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 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. For numerical stability, I use percentage returns. The fitted model is

\[ r_t = 0.743 + 0.048\sigma_t^2 + a_t, \quad \sigma_t^2 = 0.812 + 0.123a_{t-1}^2 + .854\sigma_{t-1}^2. \]

Std err of risk premium is 0.141 so that the estimate is not statistically significant at the usual 5% level.

**R demonstration**

```r
> source("garchM.R")
> mn=garchM(sp5*100)
```

```
1: 2399.822
0: 2380.0229 0.422452 0.00561297 0.806149 0.121976 0.854361
3: 2379.9525 0.427596 0.00652832 0.806107 0.121466 0.855756
6: 2378.4838 0.606807 0.0375055 0.801889 0.126193 0.846833
9: 2377.9587 0.673997 0.00166937 0.798191 0.125517 0.851915
12: 2377.8474 0.692209 0.0444464 0.802574 0.122278 0.854141
15: 2377.7922 0.742796 0.0480959 0.812187 0.122530 0.853507
```

Maximized log-likelihood: 2377.792

Coefficient(s):

|          | Estimate | Std. Error | t value | Pr(>|t|)   |
|----------|----------|------------|---------|-----------|
| mu       | 0.742796 | 0.1540336  | 4.82229 | 1.4192e-06 *** |
| gamma    | 0.0480959| 0.1408765  | 0.34140 | 0.7327991  |
| omega    | 0.8121873| 0.2858128  | 2.84168 | 0.0044877 ** |
| alpha    | 0.1225297| 0.0220596  | 5.55449 | 2.7843e-08 *** |
| beta     | 0.8535072| 0.0219079  | 38.95885| < 2.22e-16 *** |
Remarks: This R script is relatively slow. It takes longer time due to its use of recursive loop in evaluating likelihood function.

The EGARCH 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

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

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

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

Specifically, we have

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

The coeffs $(\theta + \gamma)$ & $(\theta - \gamma)$ 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):

$$r_t = 0.0105 + 0.092r_{t-1} + a_t, \quad a_t = \sigma_t \epsilon_t$$

$$\ln(\sigma_t^2) = -5.496 + \frac{g(\epsilon_{t-1})}{1 - .856B},$$

$$g(\epsilon_{t-1}) = -0.0795\epsilon_{t-1} + .2647[|\epsilon_{t-1}| - \sqrt{2/\pi}],$$

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 \[\sqrt{\frac{2}{\pi}} \approx 0.7979 \approx 0.8\], 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
\[
\sigma_t^2 = \sigma_{t-1}^2 \times 0.856 e^{-1.001} \times \begin{cases} 
 e^{0.1852 \epsilon_{t-1}} & \text{if } \epsilon_{t-1} \geq 0 \\
 e^{-0.3442 \epsilon_{t-1}} & \text{if } \epsilon_{t-1} < 0.
\end{cases}
\]

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
\[
\ln(\sigma_t^2) = \alpha_0 + \alpha_1 \frac{\vert a_{t-1} \vert + \gamma_1 a_{t-1}}{\sigma_{t-1}} + \beta_1 \ln(\sigma_{t-1}^2),
\]

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
\[
r_t = 0.012 + a_t, \quad a_t = \sigma_t \epsilon_t
\]
\[ \ln(\sigma^2_t) = -0.611 + \frac{0.231|a_{t-1}| - 0.250a_{t-1}}{\sigma_{t-1}} + 0.92\ln(\sigma^2_{t-1}). \]

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

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

A TGARCH\((s,m)\) or GJR\((s,m)\) model is defined as

\[
  r_t = \mu_t + a_t, \quad a_t = \sigma_t \epsilon_t \sigma^2_t = \alpha_0 + \sum_{i=1}^{s} (\alpha_i + \gamma_i N_{t-i})a_{t-i}^2 + \sum_{j=1}^{m} \beta_j \sigma^2_{t-j},
\]

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

\[
  r_t = \mu_t + a_t, \quad a_t = \sigma_t \epsilon_t \quad \epsilon_t \sim D(0, 1) \\
  \sigma^\delta_t = \omega + \sum_{i=1}^{s} \alpha_i (|a_{t-i}| - \gamma_i a_{t-i})^\delta + \sum_{j=1}^{m} \beta_j \sigma^\delta_{t-j} 
\]

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 \textit{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.

\textbf{R demonstration}

```r
> da=read.table("m-ibm2609.txt",header=T)
> head(da)
   date     ibm
1 19260130 -0.010381
2 19260228  0.060292
3 19260331 -0.012370
4 19260430  0.045390
5 19260529 -0.028814
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   **  |
### 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 2.775558e-15</td>
</tr>
<tr>
<td>Shapiro-Wilk Test</td>
<td>R W</td>
<td>0.9870142 8.597234e-08</td>
</tr>
<tr>
<td>Ljung-Box Test R Q(10)</td>
<td>16.90603 0.07646942</td>
<td></td>
</tr>
<tr>
<td>Ljung-Box Test R Q(15)</td>
<td>24.19033 0.06193099</td>
<td></td>
</tr>
<tr>
<td>Ljung-Box Test R Q(20)</td>
<td>31.89097 0.04447407</td>
<td></td>
</tr>
<tr>
<td>Ljung-Box Test R^2 Q(10)</td>
<td>4.591691 0.9167342</td>
<td></td>
</tr>
<tr>
<td>Ljung-Box Test R^2 Q(15)</td>
<td>11.98464 0.6801912</td>
<td></td>
</tr>
<tr>
<td>Ljung-Box Test R^2 Q(20)</td>
<td>14.79531 0.7879979</td>
<td></td>
</tr>
<tr>
<td>LM Arch Test R TR^2</td>
<td>7.162971 0.8466584</td>
<td></td>
</tr>
</tbody>
</table>

### Information Criterion Statistics:

<table>
<thead>
<tr>
<th>Statute</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)

[Data = ibm]

Conditional Distribution: std

Coefficient(s):

<table>
<thead>
<tr>
<th>mu</th>
<th>omega</th>
<th>alpha1</th>
<th>gamma1</th>
<th>beta1</th>
<th>shape</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.20476</td>
<td>3.98975</td>
<td>0.10468</td>
<td>0.22366</td>
<td>0.80711</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 1.887379e-15</td>
</tr>
<tr>
<td>Shapiro-Wilk Test</td>
<td>R W</td>
<td>0.9869698 8.212564e-08</td>
</tr>
<tr>
<td>Ljung-Box Test R Q(10)</td>
<td></td>
<td>16.91352 0.07629962</td>
</tr>
<tr>
<td>Ljung-Box Test R Q(15)</td>
<td></td>
<td>24.08691 0.06363224</td>
</tr>
<tr>
<td>Ljung-Box Test R Q(20)</td>
<td></td>
<td>31.75305 0.04600187</td>
</tr>
<tr>
<td>Ljung-Box Test R^2 Q(10)</td>
<td></td>
<td>4.553248 0.9189583</td>
</tr>
<tr>
<td>Ljung-Box Test R^2 Q(15)</td>
<td></td>
<td>11.66891 0.7038973</td>
</tr>
<tr>
<td>Ljung-Box Test R^2 Q(20)</td>
<td></td>
<td>14.18533 0.8209764</td>
</tr>
<tr>
<td>LM Arch Test R TR^2</td>
<td></td>
<td>6.771675 0.872326</td>
</tr>
</tbody>
</table>

Information Criterion Statistics:  

<table>
<thead>
<tr>
<th></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

Selection: 13

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

\[ r_t = 1.20 + a_t, \quad a_t = \sigma_t \varepsilon_t, \quad \varepsilon_t \sim t^*_{6.67} \]
\[ \sigma_t^2 = 3.99 + 0.105(\mid a_{t-1} \mid - 0.224a_{t-1})^2 + 0.807\sigma_{t-1}^2, \]

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

\[ r_t = 0.012 + a_t, \quad a_t = \sigma_t \varepsilon_t \]
\[ \sigma_t^2 = 3.99 \times 10^{-4} + 0.105(\mid a_{t-1} \mid - 0.224a_{t-1})^2 + 0.807\sigma_{t-1}^2. \]

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

\[
\frac{\sigma_t^2(-)}{\sigma_t^2(+)} = \frac{0.0004 + 0.105(0.130 + 0.224 \times 0.130)^2 + 0.807 \times 0.005}{0.0004 + 0.105(0.130 - 0.224 \times 0.130)^2 + 0.807 \times 0.005} = 1.849.
\]

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

**Stochastic volatility model**

A (simple) SV model is

\[ a_t = \sigma_t \varepsilon_t, (1 - \alpha_1 B - \cdots - \alpha_m B^m) \ln(\sigma_t^2) = \alpha_0 + v_t \]

where \( \varepsilon_t \)'s are iid \( N(0, 1) \), \( v_t \)'s are iid \( N(0, \sigma_v^2) \), \( \{\varepsilon_t\} \) and \( \{v_t\} \) are independent.
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.
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 \( N(0, \sigma^2_{\eta}) \) and independent of \( \epsilon_t, \) and \( 0 < d < 0.5. \)

The model says

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

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