Risk & Asset Allocation Assignment

John A. Dodson

October 12, 2011

This assignment will be worth half of your module grade for this semester. It is due in the Netfiles dropbox at 5:30 PM on October 26. You are welcome to provide code to support your work, but do not expect me to run it to generate your results. Results should be provided in text, Acrobat, Word, or Excel format.

Please work individually.

Problems

For this assignment, you will be working with the thirty components of the Dow Jones Industrial Average. You can read about these components on Yahoo! Finance under the symbol ^DJI.

1. Use two years of business-daily historical data to estimate the standard deviation of the value at the close of November 2, 2011 of one share invested in each of the Dow Jones Industrial Average components at the close of October 19, 2011. Assume that each value is lognormal and that the conditional variance of the daily returns is NGARCH(1,1).

   Make sure to project the one-day return variance forecasts to the ten-day investment horizon. Your results should be in units of dollars per share.

2. Use an estimator for Kendall's tau applied to the devolatized returns from above to fit the parameters of a Gaussian copula for the joint distribution of these thirty random variables.

Solution

In order to make the numerical estimator for the GARCH parameters more robust, as I recommended in class, I adapted my M-file from the previous homework for the conditional variances to accept $\log\alpha$ and $\log\beta$ instead of α and β since we only care about positive values for these parameters (γ could be negative). I also adapted it to replace a negative ω with `NaN` to prevent any negative or unreal conditional variances.

```
function [h omega forecast h0]=NGARCH2(epsilon,params)
% NGARCH(1,1) conditional variances
% params = [log(alpha) log(beta) gamma]
% epsilon(1) is the oldest residual
alpha=exp(params(1));
beta=exp(params(2));
```

```
gamma=params(3);
omega=mean(epsilon.^2)*(1-alpha*(1+gamma^2)-beta); % variance targeting
if omega<0
    omega=NaN;
end
epsilon=[epsilon;NaN]; % add on the next observation for forecasting
h=nan(size(epsilon));
h0=omega/(1-alpha*(1+gamma^2)-beta); % unconditional variance
h(1)=omega+h0*(alpha*gamma^2+beta);
% h0=omega/(1-alpha*gamma^2-beta); % assume h(0)=h(1)
% h(1)=h0;
for i=2:length(h)
    h(i)=omega+beta*h(i-1)+alpha*(epsilon(i-1)+gamma*sqrt(h(i-1)))^2;
end
forecast=h(end);
h(end)=[]; % remove forecast
```

Notice that I have included code for both the options for seeding h_0 here, although I have commented out the version based on letting $h_0 = h_1$. This version leads to somewhat different results.

This function usually returns just an array of conditional variances to correspond with the input array of residuals, but it also optionally returns the value for ω, the initial value h_0, and the one-step forecast h_{T+1}.

For the first part of my script for the solution, I load the data using a (slightly) modified version of `yahoo_prices()` (available on-line). It returns a timeseries collection of adjusted close prices and a structure of (unadjusted) final close prices[1].

I extract the column and row headers for my results in the variables `tickers` and `dates`. I exclude the index from the list of tickers, but I use it to re-sample the constituent prices in order to fill forward any missing values (presumably none).

```
%% load historical price data
[price last]=yahoo_prices({'^DJI'...
    'AA' 'AXP' 'BA' 'BAC' 'CAT' 'CSCO' 'CVX' 'DD' 'DIS' 'GE'...
    'HD' 'HPQ' 'IBM' 'INTC' 'JNJ' 'JPM' 'KFT' 'KO' 'MCD' 'MMM'...
    'MRK' 'MSFT' 'PFE' 'PG' 'T' 'TRV' 'UTX' 'VZ' 'WMT' 'XOM'},...
    '19-Oct-2009','19-Oct-2011');
tickers=gettimeseriesnames(price);
index_ticker=tickers{1};tickers(1)=[];
% remove holidays and fill in missing values based on the index
price=resample(price,price.Time(~isnan(price.(index_ticker).Data)));
dates=getabstime(price);
```

In the next part of the script, I set up the quasi-MLE objective function, a timeseries collection to hold the resulting conditional variances, and an array `z` to hold the standardized residuals.

Then I loop through the tickers and solve and store the MLE GARCH parameters, the corresponding optimal log-likelihood, the exit flag from `fminsearch()`, the timeseries of the conditional variances, and the array of the standardized residuals (in ticker and date order).

A the bottom of the loop, I use the MLE parameters and the one-day GARCH forecast to project the ten-day price standard deviation. First, I evaluate the ten-day log-return mean and variance using the techniques from our GARCH discussion. Then, I apply the "delta rule" to convert this into a price variance,

$$\operatorname{std} g(X) \approx \left| g'\left(\operatorname{E} X\right)\right| \sqrt{\operatorname{var} X}$$

[1] the previous version returned an array instead of a structure

where $g(X) = p_T e^X$.

Alternatively, I could have used the exact result for $\operatorname{var} e^X$ for $X \sim \mathsf{N}(\mu, \sigma^2)$ which is numerically similar for $\sigma \ll 1$.

```
%% fit GARCH, forecast variance, and standardize residuals
obj=@(h,epsilon)sum(log(2*pi*h)+epsilon.^2./h); % quasi-MLE objective
condvar=tscollection(dates,'Name','NGARCH conditional variance');
condvar.TimeInfo=price.TimeInfo;
z=[];
for ticker=tickers;j=ticker{:};
    y=log(price.(j).Data(2:end,:)...
        ./price.(j).Data(1:end-1,:)); % log total returns
    m=zeros(size(y)); % conditional means (zero)
    epsilon=y-m; % residuals
    [params val flag.(j)]=fminsearch(@(params)...
        obj(NGARCH2(epsilon,params),epsilon),[log(.1) log(.8) 0]);
    logL.(j)=-val; % log likelihood
    [h omega.(j) h1 h0]=NGARCH2(epsilon,params);
    condvar=addts(condvar,timeseries([h0;h],dates,'Name',j));
    alpha.(j)=exp(params(1));
    beta.(j)=exp(params(2));
    gamma.(j)=params(3);
    phi=alpha.(j)*(1+gamma.(j)^2)+beta.(j);
    varX=(1-phi^10)/(1-phi)*(h1-omega.(j)/(1-phi))...
        +10*omega.(j)/(1-phi); % ten-day log-return variance forecast
    stdP.(j)=last.(j)*sqrt(varX); % price standard deviation by Δ rule
    z=[z epsilon./sqrt(h)]; % standardized residuals
end
clear ticker j y m epsilon params val h h0 h1 phi varX
```

For the second question, I asked you to work with Kendall's tau two ways: firstly, with a sample estimator, and secondly with the population result for a gaussian copula.

$$\hat{\tau} = \binom{T}{2}^{-1} \sum_{1 \le j < k \le T} \operatorname{sgn}(z_k - z_j)' \operatorname{sgn}(z_k - z_j)$$

In the first part, I accumulated the concordances of the standardized returns for each pair of tickers for each pair of dates. I did this with a nested loop over the dates and an outer product over the tickers.

For more about this metric, see §5.5 in McNeil *et. al.*.

```
%% estimate concordances of standardized residuals
count=0;
tau=zeros(size(z,2));
for j=1:(size(z,1)-1)
    for k=(j+1):size(z,1)
        conc=sign(z(k,:)-z(j,:));
        tau=tau+conc'*conc;
        count=count+1;
    end
end
tau=tau/count;
tau=tau-diag(diag(tau))+eye(size(tau)); % adjust diagonal for ties
clear count j k conc
```

Notice that in the final expression for `tau`, I had to adjust for a small number of ties in which the concordance was neither +1 nor -1. It is reasonable to use a value of 0 for off-diagonal values of Kendall's tau in this case, but a random variable has concordance +1 with itself by definition.

Finally, we can use this estimate for Kendall's tau to calibrate our copula. To do this, we apply the population result

$$\tau = \frac{2}{\pi} \arcsin \rho$$

hence

$$\hat{\rho} \triangleq \sin \frac{\pi}{2} \hat{\tau}$$

component-wise.

```
%% identify with Kendall tau for a Gaussian copula
rho=sin(pi/2*tau);
```

One has to be a little careful with this approach, because $\hat{\rho}$ must be a positive-definite symmetric matrix (with unit diagonal). By construction, $\hat{\tau}$ is positive definite, but the component-wise transformation may destroy this property. I believe this issue is rare in practice, but McNeil *et. al.* does discuss how to handle it.

These M-files, along with a MAT file of the results, are available on-line.