

Risk & Asset Allocation

Fall Assignment

John Dodson

October 1, 2014

This assignment is not a regular homework. It is worth half of the module grade for the fall term. A solution to this problem is due in NetFiles at the beginning of the session on Wednesday, October 15.

Your solution should include source code for programs and instructions for how to run them with the software available on Math Department computers, for example ‘shelby’ in VinH 314. I recommend you use `tar` or `zip` if you will be submitting several files.

I expect you to work alone and cite your sources. Please remind yourself of the University’s definition of scholastic dishonesty.

Problems

This problem will be based on the timeseries of daily log-returns (adjusted for splits and dividends) of Apple Inc. common equity shares, ticker AAPL, for two years up to October 1, 2014.

1. Estimate the quasi-MLE values of the five GJR-GARCH(1,1) model parameters as previously defined. **(20 points)**
2. Forecast the conditional mean and volatility of the AAPL log-return for October 2, 2014. **(10 points)**
3. Using re-simulation, estimate the standard error of the forecast above. **(40 points)**

Grading Rubric

Thirty out of one hundred points will be based on the follow criteria:

- You follow the instructions. **(5 points)**
- I can reproduce your results with the code and documentation you provide. **(10 points)**
- You include adequate citations. **(10 points)**
- Your write-up is clear and professional. **(5 points)**

Solution

As previously, I will use the API to fetch the data

```
raw=yahoo_prices({'AAPL'}, '2012-10-01', '2014-10-01');
tsc=resample(raw,raw.Time(~isnan(raw.AAPL.Data))); % remove holidays
ret=diff(log(tsc.AAPL.Data)); % log-returns
```

and first fit the geometric Brownian motion model.

```
theta_GBM=fminsearch(@(theta)-GJRGARCH(ret,[theta 0 0 0]),[0 var(ret)]);
[logLike_GBM foreVar_GBM foreMean_GBM]=GJRGARCH(ret,[theta_GBM 0 0 0]);
```

Next, we proceed with GARCH model starting with the GBM parameters.

```
theta_GARCH=fminsearch(@(theta)-GJRGARCH(ret,[theta theta(end)],logLike_GBM),[theta_GBM 0 0]);
[logLike_GARCH foreVar_GARCH foreMean_GARCH]=GJRGARCH(ret,[theta_GARCH theta_GARCH(end)]);
```

Finally, fit the GJR-GARCH model, starting with the GARCH parameters.

```
theta_MLE=fminsearch(@(theta)-GJRGARCH(ret,theta,logLike_GARCH),[theta_GARCH theta_GARCH(end)]);
[logLike foreVar foreMean devolResid]=GJRGARCH(ret,theta_MLE);
```

I checked several starting points, but seemed to consistently come back to the solution in Table 1.

MLE results	GBM	GARCH	GJR-GARCH
$\hat{\mu}$	0.000338	0.000225	0.000156
$\hat{\omega}$	0.000299	0.000200	0.000198
$\hat{\beta}$		0.0419	0.0547
$\hat{\alpha}^+$		0.428	0.336
$\hat{\alpha}^-$		0.428	0.480
$E[X_{T+1} \mathcal{F}_T]$	0.000188	0.000066	-0.000008
$\text{var}[X_{T+1} \mathcal{F}_T]$	0.000299	0.000318	0.000330
$\log f_{X \theta}(x)$	1324.5	1331.0	1331.1

Table 1: Timeseries model parameters for Apple Inc. daily log-returns from October 1, 2012 to October 1, 2014.

Notice that $\hat{\alpha}^- > \hat{\alpha}^+$, which is as we might expect. The β is relatively small and the ω is relatively large. In fact it seems that about half of the unconditional variance comes from the ARCH terms.

I consider this a “quasi-MLE” fit because the residuals seem to have much fatter tails than they should, which is apparent in the Q-Q plot in Figure 1. We should probably consider other models for the residuals, but we will not attempt to address this further here.

Let’s move on to the standard error estimations. In the GARCH setting, we can no longer consider our sample to be i.i.d., so we need to resort to simulation to estimate the standard error. I wrote the following function to form the simulated log-returns from i.i.d. normal variates:

```
function [x h]=GJRGARCH_sim(z,theta)
% simulate from the GJR-GARCH timeeries model based on randn array 'z'
x=nan(size(z));
h=theta(2)/(1-theta(3)-(theta(4)+theta(5))/2);
for i=1:length(x)
    x(i)=theta(1)-h/2+sqrt(h)*z(i);
    h=theta(2)+theta(3)*h+(theta(4)+theta(5)...
        +(theta(4)-theta(5))*sign(z(i)))/2*h*z(i)^2;
end
```

which I call from my script, `GJRGARCH_sim(randn(size(ret)),theta_MLE)`.

In estimating the Fisher information, I need to calculate the gradient of the log-likelihood at each of these simulations.

For the GBM model, we have the exact result for the gradient of the log-likelihood,

$$\frac{\partial}{\partial \mu} \log f_{X|\mu,\omega}(X) = \sum_{t=1}^T \left(\frac{X_t - \mu}{\omega} + \frac{1}{2} \right) \quad (\text{GBM})$$

$$\frac{\partial}{\partial \omega} \log f_{X|\mu,\omega}(X) = \sum_{t=1}^T \left(\frac{(X_t - \mu)^2}{2\omega^2} - \frac{1}{2\omega} - \frac{1}{8} \right)$$

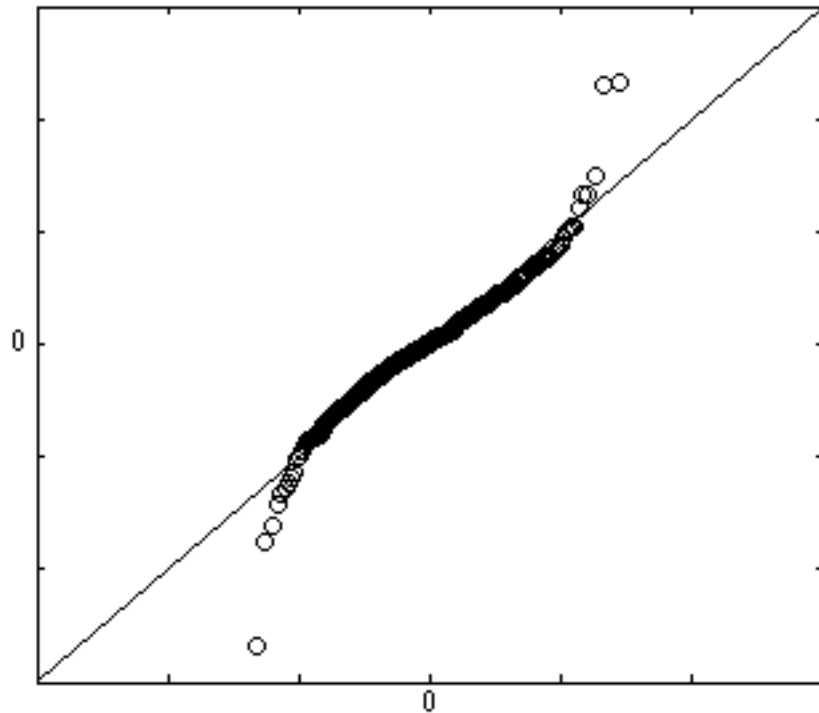


Figure 1: Normal Q-Q plot for devolatilized residuals showing obvious leptokurtosis.

For the more general GJR-GARCH, the analytic gradient is much harder to work with. I suggested you resort to a numerical approximation. To this end, I built a generic gradient calculator:

```
grad=@(f,theta)...
[ (f(theta+1E-6*[1 0 0 0 0])-f(theta-1E-6*[1 0 0 0 0]))/2E-6
  (f(theta+1E-6*[0 1 0 0 0])-f(theta-1E-6*[0 1 0 0 0]))/2E-6
  (f(theta+1E-4*[0 0 1 0 0])-f(theta-1E-4*[0 0 1 0 0]))/2E-4
  (f(theta+1E-3*[0 0 0 1 0])-f(theta-1E-3*[0 0 0 1 0]))/2E-3
  (f(theta+1E-3*[0 0 0 0 1])-f(theta-1E-3*[0 0 0 0 1]))/2E-3];
```

Note that this accepts any function of the GJR-GARCH parameters. This may not be necessary, but notice that I have also customized the step-sizes for the magnitudes of the parameter values.

I used 1000 draws to estimate the Fisher information.

```
logLike_grads=nan(1000,5);
for i=1:1000
    x=GJRGARCH_sim(randn(size(ret)),theta_MLE);
    logLike_grads(i,:)=grad(@(theta)GJRGARCH(x,theta),theta_MLE);
end
FisherInfo=cov(logLike_grads);
```

I did this several times and also tried 2000 draws, always getting similar results.

Consider first just the parameter standard errors:

```
theta_se=sqrt(diag(inv(FisherInfo)))';
```

$$\begin{aligned}\hat{\mu} &\approx 0.00016 \pm 0.00079 \\ \hat{\omega} &\approx 0.000198 \pm 0.000035 \\ \hat{\beta} &\approx 0.055 \pm 0.098 \\ \hat{\alpha}^+ &\approx 0.37 \pm 0.10 \\ \hat{\alpha}^- &\approx 0.48 \pm 0.12\end{aligned}$$

It would seem that $\hat{\omega}$, $\hat{\alpha}^+$, and $\hat{\alpha}^-$ are reliably non-zero, which $\hat{\mu}$ and $\hat{\beta}$ are not.

We can use the “Delta-rule” to estimate the standard error of the variance forecast. First, I have to write a helper-function to expose the forecast.

```
function foreVar = GJRGARCH_foreVar(x,theta)
[~,foreVar]=GJRGARCH(x,theta);
```

I apply it here:

```
foreVar_grad=grad(@(theta)GJRGARCH_foreVar(ret,theta),theta_MLE);
```

In these terms, the standard error is simply

```
foreVar_se=sqrt(foreVar_grad'/FisherInfo*foreVar_grad);
```

so¹

$$\hat{h}_{T+1} \approx 0.000330 \pm 0.000031$$

In comparison, for the geometric Brownian motion model I got

$$\hat{h}_{T+1} = \hat{\omega} \approx 0.000299 \pm 0.000018 \quad (\text{GBM})$$

Note that the simulation result for the standard error here is essentially identical to the theoretical result for the standard error of the MLE estimator for the variance of a normal sample,

$$\text{se } \hat{\sigma}^2 \approx \frac{\hat{\sigma}^2}{\sqrt{N/2}} \approx 0.000019$$

¹In daily volatility terms, this is $\sqrt{\hat{h}_{T+1}} \approx 0.0182 \pm 0.0009$.