AR モデル,GARCH モデルの最尤推定

一橋大学院経営管理研究科 (HUB)-FS
中村　信弘 *

July 3, 2018
(学内限)

概 要

非線形関数の最適化の応用として、Student-t noise をもつ AR(1) モデルと GARCH モデルの最尤推定を行う.

1 AR モデルの最尤推定

1.1 現実的な pairs trading model

問題 1.1 *pairs trading* の適当な 2 銘柄の累積リターンのスプレッドが次の時系列モデル

$$y_t = \gamma y_{t-1} + \varepsilon_t, \quad \varepsilon_t \sim \mathcal{T}(0, \sigma, \nu) \tag{1.1}$$

に従っているとする。現実の個別銘柄のスプレッド・リターンの残差系列 $\{\varepsilon_t\}_{t=1:T}$ は正規分布に従っていることは稀である。そこで、ここでは、(独立同一な)*Student-t 分布*に従うと仮定してみよう。

適当なデータセットを用意し、以下の *MATLAB sample code* を参考に、最尤法によりパラメータ $\Theta := \{\gamma, \sigma, \nu\}$ を推定せよ. ここで t 分布 $\mathcal{T}(0, \sigma, \nu)$ の確率密度関数は

$$f(x|\mu, \nu, \sigma) = \frac{\Gamma\left(\frac{\nu+1}{2}\right)}{\Gamma\left(\frac{\nu}{2}\right)\Gamma\left(\frac{1}{2}\right)} \frac{1}{\sqrt{\nu\sigma^2}} \frac{1}{\left(1 + \frac{(x-\mu)^2}{\nu\sigma^2}\right)^{(\nu+1)/2}} \tag{1.2}$$

である。ここで、$\Gamma\left(\frac{1}{2}\right) = \sqrt{\pi}$. $\mathrm{Var}[\varepsilon_t] = \frac{\nu}{\nu-2}\sigma^2 (\nu > 2)$ であるため、*parameter* σ^2 は残差の分散とスケール倍 $\frac{\nu}{\nu-2}$ だけ異なることに注意しよう。

問題 1.1 の対数尤度関数は

$$l = \sum_{t=2}^{T} l_t = \sum_{t=2}^{T} \log f(y_t | \gamma y_{t-1}, \nu, \sigma) \tag{1.3}$$

であり、これを非線形関数の最大化プログラムにより最大化することで、parameter $\Theta := \{\gamma, \sigma, \nu\}$ を最尤推定する。

♣ 最尤推定のプログラムは、大体、以下のようになる.

*本稿は 2018 年（平成 30 年度）「Computational Finance」講義資料. 著者への問い合わせは以下の通り; Email:nnakamura@hub.hit-u.ac.jp,

```
% St(nt,2): data-read from text data files

x=log(St(:,1));y= log(St(:,2));

  res = regstats(y,x);
fprintf('>>>>> log(S2) vs log(S1) <<<<<<\n');
Spd=y-(res.beta(1)+res.beta(2)*x);% 切片項も入れて、E(Spread)=0 に基準化する。


% initial parameter(guess)
gm0=regress(Spd(2:end),Spd(1:end-1));
nu0=4;
sig0=std(Spd);

paramstr={'gm','nu','sig'};
param( 1) =    gm0; % gamma
param( 2) =    nu0; % dof
param( 3) =    sig0; % sig


%%%% ------- optimization ------------
ndim=length(param);
x0 = param;% initial guess of psi, eta
options = [ ]; % Use default options
options=optimset('Display','iter');
% 許容誤差
%options=optimset(options,'TolFun',1e-8,'TolCon',1e-6,'TolX',1e-6, 'Diagnostic

%%% simple bound set
vlb(1)=0.001;   vub(1)=1;    % gm
vlb(2)=2.00001; vub(2)=50;   % dof
vlb(3)=1e-4;    vub(3)=2;    % sig

A=[];
B=[];
starttime=cputime;
[xout, fout,exitflag,output,lambda,grad,hessia]=...
 fmincon('logStudT',x0, A,B,[],[],vlb, vub,[],options, Spd);
comptime=cputime-starttime;
fprintf('comptime=%15.4f\n',comptime);

param=xout
```

♣ 目的関数は次の負の尤度関数となる.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% (2) (-1) x \sum log(t-density)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function[llk,llkt]=logStudT(param,xt)
% negative log likelihood
```

```
gm=param( 1); % gamma
nu=param( 2); % dof
sig=param( 3);% sig

nt=length(xt);
llkt=zeros(nt,1);

for t=2:nt

   zt=(xt(t)-gm*xt(t-1))/sig;
   w = gammaln((nu + 1) / 2) - gammaln(nu/2);

   llkt(t)=w-0.5*log(nu*sig^2*pi)-....;% <------ 正しい尤度関数に直すこと！

end;

llk=sum(llkt);

% MATLAB は最小化問題を解くので negative log likelihood を返すようにする！
llkt=-llkt;
llk=-llk;
```

♠ 確率密度関数 (1.2) の MATLAB program は次のようになる。この対数が上式の対数尤度に用いられる。

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% (3) t-density
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function[pdf]= StudTpdf(y,nu, mu, sig)

zt=(y-mu)/sig;
term = gammaln((nu + 1) / 2) - gammaln(nu/2);
pdf=exp(term)./(nu*sig^2*pi).^0.5./(1 + (zt.^2) ./ nu).^((nu + 1)/2);
```

1.2 注意点

- 切片項 γ_0 を入れたモデル

$$y_t = \gamma_0 + \gamma_1 y_{t-1} + \varepsilon_t, \quad \varepsilon_t \sim \mathcal{T}(0, \sigma, \nu) \tag{1.4}$$

  を推定する場合は、parameter は一つ増えて $\Theta := \{\gamma_0, \gamma_1, \sigma, \nu\}$ となる。前節の program を書き換えれば推定することができる。考えてみよ。
- 前節の program は対数株価 $\log S(t)$ を使っているが、株価 $S(t)$ でもよい。
- Spread process y_t に trend がある場合は、trend を控除した process に対してモデル (1.1)、または (1.4) をあてはめ推定せよ。

2 GARCH モデルの最尤推定

pairs trading の適当な 2 銘柄の累積リターンのスプレッドが次のような時系列モデル（残差項が不均一分散 (heteroskedastic variance) をもつ）

$$y_t = \gamma y_{t-1} + \varepsilon_t, \quad \varepsilon_t|_{\mathcal{F}_{t-1}} \sim \mathcal{N}(0, h_t) \tag{2.1}$$

に従っているとする。ここで、

$$h_t = \omega + \alpha \varepsilon_{t-1}^2 + \beta h_{t-1} \tag{2.2}$$

とすると、GARCH(1,1) model となる。
　対数尤度関数は

$$\begin{aligned} L &= \sum_{t=2}^{T} \log\left(\frac{1}{\sqrt{2\pi h_t}} \exp\left(-\frac{(y_t - \gamma y_{t-1})^2}{2h_t}\right)\right) \\ &= -\frac{1}{2}\sum_{t=2}^{T} \log(2\pi h_t) - \sum_{t=2}^{T} \frac{(y_t - \gamma y_{t-1})^2}{2h_t} \end{aligned} \tag{2.3}$$

と書くことができる。この関数を parameter $\Theta := (\gamma, \omega, \alpha, \beta)$ に関して最大化することで、parameter の最尤推定値を求めることができる。programming の際には、目的関数として、対数尤度関数 (2.3) の他に、(2.1),(2.2) の coding が必要であることに注意しよう。

♣ 最尤推定のプログラムの主要な部分は、以下のようになる.

```
paramstr={'gm','om','alp','bt'};
param( 1) =     0.8;  % gamma
param( 2) =     0.01; % om
param( 3) =     0.05; % alp
param( 4) =     0.9;  % bt

%%%% ------- optimization ------------
ndim=length(param);
x0 = param;% initial guess of parameters
options=optimset('Display','iter');

%%% simple bound set(lower,upper) of variables
vlb(1)=0.001;    vub(1)=1;        % gm
vlb(2)=0;        vub(2)=1;        % om
vlb(3)=1e-4;     vub(3)=0.2;      % alp
vlb(4)=1e-4;     vub(4)=1;        % bt

%%% No linear constraints
A=[];
B=[];
starttime=cputime;
[param, fout,exitflag,output,lambda,grad,hessia]=...
 fmincon('llkGARCH',x0, A,B,[],[],vlb, vub,[],options, Spd);
comptime=cputime-starttime;
fprintf('comptime=%15.4f\n',comptime);
```

♣ 目的関数は次の負の尤度関数となる.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% (2) loglikelihood(Normal density)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function[llk,llkt]=llkGARCH(param,Spd)
% negative log likelihood
global h1; % ht(1): initial conditional variance

gm =param( 1); % gamma
```

```
om =param( 2); % om
alp=param( 3); % alp
bt =param( 4); % bt

nt=length(xt);
llkt=zeros(nt,1);

et(1)=0;ht(1)=h1;
for it=2:nt
   et(it)=Spd(it)-gm*Spd(it-1);
   ht(it)=om+alp*et(it-1)^2+bt*ht(it-1);

   llf=-1/2*log(2*pi*ht(it))-(Spd(it)-gm*Spd(it-1))^2/(2*ht(it));

   llkt(it-1)=-llf;% negative log likelihood function for the minimizer
end;
llk=sum(llkt);
```