

分位点回帰の実装

-S+NUOPT を使用した CAViaR モデル-

東京理科大学大学院工学研究科経営工学専攻

修士課程 1 年 杉山崇則

1 はじめに

本発表では，S-PLUS(Ver.6.1 Rel.2) と S-PLUS 上で実行できる数理計画アドオンパッケージである S+NUOPT(Ver.1.6) を使用して，セミパラメトリックな分位点回帰を使った VaR(Value at Risk) の計算法である，CAViaR モデル (Conditional Autoregressive Value at Risk:条件付自己回帰 VaR) を実装する方法について紹介する．

ファイナンスの領域では，下方リスク計測に用いられる評価指標として VaR が挙げられる．VaR とは，現在ある金融資産を保有しているとして，金融資産の価値が，今後の保有期間内に，ある一定の確率の範囲内で，どの程度まで損失を被るかを，過去のデータをもとに理論的に算出された値のことである．数式で表現すると，ある時点 t での価値 (収益率) を r_t ，確率 θ で発生する最大損失額，つまり VaR を $Q_t(\theta)$ とすると，式 (1) のようになる．

$$\Pr(r_t \leq Q_t(\theta)) = \theta \quad (1)$$

VaR を計算する方法は多数提案されている．Manganelli and Engle(2004) では，それらの方法をパラメトリック，セミパラメトリック，ノンパラメトリックの 3 つに分類している．パラメトリックな方法として，分布を仮定したモンテカルロシミュレーションや，GARCH モデル等で推定したボラティリティを利用する方法，デルタ法などが挙げられる．ノンパラメトリックな方法として，ヒストリカル法が代表的な方法として挙げられる．一方，本発表で使用する CAViaR モデルはセミパラメトリックな方法として，Manganelli and Engle(2004) で提案されている．

CAViaR モデルは，分位点の予測を自己回帰モデルで扱うことから，VaR を直接モデル化できることに特徴がある．このようなことから，CAViaR モデルは金融資産の収益分布

を仮定する必要が全くないので、裾の重いような分布に対してもモデル化が可能である。また、モンテカルロシミュレーションのように、多大なシミュレーション回数は必要がないので、CAViaR モデルはそれほど計算負荷を必要としないといった特徴がある。

2 CAViaR モデルの概要

CAViaR モデルは、一般的に次のように書くことができる。

$$Q_t(\theta) = \beta_0 + \sum_{i=1}^p \beta_i Q_{t-p}(\theta) + \ell(\beta_{p+1}, \beta_{p+2}, \dots, \beta_{p+q}; \Omega_{t-1}) \quad (2)$$

ここで、 β_i はパラメータで、 Ω_t は時刻 t までに使用できる情報を表している。Manganelli and Engle(2004) では、具体的に次のようなモデルを提案している。

Symmetric Absolute Value CAViaR

$$Q_t(\theta) = \beta_0 + \beta_1 Q_{t-1}(\theta) + \beta_2 |\varepsilon_{t-1}| \quad (3)$$

Asymmetric Slope CAViaR

$$Q_t(\theta) = \beta_0 + \beta_1 Q_{t-1}(\theta) + \beta_2 (\varepsilon_{t-1})^+ + \beta_3 (\varepsilon_{t-1})^- \quad (4)$$

なお、ここで、 α, β, ω はパラメータで、 $Q_t(\theta)$ は金融資産の収益率の確率 θ 、時刻 t における分位点、 ε_t は時刻 t における条件付平均からの変動である。また、 $I[expr], (x)^+, (x)^-$ は以下のような関数である。

$$I[expr] = \begin{cases} 1 & (expr \text{ is True}) \\ 0 & (expr \text{ is False}) \end{cases} \quad (5)$$

$$(x)^+ = \max(x, 0), \quad (x)^- = -\min(x, 0) \quad (6)$$

本発表では特に、Symmetric Absolute Value CAViaR, Asymmetric Slope CAViaR の 2 つについて実装を行っている。

3 パラメータの導出

Koenker and Bassett(1978) によれば、CAViaR モデルに限らず、分位点回帰モデルのパラメータは、次の式を最小化するように決定すればよいことを示している。また、CAViaR

モデル内での比較であれば，QRSum が小さい値をとるほど，あてはまりの良いモデルであることを示している．

$$QRSum = \min \left(\sum_{t|y_t \geq Q_t(\theta)} \theta |y_t - Q_t(\theta)| + \sum_{t|y_t < Q_t(\theta)} (1 - \theta) |y_t - Q_t(\theta)| \right) \quad (7)$$

ここで， y_t は分位点を求めたいデータであるが，本発表では収益率の分位点を求めたいので $r_t = y_t$ となる．

4 CAViaR モデルの実行

4.1 データの概要

概要 JASDAQ Index 終値の日時データ

期間 1997 年 4 月から 2003 年 3 月の営業日分

なお，収益率に関する分位点を求めるため，次式のような変形を施してある．ここで， I_t が時点 t の Index の終値の値， r_t が収益率を表している．

$$r_t = \log \left(\frac{I_t}{I_{t-1}} \right) \quad (8)$$

4.2 実行結果

4.2.1 Symmetric Absolute Value CAViaR

Symmetric Absolute Value CAViaR モデルについて、係数の値と QRSum を表 1 に示し、収益率 r_t とモデルで推定した 5% 分位点 $Q_t(0.05)$ を図 1 に示す。なお、赤線が分位点を示している。

表 1 Symmetric Absolute Value CAViaR モデルのパラメータの値

| β_0 | β_1 | β_2 | QRSum |
|-------------|------------|------------|----------|
| -0.01380348 | 0.02575766 | -0.9768416 | 3.624358 |

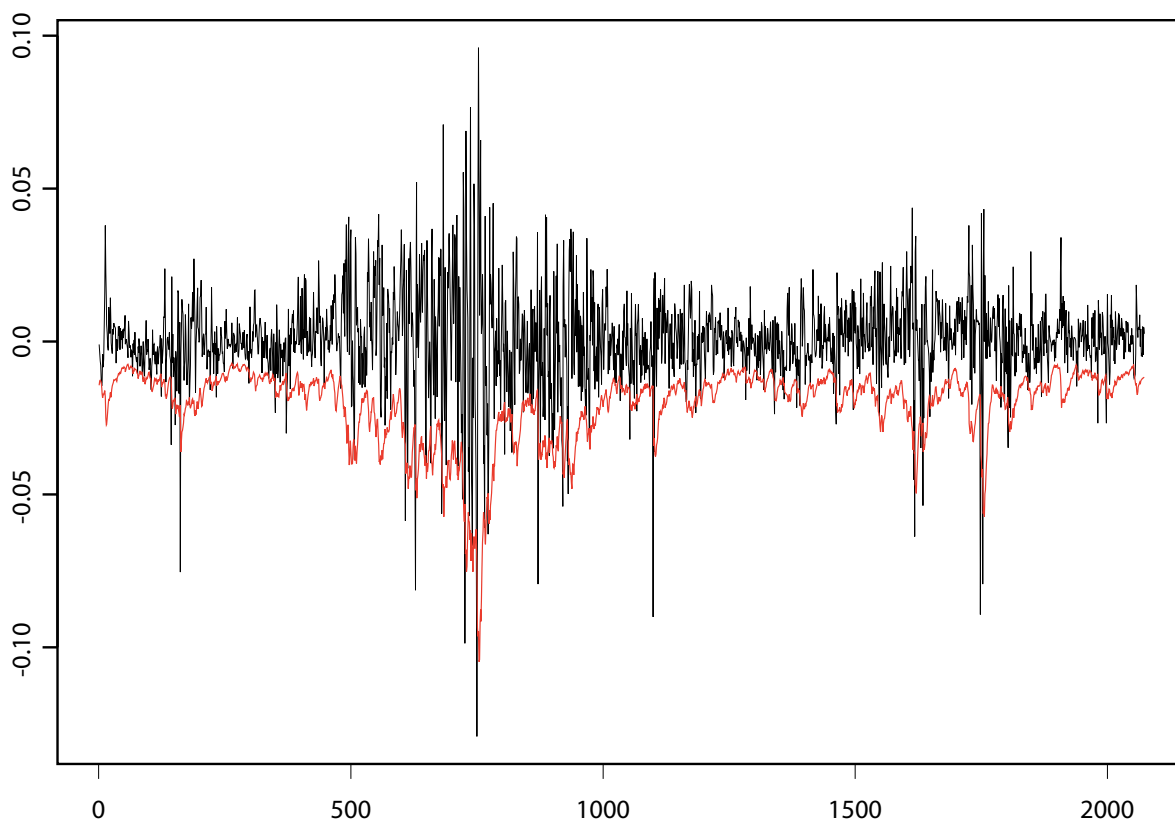


図 1 Symmetric Absolute Value CAViaR モデルの分位点

4.2.2 Asymmetric Absolute Value CAViaR

Asymmetric Absolute Value CAViaR モデルについて，係数の値と $QRSum$ を表 2 に示し，収益率 r_t とモデルで推定した 5% 分位点 $Q_t(0.05)$ を図 2 に示す．なお，赤線が分位点を示している．

表 2 Asymmetric Absolute Value CAViaR モデルのパラメータの値

| β_0 | β_1 | β_2 | β_3 | $QRSum$ |
|--------------|-----------|------------|-----------|---------|
| -0.007789031 | 0.2187208 | -0.1998446 | -1.315241 | 3.21111 |

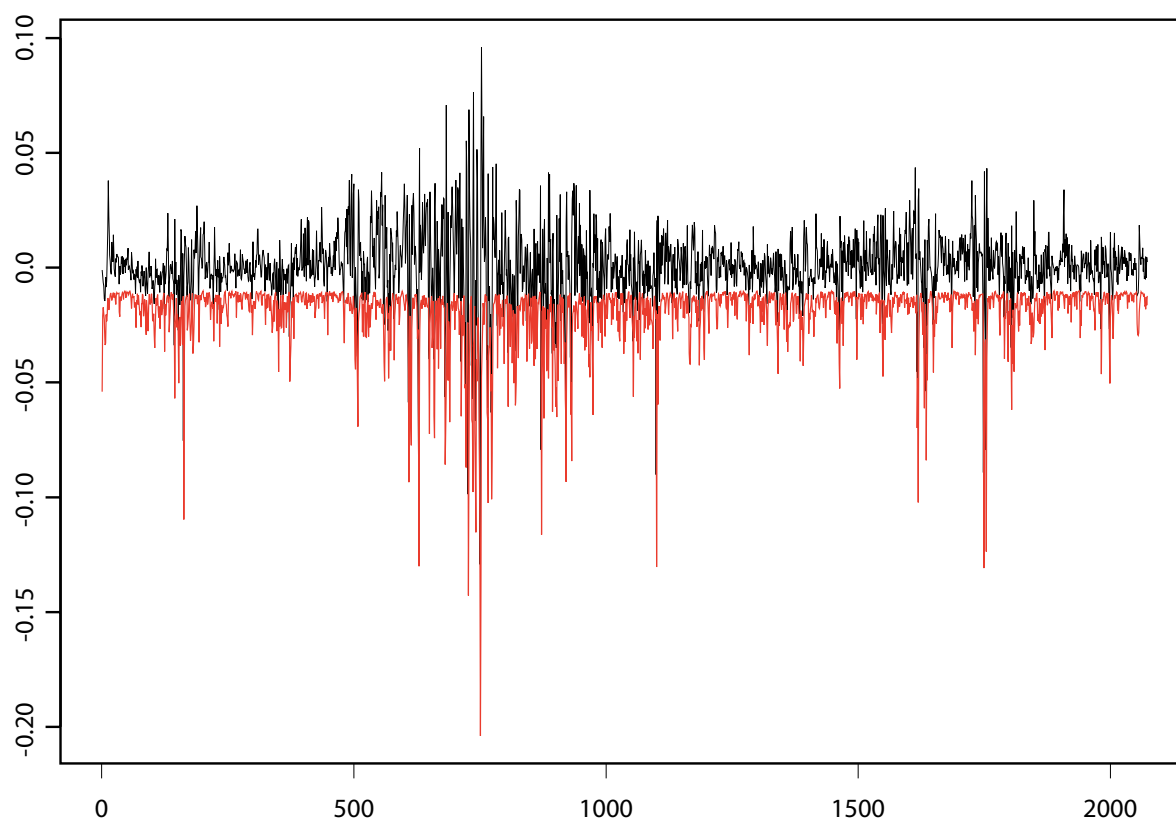


図 2 Asymmetric Absolute Value CAViaR モデルの分位点

4.2.3 考察

まず, QRSum の値を見ると, Symmetric Absolute Value CAViaR モデルより Asymmetric Absolute Value CAViaR モデルのほうが小さいことがわかる。つまり, この2つのモデルだけで比較すると, Asymmetric Absolute Value CAViaR モデルのほうがデータに対する当てはまりが良いということが分かる。係数を見てみると, 定数項と自己回帰係数には大きな差はない。しかし, Asymmetric モデルの係数を見ると, マイナス方向の変動に対して係数が大きくなっていることが分かる。モデルの当てはまりの差は, こうしたところから生じたのではないかとということが分かる。

5 まとめと今後の課題

本発表において, CAViaR モデルを実装し, 実際にデータに適用した。しかし, 他の VaR を求めるモデルとの比較をしていないので, 今後その辺を検討しなければならないと考えている。

また, Manganelli and Engle(2004) では, モデルを使った予測や, モデルの検定 (Wald 検定) の方法も示していたが, 今回はそこまで実装できなかった。さらに, 今回は Asymmetric Absolute Value CAViaR や Symmetric Absolute Value CAViaR だけを実装したが, 実はあと2つモデルが存在しており, その実装を試みたいと考えている。

参考文献

- [1] Engle, R. F., S. Manganelli: "CAViaR: Conditional autoregressive value at risk by regression quantiles", Journal of Business & Economic Statistics 22 367-381(2004).
- [2] James W. Taylor: "Generating Volatility Forecasts from Value at Risk Estimates", Management Science 51 No.5 712-725(2005).
- [3] Roger Koenker, Gilbert Bassett, Jr.: "Regression Quantiles", Econometrica 46 No.1 33-50(1978).
- [4] W. N. Venables, B. D. Ripley 著: 「S-PLUS による統計解析」, シュプリンガー・フェアラーク東京株式会社 (2005).

Appendix. CAViaR モデルのコード

本発表では, S-PLUS(Ver.6.1 Rel.2) と S-PLUS 上で実行できる数理計画アドオンパッケージである S+NUOPT(Ver.1.6) を使用している. ここでは, それらを使った CAViaR モデルを実行するためのソースを示す.

S+NUOPT のためのモデル定義関数

```
##Symmetric Absolute Slope CAViaR モデルを定義
QRSum.SASCAViaR = function(y,eps,theta){

  #添え字集合を定義
  Time <- Set()
  Slope <- Set(1:3)

  #データ長を計算
  length.time <- length(y)

  eps <- abs(eps)

  #添え字を定義
  i <- Element(set = Slope)
  t <- Element(set = Time)

  #データをモデルのパラメータとして代入
  y <- Parameter(list(1:length.time,y), index = t)
  abseps <- Parameter(list(1:length.time,eps),index = t)

  #モデルの変数を定義
  beta <- Variable(index = i)
  Q <- Variable(index = t)

  #目的関数の宣言
  obj = Objective(type="minimize")

  #制約条件を定義
  Q[t+1,t<length.time] == beta[1] + beta[2] * Q[t] + beta[3] * abseps[t]

  #目的関数を定義
```

```

    obj ~ Sum(ife(y[t] >= Q[t],theta * (y[t] - Q[t]), (theta - 1) * (y[t] - Q[t])),t)
}

##Asymmetric Slope

QRSum.ASCAViaR = function(y,eps.p,eps.m,theta){

  #添え字集合を定義
  Time <- Set()
  Slope <- Set(1:4)

  #データ長を計算
  length.time <- length(y)

  #添え字を定義
  i <- Element(set = Slope)
  t <- Element(set = Time)

  #データをモデルのパラメータとして代入
  y <- Parameter(list(1:length.time,y), index = t)
  eps.p <- Parameter(list(1:length.time,eps.p),index = t)
  eps.m <- Parameter(list(1:length.time,eps.m),index = t)

  #モデルの変数を定義
  beta <- Variable(index = i)
  Q <- Variable(index = t)

  #目的関数の宣言
  obj = Objective(type="minimize")

  #制約条件を定義
  Q[t+1,t<length.time] == beta[1] + beta[2]*Q[t] + beta[3]*eps.p[t] + beta[4]*eps.m[t]

  #目的関数を定義
  obj ~ Sum(ife(y[t] >= Q[t],theta * (y[t] - Q[t]), (theta - 1) * (y[t] - Q[t])),t)
}

```


S-PLUS 上で実行するための関数

###CAViaR モデル

```
caviar = function(x,model="Asymmetric",prob=0.05,graph=T,scaling="on",ep=1e-3){
  #S+NUOPT モジュール呼び出し
  module(nuopt)

  #データの収益率を計算
  eps = diff(log(x))

  #パラメータ推定開始
  if(model == "Asymmetric"){
    nuopt.options(maxitn = 200,method="auto",scaling=scaling,eps=ep)

    #epsilon+ と epsilon-を計算
    eps.p = ifelse(eps>0,eps,0)
    eps.m = ifelse(eps<0,abs(eps),0)

    #S+NUOPT で最適化
    result.problem = System(model=QRSum.ASCAViaR,eps,eps.p,eps.m,prob)
    result.solution = solve(result.problem)

  }else if(model == "Symmetric"){
    nuopt.options(maxitn = 200,method="auto",scaling=scaling,eps=ep)

    #S+NUOPT で最適化
    result.problem = System(model=QRSum.SASCAViaR,eps,eps,prob)
    result.solution = solve(result.problem)
  }else{
    print("error:モデル名が違います\n")
  }

  #結果をリスト形式で保存
  Q = result.solution$variable$Q$current
  beta = result.solution$variable$beta$current
  QRSum = result.solution$objective
  rslt = list(percentile = Q, beta = beta, theta = prob,QRSum = QRSum,model=model)

  #必要ならばグラフを描画
```

```
if(graph){  
  tsplit(eps)  
  lines(Q,col=6)  
}  
  
rslt  
}
```