Financial Econometrics

- Week 2, Lecture 1 -

ML Estimation for GARCH Models

VU Econometrics and Data Science

2024-2025

Paolo Gorgi

Today's class

1. ML estimation of GARCH models
   - Deriving the likelihood function
   - Asymptotic properties of the ML estimator
2. Practical implementation of Maximum Likelihood
   - Numerical optimization of the log-likelihood
   - Maximum Likelihood with R

Parameter estimation

Important: parameter values of GARCH models define stochastic properties of financial returns:

1. Temporal dependence in conditional variance;
2. Stationarity and unconditional distribution.
3. Conditional probabilities;

In practice: we do not know what is the *true parameter vector* $\theta_0 = (\omega, \alpha_1, ..., \alpha_q, \beta_1, ..., \beta_p)$ of the GARCH(p, q) that generated observed returns.

Solution: we can use the observed sample $y_1, y_2, ..., y_T$ to estimate the true parameter vector θ_0 by Maximum Likelihood.

ML estimation of GARCH models

Deriving the likelihood function (i)

Standard estimation method: *Maximum Likelihood*

Recall: *likelihood function = joint pdf* $p(y_1, \ldots, y_T; \theta)$

- The pdf is a function of data $y_1, ..., y_T$;
- The likelihood is a function of parameters $\theta = (\omega, \alpha_1, ..., \alpha_q, \beta_1, ..., \beta_p)$.

Recall: a joint pdf can *always* be factorized as

$$f(x, y) = f(x|y) \times f(y).$$

Intro to Time Series: this factorization is useful to write the likelihood function in terms of conditional densities!

Deriving the likelihood function (ii)

Factorizing the joint pdf...

$$p(y_1, \ldots, y_T; \theta) = p(y_2, \ldots, y_T | y_1; \theta) \times p(y_1; \theta).$$

$$p(y_2, \ldots, y_T | y_1; \theta) = p(y_3, \ldots, y_T | y_2, y_1; \theta) \times p(y_2 | y_1; \theta)$$

Together, this implies that

$$p(y_1, \ldots, y_T; \theta) = p(y_3, \ldots, y_T | y_2, y_1; \theta) \times p(y_2 | y_1; \theta) \times p(y_1; \theta).$$

Repeating this procedure: we write the **likelihood function** as

$$p(y_1, \ldots, y_T; \theta) = p(y_1; \theta) \prod_{t=2}^{T} p(y_t | y_{t-1}, \ldots, y_1; \theta).$$

The **log-likelihood function** is given by

$$\log p(y_1, \ldots, y_T; \theta) = \log p(y_1; \theta) + \sum_{t=2}^{T} \log p(y_t | y_{t-1}, \ldots, y_1; \theta).$$

Deriving the likelihood function (iii)

- **Question:** *why is this factorization useful?*

  Answer 1: joint pdf is very complicated or intractable!

  Answer 2: each conditional pdf is perfectly simple!

- **Example:** for ARCH and GARCH models we have seen that the distribution of $y_t|Y^{t-1}$ is Gaussian with mean zero and variance σ_t^2,

$$y_t|y_{t-1}, y_{t-2}, ... \sim N(0, \sigma_t^2).$$

  Hence: the conditional pdf is given by

$$p(y_t|y_{t-1}, y_{t-2}, ...; \theta) = \frac{1}{\sqrt{2\pi\sigma_t^2}} \exp\left\{-\frac{y_t^2}{2\sigma_t^2}\right\}.$$

Log-likelihood of the ARCH(1) model (i)

Example: ARCH(1) model

$$p(y_t|y_{t-1};\theta) = \frac{1}{\sqrt{2\pi(\omega+\alpha_1 y_{t-1}^2)}} \exp\left\{-\frac{y_t^2}{2(\omega+\alpha_1 y_{t-1}^2)}\right\}.$$

Hence: the log-likelihood function is given by

$$\begin{aligned}\log\left(p(y_1,\ldots,y_T;\theta)\right) &= \log\left(p(y_1;\theta)\right) + \sum_{t=2}^{T} \log\left(p(y_t|y_{t-1};\theta)\right) \\ &= \log\left(p(y_1;\theta)\right) - \frac{1}{2}\sum_{t=2}^{T}\left(\log(2\pi) + \log\left(\omega+\alpha_1 y_{t-1}^2\right) + \frac{y_t^2}{\omega+\alpha_1 y_{t-1}^2}\right).\end{aligned}$$

In practice: $p(y_1;\theta)$ is unknown and ignored!

Log-likelihood of the ARCH(1) model (ii)

Important: the constant term $\log(2\pi)$ can also be ignored!

Hence: the **simplified log-likelihood** takes the form

$$L(y_1, ..., y_T, \theta) = \sum_{t=2}^{T} l_t(\theta)$$

$$l_t(\theta) = -\frac{1}{2}\left(\log\left(\omega + \alpha_1 y_{t-1}^2\right) + \frac{y_t^2}{\omega + \alpha_1 y_{t-1}^2}\right).$$

Simplifications: Logarithmic transformation of the likelihood and adding constants to the log-likelihood does not change the optimization problem (i.e. the maximizer is the same, see next slide)

Maximum likelihood: simplifications

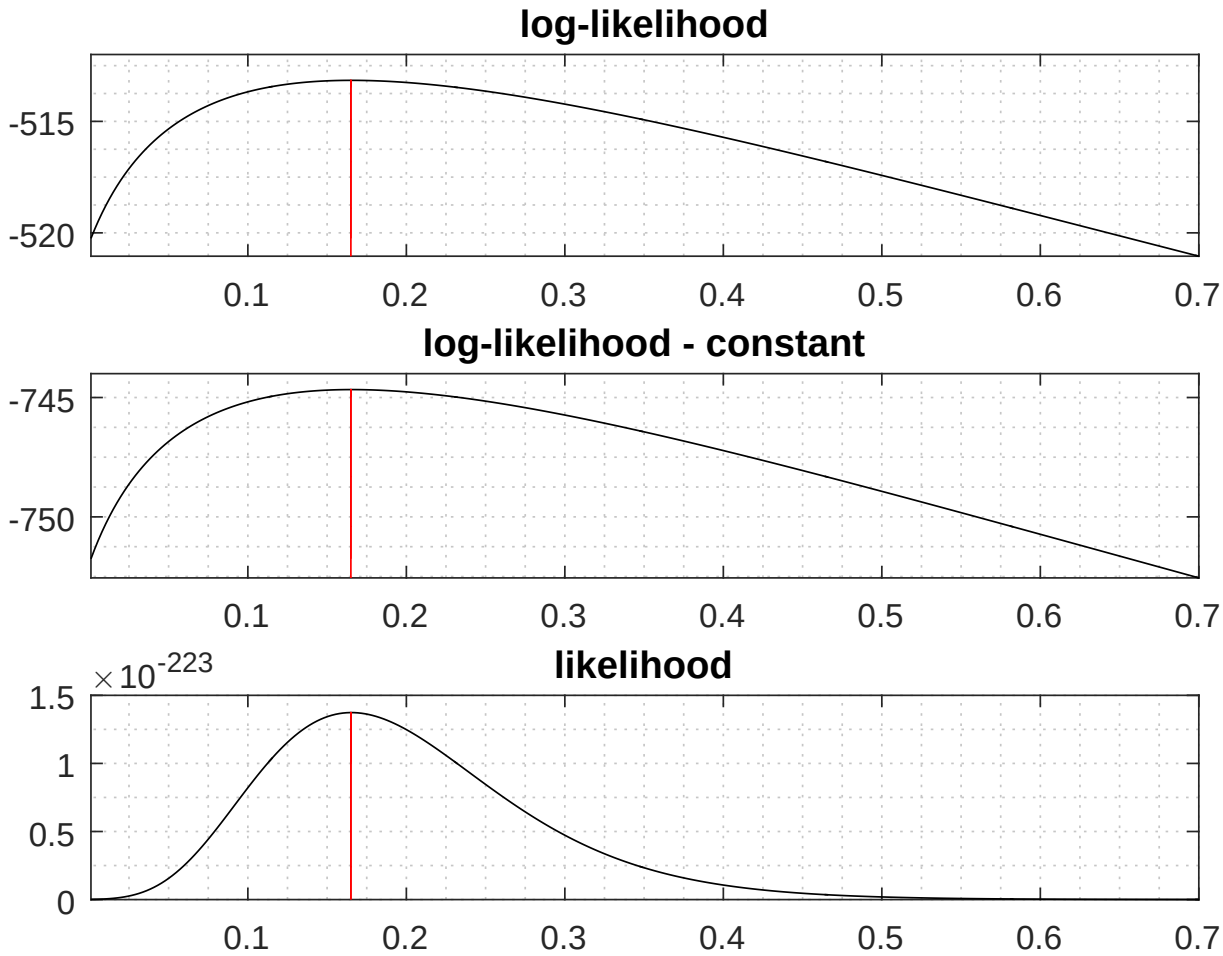

Figure: Likelihood functions for Apple daily log-returns for different values of α_1 with fixed $\omega = 2$.

Log-likelihood of the GARCH(1,1) model (i)

Another example: the GARCH(1,1) model

$$p(y_t|y_{t-1}, y_{t-2}, \ldots; \theta) = \frac{1}{\sqrt{2\pi\sigma_t^2}} \exp\left\{-\frac{y_t^2}{2\sigma_t^2}\right\}.$$

where: $\sigma_t^2 = \omega + \alpha_1 y_{t-1}^2 + \beta_1 \sigma_{t-1}^2.$

Hence: we have the log-likelihood function

$$\log\left(p(y_1, \ldots, y_T; \theta)\right) = \log\left(p(y_1; \theta)\right) + \sum_{t=2}^{T} \log\left(p(y_t|y_{t-1}, \ldots y_1; \theta)\right)$$

$$= \log\left(p(y_1; \theta)\right) - \frac{1}{2}\sum_{t=2}^{T}\left(\log(2\pi) + \log\sigma_t^2 + \frac{y_t^2}{\sigma_t^2}\right)$$

Log-likelihood of the GARCH(1,1) model (ii)

Simplified log-Likelihood:

$$L(y_1, ..., y_T, \theta) = \sum_{t=2}^{T} l_t(\theta) = \sum_{t=2}^{T} -\frac{1}{2}\left(\log \sigma_t^2 + \frac{y_t^2}{\sigma_t^2}\right),$$

where

$$\sigma_t^2 = \omega + \alpha_1 y_{t-1}^2 + \beta_1 \sigma_{t-1}^2.$$

Note: σ_1^2 must be fixed to some value!

Note: A good option is setting σ_1^2 equal to the sample variance.

Important: log-Likelihood is the same for GARCH(p, q)... only difference lies in updating equation!

Maximum Likelihood Estimator (MLE)

Recall: the MLE is defined as

$$\hat{\theta}_T = \arg\max_{\theta \in \Theta} L(y_1, ..., y_T, \theta).$$

Important: $L(y_1, ..., y_T, \theta)$ is a random function.

Every new realization $y_1, ..., y_T$ defines a new log-likelihood function to be maximized with respect to θ.

Hence: the MLE $\hat{\theta}_T$ is also a random variable!

Every new realization $y_1, ..., y_T$ defines a new point estimate of the true parameter θ_0.

MLE and asymptotic properties

Note: the MLE $\hat{\theta}_T$ is a continuous random variable.

Hence: there is zero probability that $\hat{\theta}_T = \theta_0$.

However: the MLE does have important properties!

Most important: the MLE $\hat{\theta}_T$ is *consistent and asymptotically normal* for θ_0.

Recall: $\hat{\theta}_T$ is *consistent* for θ_0 if $\hat{\theta}_T \xrightarrow{p} \theta_0$ as $T \to \infty$.

Recall: $\hat{\theta}_T$ is *asymptotically Normal* if $\sqrt{T}(\hat{\theta}_T - \theta_0) \xrightarrow{d} N(\mathbf{0}, \Omega)$ as $T \to \infty$.

Recall: Ω is called the *asymptotic variance* of $\hat{\theta}_T$.

Asymptotic distribution of the MLE

Lemma (Asymptotic distribution)

Under appropriate regularity conditions $\hat{\theta}_T$ *is consistent and asymptotically normal for* θ_0,

$$\hat{\theta}_T \xrightarrow{p} \theta_0 \quad as \quad T \to \infty,$$

$$and \quad \sqrt{T}(\hat{\theta}_T - \theta_0) \xrightarrow{d} N(\mathbf{0}, \Omega) \quad as \quad T \to \infty,$$

where $\Omega = \mathcal{I}(\theta_0)^{-1}$ *is the inverse Fisher Information matrix*

$$\mathcal{I}(\theta_0) = -\mathbb{E}\left(\frac{\partial^2 l_t(\theta)}{\partial\theta\partial\theta^\top}\right) = \frac{1}{2}\mathbb{E}\left(\frac{1}{\sigma_t^4}\frac{\partial\sigma_t^2}{\partial\theta}\frac{\partial\sigma_t^2}{\partial\theta^\top}\right).$$

Derivative process

The Fisher information $\mathcal{I}(\theta_0)$ depends on the derivative process $\frac{\partial \sigma_t^2}{\partial \theta}$;

Note that: the random sequence $\frac{\partial \sigma_t^2}{\partial \theta}$ is obtained by its own updating equation!

Example: Fort the GARCH(1,1) model we have

$$\frac{\partial \sigma_t^2}{\partial \theta} = \left[\begin{array}{ccc} \frac{\partial \sigma_t^2}{\partial \omega} & \frac{\partial \sigma_t^2}{\partial \alpha} & \frac{\partial \sigma_t^2}{\partial \beta} \end{array} \right]^\top,$$

where

$$\frac{\partial \sigma_t^2}{\partial \omega} = \frac{\partial \omega}{\partial \omega} + \frac{\partial \alpha y_{t-1}^2}{\partial \omega} + \frac{\partial \beta \sigma_{t-1}^2}{\partial \omega} = 1 + 0 + \beta \frac{\partial \sigma_{t-1}^2}{\partial \omega}$$

$$\frac{\partial \sigma_t^2}{\partial \alpha} = \frac{\partial \omega}{\partial \alpha} + \frac{\partial \alpha y_{t-1}^2}{\partial \alpha} + \frac{\partial \beta \sigma_{t-1}^2}{\partial \alpha} = 0 + y_{t-1}^2 + \beta \frac{\partial \sigma_{t-1}^2}{\partial \alpha}$$

$$\frac{\partial \sigma_t^2}{\partial \beta} = \frac{\partial \omega}{\partial \beta} + \frac{\partial \alpha y_{t-1}^2}{\partial \beta} + \frac{\partial \beta \sigma_{t-1}^2}{\partial \beta} = 0 + 0 + \sigma_{t-1}^2 + \beta \frac{\partial \sigma_{t-1}^2}{\partial \alpha}$$

GARCH(1,1): derivative process

Note: Taking all equations together, we obtain the following lemma

Lemma (Derivative process)

The conditional volatility derivative process $\{\partial\sigma_t^2/\partial\theta\}$ *of the GARCH(1,1) model satisfies the following updating equation*

$$\frac{\partial\sigma_t^2}{\partial\theta} = \begin{bmatrix} 1 \\ y_{t-1}^2 \\ \sigma_{t-1}^2 \end{bmatrix} + \beta\frac{\partial\sigma_{t-1}^2}{\partial\theta}.$$

Statistical inference (i)

Inference: for large T, the MLE $\hat{\theta}_T$ is:

- approximately Gaussian
- centered at the unknown θ_0
- with variance $\frac{1}{T}\,\mathcal{I}(\theta_0)^{-1}$ that vanishes to zero as $T \to \infty$

$$\sqrt{T}\Big(\hat{\theta}_T - \theta_0\Big) \overset{app}{\sim} N\Big(0\ ,\ \mathcal{I}(\theta_0)^{-1}\ \Big)$$

where $\overset{app}{\sim}$ denotes an 'approximate' distribution

As a result:

$$\hat{\theta}_T - \theta_0 \overset{app}{\sim} N\Big(0\ ,\ \frac{1}{T}\,\mathcal{I}(\theta_0)^{-1}\ \Big)$$

and

$$\hat{\theta}_T \overset{app}{\sim} N\Big(\theta_0\ ,\ \frac{1}{T}\,\mathcal{I}(\theta_0)^{-1}\ \Big).$$

Statistical inference (ii)

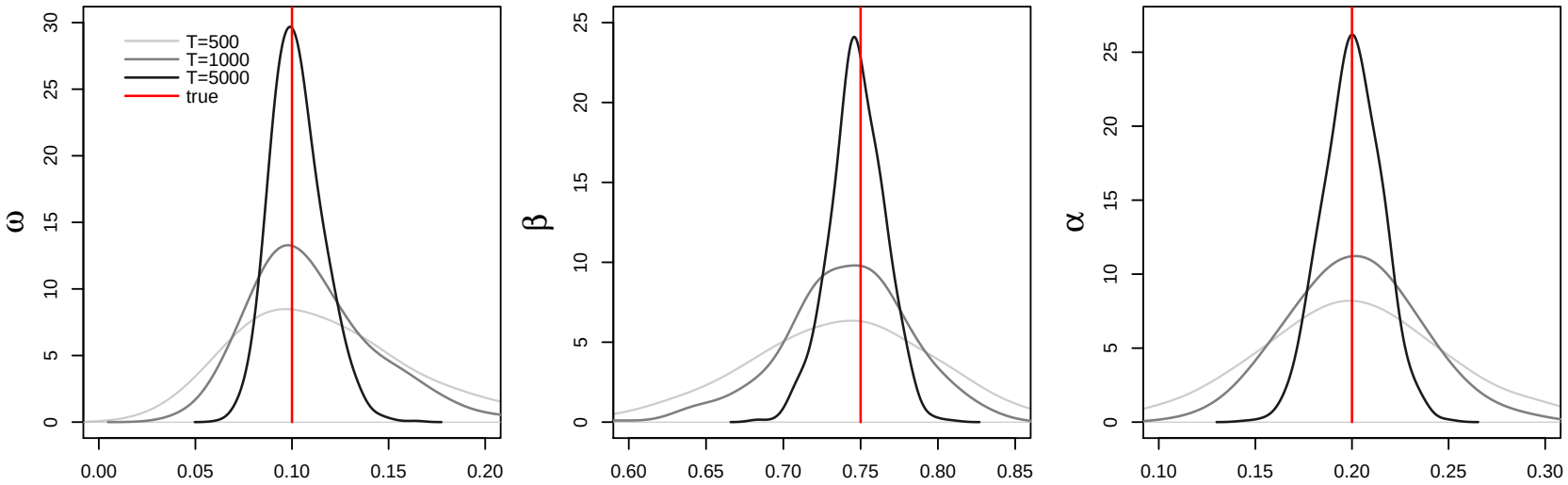

Figure: Distribution of the ML estimator $\hat{\theta}_T = (\hat{\omega}_T, \hat{\alpha}_T, \hat{\beta}_T)$ for different sample sizes T. The red line denotes the true value θ_0.

Statistical inference (iii)

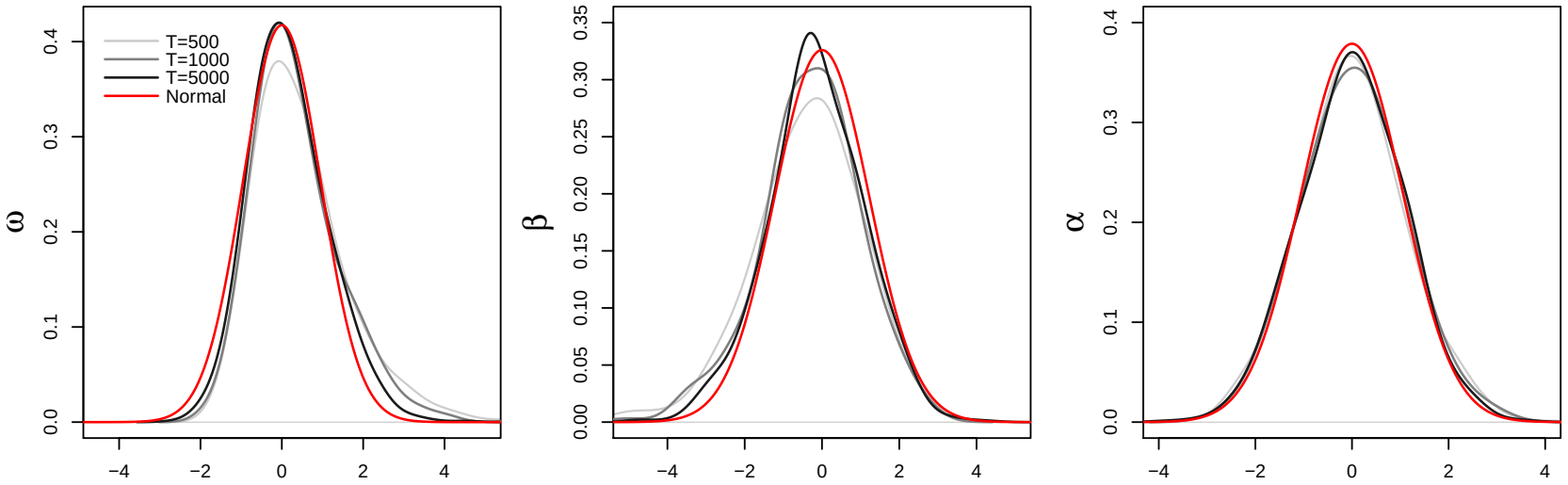

Figure: Distribution of $\sqrt{T}(\hat{\theta}_T - \theta_0)$ the for different sample sizes T. The red line denotes the normal density function.

Statistical inference in practice (i)

In practice: the Fisher information matrix $\mathcal{I}(\theta_0) = -\mathbb{E}(\partial^2 l_t(\theta)/\partial\theta\partial\theta^\top)$ is unknown since

- It depends on the unknown true parameter θ_0.
- The expectation $\mathbb{E}$ is unknown.

However: we can approximate $\mathcal{I}(\theta_0)$ by its *plug-in estimator*

$$\mathcal{I}(\theta_0) \;=\; -\mathbb{E}\left(\frac{\partial^2 l_t(\theta)}{\partial\theta\partial\theta^\top}\right) \;\approx\; -\frac{1}{T}\sum_{t=1}^{T}\frac{\partial^2 l_t(\hat{\theta}_T)}{\partial\theta\partial\theta^\top}.$$

Note 1: we replaced $\mathbb{E}$ by the sample average $1/T\sum_{t=1}^{T}$

Note 2: we replaced θ_0 by the sample estimate $\hat{\theta}_T$.

Statistical inference in practice (ii)

Final step: invert the estimate of $\mathcal{I}(\theta_0)$ and obtain an estimate of the asymptotic covariance matrix

$$\hat{\Omega} = \left(-\frac{1}{T} \sum_{t=1}^{T} \frac{\partial^2 l_t(\hat{\theta}_T)}{\partial \theta \partial \theta^\top} \right)^{-1}.$$

Note: we can use $\hat{\Omega}$ to:

1. Report standard errors;
2. Construct confidence intervals for θ_0;
3. Produce p-values.

Note: *this is all done as in your intro to statistics courses!*

Standard errors and confidence intervals

Standard errors of the ith element of the vector $\hat{\theta}_T$:

$$\text{SE}(\hat{\theta}_T^i) = \sqrt{\frac{1}{T}\hat{\Omega}_{ii}}$$

Approximate 95% confidence interval for the ith element of θ_0:

$$\left[\hat{\theta}_T^i - 1.96 \times \sqrt{\frac{1}{T}\hat{\Omega}_{ii}} \quad , \quad \hat{\theta}_T^i + 1.96 \times \sqrt{\frac{1}{T}\hat{\Omega}_{ii}}\right]$$

Practical implementation of Maximum Likelihood

Numerical optimization of the log-likelihood (i)

Question: how can we find the maximizer $\hat{\theta}_T$?

Intro Econometrics:

1. Take derivative of the log-likelihood;
2. Set derivative to zero;
3. Solve for $\hat{\theta}_T$

Example: linear Gaussian regression model

$$\text{For the model} \quad y_t = \beta x_t + \epsilon_t \quad \text{we obtained} \quad \hat{\beta} = \frac{\sum_{t=1}^{T} y_t x_t}{\sum_{t=1}^{T} x_t^2}$$

Example: Linear Gaussian AR(1) model

$$\text{For the model} \quad x_t = \rho x_{t-1} + \epsilon_t \quad \text{we obtained} \quad \hat{\rho} = \frac{\sum_{t=1}^{T} x_t x_{t-1}}{\sum_{t=1}^{T} x_{t-1}^2}$$

Numerical optimization of the log-likelihood (ii)

Problem: GARCH models are too complicated!

Problem: We cannot solve analytically for $\hat{\theta}_T$

Solution: We must proceed numerically!

Idea: We can evaluate the log-likelihood function for several values of θ and pick the one that attains the maximum value! (table below)

Of course: There are better methods! Newton-type algorithms evaluate the log-Likelihood sequentially and step in the most promising direction

Numerical optimization of the log-likelihood (iii)

Table: Log-likelihood function of the GARCH(1,1) model for daily Apple log-returns evaluated at different values of θ.

parameter value $\theta = (\omega, \alpha_1, \beta_1)$	log-lik value
$(0.30, 0.10, 0.70)$	-2962.5
$(0.20, 0.10, 0.70)$	-3090.0
$(0.20, 0.07, 0.70)$	-3211.4
$(0.20, 0.07, 0.80)$	-2941.6
$(0.20, 0.07, 0.85)$	**-2882.3**

The Newton-Raphson algorithm (i)

A simple and more efficient algorithm...

Definition: Newton-Raphson Algorithm

Starting from an initial value $\theta^{(1)}$*, update the parameter vector as follows*

$$\theta^{(k+1)} = \theta^{(k)} - \nabla L_T(\theta^{(k)}) \Big(\nabla^2 L_T(\theta^{(k)}) \Big)^{-1}$$

where $\nabla L_T(\theta^{(k)})$ *and* $\nabla^2 L_T(\theta^{(k)})$ *denote the Gradient vector and the Hessian matrix respectively*

$$\nabla L_T(\theta^{(k)}) = \frac{\partial L_T(\theta^{(k)})}{\partial \theta} \quad \text{and} \quad \nabla^2 L_T(\theta^{(k)}) = \frac{\partial^2 L_T(\theta^{(k)})}{\partial \theta \partial \theta^\top}.$$

The Newton-Raphson algorithm (ii)

Lemma (numerical optimization)

Under appropriate regularity conditions, the Newthon-Raphson algorithm converges to the MLE as the number of iterations k go to infinity; i.e. $\theta^{(k)} \to \hat{\theta}_T$ as $k \to \infty$.

Note: you can try to implement the Newton-Raphson and other algorithms in R to practice and have fun!

However: strictly speaking, you do not have to do this...

Efficient Newton-type algorithms are already implemented in most software packages

Parameter estimation with R (i)

Estimation with R: we first create an R *function* to evaluate the log-likelihood function of the GARCH(1,1).

```
llik_fun_GARCH <- function(par, x){
```

Note: We call this function `llik_fun_GARCH`

- `llik_fun_GARCH` takes as input a vector of data labeled `x` and a parameter vector labeled `par`
- `llik_fun_GARCH` returns as output the average log-likelihood

Note: You can find this function in the course's R folder!

Parameter estimation with R (ii)

First: We define the sample size and parameter values from the inputs `par` and `x`.

```
n <- length(x)
omega <- exp(par[1])
alpha <- exp(par[2])/(1+exp(par[2]))
beta <- exp(par[3])/(1+exp(par[3]))
```

Note that: The input vector `par` is transformed through link functions to impose some restriction on ω, α and β, which are useful to avoid numerical problems. In particular, the link function $\exp(p)$ ensures $\omega > 0$ and the logistic link function $\text{logistic}(p) = \exp(p)/(1+\exp(p))$ ensures $0 < \alpha, \beta < 1$.

Parameter estimation with R (iii)

Next: We obtain the conditional variance $\sigma_t^2(\theta)$ (labeled `sig2`) using a *for loop*

```
sig2 <- rep(0,n)
sig2[1] <- var(x)

for(t in 2:n){
   sig2[t] <- omega + alpha*x[t-1]^2 + beta*sig2[t-1]
}
```

Note that: The updating equation of $\sigma_t^2(\theta)$ is initialized using the sample variance. Alternative initializations may be considered.

Parameter estimation with R (iv)

Finally: We calculate the log-likelihood contribution of each observation y_t and the average log-likelihood value `llik`, which is returned as output of the R function.

```
l <- -(1/2)*log(2*pi) - (1/2)*log(sig2) - (1/2)*x^2/sig2
llik <- mean(l)
return(llik)
```

Note that: `l` is a vector containing the log-likelihood values $l_t(\theta)$ from $t = 1$ to $t = T$

Parameter estimation with R (v)

Stacking all the code together: we obtain the script

```
llik_fun_GARCH <- function(par,x){
  n <- length(x)
  omega <- exp(par[1])
  alpha <- exp(par[2])/(1+exp(par[2]))
  beta <- exp(par[3])/(1+exp(par[3]))
  sig2 <- rep(0,n)
  sig2[1] <- var(x)
  for(t in 2:n){
    sig2[t] <- omega + alpha*x[t-1]^2 + beta*sig2[t-1]
  }
  l <- -(1/2)*log(2*pi) - (1/2)*log(sig2) - (1/2)*x^2/sig2
  llik <- mean(l)
  return(llik)
}
```

Optimizing the likelihood function (i)

We are now ready to optimize the log-likelihood function!

Note: The code for estimation of GARCH(1,1) is available in the R file `Estimate_ML_GARCH.R`

First: We load the series of interest and store it in the object `x`.

Next: We define initial value for θ for the optimization

```
a <- 0.2
b <- 0.6
omega <- var(x)*(1-a-b)
par_ini <- c(log(omega),log(a/(1-a)),log(b/(1-b)))
```

Note: We set the initialization `par_ini` by using the inverse of the link functions we have used for the likelihood.

Optimizing the likelihood function (ii)

Finally: We obtain the point estimate $\hat{\theta}_T$ by optimizing the log-likelihood function.

Note: We use `optim()` to minimize the *negative* of the log-likelihood function. We set `par_ini` as initialization and we select the algorithm `BFGS`.

```
est <- optim(par=par_ini, fn=function(par) - llik_fun_GARCH(par,x),
             method = "BFGS")
```

The output of the optimization is stored in `est`.

Optim output

Note: The output of `optim()` includes

1. The point estimates $\hat{\theta}_T$ (`est$par`).
2. The value of negative average log-likelihood evaluated at $\hat{\theta}_T$ (`est$value`).
3. Exit flag that indicates if the optimization has been successful (`est$convergence`). The value zero indicates a successful optimization (see `help(optim)`).

Standard errors and confidence intervals (i)

We can obtain an estimate of the covariance matrix of the MLE $\hat{\Omega}$ from the hessian matrix of the average negative log-likelihood function (2nd derivative matrix).

Note: The function `Hess_fun_GARCH()` gives the average log likelihood without transforming the parameter input with the link functions (see the R file `Hess_fun_GARCH.R`). This is needed because we want the hessian with respect to the original parameters not through the link functions.

First: We obtain the hessian matrix using the R function `optimHess()` as follows:

```
hessian <- optimHess(par=theta_hat,
                     fn=function(par)-Hess_fun_GARCH(par,x))
```

Standard errors and confidence intervals (ii)

Next: We obtain $\hat{\Omega}$ by inverting the hessian of the average negative log-likelihood.

```
SIGMA <- solve(hessian)
```

Finally: We obtain a 95% level confidence interval for β.

```
lb_beta <- theta_hat[3]-1.96*sqrt(SIGMA[3,3])/sqrt(length(x))
ub_beta <- theta_hat[3]+1.96*sqrt(SIGMA[3,3])/sqrt(length(x))
ci_beta <- c(lb_beta, ub_beta)
```

Summary of the lecture

- The parameter vector $\theta_0 = (\omega, \alpha_1, ..., \alpha_q, \beta_1, ..., \beta_p)$ of GARCH models can be estimated by ML.
- The log-likelihood function can be obtained as the sum of the conditional log-densities.
- The MLE is consistent and asymptotically Normal with covariance matrix given by the inverse on the Fisher Information.
- In practice, the log-likelihood function needs to be optimized using numerical algorithms (such as Newton-Raphson). This can be done in R using `optim()`.