Rで学ぶベイズ統計学入門

第5章ベイズ計算入門

５．６　事例
５．７　積分計算のためのモンテカルロ法
５．８　棄却サンプリング
５．９　重点サンプリング
５．１０　サンプリング重点リサンプリング（SIR）

茨城大学情報工学科
倉持辰洋

laplace関数

- ベータ･二項モデルの事例にlaplace()関数を適用する
- 図５．２等高線図にもとづき、まずネルダー－ミード法に初期の推定値

$$(logit(\eta), logK)=(-7,6)$$

を与える

Rでの実行（laplace関数の適用）

```
> fit <- laplace(betabinexch, c(-7,6), cancermortality)
> fit
$mode
[1] -6.819793  7.576111

$var
           [,1]       [,2]
[1,]  0.07896568 -0.1485087
[2,] -0.14850874  1.3483208

$int
[1] -570.7743

$converge
[1] TRUE
```

- 事後モードが

$$(-6.82, 7.58)$$

だと分かる

- 出力から $(logit(\eta), logK)$ が平均ベクトルfit$modeで分散共分散行列fit$varの2変数正規分布の近似となっている

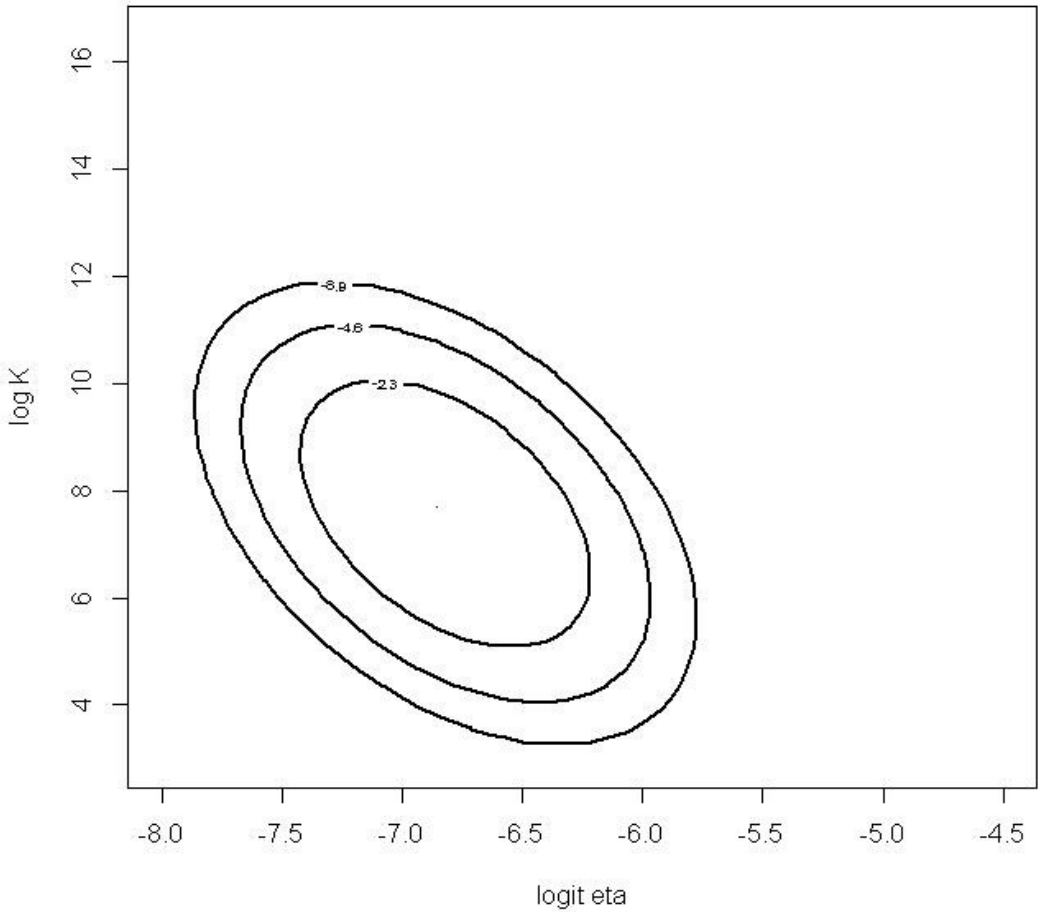
近似の正規密度の等高線

- 2変量対数正規分布のlibnorm()関数をmycontour()関数で指定して作成した図５．３は近似の正規密度の等高線を表している

- 図５．２と見比べると正確な事後密度と近似の正規事後密度の等高線の違いがはっきりわかる

Rでの実行
（近似の正規密度の等高線の作成）

```
> npar <- list(m = fit$mode, v = fit$var)
> mycontour(lbinorm, c(-8, -4.5, 3, 16.5),
+ npar, xlab = "logit eta", ylab = "log K")
```



ベータ･二項モデル問題の $logit(\eta)$ と $logK$ の正規近似の等高線図

アルゴリズムのメリット

- 多変量正規近似を使うことでパラメータについての要約をすぐ求められる

  分散共分散行列の対角成分を使いおおよその確立区間を構成できる
- 例：パラメータの90%確立区間を構成する方法

```
> se <- sqrt(diag(fit$var))
> fit$mode - 1.645 * se
[1] -7.282052  5.665982
> fit$mode + 1.645 * se
[1] -6.357535  9.486239
```

- $logit(\eta)$ の90%区間推定値：(-7.28,-6.36)
- $\log K$ の90%区間推定値：(5.67,9.49)

積分計算のためのモンテカルロ法（1）

- 事後分布要約する二つ目の一般的なアプローチはシミュレーション

  θの事後密度：g(θ|y)　のとき

  パラメータ関数h(θ)について検討する

- h(θ)の事後平均は

$$E(h(\theta)|y)=\int h(\theta)g(\theta|y)d\theta$$

  で与えられる

積分計算のためのモンテカルロ法（2）

- 事後密度から独立した標本$\theta^1,\ldots,\theta^m$をシミュレーションできると仮定すると以下のように推定できる
- 事後平均のモンテカルロ平均値:

$$\bar{h}=\frac{\sum_{j=1}^{m} h(\theta^j)}{m}$$

- 推定値の標準誤差:

$$se_{\bar{h}}=\sqrt{\frac{\sum_{j=i}^{m}\left(h(\theta^j)-\bar{h}\right)^2}{(m-1)m}}$$

モンテカルロ法の2.4節の例への利用

- 正確な事後分布からシミュレーションした標本が得られる場合に効果的
- ２．４節の例のシミュレーション結果を利用

  睡眠の足りてる学生の割合：p

  pの事後分布：beta(14.26,23.19)
- 事後平均：p^2 を知りたいとすると

  シミュレーション標本：$\{p^j\}$ とする

Rでの実行（モンテカルロ法）

- 1000個の値でシミュレーションする：j = 1000

```
> p <- rbeta(1000, 14.26, 23.19)
> est <- mean(p^2)
> se <- sd(p^2)/sqrt(1000)
> c(est, se)
[1] 0.149250359 0.001949135
```

モンテカルロ推定値：0.149

シミュレーション標準誤差：0.002

棄却サンプリング

- 多くの場合、たとえばベータ・二項分布では周知の関数形で表されない

  ⇒別のアルゴリズムが必要

  ここでは棄却サンプリングを紹介

- 棄却サンプリングとは

  ⇒所与の確立分布からランダム標本をシミュレーションする汎用的なアルゴリズム

確立密度p(θ)の探索

- 正規化定数が未定の事後密度g(θ|y)から独立標本を抽出したい
- そのために以下の条件を満たす別の確率密度p(θ)を探索する

  条件１：pからの標本のシミュレーションが容易である

  条件２：密度pは、目的とする事後密度gに近い位置と広がりを持つ

  条件３：すべてのθとある定数cについて、
  g(θ|y)<=cp(θ)である

アルゴリズム

- 条件を満たす密度pが見つかるとする

  ⇒次の受理と棄却のアルゴリズムを使ってgから抽出を行う

  1.pからθを、また単位区間上で一様分布に従う乱数Uを独立にシミュレーションする

  2.U <= g(θ|y) / (cp(θ))であれば、θを密度gからの一つの標本として受理する、そうでないなら棄却する

  3.1と2のアルゴリズムを「受理された」θが十分な数になるまで繰り返す

棄却サンプリングの特徴

- 様々な分布から標本をシミュレーションするもっとも有効な方法の一つ

- 適切な提案密度pと定数cを見つけることが課題

  効率的な棄却サンプリングアルゴリズムは受理率が高い

  pの受理率：$g(\theta|y) / (cp(\theta))$

棄却サンプリングを利用した標本シミュレーション

- θ=(logit(η),log K)から標本をシミュレートする
- 提案密度pの選択肢

  2変量正規密度:

  平均と分散はlaplace()関数の出力を利用できる

  正規密度の裾は比較的狭い⇒比g(θ|y) / p(θ)は有界にならないのが普通

  多変量t分布:

  平均と尺度行列を事後分布に適合するように選び、自由度も小さくなるようにする

  自由度が小さいと裾が広がる⇒比g(θ|y) / p(θ)の有界も見つけやすくなる

多変量t分布の設定

- 5.6節でlaplace()関数によって求めた事後平均(fit$mode)、分散共分散行列(fit$var)より

  位置パラメータ：fit$mode

  尺度行列：2 fit$var

  自由度：4

  とした多変量t密度を利用する

有界定数cの探索

- 次の条件を満たすcを見つける必要がある

  すべてのθについて　$g(\theta|y) \le cp(\theta)$

- gは対数尺度でプログラム化されるので、次を満たす定数d = logcを見つけたい

  すべてのθについて　$\log g(\theta|y) - \log p(\theta) \le d$

定数dを探索するアルゴリズム

- $\log g(\theta|y) - \log p(\theta)$をすべてのθ上で最大化したい、簡単な方法がlaprace()関数を利用することである
- ここで、この差関数を計算する関数betabinTを作成する

```
> betabinT <- function(theta, datapar)
+ {
+ data <- datapar$data
+ tpar <- datapar$par
+ d <- betabinexch(theta, data) - dmt(theta, mean =
c(tpar$m),
+ S <- tpar$var, df = tpar$df, log = TRUE)
+ return(d)
+ }
```

パラメータ：theta　リスト：datapar

リストdataparを定義する

- ここでは以下のようにt提案密度のパラメータとdataparリストを定義する

```
> tpar <- list(m = fit$mode, var = 2 * fit$var, df = 4)
> datapar <- list(data = cancermortality, par = tpar)
```

betabinT関数の実行

- laplace()関数を作成した関数とともに実行する、その際賢明な初期値を指定する

```
> start <- c(-6.9, 12.4)
> fit1 <- laplace(betabinT, start, datapar)
> fit1$mode
[1] -6.888963 12.421993
```

- 最大値dはθ=(-6.889, 12.422)で得られる

  このθはシミュレーションした標本空間の極端な位置に無いので最大値の近似を本当に見つけられたと考えられる

定数dを求める

- 実際のdの値は以下のコードで求まる

```
> betabinT(fit1$mode, datapar)
[1] -569.2829
```

dの値は-569.2829となる

rejectsampling()関数を利用しての実装

- 棄却サンプリングをrejectsampling()関数を利用して実装する
- 与える引数

  対数事後密度定義：logf

  tをカバーする密度パラメータ：tpar

  dの最大値：dmax

  シミュレーションされる候補値の数：n

  対数密度関数へのデータ；data

rejectsampling()関数について

- rejectsampling()関数は以下の４つのステップを踏んでいる

  1.提案密度からθベクトルをシミュレーション

  2.シミュレーションされた標本を元にlog gとlog fの値を計算

  3.log gとlog fから受理確立を計算

  4.返り値として一様標本が受理確立よりも小さい場合のθのシミュレーション値を与える

rejectsampling()の実行

- 求めたdの値を利用し実行する

  提案密度から10000個の値をシミュレーションする

```
> theta <- rejectsampling(betabinexch, tpar, -569.2829, 10000, cancermortality)
> dim(theta)
[1] 2403    2
```

出力されたthetaの行数は2403なので、アルゴリズムの受理率は2403/10000=0.24となり低い。

提案密度は簡単にきまったが効率的とはいえない

シミュレーションした標本の分布

```
> mycontour(betabinexch, c(-8, -4.5, 3, 16.5), cancermortality,
+ xlab = "logit eta", ylab = "log K")
> points(theta[,1],theta[,2])
```

- 抽出された標本は正確な密度の等高線内部に納まっている