第9章 GLMベイズ化・事後分布推定

2015/03/03 高倉担当

前章の内容：MCMCで事後分布が推定できる

- 事後分布∝尤度×事前分布
  - 事前分布が一定なら、事後分布∝尤度
  - だから、結果は最尤推定と同じになる
- パラメータを分布として推定？
  - ベイズ統計の中ではおかしくない考え方

この章：7章までのモデリングの話
＋ MCMCによる推定を合体させる

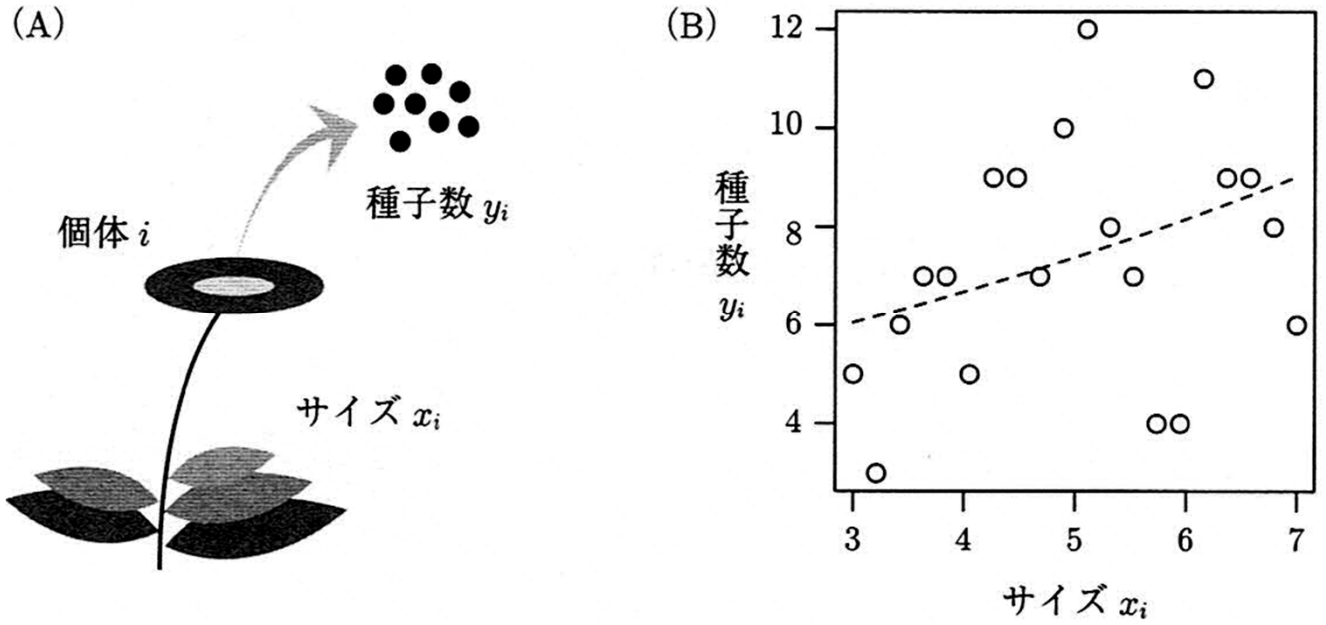
9.1例題 種子数のポアソン回帰（個体差なし）

ごく簡単な（GLMで事足りる）例で考える

- 20個体でサイズx_iと種子数y_iの関係

  真（未知）の関係：$\lambda=\exp(1.5 + 0.1x)$

図9.1



9.1例題 種子数のポアソン回帰（個体差なし）

ごく簡単な（GLMで事足りる）例で考える

- 20個体でサイズx_iと種子数y_iの関係

GLMで解析するなら

glm(y ~ x, family = poisson, data= …)

で終わり

9.2 GLMのベイズモデル化

ベイズ化しても、中身は同じごく簡単な

- 線形予測子：$\beta_1+\beta_2 x_i$
- リンク関数：log()

➔平均$\lambda_i=\exp(\beta_1+\beta_2 x_i)$

尤度関数$L(\beta_1, \beta_2)$は

$$L(\beta_1, \beta_2) = \prod_i p(y_i|\lambda_i) = \prod_i p(y_i|\beta_1, \beta_2, x_i)$$
$$= p(\boldsymbol{Y}|\beta_1, \beta_2)$$

9.2 GLMのベイズモデル化

事後分布∝尤度×事前分布だから

$$p(\beta_1, \beta_2|\boldsymbol{Y}) \propto p(\boldsymbol{Y}|\beta_1, \beta_2)p(\beta_1)p(\beta_2)$$

この事前分布には何を使うのか？

事前分布：データが得られてない時の
パラメータの分布
？？？

9.3 無情報事前分布

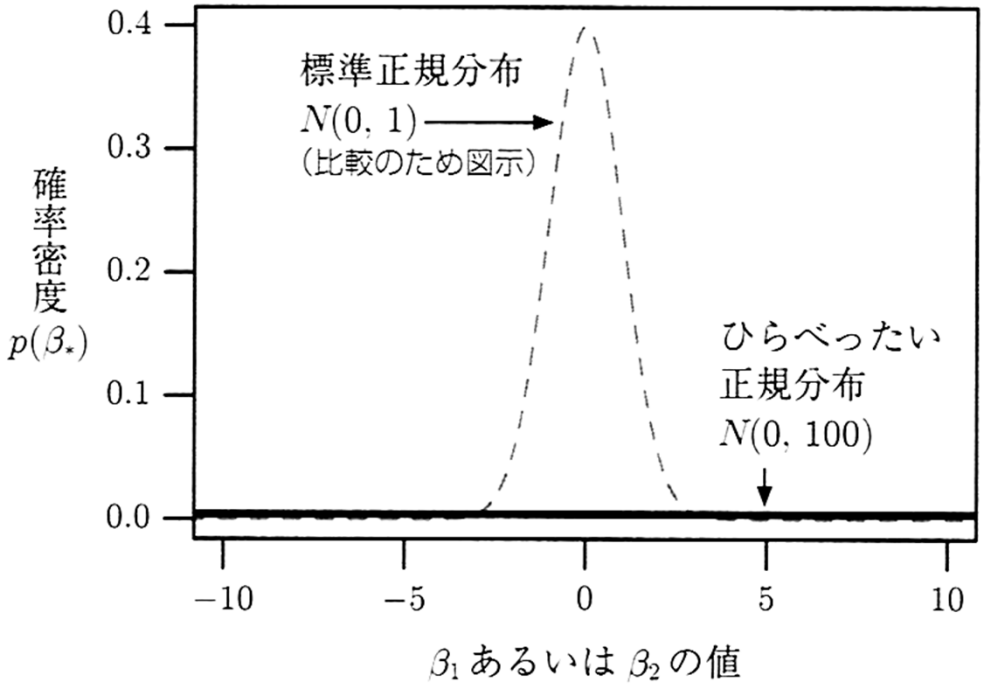
この事前分布には何を使うのか？
「なんでもOK」な分布を当てはめておく

無情報事前分布 non-informative (flat) prior

この場合、平べったい正規分布を与えておこう

9.3 無情報事前分布

無情報事前分布 non-informative (flat) prior
平べったい正規分布, $N(0, 100)$



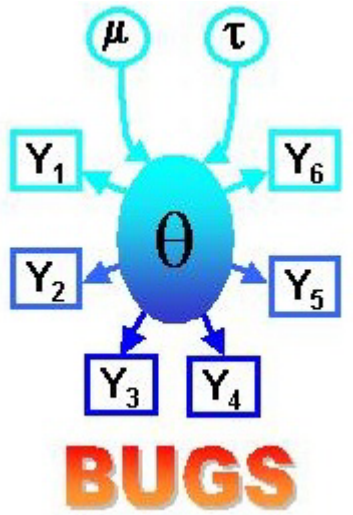
第8章では、0～1の
範囲で、一様な分布を
事前分布として考えた。
同じ考え方。

9.4 ベイズモデルの事後分布の推定

専用のソフトウェアを使うほうが早い

WinBUGSを使います

- MCMCサンプリングする
- BUGSコードで記述
- 柔軟・簡単にモデルを記述
- アルゴリズム指定など不要



9.4.1 ベイズモデルのコーディング

```
model
{
  for (i in 1:N) {
    Y[i] ~ dpois(lambda[i])
    log(lambda[i]) <- beta1 + beta2 * (X[i] -
Mean.X)
  }
  beta1 ~ dnorm(0, 1.0E-4)
  beta2 ~ dnorm(0, 1.0E-4)
}
```

9.4.1 ベイズモデルのコーディング

model {}内（ブロック）がモデルを記述していることを示す

```
model
{
  for (i in 1:N) {
    Y[i] ~ dpois(lambda[i])
    log(lambda[i]) <- beta1 + beta2 * (X[i] -
Mean.X)
  }
  beta1 ~ dnorm(0, 1.0E-4)
  beta2 ~ dnorm(0, 1.0E-4)
}
```

9.4.1 ベイズモデルのコーディング

```
model
{
  for (i in 1:N) {
    Y[i] ~ dpois(lambda[i])
    log(lambda[i]) <- beta1 + beta2 * (X[i] -
Mean.X)
  }
  beta1 ~ dnorm(0, 1.0E-4)
  beta2 ~ dnorm(0, 1.0E-4)
}
```

ブロック内の記述を繰り返す。
最初はi=1の設定で実行する。
以後、iを1ずつ繰り上げながら
Nになるまで繰り返す。

9.4.1 ベイズモデルのコーディング

```
model
{
  for (i in 1:N) {
    Y[i] ~ dpois(lambda[i])
    log(lambda[i]) <- beta1 + beta2 * (X[i] -
Mean.X)
  }
  beta1 ~ dnorm(0, 1.0E-4)
  beta2 ~ dnorm(0, 1.0E-4)
}
```

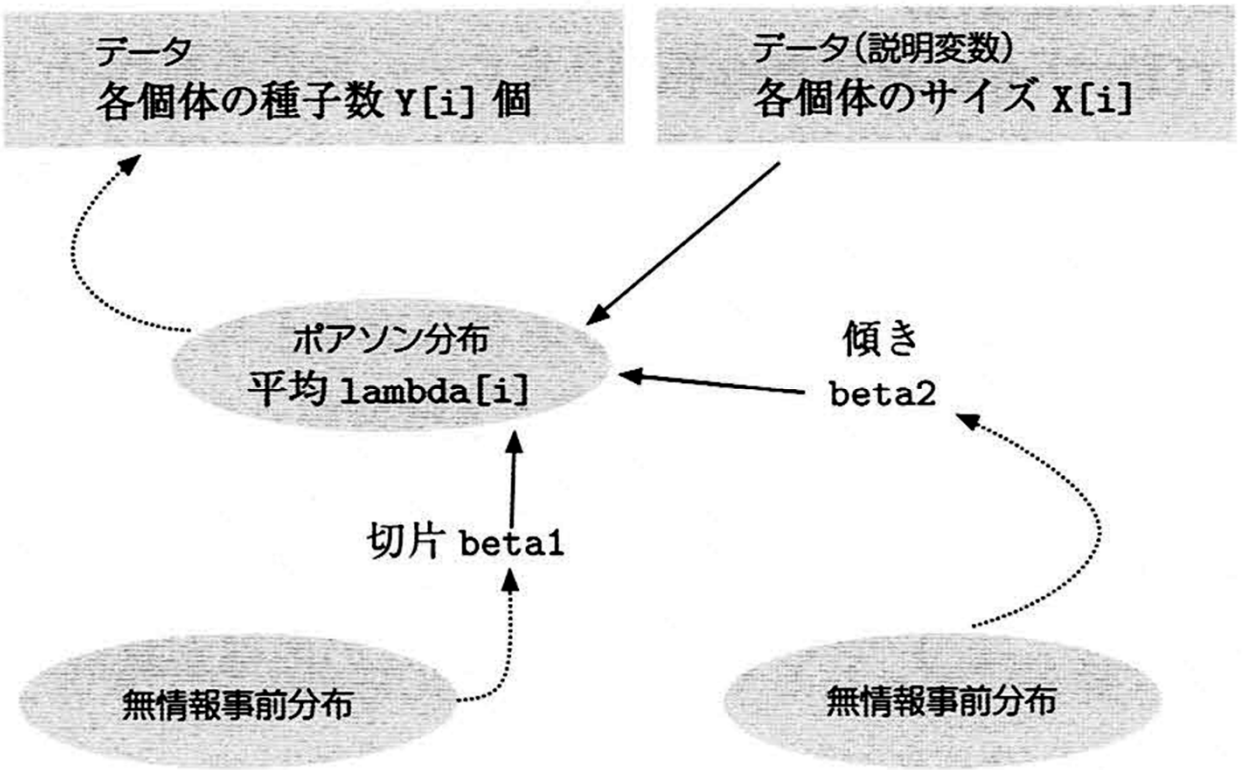
リンク関数log
線形予測子$\beta_1+\beta_2 x_i$
xは中央化してある
（処理速度向上のため）

2つの演算子～と<-が
使い分けられている
ことに注意

9.4.1 2つの演算子～と<-

～：確率論的な関係（実線矢印）

<-:決定論的な関係（破線矢印）



9.4.1 ベイズモデルのコーディング

```
model
{
  for (i in 1:N) {
    Y[i] ~ dpois(lambda[i])
    log(lambda[i]) <- beta1 + beta2 * (X[i] -
Mean.X)
  }
  beta1 ~ dnorm(0, 1.0E-4)
  beta2 ~ dnorm(0, 1.0E-4)
}
```

パラメータβ1,β2の
事前分布を指定している
平べったい正規分布

WinBUGSでは
ばらつきの大きさを
sdではなく1/Vで記述する