SD Superimposed
4: ACF of Observations
5: ACF of Squared Observations
6: Cross Correlation
7: Residuals
8: Conditional SDs
9: Standardized Residuals
10: ACF of Standardized Residuals
11: ACF of Squared Standardized Residuals
12: Cross Correlation between r^2 and r
13: QQ-Plot of Standardized Residuals
### The APARCH model with \( \delta = 2 \) can also be fitted by using the subcommand ‘leverage=T’ as below:

```r
mm <- garchFit(~garch(1,1), data=ibm, trace=F, leverage=T)
```

For the percentage log returns of IBM stock from 1926 to 2009, the fitted GJR model is

\[
\begin{align*}
    r_t &= 1.20 + a_t, \quad a_t = \sigma_t \epsilon_t, \quad \epsilon_t \sim t^{*}_{6.67} \\
    \sigma_t^2 &= 3.99 + 0.105(|a_{t-1}| - 0.224a_{t-1})^2 + .807\sigma_{t-1}^2,
\end{align*}
\]

where all estimates are significant, and model checking indicates that the fitted model is adequate.

Note that, we can obtain the model for the log returns as

\[
\begin{align*}
    r_t &= 0.012 + a_t, \quad a_t = \sigma_t \epsilon_t \\
    \sigma_t^2 &= 3.99 \times 10^{-4} + 0.105(|a_{t-1}| - 0.224a_{t-1})^2 + .807\sigma_{t-1}^2.
\end{align*}
\]

The sample variance of the IBM log returns is about 0.005 and the empirical 2.5% percentile of the data is about \(-0.130\). If we use these two quantities for \( \sigma_{t-1}^2 \) and \( a_{t-1} \), respectively, then we have

\[
\begin{align*}
    \sigma_t^2(-) &= 0.0004 + 0.105(0.130 + 0.224 \times 0.130)^2 + 0.807 \times 0.005 \\
    \sigma_t^2(+) &= 0.0004 + 0.105(0.130 - 0.224 \times 0.130)^2 + 0.807 \times 0.005 \\
    &= 1.849.
\end{align*}
\]

In this particular case, the negative prior return has about 85% higher impact on the conditional variance.

**Stochastic volatility model**
Figure 1: Normal probability plot for TGARCH(1,1) model fitted to monthly percentage log returns of IBM stock from 1926 to 2009
Figure 2: QQ plot for TGARCH(1,1) model fitted to monthly percentage log returns of IBM stock from 1926 to 2009.
A (simple) SV model is

\[ a_t = \sigma_t \epsilon_t, \quad (1 - \alpha_1 B) \ln(\sigma_t^2) = \alpha_0 + v_t \]

where \( \epsilon_t \)'s are iid \( N(0,1) \), \( v_t \)'s are iid \( N(0, \sigma_v^2) \), \( \{\epsilon_t\} \) and \( \{v_t\} \) are independent.

**Estimation** of SV model. May use the R package *stochvol* to estimate SV models.

**Long-memory SV model**

A simple LMSV is

\[ a_t = \sigma_t \epsilon_t, \quad \sigma_t = \sigma \exp(u_t/2), \quad (1 - B)^d u_t = \eta_t \]

where \( \sigma > 0 \), \( \epsilon_t \)'s are iid \( N(0,1) \), \( \eta_t \)'s are iid \( N(0, \sigma_\eta^2) \) and independent of \( \epsilon_t \), and \( 0 < d < 0.5 \).

The model says

\[
\ln(a_t^2) = \ln(\sigma^2) + u_t + \ln(\epsilon_t^2) \\
= [\ln(\sigma^2) + E(\ln(\epsilon_t^2))] + u_t + [\ln(\epsilon_t^2) - E(\ln(\epsilon_t^2))] \\
\equiv \mu + u_t + \epsilon_t.
\]

Thus, the \( \ln(a_t^2) \) series is a Gaussian long-memory signal plus a non-Gaussian white noise; see Breidt, Crato and de Lima (1998).

**Application**

See Examples 3.4 & 3.5 of the textbook