MODELING FAT TAILS IN FINANCIAL RETURNS:
The Student-t distribution

Boston University Questrom School of Business

MF840 – Spring 2018

Eric Jacquier

1. Motivation: Student-t is a natural extension for fat-tailed distributions

2. Estimation by MLE difficult in small sample: likelihood surface uninformative

1 Motivation

Need to predict tail events for risk management:
Need to compute VAR

1) Pure simulation approach:

- Bootstrapping the available returns and compute sample quantiles
- Very imprecise for tail computations:
  Not enough information in the data about extreme events

2) Parametric approach: Normal distribution

- $R \sim N(\mu, \sigma)$ => $\Pr(R < \mu - 1.96\,\sigma) = 0.05$
- Financial returns exhibit far more extreme events than predicted by the Normal tail probabilities
- To use the parametric approach, we need distributions that reflects better the tail properties of financial series

Tail properties of financial series

- Fat Tails 肥尾
- Asymmetric index returns. 收益不对称

Modeling fat tails

- Fat tail distribution: Student-t
- Time varying variance: $R(t) \sim N(\mu, \sigma(t))$
  $\sigma(t) \sim$ GARCH or SV (stochastic volatility)
- Even with GARCH or SV, we may need extra fat-tailness for the conditional distribution 有GARCH或SV也可能还有fat tail
- So the Student-t is useful either for the unconditional distribution or for the conditional distribution of returns
- Alternative: Jump processes, regime shifts

Student-t Distribution Estimation

- Maximum Likelihood
- Bayesian Methods
  - Standard Bayesian is hard, not much better than MLE, MC solves the problem
  - MC: Data augmentation, model the Student-t as a mixture
  - Can be incorporated in a larger model if time varying variance

Student-t Distribution

- Standard t: $t \sim \dfrac{N(0,1)}{\sqrt{\chi^2(\nu)/_{\nu}}}$

- It comes up as integration of the sample variance of a conditionally normal mean, or regression slope coefficient, or forecast,

- ν = 1: Cauchy Distribution, $\nu = \infty$: Normal distribution

Same as 793

- Moments of order ν or higher do not exist: Var(t) = ν / ν-2

- Textbook formulation uses standard Student-t, Returns are **not** standardized

  - Density: $R \sim \sigma\, t(\nu)$ $$p(R_t \,|\sigma\,, \nu\,) \propto \frac{1}{\sigma}\frac{\Gamma\left(\frac{\nu+1}{2}\right)}{\Gamma\left(\frac{\nu}{2}\right)\Gamma\left(\frac{1}{2}\right)}\frac{1}{\sqrt{\nu}}\left(1+\frac{R_t^2}{\nu\sigma^2}\right)^{-\frac{\nu+1}{2}}$$

  - Variance: V(R) = $\sigma^2 \nu / (\nu-2)$

  - Kurtosis: K(R) = $3\, \sigma^4\, \nu^2 / (\nu-2)(\nu-4)$

2 Estimation by Maximum Likelihood

Sample of returns: $R_t \sim$ i.i.d σ $t(\nu)$, $t = 1, \ldots, T$

We will only get an estimate of (ν, σ)

Likelihood:

$$p\,(R\,|\sigma\,,\nu\,) \propto \frac{1}{\sigma^T}\left(\frac{\Gamma\left(\frac{\nu+1}{2}\right)}{\Gamma\left(\frac{\nu}{2}\right)}\right)^T \nu^{-\frac{T}{2}} \prod_t \left(1+\frac{R_t^2}{\nu\sigma^2}\right)^{-\frac{\nu+1}{2}}$$

Likelihood Function

```
#  Function computes likelihood of data(TT) ~ sd * Student(df)

t.loglik <- function(data,TT,df,sd){
loglik <- TT*df*log(sd) + sum(log(dt(data,df)))
loglik <- loglik -
0.5*(df+1)*sum(log((df*sd^2+data^2)/(df+data^2)))
loglik }
```

Simulating Student-t Data

```
# Simulate TT data Student-t(nudraw,sdraw)

TT   <- 10000
nu   <- 10                    # Try low and high nu.
sig  <- 0.2/sqrt(52)          # Weekly US index standard deviation
rm(xx)
xx    <- rt(TT,nu)*sig        # simulated data
```

Plotting the Likelihood

```
# Grid of values for likelihood plot
# Making sure we cover large enough range, this is overkill

shat <- sd(xx)                                # sample standard deviation too high
shigh<- shat * sqrt(qchisq(0.99,TT)/TT)       # bounds for grid
slow <- shat * sqrt(qchisq(0.01,TT)/TT) / sqrt(4/(4-2))
nuvec   <- seq(4, 40,length=70)
sigvec  <- seq(slow,shat, length=35)          # bottom half
sigvec  <- c(sigvec,seq(shat+(shigh-shat)/(35-1),shigh,length=35)) # add top
```

Plot the likelihood versus ν and σ

```
rm(loglikmat); loglikmat<- matrix(0,nrow=70,ncol=70)
inu<-1;  isig<-1;  i<-1
for (inu in 1:70){  for (isig in 1:70){
loglikmat[inu,isig]<-t.loglik(xx,TT,nuvec[inu],sigvec[isig])   }}

contour(nuvec,sigvec,loglikmat,nlevels=40)
abline(v= nuvec[which(loglikmat==max(loglikmat),arr.ind=T)[1]])
abline(h= sigvec[which(loglikmat==max(loglikmat),arr.ind=T)[2]])

persp(nuvec,sigvec,loglikmat,theta=-20,phi=25)

plot(sigvec,colMaxs(loglikmat))       # requires package timeSeries
plot(nuvec,colMaxs(t(loglikmat)))

persp(nuvec,sigvec,loglikmat,theta=-20,phi=25)
```

- Likelihood is very uninformative with respect to ν

Estimate parameters by MLE

Same as t.loglik but structured to be acceptable by nlminb

```
t.loglikmle <- function(p,xx,TT){      # routine computes minus the likelihood
loglik <- TT*p[2]*log(p[1]) + sum(log(dt(xx,p[2])))
loglik <- loglik - 0.5*(p[2]+1)*sum(log((p[2]*p[1]^2+xx^2)/(p[2]+xx^2)))
loglik <- -loglik
loglik   }

# MLE in R:  nlminb (no numerical Hessian)
#            optim  (   numerical Hessian)
#            fitdistr in MASS uses optim
# ind<-which(loglikvec==max(loglikvec),arr.ind=T)

xx    <- rt(TT,nu)*sig    # simulated data

out1<-nlminb(c(shat,30),t.loglikmle,xx=xx,TT=TT,lower=c(0,3),upper=c(Inf,100))
out1$par

out2<-optim(c(shat,30),xx=xx,TT=TT,t.loglikmle, lower=c(slow,4), upper=c(shigh,50),
hessian=T,method = "L-BFGS-B")

out2$value; out2$par; sqrt(diag(solve(out2$hessian)))

fitdistr(rt(500,5)*0.2/sqrt(52),"t",list(m=0,s=1,df=20))
```

- uninformative for high nu: Not a real problem, a fact of life

- ν and σ related: especially for low ν

- Even with large sample size (try 500, 1000) optimization often fails to produce reliable standard errors.

- It takes a LOT of observations to learn about ν.