

Bayes

ベイズ法に関連するプログラム等。

最終更新時間：2009年12月11日13時23分28秒

ベイズ法について

ベイズ法または、伝統的な統計学と異なり統計的推論にベイズの定理を使う統計学の手法です。古くから、優れた手法であることは知られていましたが、推定に必要な積分の計算が困難だったことから、統計学の主流とはなりませんでしたが、近年、コンピュータの処理速度の進歩とMCMC法の確立により、多くのモデルが推定可能となり、さまざまな分野への応用が進んでいます。

パッケージソフト

BUGS

ベイズ法で事後分布を積分する際に必要となるモンテカルロ積分を実行するソフトです。MCMC法のうちギブス・サンプラーを使っています。いまのところ、もっとも多様なモデルに対応しているようです。

BUGS系のソフトのうち、Windows用のものがWinBUGSです。

- ・ [RjpWiki: Rでベイズ統計学: BUGS/WinBUGS](#)
- ・ BUGSコードについては下記参照

RのMCMCpackパッケージ

BUGSを使わなくても、Rの内部である程度はできます。他にも、bayesmパッケージやMNPパッケージがあります。

- ・ [Econometrics](#) に実例を掲載

パッケージソフトは使わない

もともと、パッケージソフトを使う目的は、時間の短縮のためです。統計学の方法の理解のためには、自分でプログラムを組むという作業が不可欠です。また、マルコフ切り替えモデルなど複雑なモデルの推定を行うには、パッケージソフトが対応していないので、自分でプログラミングを行う必要があります。

プログラミングには、一般的にMATLAB/Gaussが使われているようです。ただし、これらはインタープリタ型なので、スピードは遅めです。

Rの組み込み関数だけでMCMCを適用すると、スピードはかなり遅くなります。C/C++と組み合わせる必要があります。

- ・ [RipWiki: Rから他言語利用](#)

BUGS コード

実例：株価の変化率のt分布への当てはめ

- ・ データセット：[topix.csv](#)
- ・ BUGSのモデル：[model_stock.txt](#)

データセットと一緒にワーキングディレクトリにおいておきます。

```
model{
  for(i in 1:n){
    e[i] dt(mu, tau, nu)
  }
  mu dnorm(0, tau2)
  tau dgamma(5,4)
  tau2 <- 10*tau
  nu dgamma(2, 0.5)
}
```

- ・ R プログラム

```
# ライブラリ "R2WinBUG" の読み込み。事前にインストールしておく必要があります。
library(R2WinBUGS)
# データセットの読み込み。
topix <- read.csv("topix.csv", header=F)

topix.ts <- ts(topix[[2]])
e <- c(100*diff(log(topix.ts)))
n <- NROW(e)
in1 <- list(mu=0, tau=2, nu=10)
in2 <- list(mu=0, tau=2, nu=15)
in3 <- list(mu=0, tau=2, nu=7.5)
inits <- list(in1, in2, in3)
data <- list("n", "e")
parameters <- c("mu", "tau", "nu")

#MCMCの実行
topix.sim <- bugs(data, inits=inits, parameters, model.file="model_stock.txt",
  n.chains = 3, n.iter = 5000, debug=F,
  bugs.directory = "c:/Program Files/WinBUGS14/",
  working.directory = NULL)

# 結果の表示
print(topix.sim, digits=3)

# グラフ
mu <- topix.sim$mean$mu
tau <- topix.sim$mean$tau
df <- topix.sim$mean$nu
temp <- c(-10000:10000)/1000

hist(e, breaks=50, probability=1)
lines(temp/tau+mu, dt(temp, df=df)*tau, col=1)
```

1. Normal Linear Model

以下、下記Lncasterより転載(未テスト)

```

model{
  for (i in 1:n){
    y[i] dnorm(mu[i], tau)
    mu[i] <- b0 + b1*x[i] + b2*z[i]
  }
  b0 dnorm(0, 1.0E-6)
  b1 dnorm(0, 1.0E-6)
  b2 dnorm(0, 1.0E-6)
  tau dgamma(1.0E-3, 1.0E-3)
}

```

2. Heteroscedastic Regression

```

model{
  for (i in 1:n){ y[i], dt(mu[i], tau, d)
    mu[i] <- b0 + b1*x[i] + b2*z[i]
  }
  d <- 4
  b0 dnorm(0, 1.0E-6)
  b1 dnorm(0, 1.0E-6)
  b2 dnorm(0, 1.0E-6)
  tau dgamma(1.0E-3, 1.0E-3)
}

```

3. Regression with Autocorrelated Errors

```

model{
  for (i in 2:n){
    y[i] dnorm(mu[i], tau)
    mu[i] <- rho*y[i-1] + b0*(1-rho) + b1*x[i] - rho*b1*x[i-1]
  }
  b0 dnorm(0, 1.0E-6)
  b1 dnorm(0, 1.0E-6)
  tau dgamma(1.0E-3, 1.0E-3)
  rho dunif(-0.999, 0.999)
}

```

4. CES Production Function

```

model{
  for (i in 1:n){
    y[i] dnorm(mu[i], tau)
    mu[i] <- a - (1/b) * log(d+(1-d)*exp(-b*k[i]))
  }
  b <- (1/sig) - 1
  sig dgamma(2,2)
  a dnorm(0, 0.001)
  d dbeta(2,2)
  tau dgamma(1.0E-3, 1.0E-3)
}

```

5. Probit Model

```

model{
  for (i in 1:n){
    y[i] dbin(p[i], 1)
    p[i] <- phi(b0 + b1*x1[i] + b2*x2[i])
  }
  b0 dnorm(0, 1.0E-3)
  b1 dnorm(0, 1.0E-3)
  b2 dnorm(0, 1.0E-3)
}

```

6 Tobit Model

```

model{
  for (i in 1:n){

```

```

ones[i] <- 1
ones[i] ~ dbern( p[i] )
term1[i] <- 0.39894*pow(tau, 1/2)*exp(-0.5*tau*pow(y[i] - mu[i], 2))
term2[i] <- phi(-mu[i]*pow(tau, 1/2))
p[i] <- pow(term1[i], ind[i])*pow(term2[i], 1-ind[i])/10000
mu[i] <- b0 + b1*x[i]
}
b0 ~ dnorm(0, 1.0E-3)
b1 ~ dnorm(0, 1.0E-3)
tau ~ dgamma(1.0E-3, 1.0E-3)
}

```

7. Truncated Normal

```

model{
for (i in 1:n){
ones[i] <- 1
ones[i] ~ dbern( p[i] )
term1[i] <- 0.39894*pow(tau, 1/2)*exp(-0.5*tau*pow(y[i] - mu[i], 2))
term2[i] <- phi(-mu[i]*pow(tau, 1/2))
r[i] <- term1[i]/term2[i]
p[i] <- pow(r[i]-ind[i])/10000
mu[i] <- b0 + b1*x[i]
}
b0 ~ dnorm(0, 1.0E-3)
b1 ~ dnorm(0, 1.0E-3)
tau ~ dgamma(1.0E-3, 1.0E-3)
}

```

8. Ordered Probit

```

model{
for (i in 1:n){
y[i] ~ dcat( p[i,] )
p[i, 1] <- phi(r-mu[i])
p[i, 4] <- 1 - phi(r+d+e-mu[i])
p[i, 2] <- phi(r+d-mu[i]) - phi(r-mu[i])
p[i, 3] <- phi(r+d+e-mu[i]) - phi(r+d-mu[i])
mu[i] <- b*x[i]
}
r ~ dnorm(0, 1.0E-3)
d ~ dgamma(0.1, 0.1)
e ~ dgamma(0.1, 0.1)
b ~ dnorm(0, 1.0E-3)
}

```

9. Poisson Regression

```

model{
for(i in 1:n){
y[i] ~ dpois(mu[i])
mu[i] ~ exp(b0 + b1 * x[i])
}
b0 ~ dnorm(0, 0.001)
b1 ~ dnorm(0, 0.001)
}

```

10. Heterogeneous Poisson Regression

```

model{
for(i in 1:n){
y[i] ~ dpois(mu[i])
mu[i] ~ exp(b0 + b1 * x[i]) * v[i]
v[i] ~ dgamma(a, a)
}
a <- 1/ra
ra ~ dgamma(1, 1)
b0 ~ dnorm(0, 0.001)
b1 ~ dnorm(0, 0.001)
}

```

11. Right Censored Weibull Data

```
model{
  for(i in 1:n){
    y[i] dweib(alpha, r[i])|(cenind[i], )
    mu[i] exp(b0 + b1 * x[i])
  }
  b0 dnorm(0, 0.001)
  b1 dnorm(0, 0.001)
  alpha dgamma(1, 1)
}
```

12. A Censored Heterogeneous Weibull Model

```
model{
  for(i in 1:n){
    y[i] dweib(alpha, r[i])|(cenind[i], )
    mu[i] v[i] * exp(b0 + b1 * x[i])
    v[i] dgamma(del, del)
  }
  del dgamma(1, 1)
  alpha dgamma(1, 1)
  b0 dnorm(0, 0.001)
  b1 dnorm(0, 0.001)
  vari <- 1/del
}
```

13. An Over-Identified Recursive Equations Model

```
model{
  for (i in 1:n){
    y[i,1:2] dnorm(mu[i,], R[,])
    mu[i,1] <- b0 + b1*c0*z[i,1] + b1*c2*z[i,2]
    mu[i,2] <- c0 + c1*z[i,1] + c2*z[i,2]
  }
  R[1:2,1:2] dwish(Omega[,],2)
  b0 dnorm(0, 1.0E-4)
  b1 dnorm(0, 1.0E-4)
  c0 dnorm(0, 1.0E-4)
  c1 dnorm(0, 1.0E-4)
  c2 dnorm(0, 1.0E-4)
}
```

14. An Just Identified Simultaneous Equations Model

```
model{
  for (i in 1:n){
    y[i,1:2] dnorm(mu[i,], R[,])
    mu[i,1] <- ap + bp[1]*zd[i] + bp[2]*zs[i]
    mu[i,2] <- aq + bq[1]*zd[i] + bq[2]*zs[i]
  }
  ap dnorm(0, 1.0E-3)
  aq dnorm(0, 1.0E-3)
  for (i in 1:2){
    bp[i] dnorm(0, 1.0E-3)
    bq[i] dnorm(0, 1.0E-3)
  }
  R[1:2,1:2] dwish(Omega[,],2)
}
```

15. A Panel Data Linear Model

```
model{
  for (i in 1:N){for (t in 1:T){
    y[i,t] dnorm(mu[i,t],tau)
    mu[i,t] <- beta*x[i,t] + alpha[i]
  }}
  for (i in 1:N){ alpha[i] dnorm(alphabar,phi) }
  beta dnorm(0, 1.0E-4)
}
```

```

tau dgamma(0.1, 0.1)
phi dgamma(0.1, 0.1)
}

```

16. A Second Order Autoregression

```

model{
for (i in 3:n){
y[i] dnorm(mu[i],tau)
mu[i] <- b0 + b1*y[i-1] + b2*y[i-2]
}
y[1] dnorm(0, 1.0E-4)
y[2] dnorm(0, 1.0E-4)
b0 dnorm(0, 1.0E-4)
b1 dnorm(0, 1.0E-4)
b2 dnorm(0, 1.0E-4)
tau dgamma(1.0E-3, 1.0E-3)
}

```

17. Stochastic Volatility

```

model{
for (i in 1:n){
y[i] dnorm(0,p[i])
p[i] <- exp(-h[i])
}
h[1] dnorm(muh,qh)
muh <- alpha/(1-rho)
qh <- qv*(1-rho*rho)
for(j in 2:n){
h[j] dnorm(mu2[j],pv)
mu2[j] <- alpha + rho*h[j-1]
alpha dnorm(0, 1.0E-4)
tau dgamma(1.0E-3, 1.0E-3)
rho dunif(-0.999, 0.999)
}
}

```

BUGS コード (2)

マルコフ切替モデル (単位根検定)

```

model{
for(i in 2:n){
s[i] dcat(pi[s[i-1],])
mu[i] <- mu0[i,s[i]]
x[i] dnorm(mu[i],tau[s[i]])
mu0[i,1] <- phi[2] + (phi[4]+1)*x[i-1]
mu0[i,2] <- phi[1] + (phi[3]+1)*x[i-1]
}
# priors
phi[1] dnorm(0, 0.001)
phi[2] <- 0
phi[3] <- 0
phi[4] dnorm(0, 0.001)
for(j in 1:2){
pi[j,1:2] ddirch(alpha[j])
}
tau0 dgamma(0.01,0.01)
omega dgamma(0.01,0.01)
tau[1] <- tau0
tau[2] <- tau0*omega
s[1] dcat(pi0[])
alpha[1] <- 1
alpha[2] <- 1
}

```

関連文献およびリンク

- [The BUGS Project](#)

- Koop, Gary. Bayesian Econometrics. Wiley, 2003.
- Luncaster, Tony. An Intoroduction to Modern Bayesian Econometrics. Blackwell, 2004.

2冊ともそれほど厚くはありませんが、入門から応用まで広い内容を扱っています。また、それぞれ扱っている内容や使うプログラミング・ソフトが異なるので、重宝します。使うソフトはKoopがMATLAB、LuncasterはRとWinBUGSです。ただし、実例自体は多くないので別個論文等で補う必要はあるでしょう。

- 中妻照雄『ファイナンスのためのMCMC法によるベイズ分析』三菱経済研究所, 2003年
- 和合肇編著『ベイズ計量経済分析』東洋経済新報社, 2005.

基礎から応用まで。基礎編を入門としてつかうには厳しいですが、応用編は論文集になっていて、まだベイズ分析が少数派の現状にあっては、とても参考になります。