空間の統計学(9)：ベイズ空間計量経済学①

慶應義塾大学総合政策学部准教授
古谷　知之（Furutani Tomoyuki）

■兵庫県生まれ。2001年東京大学大学院工学系研究科博士課程修了。博士（工学）。東京大学大学院助手、慶應義塾大学環境情報学部専任講師を経て、07年４月より現職。専門分野：空間統計学、都市交通計画、観光政策。

1. はじめに

今回から、第79、80回（2010年２、３月号）で紹介した空間計量経済モデルのベイズ推定について紹介します。地域間・地点間の空間依存性に関する構造を内包する空間計量経済モデルが、空間に依存した効果や空間的な異質性を表現するモデルであることは、これまでにも紹介してきました。空間に依存した効果は空間重み付け行列などで表現されます。空間的な異質性は、誤差項の分散不均一を仮定したり、未知パラメータに空間的な変量効果を組み込んだりすることで表現されます。ところが、こうした空間効果を考慮してモデル推定しようとすると、最尤推定法などの方法では、直接推定できないことがあります。

近年、空間効果を考慮した計量経済モデル推定に、ベイズ推定法が用いられるようになってきています。ベイズ推定法は、ベイズの定理を用いて、与えられた事前情報をもとに事後情報としてモデルパラメータを推定する方法です。マルコフ連鎖モンテカルロ法（MCMC）などのシミュレーション手法を用いて、事後情報を次々に生成し、得られたモデルパラメータを確率分布として捉えます。最小二乗法や最尤推定法によりモデルパラメータを点推定する手法に慣れていると、ベイズ推定法が奇異に感じられるかもしれません。しかし、空間的な異質性や空間効果を考慮した柔軟なモデリングを行うには、最小二乗法や最尤推定法よりも、MCMCなどによるベイズ推定法の方が、より適していると言えます。

そこで今回から、空間計量経済モデルのベイズ推定について紹介します。今回はまず、線形回帰モデルを中心とする基本的なモデルのベイズ推定について紹介することにします。統計モデルのベイズ推定についての説明は、ここでは詳述できないため、参考文献（例えば、[1]など）を参照してください。

2. RとJAGS

MCMCによるベイズ推定を行うための統計

ソフトも、様々なものが利用可能です。Rでも、ベイズ推定を行うパッケージがいくつか用意されています。代表的なパッケージとして、bayesmやMCMCpackなどがあり、線形回帰モデルをはじめ、様々な回帰モデルを推定できます。また、MCMCの代表的な手法であるギブス・サンプラーにより、事後情報を生成するソフトとして、WinBUGS[2]やJAGSが知られています。これらのソフトを使うと、尤度関数を定義することにより、空間効果などを考慮した柔軟なモデリングを行うことができます。WinBUGSはMac OSで利用することが困難なため、今回はJAGSを用いた方法を紹介します。JAGSのコードはRやWinBUGSとよく似ているため、Rを使ったプログラミングに慣れている人には習得しやすいでしょう。

RではR2jagsパッケージを用いて、JAGSコードを使ったMCMCによるモデリングができます。

JAGSは、以下の手順によりインストールします。まず、JAGSホームページ（http://calvin.iarc.fr/~martyn/software/jags/）にアクセスします。次に、JAGSをダウンロードし、インストールします。Windowsの場合は「.exe」ファイルを、Mac OS X 10.4-6の場合は「.dmg」ファイルをダウンロードします。

次に、R上でR2jagsパッケージをインストールします。JAGSでは、ギブス・サンプラーを実行するためのコードファイルをテキスト形式で作成します。作成したJAGSファイルは、Rインストール先フォルダの下にある、R2jagsフォルダ下のmodelフォルダの下に置いて、R2jagsパッケージで読み込みます。Windowsの場合、例えば「C:¥Program Files¥R¥R-2.x.x¥library¥R2jags¥model」の下にファイルを置きます。Macintoshの場合、「Macintosh HD」→「ライブラリ」→「Frameworks」→「R.frameworks」→「Resources」→「library」→「R2jags」→「model」にファイルを置きます。

3. 線形回帰モデルのベイズ推定

演習には、今回も前回同様、首都圏の市区町村別地価データ（住宅地標準地地価の平均価格）並びに夜間人口密度及び第三次産業従業人口密度データを用いて、地価モデルの推定を例に挙げます。

まずは、データを読み込み、最小二乗法による線形回帰モデルの推定を行いましょう。

被説明変数である地価$Y=(y_1, \cdots, y_i, \cdots, y_n)$、説明変数である夜間人口密度$X_1=(x_{11}, \cdots, x_{1i}, \cdots, x_{1n})$、第三次産業従業人口密度$X_2=(x_{21}, \cdots, x_{2i}, \cdots, x_{2n})$と表すことにします。$n$は地区数、$i$は地区番号を意味します。

このとき、線形回帰モデルは次式のように表すことができます。

$$Y=\beta_0+\beta_1X_1+\beta_2X_2+\varepsilon,$$
$$\varepsilon \sim N(0, \sigma^2)$$

ここで、$\beta_0 \sim \beta_2$は未知パラメータ、σ^2は誤差項の分散を意味します。

```
lph <- read.table("lph.csv",
sep=",", header=T)
summary(lph)
# 変数の指定
# 被説明変数
y <- as.vector(lph$LPH)
# 地域数
n <- length(y)
# 説明変数
```

```
x <- as.matrix(cbind(
lph$POPD, lph$EMP3D ))
# 線形回帰モデルの誤差項（最小二乗法）
model.lm <- lm(y~x)
summary(model.lm)
```

線形回帰モデルの最小二乗推定結果

```
> summary(model.lm)

Call:
lm(formula = y ~ x)

Residuals:
     Min       1Q   Median       3Q      Max
-85.1101  -1.8732  -0.5808   1.2305  67.0602

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  2.55112    0.56647   4.504 9.05e-06 ***
x1           1.68162    0.10611  15.848  < 2e-16 ***
x2           2.24666    0.07841  28.654  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’
0.1 ‘ ’ 1

Residual standard error: 8.188 on 359 degrees of
freedom
Multiple R-squared: 0.8289,  Adjusted R-squared:
0.8279
F-statistic: 869.6 on 2 and 359 DF,  p-value: <
2.2e-16
```

線形回帰モデルの尤度は、説明変数X_1とX_2が与えられた条件の下で、被説明変数Yが与えられている条件付き確率として表されます。またこのとき、被説明変数Yは正規分布に従うことから、正規分布の平均をμ、分散をτ^2とすると、尤度は次式のようになります。

$$Y \mid \beta_0 + \beta_1 X_1 + \beta_2 X_2 \sim N(\mu, \tau^2)$$

地区iに対する尤度は、次式のように表せます。

$$y_i \mid \beta_0 + \beta_1 x_{1i} + \beta_2 x_{2i} \sim N(\mu_i, \tau^2),$$
$$\mu_i = \beta_0 + \beta_1 x_{1i} + \beta_2 x_{2i}$$

線形回帰モデルをベイズ推定する際には、未知パラメータに対する事前情報を与えます。ここでβ_0、β_1、β_2にはそれぞれ正規分布を、τ^2には逆ガンマ関数$\tau^2 \sim \Gamma(a, b)$を事前情報として与えます。

以下の手順により、未知パラメータを生成し、その事後情報（事後確率）を求めます。

1） まず、MCMCの生成回数sに対して、$\mu = 0$、$\tau = 0.000001$、$a = 0.01$、$b = 0.01$として、初期値を事前情報として以下のように与えます。

$$\beta_0^{s=0} \sim N(0, 0.000001)$$
$$\beta_1^{s=0} \sim N(0, 0.000001)$$
$$\beta_2^{s=0} \sim N(0, 0.000001)$$
$$\tau^{s=0} \sim \Gamma^{-1}(0.01, 0.01)$$

2） 次に、事前情報が与えられた条件付きで尤度及び事後情報を以下のように生成します。

$$① p(\beta_0^{s+1} \mid \beta_1^s, \beta_2^s, \tau_s^2, Y, X_1, X_2) \sim N(\mu, \tau^2)$$
$$② p(\beta_1^{s+1} \mid \beta_0^{s+1}, \beta_2^s, \tau_s^2, Y, X_1, X_2) \sim N(\mu, \tau^2)$$
$$③ p(\beta_2^{s+1} \mid \beta_0^{s+1}, \beta_1^{s+1}, \tau_s^2, Y, X_1, X_2) \sim N(\mu, \tau^2)$$
$$④ p(\tau_{s+1}^2 \mid \beta_0^{s+1}, \beta_1^{s+1}, \beta_2^{s+1}, Y, X_1, X_2) \sim \Gamma^{-1}(a, b)$$

3） 上記2）を、あらかじめ指定したMCMCの生成回数分だけ繰り返します。

4） 稼働検査期間に相当する生成回数を除いた分を、事後情報とします。

MCMCのチェーン数を複数生成する場合にも、それぞれのチェーンに対して上記1）～4）の手順で事後情報を生成し、得られた事後情報をまとめて、モデル全体の事後情報として扱います。

線形回帰モデルのJAGSコードは、以下のように記述できます。モデル全体をmodel{ }で括り、モデルの尤度をfor文の中で表現しています。

```
model{
# 尤度
for(i in 1:n){
y[i] ~ dnorm(mu[i], tau)
mu[i] <- b0+b1*x[i,1]+b2*x[i,2]
}
# 事前情報
b0 ~ dnorm(0,1.0E-6)
b1 ~ dnorm(0,1.0E-6)
b2 ~ dnorm(0,1.0E-6)
tau ~ dgamma(0.01, 0.01)
sigma <- 1/tau
}
```

次に、パッケージを読み込み、ギブス・サンプラーを実行するためのデータ、初期値、パラメータ、モデルファイルを指定します。その上で、**jags**( )関数を使って上述のjagsコードを実行します。jags( )関数では、データ、初期値、パラメータ、モデルファイルに加え、MCMCの生成回数、稼働検査期間、チェーン数などを指定します。

ここでは、MCMC初期値（事前情報）として、最小二乗法による推定結果を用いることにします。また、MCMCの生成回数を10,000回、稼働検査期間を1,000回、チェーン数を3としています。

```
library(R2jags)
# JAGS変数設定
# データ
data <- list("n", "y", "x")
# MCMC初期値（事前情報）
in1 <-
list(b0=model.lm$coefficients[1],
b1=model.lm$coefficients[2],
b2=model.lm$coefficients[3],
tau=1)
in2 <-
list(b0=model.lm$coefficients[1],
b1=model.lm$coefficients[2],
b2=model.lm$coefficients[3],
tau=1)
in3 <-
list(b0=model.lm$coefficients[1],
b1=model.lm$coefficients[2],
b2=model.lm$coefficients[3],
tau=1)
inits <- list(in1,in2,in3)
# パラメータ
parameters <- c("b0", "b1", "b2", "tau", "sigma")
# モデルファイル
model.file <- system.file(
package="R2jags", "model",
"lm.txt")
# MCMC
lm.jags <- jags(data=data,
inits=inits, parameters,
n.iter=10000,
n.burnin=1000, n.chains=3,
model.file=model.file)
# 推定結果の表示
print(lm.jags, n.burnin=1000,
digits=3)
plot(lm.jags)
traceplot(lm.jags)
```

線形回帰モデルのベイズ推定結果は、以下のように出力されます。meanは各パラメータの事後平均、sdは標準偏差を意味します。2.5%値から97.5%値までの範囲を、95%信頼区間といい、事後分布の符号判定やばらつきの確からしさの判断指標となります。

deviance（D）は、モデルがどの程度データに当てはまるかを意味する指標です。MCMC生成中に計算される尤度をもとに、−2×対数尤度として計算されます。

$$D = -2\log[p(\beta_0, \beta_1, \beta_2, \tau | Y, X)]$$

$\overline{\beta}_0, \overline{\beta}_1, \overline{\beta}_2, \overline{\tau}$を$\beta_0, \beta_1, \beta_2, \tau$の事後平均とすると、$\overline{\beta}_0, \overline{\beta}_1, \overline{\beta}_2, \overline{\tau}$から得られるdevianceの点推定値$\hat{D}$は、$\hat{D} = -2\log[p(\overline{\beta}_0, \overline{\beta}_1, \overline{\beta}_2, \overline{\tau} | Y, X)]$となります。$\overline{D}$を$D$の事後平均とすると、モデルの複雑さを評価する有効なパラメータ数pDは、次のように定義されます。

$$pD = \overline{D} - \hat{D}$$

このとき、モデルの当てはまりの良さを示す指標*DIC*は次式のように表されます。

$$DIC = \overline{D} + pD$$

*DIC*が最も小さいモデルがよいモデルとされます。

MCMCの収束判定方法として、いくつかの手法が提案されていますが、ここではGelman-Rubin統計量がRhatとして表示されます。Rhatは、1に近いほどMCMCが収束していると判断しますが、実用上Rhatが1.05以下であれば、MCMCが収束したと判断することがあります。

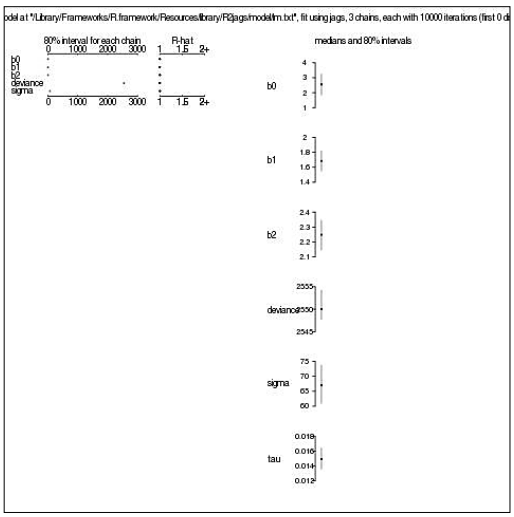
線形回帰モデルのベイズ推定結果

```
> print(lm.jags, digits=3)
Inference for Bugs model at "/Library/Frameworks/R.framework/
Resources/library/R2jags/model/lm.txt", fit using jags,
 3 chains, each with 10000 iterations (first 0 discarded),
n.thin = 9
 n.sims = 3000 iterations saved
             mean    sd     2.5%      25%      50%
b0          2.543 0.550    1.490    2.170    2.545
b1          1.684 0.107    1.478    1.609    1.687
b2          2.247 0.079    2.096    2.192    2.249
deviance 2550.610 2.820 2547.098 2548.534 2549.984
sigma      67.246 5.056   57.910   63.668   66.997
tau         0.015 0.001    0.013    0.014    0.015
              75%    97.5%  Rhat n.eff
b0          2.911    3.623 1.001  3000
b1          1.758    1.884 1.001  3000
b2          2.300    2.404 1.002  1800
deviance 2551.986 2557.974 1.002  1800
sigma      70.535   77.594 1.002  2900
tau         0.016    0.017 1.002  2900

For each parameter, n.eff is a crude measure of effective
sample size,
and Rhat is the potential scale reduction factor (at
convergence, Rhat=1).

DIC info (using the rule, pD = var(deviance)/2)
pD = 4.0 and DIC = 2554.6
DIC is an estimate of expected predictive error (lower
deviance is better).
```

plot( )関数を用いて、モデルパラメータの分布を示してみましょう（図1）。

また、**traceplot**( )関数を用いて、MCMCの挙動を図示できます（図2～5）。

図1　モデルパラメータのベイズ推定結果



update( )関数または**autojags**( )関数を用いれば、モデルが収束するようにギブス・サンプラーを修正します。

```
lm.fit <- update(lm.jags)
print(lm.fit, digits=3)
```

update( )関数の適用結果

```
> print(lm.fit, digits=3)
Inference for Bugs model at "/Library/Frameworks/R.framework/
Resources/library/R2jags/model/lm.txt", fit using jags,
 3 chains, each with 1000 iterations (first 0 discarded)
 n.sims = 3000 iterations saved
             mean    sd     2.5%      25%      50%
b0          2.571 0.562    1.439    2.180    2.567
b1          1.677 0.103    1.471    1.606    1.678
b2          2.247 0.077    2.099    2.193    2.247
deviance 2550.576 2.699 2547.156 2548.570 2550.012
sigma      67.296 5.022   58.057   63.781   67.130
tau         0.015 0.001    0.013    0.014    0.015
              75%    97.5%  Rhat n.eff
b0          2.960    3.654 1.002  1100
b1          1.746    1.881 1.004   630
b2          2.303    2.392 1.001  3000
deviance 2551.945 2557.333 1.001  3000
sigma      70.483   77.776 1.001  3000
tau         0.016    0.017 1.001  3000

For each parameter, n.eff is a crude measure of effective
sample size,
and Rhat is the potential scale reduction factor (at
convergence, Rhat=1).

DIC info (using the rule, pD = var(deviance)/2)
pD = 3.6 and DIC = 2554.2
DIC is an estimate of expected predictive error (lower
deviance is better).
```

図2　MCMCのトレース結果出力例（β_0）

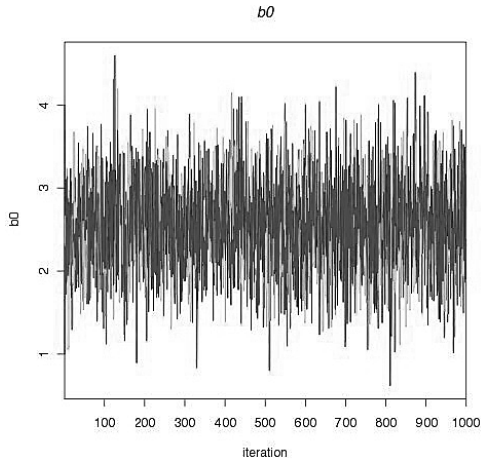


図3　MCMCのトレース結果出力例（β_1）

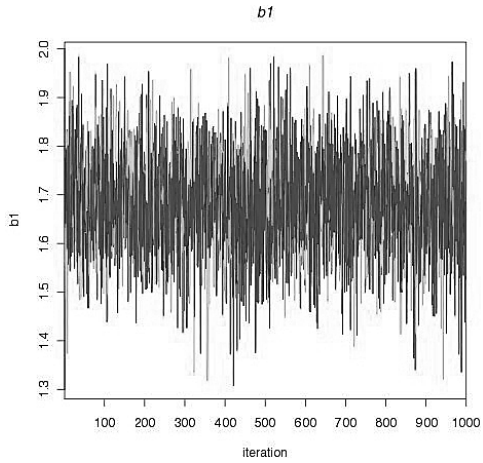


図4　MCMCのトレース結果出力例（β_2）

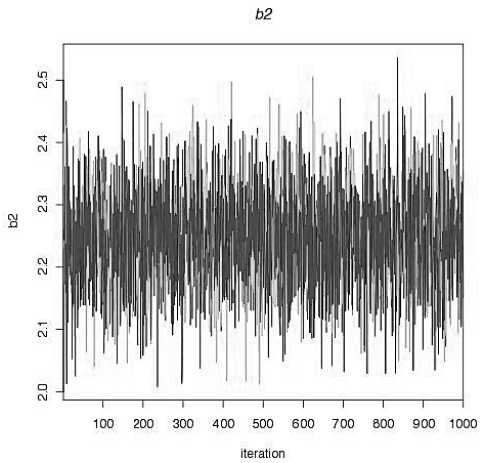
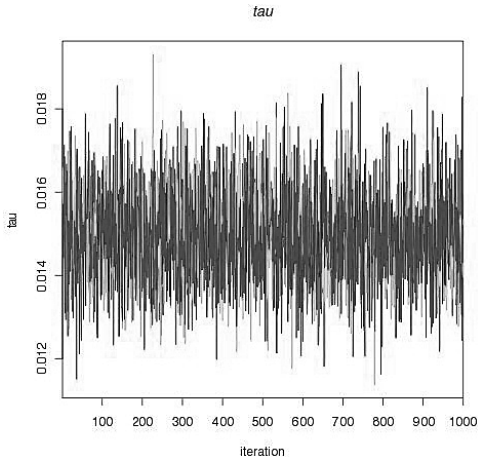


図5　MCMCのトレース結果出力例（τ）



今回は、この連載でベイズ推定を扱うのがはじめてであることもあり、また紙面の都合上、jagsの使い方の解説を中心に、線形回帰モデルのベイズ推定の紹介にとどまりました。次回から、空間計量経済モデルのベイズ推定について、詳しく紹介する予定です。

＊図2～5を、統計情報研究開発センターのホームページ（http://www.sinfonica.or.jp/）内の［刊行物］>［ESTRELA］>［参考］にカラーで掲載しています。

＊参考文献

[1] Jim Albert（2009）：Bayesian Computation with R：Springer.
[2] Ioannis Ntzoufras（2009）：Bayesian Modeling Using WinBUGS：Wiley.