Application of Matlab for Finance
Week 7

Shi, Yining

October 21, 2017

Today's Class

- Ordinary Least Squares (OLS)

- Rolling Windows Estimation
- Maximum Likelihood Estimator (MLE)

Basic Linear Regression: OLS

$$Y = X\beta + \epsilon, \quad \epsilon \sim N(0, \sigma)$$

- Y: $T \times 1$ vector of dependent variables
- X: $T \times M$ matrix of regressors (or $[1, X]$ regression with constant)
- Coefficient: $\hat{\beta} = (X'X)^{-1}X'Y$, i.e., $\hat{\beta} \to \min(Y - X\beta)'(Y - X\beta)$
- Residual: $\hat{\epsilon} = Y - \hat{Y} = Y - X\hat{\beta}$
- Mean squared error (MSE)/variance of residual: $s^2 = \frac{\hat{\epsilon}'\hat{\epsilon}}{T-M}$
- Goodness-of-fit against the benchmark: historical average ($\bar{Y}$)

$$R^2 = 1 - \frac{\sum_{i=1}^{T}(Y_i - \hat{Y}_i)^2}{\sum_{i=1}^{T}(Y_i - \bar{Y})^2} = 1 - \frac{\hat{\epsilon}'\hat{\epsilon}}{(Y - \bar{Y})'(Y - \bar{Y})}$$

$$= 1 - \frac{Sum\ Squares\ of\ Residual}{Sum\ Squares\ of\ Total}, \qquad \bar{Y} = \frac{\sum_{i=1}^{T} Y_i}{T}$$

OLS Regression Code

- `b = regress(y,X)` estimates the M-by-1 β coefficient estimations of regression $y = X\beta$ and stored in output variable `b`.
- `[b,bint,r,rint,stats] = regress(y,X)`:
  - `b`: M-by-1 matrix of the estimations of β

  - `bint`: M-by-2 matrix of the 95% confidence intervals for the coefficient estimates
  - `r`: T-by-1 matrix of the residual
  - `rint`: T-by-2 matrix of intervals that can be used to diagnose outliers (if zero is not inside).

  - `stats`: 1-by-4 vector contains the R2 statistic, the F statistic and its p-value, and the estimate of the error variance (MSE).

Exercise 1: Basic Linear Regression

- Estimate the coefficients of the Fama-French Three-Factor model

$$r_i = \beta_i^m r_M + \beta_i^s SMB + \beta_i^v HML + \epsilon_i$$

- r_i is the excess return of portfolio i
- r_M is the excess return of the market portfolio
- SMB is the size factor
- HML is the value factor

- r_M, SMB, HML are stored in sheet `Factors` inside excel file `FF_Data.xlsx`.
- Individual portfolio returns are stored in the `IndusPort` worksheet.

Exercise 1: Fama French Model


```
% Estimate the model on S, V1
% alternative code of reading from excel file
[ff_fact, ff_txt]= xlsread('FF_Data.xlsx','Factors');
[ff_port, ff_port_txt]= xlsread('FF_Data.xlsx','IndusPort');

% Assign X: Mkt-RF, SMB, HML (last col as rf)
x = ff_fact(:,1:3);
rf = ff_fact(:,4);
% Assign y: Excess returns
y = ff_port - rf;

% Regress with loop for each asset: the coefficients for each asset are
% stored in each row of beta
[T,K] = size(y);
for i = 1: K
    beta(i,:) = regress(y(:,i),x);
end

% perform the same regression: read beta estimations and the statistics
% of the model
for i = 1: K
    [beta(i,:),¬,¬,¬,stats(i,:)] = regress(y(:,i),x);
end
```

Linear Model Regression

```
res = fitlm(X,y,modelspec)
```


- `res`: the LinearModel Object for the responses `y` to be fit to data matrix `X`;
- `modelspec`: model specification for the linear model with 'constant', 'linear', 'interactions', 'quadratic'.

- Refer relevant items in an objective variable, use the period sign `.`
  - `res.Coefficients.Estimate` returns the beta coefficient
  - `res.Coefficients.pValue` returns the significance of beta estimation

- The `fitlm` function assumes a constant term, to remove it:
  - Original model `y` $\sim$ `1 + x1 + x2 + x3`
  - `terms = '1'`
  - `res = removeTerms(res,terms)`
  - New model `y` $\sim$ `x1 + x2 + x3`

Exercise 1: Fama-French Model


```
%% Regression with lmfit with consumer sector as example
y_cnsmr = y(:,1);
res = fitlm(x,y_cnsmr,'linear')
terms = '1'
res = removeTerms(res,terms)
beta_full = res.Coefficients.Estimate
```

Rolling and Recursive Regressions

- Static coefficients may fail to adjust changes in the economy

$$Y = X\beta + \epsilon, \quad \epsilon \sim N(0, \sigma)$$

- Time-varying parameters

$$Y_t = X_t\beta_t + \epsilon_t, \quad \epsilon_t \sim N(0, \sigma)$$

- Timely adjusted coefficient shall reflect more updated information;
- If coefficient change gradually, then similar coefficient in adjacent time periods (seasonal effect)

- The rolling estimates is a combination of true coefficients and sampling errors
  - True coefficient is trending: estimates display trend and noise
  - True coefficient is constant: estimates display random fluctuation and noise

Rolling and Recursive Regressions

- For a given window width τ, using the τ observations $(t = 1, ..., \tau)$ for estimation
- Advance one observation at a time $(t = 2, ..., \tau + 1)$, repeat the estimation

- Plot the estimated coefficient against time to visualize the changes
- For a sample of observations k, it shall be $k - \tau + 1$ estimates of β

Exercise: 2

- For the same portfolios in the previous exercise, compute rolling βs using 60 consecutive observations
- Do the coefficient for market exposure appear constant?

Exercise 2: Rolling Estimation

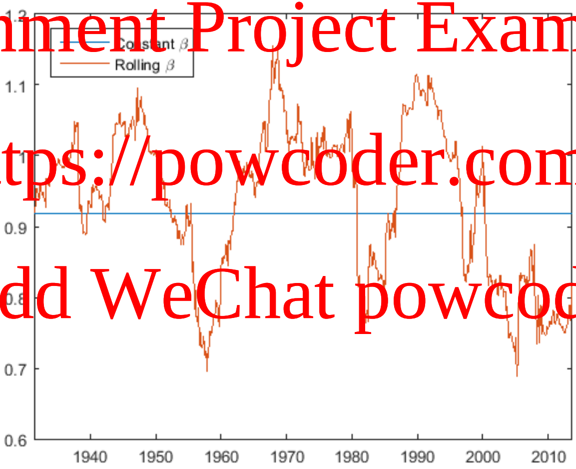

```
% Rolling Regressions for 1 asset
% Read Data Data
dd = ff_txt(2:end,1);
ddnum = datenum(dd,'dd/mm/yyyy');

% Read the full sample estimation of beta for Mkt factor

% Define the Window
tau = 60;
% Loop from 60 to the end
for t=61:T
    x_roll = x(t—tau:t,:);
    y_roll = y_cnsmr(t—tau:t);
    beta_roll(t,:)= regress(y_roll,x_roll);
end

% only plot the exposure to the market risk:
plotdd = ddnum(tau+1:end,1);
plotb_full = beta_full(1)*ones(length(plotdd),1);
plotb_roll = beta_roll(tau+1:end,1);
% Plot the data
plot(plotdd, plotb_full, plotdd, plotb_roll);
datetick('x');
xlim([plotdd(1), plotdd(end)]);
% Add a legend
legend('Constant \beta','Rolling \beta','Location','NorthWest');
```

Estimated Beta



Maximum Likelihood

- Maximum likelihood is a popular method to estimate parameters in econometric models. In many cases, closed form estimators are not available and so non-linear optimizers are required.

- Suppose $\mathbf{X} = (x_1, x_2, ..., x_n)$ are the samples taken from a random variable $\mathbf{X}$ with probability distribution function (pdf) $f_n(x_n; \theta)$, where $\theta = [\mu, \sigma]'$.

- For example: $X \sim N(\mu, \sigma^2)$: $f(\theta; X) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$
- The joint density function of the sample is, by independence, equal to the product of the marginal densities, which is also known as the likelihood function with respect to the parameter set θ,

$$L(\theta; X) = \prod_{i=1}^{n} f_i(x_i; \theta) = f_1(x_1; \theta) f_2(x_2; \theta.)..f_n(x_n; \theta)$$

- The Maximum Likelihood Estimator (MLE) is the parameter set $\hat{\theta}$ that maximizes the likelihood function $L(\theta; X)$

$$L(\hat{\theta}; X) = \max_{\theta \in \Theta} L(\theta; X)$$

Log Maximum Likelihood

- It is often rather difficult to directly maximise the $L(\theta; X)$. It is much easier to maximise the log-likelihood function since $ln(.)$ is a monotonic function that maximising $L(\theta; X)$ is equivalent to maximising $\ln L(\theta; X)$

$$L(\theta; X) \to \max \quad \Longleftrightarrow \quad \ln L(\theta; X) \to \max$$

- The Log-likelihood function

$$\log L(\theta; X) = \sum_{i=1}^{n} \ln f_i(x_i; \theta)$$

- The crucial is to have explicit pdf function to maximize. MATLAB contains various options for different distribution

$$\ln L(\theta; X) = \sum_{i=1}^{n} (\ln \frac{1}{\sqrt{2\pi}} - \ln \sigma - \frac{(X-\mu)^2}{2\sigma^2})$$

$$= n \ln \frac{1}{\sqrt{2\pi}} - n \ln \sigma - \frac{1}{2\sigma^2} \sum_{i=1}^{n} (X-\mu)^2$$

Log Maximum Likelihood: Exercises

- Use help to find the `mle` function in MATLAB
- Use the `mle` function to estimate the mean and variance on the monthly FF consumer sector returns

- Compare the estimated value with the mean and variance of the sample
- Note: the `mle` function only allows one time serie (vector observations) at a time.

Exercise 3: MLE


```
% Maximum Likelihood:
% calculate mle of mean and variance of consumer sector excess return
ret_cnsmer = y(:,1);
parm_est = mle(ret_cnsmer)
% Calculate the sample statistics
parm_data = [mean(ret_cnsmer), std(ret_cnsmer)];
% Display
disp('Consumption Industry Returns')
disp('MLE estimates of mu, sigma')
disp(parm_est)
disp('sample estimates of mu, sigma')
disp(parm_data)
```

Extra: AR, MA and ARMA code

- Autoregressive Model: AR(p)

$$Y_t = c + \sum_{i=1}^{p} \varphi_i Y_{t-i} + \epsilon_t$$

- Moving-Average Model: MA(q)

$$Y_t = \mu + \epsilon_t + \sum_{i=1}^{q} \theta_i \epsilon_{t-i}$$

- ARMA(p,q) Model (set $\mu = 0$)

$$Y_t = c + \epsilon_t \sum_{i=1}^{p} \varphi_i Y_{t-i} + \sum_{i=1}^{q} \theta_i \epsilon_{t-i}$$

- If Y_t is integrated, use the ARIMA(p,D,q) model, where D is the number of difference order

Extra: AR, MA and ARMA code

- Code in matlab: `arima(p,D,q)`
  - AR(1): `arima(1,0,0)`
  - MA(2): `arima(0,0,2)`
  - ARMA(2,3): `arima(2,0,3)`
- The presence of AR process violates the assumption of stationary for the linear regression analysis (to apply the central limit theorem)
- To test whether the underlying time series is stationary, the Augmented Dicky-Fuller test `adftest` is applied to test the presence of unit-root in the underlying return sequence.
- `res = adftest(y)`
  - $res = 0$: fail to reject the null hypothesis of a unit root against the autoregressive alternative.
  - $res = 1$: reject the null hypothesis and conclude that the y is stationary.