Financial Time Series Lecture 7:
Simple Nonlinear Models & Market Micro-structure

Does nonlinearity exist in financial TS?
Yes, especially in volatility modeling & high-frequency data analysis

We focus on simple nonlinear models & neural networks
What is a linear time series?

$$x_t = \mu + \sum_{i=0}^{\infty} \psi_i a_{t-i}$$

where μ is a constant, ψ_i are real numbers with $\psi_0 = 1$, and $\{a_t\}$ is an iid $(0, \sigma_a^2)$.
General concept: Let F_{t-1} denote the information available at time $t-1$.
Conditional mean:

$$\mu_t = E(x_t|F_{t-1}) \equiv g(F_{t-1}),$$

Conditional variance:

$$\sigma_t^2 = \mathrm{Var}(x_t|F_{t-1}) \equiv h(F_{t-1})$$

where $g(.)$ and $h(.)$ are well-defined functions with $h(.) > 0$.
For a linear series, $g(.)$ is a linear function of F_{t-1} and $h(.) = \sigma_a^2$.

Statistics literature: focuses on $g(.)$
See the book by Tong (Oxford University Press, 1990)
Financial econometrics literature: focuses on $h(.)$

Some specific models
TAR model: a piecewise linear model in the space of a threshold variable.

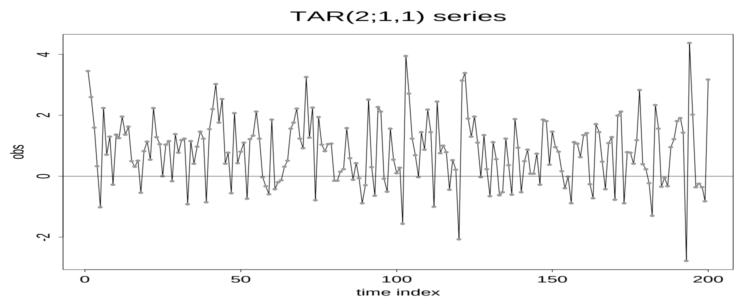


Figure 1: A simulated two-regime TAR process

Example: 2-regime AR(1) model

$$x_t = \begin{cases} -1.5x_{t-1} + a_t & \text{if } x_{t-1} < 0, \\ 0.5x_{t-1} + a_t & \text{if } x_{t-1} \geq 0, \end{cases}$$

where a_t's are iid $N(0,1)$.
Here the delay is 1 time period, x_{t-1} is the **threshold** variable, and the threshold is 0. The threshold divides the range (or space) of x_{t-1} into two regimes with Regime 1 denoting $x_{t-1} < 0$.
What is so special about this model? See the time plot.
Special features of the model: (a) asymmetry in rising and declining patterns, (more data points are positive than negative) (b) the mean of x_t is not zero even though there is no constant term in the model, (c) the lag-1 coefficient may be greater than 1 in absolute value.
Financial applications:
(A) Nonlinear Market Model: Consider monthly log returns of GM stock and S&P composite index from 1967 to 2008. The Market model is

$$r_t = \alpha + \beta r_{m,t} + \epsilon_t.$$

A simple nonlinear model:

$$r_t = \begin{cases} \alpha_1 + \beta_1 r_{m,t} + \epsilon_t, & \text{if } r_{m,t} \le 0 \\ \alpha_2 + \beta_2 r_{m,t} + \epsilon_t, & \text{if } r_{m,t} > 0. \end{cases}$$

```
> da=read.table("m-gmsp6708.txt",header=T)
> head(da)
      Date        GM        SP
1 19670331  0.053541  0.039410
 .....
6 19670831 -0.004720 -0.011715
> gm=log(da$GM+1)
> sp=log(da$SP+1)
> m1=lm(gm~sp)   % Market model
> summary(m1)
Call: lm(formula = gm ~ sp)

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.004861   0.003434  -1.415    0.158
sp           1.072508   0.077177  13.897   <2e-16 ***
---
Residual standard error: 0.07652 on 500 degrees of freedom
Multiple R-squared: 0.2786,     Adjusted R-squared: 0.2772

> length(gm)
[1] 502
> idx=c(1:502)[sp <= 0]  % Locate all non-positive market returns
> nsp=rep(0,502)  % Create the variable of non-positive market returns
> nsp[idx]=sp[idx]
> c1=rep(0,502)   % Create a variable for intercept of non-positive market returns.
> c1[idx]=1
> xx=cbind(gm,sp,c1,nsp)   % Show the resulting variables
> head(xx)
               gm          sp c1         nsp
[1,]  0.052156871  0.03865324  0  0.00000000
[2,]  0.126126796  0.04137128  0  0.00000000
[3,] -0.083130553 -0.05386607  1 -0.05386607
[4,] -0.024098039  0.01736043  0  0.00000000
[5,]  0.097524998  0.04434602  0  0.00000000
[6,] -0.004731174 -0.01178416  1 -0.01178416

> m2=lm(gm~c1+sp)   % with different intercepts
> summary(m2)
Call:  lm(formula = gm ~ c1 + sp)
```

```
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.014971   0.005931  -2.524   0.0119 *
c1           0.021994   0.010538   2.087   0.0374 *
sp           1.258037   0.117556  10.702   <2e-16 ***
---
Residual standard error: 0.07626 on 499 degrees of freedom
Multiple R-squared: 0.2849,     Adjusted R-squared: 0.282

> m3=lm(gm~sp+nsp)
> summary(m3)
Call: lm(formula = gm ~ sp + nsp)

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 0.002329   0.005288   0.440   0.6598
sp          0.848133   0.147421   5.753 1.53e-08 ***
nsp         0.421989   0.236424   1.785   0.0749 .
---
Residual standard error: 0.07635 on 499 degrees of freedom
Multiple R-squared: 0.2832,     Adjusted R-squared: 0.2803

> m4=lm(gm~sp+c1+nsp)
> summary(m4)
Call: lm(formula = gm ~ sp + c1 + nsp)

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.007778   0.007369  -1.055   0.2917
sp           1.041129   0.176838   5.887 7.21e-09 ***
c1           0.020713   0.010550   1.963   0.0502 .
nsp          0.387630   0.236399   1.640   0.1017
---
Residual standard error: 0.07613 on 498 degrees of freedom
Multiple R-squared: 0.2887,     Adjusted R-squared: 0.2844
```

(B) Modeling the leverage effect in volatility: Recall EGARCH, GJR, TGARCH, and APARCH models.

Markov Switching models

Two-state MS model:

$$x_t = \begin{cases} c_1 + \Sigma_{i=1}^{p} \phi_{1,i} x_{t-i} + a_{1t} & \text{if } s_t = 1, \\ \\ c_2 + \Sigma_{i=1}^{p} \phi_{2,i} x_{t-i} + a_{2t} & \text{if } s_t = 2, \end{cases}$$

where s_t assumes values in $\{1,2\}$ and is a first-order Markov chain with trans. prob.

$$P(s_t = 2|s_{t-1} = 1) = w_1, \quad P(s_t = 1|s_{t-1} = 2) = w_2,$$

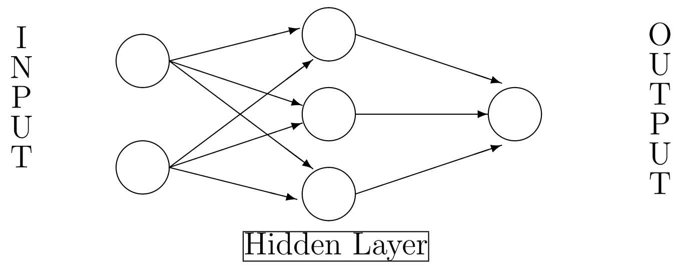
where $0 \leq w_1 \leq 1$ is the probability of switching out State 1 from time $t-1$ to time t. A large w_1 means that it is easy to switch out State 1, i.e. cannot stay in State 1 for long. The inverse, $1/w_1$, is the expected duration (number of time periods) to stay in State 1. Similar idea applies to w_2.

Example: Growth rate of US quarterly real GNP 47-91.
See Figure 4.4 of the textbook (p.188).

State 1							
Par	c_i	ϕ_1	ϕ_2	ϕ_3	ϕ_4	σ_i	w_i
Est	0.909	0.265	0.029	−0.126	−0.110	0.816	0.118
S.E	0.202	0.113	0.126	0.103	0.109	0.125	0.053
State 2							
Est	−0.420	0.216	0.628	−0.073	−0.097	1.017	0.286
S.E	0.324	0.347	0.377	0.364	0.404	0.293	0.064

Discussion

- Regime 2, which has a negative expectation (or growth), denotes "recession" periods. The S.E. of the estimates are large due to the small number of data in the regime.



- The expected durations for Regime 1 and 2 are 8.5 and 3.5 quarters, respectively. $(1/w_i)$

Discussion: Threshold model vs Markov switching model. Deterministic switching vs stochastic switching. They are basically trying to handle similar nonlinearity in a time series.

Empirical analysis

1. You may use the R package `TSA` to fit TAR models. The subcommand is `tar`. Commodity prices tend to be nonlinear. For instance, I use TAR models to study the annual price of copper from 1800 to 1996. A two-regime TAR(1,2) model with delay d = 1 fits the data better than an AR(12) model.

2. You may use the R package `MSwM` to fit Markov switching models. The command is `msmFit`. The package uses EM-algorithm to perform estimation and simulation to produce forecasts. I use the package to model the U.S. GDP growth rate. (Quarterly data.)

Neural networks and Deep learning

- a semi-parametric approach to data analysis
- Structure of a network:

- Output layer
- Input layer
- Hidden layer
- Nodes

- Activation function:
  - Logistic function:

$$\ell(z) = \frac{\exp(z)}{1 + \exp(z)}$$

  - Heaviside (or threshold) function:

$$H(z) = \begin{cases} 1 & \text{if } z > 0 \\ 0 & \text{if } z \leq 0 \end{cases}$$

- Use $\ell(z)$ for the hidden layer

Feed-forward neural network:
Hidden node:

$$x_j = f_j(\alpha_j + \sum_{i \to j} w_{ij} x_i)$$

where $f_j(.)$ is an activation function which is typically taken to be the logistic function

$$f_j(z) = \frac{\exp(z)}{1 + \exp(z)},$$

α_j is called the bias, the summation $i \to j$ means summing over all input nodes feeding to j, and w_{ij} are the weights.
Output node:

$$y = f_o(\alpha_o + \sum_{j \to o} w_{jo} x_j),$$

where the activation function $f_o(.)$ is either linear or a Heaviside function. By a **Heaviside function**, we mean $f_o(z) = 1$ if $z > 0$ and $f_o(z) = 0$, otherwise.
General form:

$$y = f_o\left[\alpha_o + \sum_{j \to o} w_{jo} f_j\left(\alpha_j + \sum_{i \to j} w_{ij} x_i\right)\right].$$

With direct connections from the input layer to the output layer:

$$y = f_o\left[\alpha_o + \sum_{i \to o} w_{io} x_i + \sum_{j \to o} w_{jo} f_j\left(\alpha_j + \sum_{i \to j} w_{ij} x_i\right)\right],$$

Training and forecasting
Divide the data into training and forecasting subsamples.
Training: build a few network systems
Forecasting: based on the accuracy of out-of-sample forecasts to select the "best" network.

Example: Monthly log returns of IBM stock 26-99.
See text for details.

Some R commands: with `nnet` package

```
library(nnet)
x=scan(‘‘m-ibmln2699.txt’’)
y=x[4:864]   % select the output: r(t)
# obtain the input variables: r(t-1), r(t-2), and r(t-3)
ibm.x=cbind(x[3:863],x[2:862],x[1:861])
# build a 3-2-1 network with skip layer connections
# and linear output.
ibm.nn=nnet(ibm.x,y,size=2,linout=T,skip=T,maxit=10000,
decay=1e-2,reltol=1e-7,abstol=1e-7,range=1.0)
# print the summary results of the network
summary(ibm.nn)
# compute \& print the residual sum of squares.
sse=sum((y-predict(ibm.nn,ibm.x))^2)
print(sse)

# setup the input variables in the forecasting subsample
```

```
ibm.p=cbind(x[864:887],x[863:886],x[862:885])
# compute the forecasts
yh=predict(ibm.nn,ibm.p)
# The observed returns in the forecasting subsample
yo=x[865:888]
# compute \& print the sum of squares of forecast errors
ssfe=sum((yo-yh)^2)
print(ssfe)
```

Remark: One-step ahead Out-of-sample-forecasts using **nnet** command. A R script, **backnnet.R**, is developed to carry out the evaluation of 1-step ahead out-of-sample forecasts. For illustration,

```
> source(``backnnet.R'')
> m3=backnnet(x,y,nsize,orig,nl,nsk,miter)
```

A reference book: *Neural Networks in Finance: Gaining Predictive Edge in the Market* by Paul D. McNelis (2005, Elsevier). It uses Matlab.

Analysis of High-Frequency Financial Data & Market Microstructure

Market microstructure: Why is it important?

1. Important in market design & operation, e.g. to compare different markets (NYSE vs NASDAQ)
2. To study price discovery, liquidity, volatility, etc.
3. To understand costs of trading
4. Important in learning the consequences of institutional arrangements on observed processes, e.g.
    - Nonsynchronous trading