ボラティリティ予測精度の比較
- CAViaRモデルとGJRGARCHモデルの比較 -

東京理科大学工学研究科
鈴木 直明

発表構成

1. はじめに
2. 本研究に用いるモデル
3. 期待値仮定変更の提案
4. 分析
5. まとめ
6. 参考文献

Appendix

ボラティリティとは

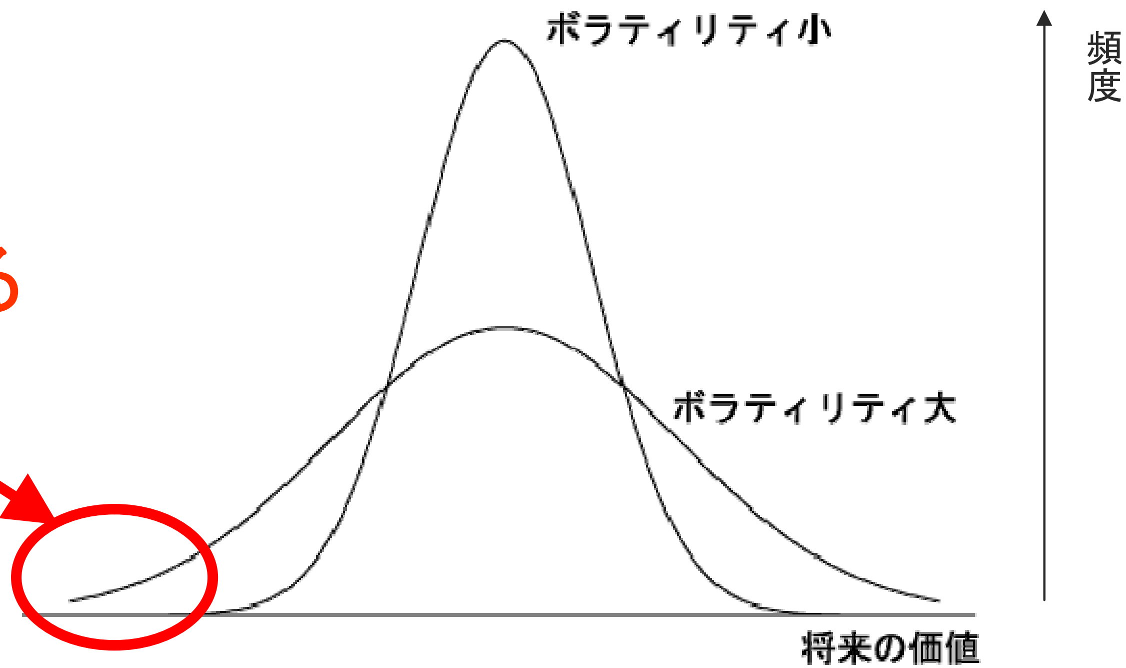
- 株価のボラティリティ
  - 統計における標準偏差（分散）のこと
  - 直接株価の上下の情報は与えない
  - どのくらい株価が変動したかを表す指標である

- ボラティリティの重要性
  - 金融機関の視点
    - ✓ 株式や社債など金融商品の
適切な価格付けに用いられている

ボラティリティの重要性（投資家の視点）

- 投資家は複数の投資案から投資先を選択する
  - 投資案の順序付けをどのように行うか
- 順序付けの基準のひとつとしてボラティリティがある
  - ボラティリティの大きい投資案は選択されない

大きな損失を被る可能性が大きい



ボラティリティ予測の必要性

- ボラティリティを把握することは重要
- ボラティリティを予測することができるならば
  - 適切な価格で金融商品を販売できる
  - 将来のリスクをコントロールできる

このため

- ボラティリティを予測するためのモデルが提唱されてきた

Taylorの研究[3]

- ボラティリティの予測に広く用いられているGARCHモデル

  （Generalized AutoRegressive

  Conditional Hetroskedasticity : 一般化分散不均一）
- GARCHモデルには次の特徴がある
  - GARCHモデルによる予測は分布の仮定に依存している
  - 時間とともに仮定した分布に狂いが生じると予測精度低下につながる

そこで

- 分布の仮定を必要としないモデルを用いてボラティリティ予測を行いたい

Taylorの研究[3]

- TaylorによってCAViaRモデル[1]を用いてボラティリティを予測する方法が提唱された
  （Conditional Autoregressive
  Value at Risk : 条件付自己回帰VaR）
  - ✓ 自己回帰モデル
    過去の実現値を説明変数とした回帰モデル

- TaylorによりCAViaRモデルによる予測精度がGARCHモデルに匹敵することが示された

鈴木らの研究[10]

- CAViaRモデルによる予測精度の向上を目的とした研究
  - CAViaRモデルに過去の情報をより多く取り込む

そのために

  - CAViaRモデルに高次の自己回帰を適用した次数拡張CAViaRモデルを提案
  - 次数拡張モデルを用いた予測方法と既存モデルによる予測精度を比較
- 次数拡張モデルが有効であることが示された

研究目的

- 本研究では次数拡張CAViaRモデルによるボラティリティ予測精度向上を目的とする

- 研究内容は次の通りである
  - 次数拡張モデルの期待値の定義を変更したモデルを提案する
  - 既存モデル（GJRGARCHモデル，TaylorＴの用いたCAViaRモデル及び鈴木らの用いた次数拡張モデル）と予測精度を比較する

研究に用いるモデル

- 本研究に関係するモデルを紹介する
  - VaR (Value at Risk) モデル
  - ARCH
    (AutoRegressive Conditional Hetroskedasticity
    : 分散不均一) モデル
  - GARCH型モデル
  - ASCAViaRモデル
  - 次数拡張モデル
  - 提案モデル
    - ✓ AR model
- 本研究では株価の収益率(r_t)として対数収益率を用いる

$$\text{対数収益率}(r_t) = \log \frac{t\,\text{時点の株価}}{t-1\,\text{時点の株価}}$$

VaRモデル

VaR

- ある信頼区間の中で最低の収益率を分位点によって表現する

分位点

- 数値を小さい順に並べ替えたときに小さいほうから数えて総度数の θ%に当たる値
- θ%分位点は以降 $Q(\theta)$ と表現する

ARCHモデル

ARCHモデル

- (1)式から(3)式で表現される
- (3)式がボラティリティを表す
  - ✓ 過去の予期せざる変動を線形関数化している

$$r_t = \mu + \varepsilon_t \qquad (1)$$

$$\varepsilon_t = \sigma_t u_t \qquad (2)$$

$$\sigma_t^2 = \omega + \sum_{i=1}^{p} \alpha_i \varepsilon_{t-i}^2 \qquad (3)$$

$\mu = E[r_t]$, ε_t : 誤差項

σ_t : ボラティリティ

$u_t : i.i.d \sim N(0, 1^2)$

$\omega, \alpha_i \geq 0$: パラメータ

- ボラティリティの高い(低い)時期が継続する現象を捉える

GARCH型モデル（GARCHモデル）

GARCHモデル

- ARCHモデルを一般化したモデル
- ARCHモデルの(3)式に自己回帰を組み込む

$$\sigma_t^2 = \omega + \sum_{i=1}^{p} \alpha_i \varepsilon_{t-i}^2 + \sum_{j=1}^{q} \beta_j \sigma_{t-j}^2 \qquad (4)$$

$\beta_j \geq 0$: パラメータ

GARCH型モデル（GJRGARCHモデル）

GJRGARCHモデル

- GARCHモデルは変動の非対称性を捉えられない
- 株価が下がった後にボラティリティは大きくなる
- GJRGARCHモデルは
  (4)式に右辺第3項を加えることにより反映

$$\sigma_t^2 = \omega + \sum_{i=1}^{p} \alpha_i \varepsilon_{t-i}^2 + \sum_{i=1}^{p} \gamma_i S_{t-i} \varepsilon_{t-i}^2 + \sum_{j=1}^{q} \beta_j \sigma_{t-j}^2 \qquad (5)$$

$\gamma_i \geq 0$: パラメータ

$$S_{t-i} = \begin{cases} 1 & \varepsilon_{t-i} < 0 \\ 0 & \varepsilon_{t-i} \geq 0 \end{cases}$$

CAViaRモデル

- CAViaR[1]モデルの特徴
  - ヒストリカルな手法である
  - 自己回帰モデルである

- 本研究では，Asymmetric Slope CAViaR（以降ASCAViaRと表現する）モデルを用いる
  - 予測に用いる過去の収益率がプラスかマイナスかによって予測結果が異なる
  - 予測に用いる過去の収益率がマイナスの方が予測に大きな影響を与える

ASCAViaRモデルのモデル式

- ASCAViaRモデルは(6)式で定式化される

$$Q_t(\theta) = \omega + \alpha_1(\varepsilon_{t-1})^+ + \alpha_2(\varepsilon_{t-1})^- + \beta Q_{t-1}(\theta) \quad (6)$$

- 右辺第2項:*t-1*時点における収益率と平均の正の差とパラメータの積
- 右辺第3項:*t-1*時点における収益率と平均の負の差とパラメータの積
- 右辺第4項:*t-1*時点の分位点とパラメータの積

$Q_t(\theta)$: θによって与えられる分位点

ω , $\alpha 1$, $\alpha 2$, β :パラメータ

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

$$\varepsilon_t = r_t - E[r_t]$$

鈴木らの研究[10]では一定を仮定

分位点回帰最小化法

- CAViaRモデルのパラメータの推定方法
  - (7)式を分位点回帰[8]と呼ぶ
  - QRSumの値を最も小さくするものをパラメータとして選ぶ[8]
- QRSum は値が小さいほどモデルの当てはまりがよいことを示す

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

t :過去の実測値（1 時点～ t-1 時点）
y_t :実現値の集合

ASCAViaRモデルを用いたボラティリティの予測

- ボラティリティは分位点を用いて(8)式で近似できる
- いかなる分布においても頑健であるとされている[3]

$$\sigma^2 \approx \left(\frac{Q(0.95) - Q(0.05)}{3.25} \right)^2 \qquad (8)$$

拡張モデル（ASCAViaR(*p*,*q*)モデル）

- Taylorが用いたASCAViaRモデルは1次の自己回帰
- 拡張モデルは高次の自己回帰
  - 過去の情報をより多く取り入れることができる
- 拡張モデル[10]は(9)式で定式化される

$$Q_t(\theta) = \omega + \underbrace{\sum_{i=1}^{p} \alpha_{1i}(\varepsilon_{t-i})^{+} + \sum_{i=1}^{p} \alpha_{2i}(\varepsilon_{t-i})^{-}}_{p\text{ 時点前までの情報を取り入れられる}} + \underbrace{\sum_{j=1}^{q} \beta_j Q_{t-j}(\theta)}_{q\text{ 時点前までの情報を取り入れられる}} \qquad (9)$$

期待値仮定の変更

- 鈴木らの研究[10]では期待値として $E[r_t] = \mu$ という一定の仮定をおいていた
- 本研究では期待値に時系列的なモデルを適用する
- 期待値にAR(AutoRegressive)モデルを適用する
- ARモデル
  - $\{X_t\}$ は(10)式を満たす系列である

$$X_t = \sum_{i=1}^{p} \phi_i X_{t-i} + \varepsilon_t \quad \{\varepsilon_t\} \sim WN(0, \sigma^2) \qquad (10)$$

期待値仮定の決定

- 期待値として用いるARモデルは各時点ごとに更新
- モデルのパラメータは
  AIC(Akaike’s Information Criterion)基準で選択
- 最もAICの低いモデルを用いて1期先の予測を行い
  その値を期待値とする

分析対象データ

- 内容
  - JASDAQ Index，TOPIX，米国S&P Index
  - 日次の終値の時系列データ
- 期間
  - 1997年4月1日～2003年5月23日
- 学習用データ
  - 1997年4月1日から1001営業日分のデータを
    モデルの学習用データとする
- 評価用データ
  - 学習用データ終了後からのデータを
    モデルの評価用データとする

分析概要

- 学習用データを用いて提案モデル，既存モデルを構築する
- 次数p,qは1から3まで独立に変えて，9通り評価する
  - パラメトリックな方法であるGJRGARCHモデルはAIC基準などにより最適な次数が選択できる
  - ヒストリカルな方法であるASCAViaRモデルは次数を選択する基準がないため総当りで最適次数を評価
- 10，20営業日先までのボラティリティを予測する
- 予測精度の評価には(11)式の誤差を利用する

$$\varepsilon_t = \left| r_t - E[r_t] \right| \qquad (11)$$

- 予測したボラティリティと誤差の寄与率を求める

分析概要

- 寄与率は相関係数の2乗
- 寄与率は%で表現する
- 寄与率の値は0%から100%の間の値をとる
- 寄与率が高いほうが，予測の精度がよいとする
- 提案モデルと既存モデルの結果の比較を行う
- 各Index，予測期間，モデルで一番高い寄与率を青枠で示した
- 各Index，予測期間で平均寄与率が一番大きいモデルを赤枠で示した

分析結果（JASDAQ Index）

- JASDAQ Indexについて予測した結果を表1と表2に示す
  - JASDAQ Index・10日先の予測
    - ✓ 提案ASCAViaRモデルが最大寄与率であった
      - $(p,q) = (2,1)$ で寄与率37.5%
    - ✓ 提案ASCAViaRモデルが最大平均寄与率であった
      - 平均寄与率15.5%
  - JASDAQ Index・20日先の予測
    - ✓ 既存ASCAViaRモデルが最大寄与率であった
      - $(p,q) = (1,2)$ で寄与率17.6%
    - ✓ 既存ASCAViaRモデルが最大平均寄与率であった
      - 平均寄与率11.8%

JASDAQ Index･10日先の予測結果

表1:JASDAQ Index･10日先の予測結果

p	q	提案ASCAViaR	既存ASCAViaR	GJRGARCH
1	1	3.3	4.4	32.2
1	2	4.9	12.9	10.9
1	3	5.3	10.4	9.2
2	1	**37.5**	9.0	13.5
2	2	7.1	5.5	9.2
2	3	0.7	7.9	9.3
3	1	17.7	7.3	16.1
3	2	31.5	9.9	11.7
3	3	31.9	8.0	16.4
average		**15.5**	8.4	14.8

JASDAQ Index･20日先の予測結果

表2:JASDAQ Index･20日先の予測結果

p	q	提案ASCAViaR	既存ASCAViaR	GJRGARCH
1	1	8.8	4.3	15.7
1	2	11.1	**17.6**	9.5
1	3	10.7	14.5	9.6
2	1	3.7	10.9	11.7
2	2	1.7	6.2	9.4
2	3	4.5	8.8	10.1
3	1	7.2	13.5	12.3
3	2	3.4	15.2	9.2
3	3	10.2	14.8	9.5
average		6.8	**11.8**	10.8

分析結果(TOPIX)

- TOPIXについて予測した結果を表3と表4に示す
  - TOPIX･10日先の予測
    - ✓ GJRGARCHモデルが最大寄与率であった
      - (p,q) = (2,3) で寄与率31.6%
    - ✓ GJRGARCHモデルが最大平均寄与率であった
      - 平均寄与率20.2%
  - TOPIX･20日先の予測
    - ✓ GJRGARCHモデルが最大寄与率であった
      - (p,q) = (3,1) で寄与率15.4%
    - ✓ GJRGARCHモデルが最大平均寄与率であった
      - 平均寄与率4.8%

TOPIX･10日先の予測結果

表3:TOPIX･10日先の予測結果

p	q	提案ASCAViaR	既存ASCAViaR	GJRGARCH
1	1	7.5	6.9	28.8
1	2	0.0	4.7	25.4
1	3	0.3	4.6	10.7
2	1	0.2	0.0	2.4
2	2	0.2	0.2	29.2
2	3	0.1	0.1	**31.6**
3	1	0.0	0.3	9.7
3	2	0.0	0.4	13.3
3	3	0.0	0.1	30.6
average		0.9	1.9	**20.2**

TOPIX･20日先の予測結果

表4:TOPIX･20日先の予測結果

p	q	提案ASCAViaR	既存ASCAViaR	GJRGARCH
1	1	10.2	4.3	0.4
1	2	0.9	2.1	0.0
1	3	0.0	3.1	11.1
2	1	0.0	0.3	9.0
2	2	1.1	0.9	0.0
2	3	0.8	0.6	0.0
3	1	1.4	1.4	**15.4**
3	2	0.3	1.2	7.3
3	3	0.4	0.4	0.0
average		1.7	1.6	**4.8**

分析結果（S&P Index）

- S&P Indexについて予測した結果を表5と表6に示す
  - S&P Index・10日先の予測
    - 提案ASCAViaRモデルが最大寄与率であった
      - $(p,q) = (3,1)$ で寄与率62.8%
    - 既存ASCAViaRモデルが最大平均寄与率であった
      - 平均寄与率31.1%
  - S&P Index・20日先の予測
    - 提案ASCAViaRモデルが最大寄与率であった
      - $(p,q) = (1,3)$ で寄与率18.1%
    - 提案ASCAViaRモデルが最大平均寄与率であった
      - 平均寄与率8.5%

S&P Index･10日先の予測結果

表5:S&P Index･10日先の予測結果

p	q	提案ASCAViaR	既存ASCAViaR	GJRGARCH
1	1	14.1	37.2	8.9
1	2	17.4	24.2	8.5
1	3	0.3	40.1	20.2
2	1	37.5	27.4	9.4
2	2	52.4	32.8	9.9
2	3	15.6	32.6	18.4
3	1	**62.8**	24.8	5.7
3	2	29.0	29.7	9.4
3	3	46.7	31.3	6.7
average		30.6	**31.1**	10.8

S&P Index・20日先の予測結果

表6 :S&P Index・20日先の予測結果

p	q	提案ASCAViaR	既存ASCAViaR	GJRGARCH
1	1	11.8	9.7	2.1
1	2	11.1	5.8	2.1
1	3	**18.1**	0.8	3.8
2	1	6.8	5.2	2.2
2	2	7.9	8.3	1.5
2	3	7.1	10.1	3.5
3	1	9.9	10.9	2.6
3	2	1.7	11.1	2.4
3	3	2.3	8.0	2.8
average		**8.5**	7.8	2.6

考察

- 最高寄与率は6通りの予測中
  - 提案ASCAViaR: 3通り
  - 既存ASCAViaR: 1通り
  - GJRGARCH: 2通り

最高寄与率の数で比較すると
提案ASCAViaRモデルを用いた予測精度が最もよい

- 平均最高寄与率は6通りの予測中
  - 提案ASCAViaR: 2通り
  - 既存ASCAViaR: 2通り
  - GJRGARCH: 2通り

平均最高寄与率の数で比較すると予測精度に
差がなかった

考察

- GJRGARCHモデルを除いた場合
  最高寄与率は6通りの予測中
  - 提案ASCAViaR: 5通り
  - 既存ASCAViaR: 1通り

最高寄与率の数で比較すると
提案ASCAViaRモデルを用いた予測精度が最もよい

- GJRGARCHモデルを除いた場合
  平均最高寄与率は6通りの予測中
  - 提案ASCAViaR: 3通り
  - 既存ASCAViaR: 3通り

平均最高寄与率の数で比較すると予測精度に
差がなかった

考察

- GJRGARCHモデルを除いた場合
  最高寄与率は6通りの予測中
  - 提案ASCAViaR:　　5通り
  - 既存ASCAViaR:　　1通り

最高寄与率の数で比較すると
提案ASCAViaRモデルを用いた予測精度が最もよい

- GJRGARCHモデルを除いた場合
  平均最高寄与率は6通りの予測中
  - 提案ASCAViaR:　　3通り
  - 既存ASCAViaR:　　3通り

平均最高寄与率の数で比較すると予測精度に
差がなかった

まとめ

- 最高寄与率で比較すると提案ASCAViaRモデルは有効であるといえる
- 平均最高寄与率で比較すると提案ASCAViaRモデルは既存モデルと同等であるといえる

結論として

- 最高寄与率を出す(p,q)の組が特定できるならば提案ASCAViaRモデルによるボラティリティ予測は既存モデルよりも優れているといえる

今後の課題

- 最高寄与率を出す(p,q)の組が特定する方法を検討する
- ヒストリカルな方法(CAViaRモデル)とパラメトリックな方法(ARモデル)を両方用いることがよいかを検討する
- 期待値の仮定にARMAやARIMAなどのモデルを用いてみる
- 期間の影響を受けていないかを検討する

参考文献1

[1] Engle, R. F. , S. Manganelli (2004) “CAViaR: Conditional autoregressive value at risk by regression quantiles”, J. Bus. Econm. Statist, Vol 22, 367-381

[2] Eric Zivot, Jiahui Wang(2001)「Modeling Finacial Time Series with S-PLUS」, Insightful

[3] James W. Taylor (2005) “Generating Volatility Forecasts from Value at Risk Estimates”, MANAGEMENT SCIENCE, Vol 51, 712-725

[4] ジャスダック証券取引所:「JASDAQ INDEX 算出要領」, http://www.jasdaq.co.jp/data/JASDAQ_INDEX_Summary170204.pdf , 最終閲覧日2005年10月1日

[5] 刈屋武明, 矢島美寛, 田中勝人, 竹内啓(2003) :「金融時系列の統計」, 岩波書店

参考文献2

[6] Peter J. Brockwell, Richard A. Davis (2004)「入門時系列解析と予測」, 逸見功, 田中稔, 宇佐美嘉弘, 渡辺則生訳 シーエーピー出版
[7] R.A.ベッカー, J.M.チェンバース, A.R.ウィルクス(1991)「S言語Ⅱ」, 共立出版株式会社
[8] Roger Koenker, Gilbert Bassett, Jr (1978) "Regression Quantiles", Econometrica, Vol 46, No.1, 33-50
[9] 杉山崇則(2005) "分位点回帰の実装 -S+NUOPTを使用したCAViaRモデル-", S-PLUS学生研究奨励賞佳作論文
[10]鈴木直明, 朝日弓未, 山口俊和(2006) "次数を拡張した**CAViaR** モデルによるボラティリティの予測",日本オペレーションズ・リサーチ学会2006年度秋季研究発表会アブストラクト集,194-198
[11]渡辺敏明(2000)「ボラティリティ変動モデル」, 朝倉書店
[12]廣松毅, 浪花貞夫(1990)「経済時系列分析」, 朝倉書店

Appendix
S-PLUS code

Appendix A　CAViaR modelの推定

```
###QRSum 最小化モデル

##Asymmetric Slope

QRSum.ASCAViaR = function(y,eps.p,eps.m,theta,pqnum=c(1,1)){

          Time <- Set()
          Slope.alpha <- Set(seq(from=1,to=pqnum[1]))
          Slope.beta <- Set(seq(from=1,to=pqnum[2]))
          Slope.omega <- Set(1:1)

          length.time <- length(y)
          paramlength <- max(pqnum)

          i <- Element(set = Slope.alpha)
          j <- Element(set = Slope.beta)
          k <- Element(set = Slope.omega)
          t <- Element(set = Time)

          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)
```

```
        alpha <- Variable(index = i)
        beta <- Variable(index = j)
        gamma <- Variable(index = j)
        omega <- Variable(index = k)
        Q <- Variable(index = t)

        obj = Objective(type="minimize")

        Q[t+1,t<length.time,t>paramlength] ==
        omega[1] + Sum(alpha[i] * Q[t-i],i) + Sum(beta[j] * eps.p[t-j],j) + Sum(gamma[j]
* eps.m[t-j],j)

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

```
##CAViaRモデル

caviar.suzuki2 =
function(x,y,model="Asymmetric",prob=0.05,pqnum=c(1,1),graph=T,scaling="on",ep=1e-3){
          #モジュール呼び出し
          module(nuopt)

          #元データ編集
          eps = diff(log(x))
          E.eps = y

          #パラメータ推定開始

                    nuopt.options(maxitn = 200,method="auto",scaling=scaling,eps=ep)

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

                    #最適化
                    result.problem = System(model=QRSum.ASCAViaR,eps,eps.p,eps.m,prob,pqnum)
                    result.solution = solve(result.problem)
```

```
Q = result.solution$variable$Q$current
                    alpha = result.solution$variable$alpha$current
                    beta = result.solution$variable$beta$current
                    gamma = result.solution$variable$gamma$current
                    omega = result.solution$variable$omega$current

                    QRSum = result.solution$objective
                    rslt = list(percentile = Q,omega = omega,alpha = alpha, beta = beta,
gamma = gamma,theta = prob,QRSum = QRSum,pqnum=pqnum,model=model)

          if(graph){
                    tsplot(eps)
                    lines(Q,col=6)
          }

          rslt
}
```

Appendix B　期待値（AR modelの最適次数検索）

```
search.order = function(x){
        #データの定義
        data = x
        #対数収益率に変換
        log.return = diff(log(data))
        #収益率から平均を引いたものを定義
        eps = log.return - mean(log.return)
        #長さを確認
        eps.length = length(eps)
        #変数の用意
        data.order = rep(0, eps.length)

        #モデルのあてはめ　　　#データ2つ以下ではARmodel算出不可
        for(i in 3:eps.length){
                data.ar = ar(eps[1:i], order.max = min(i-1,30))
                data.order[i] = data.ar$order
        }
data.order
}
```

Appendix C　期待値算出

```
expect = function(x,y){

            #元データ編集
            eps = diff(log(x))
            eps.length = length(eps)
            data.order = y

            #変数の用意
            E.eps = rep(0,eps.length)

            #期待値過程
                        #1期はepsilon^+およびepsilon^-は0
                        #2期および最適AR過程のorder0の場合は通常の平均をとる

            E.eps[1] = eps[1]
            E.eps[2] = eps[1]
            for(i in 3:eps.length){
                        if(data.order[i] == 0){
                                    E.eps[i] = mean(eps[1:i-1])
                        }
                        else{
                                    eps.ar = arima.mle(eps[1:i]-mean(eps[1:i]),
                                    model=list(order=c(data.order[i],0,0)))
                                    eps.fore = arima.forecast(eps[1:i]-mean(eps[1:i]), n=1,
                                    model=eps.ar$model)
                                    E.eps[i] = eps.fore$mean + mean(eps[1:i])
                        }
            }
            E.eps
}
```

Appendix D　ASCAViaR予測精度算出

```
caviar.predictpq = function(x,y,z,w,pqnum=c(1,1)){

          ##データ準備

          eps = diff(log(x))
          lst.05 = y
          lst.95 = z
          k = pqnum[2]
          j = pqnum[1]

          q05 = c(lst.05$percentile,rep(0,20))
          q95 = c(lst.95$percentile,rep(0,20))

          eps.p = ifelse(eps - w > 0, eps - w, 0)
          eps.m = ifelse(eps - w < 0, abs(eps - w), 0)

          ##比較対象

          epsilon = rep(0,1019)
          for(i in 1:1019){
                    epsilon[i] = abs(eps[i] - w[i])
          }
```

```
##予測

for(i in 1:20){
        q05[i+999] = lst.05$omega
        + sum(lst.05$alpha * q05[(i+999-1):(i+999-k)])
        + sum(lst.05$beta * eps.p[(i+999-1):(i+999-j)])
        + sum(lst.05$gamma * eps.m[(i+999-1):(i+999-j)])
        q95[i+999] = lst.95$omega
        + sum(lst.95$alpha * q05[(i+999-1):(i+999-k)])
        + sum(lst.95$beta * eps.p[(i+999-1):(i+999-j)])
        + sum(lst.95$gamma * eps.m[(i+999-1):(i+999-j)])
}

volatility = ((q95[1000:1019] - q05[1000:1019])/3.25)^2

##寄与率

corr.10 = cor(volatility[1:10],epsilon[1000:1009])
r.2.10 = (corr.10)^2

corr.20 = cor(volatility[1:20],epsilon[1000:1019])
r.2.20 = (corr.20)^2

kiyo = c(round(r.2.10*100,1),round(r.2.20*100,1))
}
```