Useful Toolkit for Trading & Risk Management

Hui Gong

University College London
Crescent Quant Co., Ltd
http://www.homepages.ucl.ac.uk/~ucahgon/

17 December, 2015

Financial Industry

- Finance: Putting the 'RIGHT' money in the 'RIGHT' place with the 'RIGHT' amount for the 'RIGHT' price.
- Financial Market
  - Based on Market levels
    - Primary Market
    - Secondary Market
  - Based on security types
    - Money Market (Purely Short-term Funds)
    - Capital Market (Equity Market/Debt Market)
    - Derivative Market
    - Financial Service Market (ATM, Credit Cards, Credit Rating, Stock Broking etc.)
    - Depository Market (Giving loans or purchasing other debt instruments such as treasure bills.)
    - Non-Depository market (Mutual Funds, Insurance Companies, Pension Funds, Brokerage Firms etc.)

Financial Products and its Stochastic Modelling

- Commodities [Ornstein-Uhlenbeck type models (with Jumps)]
- Currencies [Stochastic Volatility Models: The Heston Model, The Stein and Stein Model, Longstaff's double square root Model, Scott's exponential Ornstein-Uhlenbeck model, The SABR model]
- Indices/ETFs [GBM with Jump Diffusion Models, SV models: GARCH Models]
- Shares/Equities [Similar to Indices/ETFs]
- Treasuries/Bonds [Short-rate model: Merton's model, The Vasicek model, The Rendleman-Bartter model, The Cox-Ingersoll-Ross model, The Ho-Lee model, The Hull-White model, The Black-Derman-Toy model, The Black-Karasinski model, The Kalotay-Williams-Fabozzi model , The Longstaff-Schwartz model, The Chen model]

Equities/ETFs and their Derivatives

- Flow of funds (Assets/Liabilities)
  - What is the role of Investment Banks?
  - What is the role of Exchanges and Dealers?
  - What is the role of Hedge Funds/Asset Managers?
- Sell Side
  - Derivative(Option) Pricing & Risk Sensitivities
- Buy Side
  - Portfolios Optimisation

Downloading the data from Yahoo! Finance

- Yahoo! Finance
- Ticker Symbol/Stock Symbol
- Date, Open, High, Low, Close, Volume, Adj Close
- Start/End Date
- Try `price2ret` & `ret2price` !

```
% Connect to Yahoo! Finance.
c = yahoo;
% Obtain the adjusted closing price for the 'Apple Inc.' equity from 01/01/2007
     to 01/12/2015.
ClosePrice = fetch(c,'AAPL','Adj Close','01/01/2007','01/12/2015');
% Set the price equal to the adjusted closing price in the second column
% and flip array in up/down direction
Price = flipud(ClosePrice(:,2));
% Close Yahoo! connection.
close(c)
```

The Geometric Brownian Motion

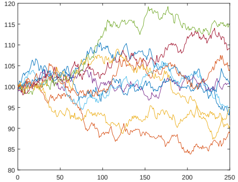

Figure: MATLAB Simulate GBM Sample Paths.

```
function S= GBM_simulation(N_Sim,T,dt,mu,sigma,S0)
mean=(mu-0.5*sigma^2)*dt/T;
S=S0*ones(N_Sim,T + 1);
BM=sigma*sqrt(dt/T)*normrnd (0,1,N_Sim,T);
S(:,2:end)=S0* exp(cumsum(mean+BM,2));

for i=1:N_Sim
    plot([0:dt:T],S(i,:));
    hold on;
end
hold off;
end

%S= GBM_simulation(10 , 250 , 1 , 0.01 , 0.1 , 100);
```

Parameter Estimation

- Time Series Model
- Autocorrelation function (ACF)
- Partial Auto-Correlation function (PACF)
- Maximum Likelihood Estimation

```
function [mu sigma]=GBM_calibration(Price,T,dt,params)
Ret=price2ret(Price);
n=length(Ret);
options=optimset ('MaxFunEvals',100000,'MaxIter',100000);
fminsearch(@normalLL,params,options);
% The MLE Function for iid Normal Distribution.
    function mll=normalLL (params)
        mu=params(1);
        sigma=abs(params(2));
        ll=n* log (1/ sqrt (2*pi* dt/T)/ sigma )+ sum(-(Ret -mu*dt/T).^2/2/(dt/T*sigma ^2));
        mll=-ll;
    end
end
% Initial params=[0 0];
```

Option Pricing via BSM model

- Pricing Equity Derivatives and Greeks Calculation

```
% To illustrate toolbox Black-Scholes functions (check assumptions)
[OptCall, OptPut] = blsprice(100, 95, 0.10, 0.25, 0.50, 0);
% Delta of a derivative security is the rate of change of its price relative to
% the price of the underlying asset.
[CallVal, PutVal] = blsdelta(100, 95, 0.10, 0.25, 0.50, 0);
% Gamma of a derivative security is the rate of change of delta relative to
% the price of the underlying asset.
GammaVal = blsgamma(100, 95, 0.10, 0.25, 0.50, 0);
% Vega is the rate of change in the price of a derivative security relative to
% the volatility of the underlying security.
VegaVal = blsvega(100, 95, 0.10, 0.25, 0.50, 0);
% Lambda, also known as the elasticity of an option, represents the percentage
    change
% in the price of an option relative to a 1% change in the price of the
    underlying security.
[LamCall, LamPut] = blslambda(100, 95, 0.10, 0.25, 0.50, 0);
% The implied volatility of an option is the standard deviation that makes
% an option price equal to the market price.
Volatility = blsimpv(100, 95, 0.10, 0.25, OptCall);
```

- Try the ‘AAPL’ data and price the one year option @ K=120.
- Check the value in Thomson Reuters and calculate the Implied Vol.

Portfolio Optimization and Performance Backtesting

- In this section, we are going to find out how to construct a optimal portfolio in a universe of 30 US stocks and backtest the performance of several portfolio strategies using historical data.
- The dataset 'equity_ dataset_ 30.csv' can be downloaded from http://www.homepages.ucl.ac.uk/~ucahgon/. It contains the daily closing prices (adjusted for stock splits and cash/stock dividends) for 30 blue-chip stocks over past 10 years. The dataset has 31 columns in total, with the first column being the date index in ISO format (yyyy-mm-dd) and the rest 30 columns containing price data for 30 stocks respectively.

Portfolio Optimization

Step 1 Import the csv data file 'equity_ dataset_ 30.csv' into Matlab, and extract the numeric price data into a variable named 'px_ mat'. 'px_ mat' should be a T-by- N matrix where T = 2641 and N = 30.

Step 2 Calculate the log return series for all 30 stocks according to formula $R_{i,t} = \log(P_{i,t}) - log(P_{i,t-1})$, where $i \in [1, 30]$ and $t \in [2, 2641]$.

Step 3 Split the whole sample period into In-Sample (training dataset, from 2005-01-01 to 2012-12-31) and Out-of-Sample (testing dataset, from 2013-01-01 to 2015-06- 30) periods. Use two variables 'ret_ mat_ is' and 'ret_ mat_ oos' to store the stock return matrix for In-Sample and Out-of-Sample periods respectively.

Step 4 Calculate the historical average daily return for each stock and the historical covariance matrix by using only the **In-Sample dataset**.

Data Preprocessing

```
% Portfolio Optimization and Performance Backtesting %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Step 1
% import the csv data file
Table = readtable('equity_dataset_30.csv','ReadVariableNames',true,'ReadRowNames',true);
% extract the numeric price data
px_mat = table2array(Table);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Step 2
% calculate the log return. Note: here T will change to 2640
ret_mat = diff(log(px_mat));
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Step 3
% calculate the in sample size of In-Sample
date = Table.Properties.RowNames;
in_length = length(find(datenum(date)<datenum('2013-01-01')));
% split the sample into In-Sample and Out-of-Sample, delete the first day
ret_mat_is = ret_mat(1:in_length-1,:);
ret_mat_oos = ret_mat((in_length):end,:);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Step 4
% calculate the In-Smaple mean
M_is = mean(ret_mat_is, 1);
% calculate the In-Smaple covariance
C_is = cov(ret_mat_is);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Portfolio Optimization and Performance Backtesting

Step 5 Consider the following 4 portfolios:

- Benchmark 1/N portfolio: allocate capital equally.
- Portfolio 1: Maximize Sharpe ratio (short-selling is allowed).
- Portfolio 2: Maximize Sharpe ratio (no short-selling).
- Portfolio 3: Minimize portfolio variance (short-selling is allowed).

Step 6 Now we have 4 portfolio trading strategies, that is we can allocate capital according to the optimal weights calculated in Step 5. Assume that we can buy/sell any fraction of shares and ignore the transaction cost associated with rebalancing portfolio daily. Backtest the strategy performances of benchmark 1/N portfolio and optimized portfolio 1, 2 and 3 using In-sample dataset. Construct and plot the equity curve and drawdown curve. Calculate annualized Sharpe Ratio and annualized Cumulative Average Return (CAR) for each strategy.

Step 7 Repeat the same backtesting process in Step 6 using out-of-sample dataset. Compare the out-of-sample performance with in-sample performance for each portfolio strategy.

Portfolio Optimization (1)

```
% Step 5
% Benchmark 1/N portfolio
w0 = repmat(1/length(M_is),length(M_is),1);
% Portfolio 1: Maximize Sharpe ratio
% short-selling is allowed, i.e. weights can be negative
% constraint: sum of weights is 1
Aeq = ones(1,length(M_is)); beq = 1;
% suppress optimization message
options = optimset('Display', 'off');
% NOTE: use - to convert max optimization to a min optimization problem
w1 = fmincon(@(w)-compute_sharpe(w, M_is*252, C_is*252), w0, [], [], Aeq, beq,
     [], [], [], options);
sr1 = compute_sharpe(w1, M_is*252, C_is*252);
fprintf('The maximized Sharpe Ratio is %.4f.\n', sr1);
% Portfolio 2: Maximize Sharpe ratio
% short-selling is not allowed, i.e. weights should be non-negative
% constraint: sum of weights is 1
Aeq = ones(1,length(M_is)); beq = 1;
% constraint: weights are non-negative
lb = zeros(1,30);
ub = ones(1,length(M_is));
% suppress optimization message
options = optimset('Display', 'off');
% NOTE: use - to convert max optimization to a min optimization problem
w2 = fmincon(@(w)-compute_sharpe(w, M_is*252, C_is*252), w0, [], [], Aeq, beq,
     lb, ub, [], options);
sr2 = compute_sharpe(w2, M_is*252, C_is*252);
fprintf('The maximized Sharpe Ratio is %.4f.\n', sr2);
```

Portfolio Optimization (2)

```
% Portfolio 3: Minimize portfolio variance
% short-selling is allowed, i.e. weights can be negative
% constraint: sum of weights is 1
Aeq = ones(1, length(M_is)); beq = 1;
% suppress optimization message
options = optimset('Display', 'off');
% column vector for initial weight
w3 = fmincon(@(w)compute_pvar(w, C_is*252), w0, [], [], Aeq, beq, [], [], [],
     options);
pvar = compute_pvar(w3, C_is*252);
fprintf('The minimum portfolio variance is %.4f.\n', sqrt(pvar));
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Step 6
% Aunualized Sharpe Ratio
sr0 = compute_sharpe(w0, M_is*252, C_is*252);
sr3 = compute_sharpe(w3, M_is*252, C_is*252);
Sharpe_Ratio_is = [sr0, sr1, sr2, sr3]
% Aunualized Cumulative Average Return (CAR), i.e. construct Equity Curve
car0 = cumprod(1 + ret_mat_is * w0);
car1 = cumprod(1 + ret_mat_is * w1);
car2 = cumprod(1 + ret_mat_is * w2);
car3 = cumprod(1 + ret_mat_is * w3);
CAR_is = [car0(end), car1(end), car2(end), car3(end)]
% construct Drawdown Curve
dd0 = drawdown(car0);
dd1 = drawdown(car1);
dd2 = drawdown(car2);
dd3 = drawdown(car3);
```

Performance Checking

```
% Plot the equity curve
% Setup the In-sample date
date_is = datenum(date(2:(in_length)));

% figure 1 Benchmark 1/N portfolio
figure(1);
subplot(2,1,1)
plot(date_is, car0)
datetick('x')
title('Equity Curve (Benchmark 1/N portfolio)')
subplot(2,1,2)
plot(date_is, dd0)
datetick('x')
title('Drawdown Curve')

% figure 2 Portfolio 1
figure(2);
subplot(2,1,1)
plot(date_is, car1)
datetick('x')
title('Equity Curve (Portfolio 1)')
subplot(2,1,2)
plot(date_is, dd1)
datetick('x')
title('Drawdown Curve')
```

```
% figure 3 Portfolio 2
figure(3);
subplot(2,1,1)
plot(date_is, car2)
datetick('x')
title('Equity Curve (Portfolio 2)')
subplot(2,1,2)
plot(date_is, dd2)
datetick('x')
title('Drawdown Curve')

% figure 4 Portfolio 3
figure(4);
subplot(2,1,1)
plot(date_is, car3)
datetick('x')
title('Equity Curve (Portfolio 3)')
subplot(2,1,2)
plot(date_is, dd3)
datetick('x')
title('Drawdown Curve')
% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Performance Backtesting

```
% Setp 7
% calculate the Out-of-Smaple mean
M_oos = mean(ret_mat_oos, 1);
% calculate the Out-of-Smaple covariance
C_oos = cov(ret_mat_oos);
% Aunualized Sharpe Ratio
sr0_oos = compute_sharpe(w0, M_oos*252, C_oos*252);
sr1_oos = compute_sharpe(w1, M_oos*252, C_oos*252);
sr2_oos = compute_sharpe(w2, M_oos*252, C_oos*252);
sr3_oos = compute_sharpe(w3, M_oos*252, C_oos*252);
Sharpe_Ratio_oos = [sr0_oos, sr1_oos, sr2_oos, sr3_oos]
% Aunualized Cumulative Average Return (CAR), i.e. construct Equity Curve
car0_oos = cumprod(1 + ret_mat_oos * w0);
car1_oos = cumprod(1 + ret_mat_oos * w1);
car2_oos = cumprod(1 + ret_mat_oos * w2);
car3_oos = cumprod(1 + ret_mat_oos * w3);
CAR_oos = [car0_oos(end), car1_oos(end), car2_oos(end), car3_oos(end)]
% construct Drawdown Curve
dd0_oos = drawdown(car0_oos);
dd1_oos = drawdown(car1_oos);
dd2_oos = drawdown(car2_oos);
dd3_oos = drawdown(car3_oos);
```

Functions Used (1)

- Calculate the Portfolio Variance

```
function [ result ] = compute_pvar(weights, Covariance)
% weights are suppose to be a column vector
result = weights' * Covariance * weights;
end
```

- Calculate the Portfolio Sharpe Ratio

```
function [ result ] = compute_sharpe( weights, mean, covarience )
% weights are suppose to be a column vector
pvar = weights' * covarience * weights;
pret = mean * weights;
% assumes rf = 3% annually
result = (pret - 0.03) / sqrt(pvar);
end
```

Functions Used (2)

- Construct Drawdown Curve

```
function [ dd_curve ] = drawdown( equity_curve )

high_water_mark = 1;
dd_curve = zeros(length(equity_curve),1);

for i = 1:length(equity_curve)

    if equity_curve(i) > high_water_mark
        high_water_mark = equity_curve(i);
    end

    drawdown = (high_water_mark - equity_curve(i)) / equity_curve(i);
    dd_curve(i) = drawdown;

end
```

Additional Exercise

- Simulate GBM with NGARCH vol.
- Estimate Parameters for GBM with NGARCH.
- Simulate GBM with Jumps.
- Estimate Parameters GBM with Jumps.
- Find out the Moving-Average Crossover Strategy for Buy Side Portfolio Optimisation Case.

Feel free to email me (h.gong.12@ucl.ac.uk) and Join the Crescent Quant for the additional materials. (`http://www.crescentquant.com`)