階層モデルあてはめの実例

伊東宏樹

2022-09-04

本日の内容

はじめに

サイト 占有モデル

（二項）N 混合モデル

おわりに

はじめに

階層モデル（復習）

- 生態過程（潜在状態）
  - 観測できないシステム
- 観測過程
  - 生態過程を観測（観測誤差を含む）

統計ソフトウェアによるあてはめ

今回使用するソフトウェア

- JAGS
- NIMBLE
- Stan

いずれも、MCMC によるベイズ推定が可能

このほか、R の unmarked パッケージでは最尤推定が可能

JAGS

- https://mcmc-jags.sourceforge.io/
- スタンドアロンのプログラム
  - R からは、rjags などのパッケージから利用可能
- モデル記述に BUGS 言語を使用

NIMBLE

- https://r-nimble.org/
- R パッケージ
- BUGS を拡張したモデル記述言語
- モデルコードを C++に変換後、コンパイルして実行
- 生態学に特化した nimbleEcology パッケージもあり

Stan

- https://mc-stan.org/
- スタンドアロンのプログラム
  - R からは、Rstan または cmdstanr パッケージから利用可能
- Stan 独自のモデル記述言語
- モデルコードを C++に変換後、コンパイルして実行
- 離散パラメータの推定はできない

サイト 占有モデル

サイト 占有モデル

あるサイトに、ある生物の個体がいる（在、占有）か、いない（不在、占有していない）かを推定する

- 潜在状態: サイト m では、占有確率 ψ_m に応じて個体の在・不在 z_m が決定される
  - z_m ~ Bernoulli(ψ_m)
- 観測過程: j 回目の観測では潜在状態 z_m と発見確率 p_{mj} に応じて個体の発見・不発見 Y_{mj} が決定される
  - Y_{mj} ~ Bernoulli($z_m p_{mj}$)

偽陽性の誤差（ないものをあるとしてしまう）はないとする

模擬データの作成

```
occ_data <- AHMbook::simOcc(
  M = 200,                  # サイト数
  J = 3,                    # 調査回数
  mean.occupancy = 0.6,     # 共変量が 0 のときの平均占有確率
  beta1 = -2,               # 占有確率に対する標高の係数
  beta2 = 2,                # 占有確率に対する森林被覆率の係数
  beta3 = 0,                # 占有確率に対する交互作用の係数
  mean.detection = 0.3,     # 共変量が 0 のときの平均発見確率
  time.effects = c(0, 0), # 時間効果
  alpha1 = -1,              # 発見確率に対する標高の係数
  alpha2 = -3,              # 発見確率に対する風速の係数
  alpha3 = 0,               # 発見確率に対する交互作用の係数
  sd.lp = 0,                # サイトの変量効果
  b = 0,                    # 行動反応
  show.plot = FALSE)
```

データの確認

真の占有サイト数（$\sum_{m=1}^{M} z_m$）

```
> occ_data$sumZ
```

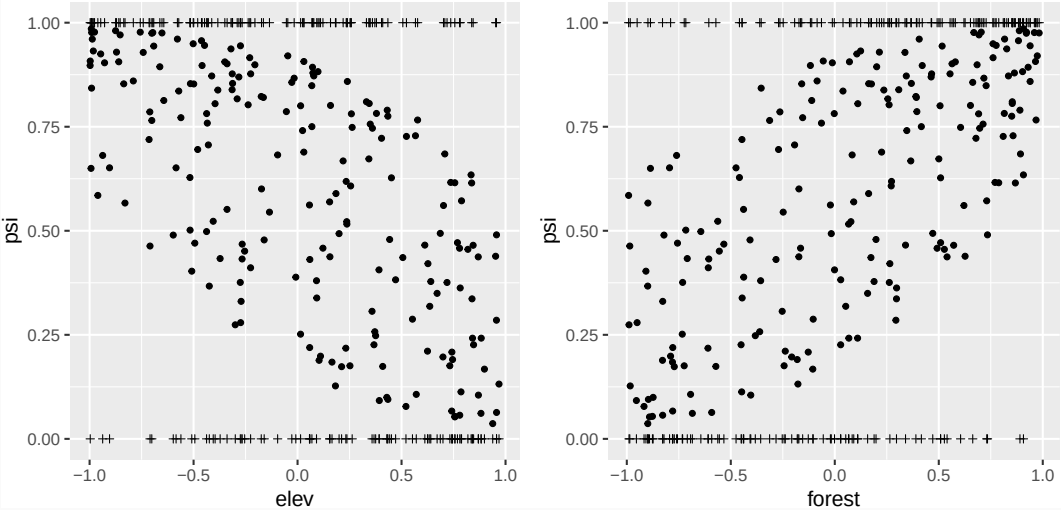
```
[1] 113
```

観察された占有サイト数（少なくとも 1 回は発見のあったサイト数）

```
> occ_data$sumZ.obs
```

```
[1] 92
```

共変量と占有確率との関係



BUGS モデル

```
model {
  for (m in 1:M) {
    logit(psi[m]) <- beta[1] + beta[2] * elev[m] + beta[3] * forest[m]
    z[m] ~ dbern(psi[m])
    for (j in 1:J) {
      logit(p[m, j]) <- beta[4] + beta[5] * elev[m] + beta[6] * wind[m, j]
      Y[m, j] ~ dbern(z[m] * p[m, j])
    }
  }
  for (i in 1:6) {
    beta[i] ~ dnorm(0, 1.0e-4)
  }
  Nocc <- sum(z[])
}
```

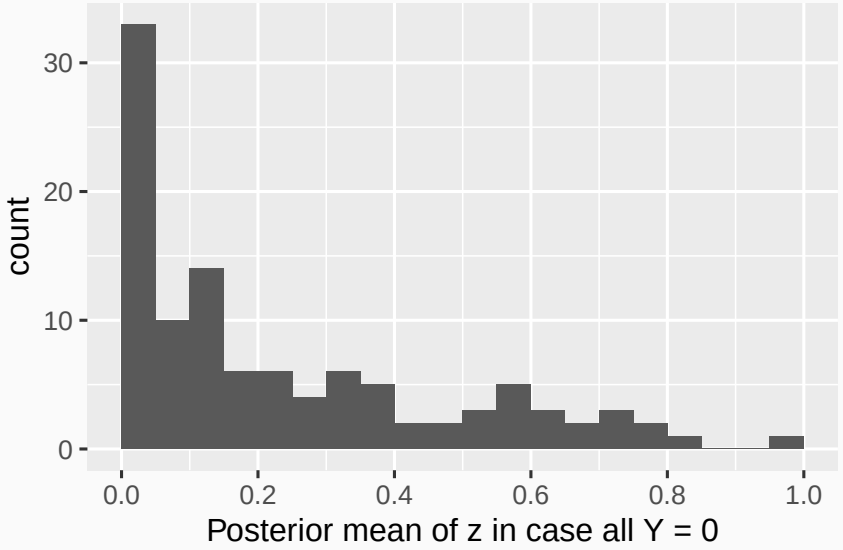
JAGS によるあてはめ

```
model <- jags.model(occ_bugs,
                    data = list(M = occ_data$M, J = occ_data$J,
                                Y = occ_data$y, elev = occ_data$elev,
                                forest = occ_data$forest,
                                wind = occ_data$wind),
                    inits = list(z = apply(occ_data$y, 1, max)),
                    n.chains = 3, n.adapt = 1000)
update(model, n.iter = 1000)
fit1 <- coda.samples(model, variable = c("beta", "Nocc", "z"),
                     n.iter = 1000)
```

結果

	Mean	SD	Naive SE	Time-series SE
Nocc	117.2963333	6.1691603	0.112632943	0.318387458
beta[1]	0.4630851	0.3076713	0.005617284	0.015161590
beta[2]	-2.4026271	0.6026232	0.011002344	0.028215890
beta[3]	2.7992588	0.5783976	0.010560047	0.026732731
beta[4]	-0.8564236	0.1856286	0.003389099	0.008430319
beta[5]	-0.7919187	0.3256273	0.005945114	0.012924235
beta[6]	-2.9607600	0.3306082	0.006036053	0.009555262

発見のなかったサイトでの z の事後平均の頻度分布



NIMBLE によるあてはめ

同じ BUGS コードを使用

```
data <- list(M = occ_data$M, J = occ_data$J,
             Y = occ_data$y, elev = occ_data$elev,
             forest = occ_data$forest, wind = occ_data$wind)
init <- list(beta = runif(6, -2, 2), z = apply(occ_data$y, 1, max))
model <- readBUGSmodel(occ_bugs, data = data, inits = init)
fit2 <- nimbleMCMC(model = model, monitors = c("beta", "Nocc"),
                   niter = 2000, nburnin = 1000, nchain = 3,
                   summary = TRUE)
```

結果

	Mean	Median	St.Dev.	95%CI_low	95%CI_upp
Nocc	117.8463333	117.0000000	6.3870430	107.00000000	131.0000000
beta[1]	0.4963988	0.4670631	0.3245141	-0.05523859	1.1560600
beta[2]	-2.4131694	-2.3754546	0.6683046	-3.79873697	-1.2802603
beta[3]	2.7875865	2.7413316	0.5889849	1.81318831	4.1165894
beta[4]	-0.8665529	-0.8604129	0.1909911	-1.24532476	-0.5018154
beta[5]	-0.7917638	-0.7943563	0.3361115	-1.44344513	-0.1195437
beta[6]	-2.9568061	-2.9515492	0.3382397	-3.58193998	-2.2896453

Stan

離散パラメータの z を消去する

- 各サイトについて
  - Y_j がすべて 0 のとき

$$L(\psi, \boldsymbol{p} \mid \boldsymbol{Y}) = \mathrm{Bern}(0 \mid \psi) + \mathrm{Bern}(1 \mid \psi) \prod_{j=1}^{J} \mathrm{Bern}(0 \mid p_j)$$

  - Y_j の少なくとも 1 つが 1 のとき

$$L(\psi, \boldsymbol{p} \mid \boldsymbol{Y}) = \mathrm{Bern}(1 \mid \psi) \prod_{j=1}^{J} \mathrm{Bern}(Y_j \mid p_j)$$

Stan のモデル

data ブロック

```
data {
  int<lower=0> M;
  int<lower=0> J;
  array[M, J] int<lower=0, upper=1> Y;
  vector[M] Elev;
  vector[M] Forest;
  matrix[M, J] Wind;
}
```

parameters ブロックと transformed parameters ブロック

```
parameters {
  array[6] real beta;
}

transformed parameters {
  vector<lower=0, upper=1>[M] psi;
  matrix<lower=0, upper=1>[M, J] p;
  psi = inv_logit(beta[1] + beta[2] * Elev + beta[3] * Forest);
  for (m in 1:M)
    p[m] = inv_logit(beta[4] + beta[5] * Elev[m] + beta[6] * Wind[m]);
}
```

model ブロック

```
model {
  for (m in 1:M) {
    if (sum(Y[m]) == 0) // not detected
      target += log_sum_exp(bernoulli_lpmf(0 | psi[m]),
                            bernoulli_lpmf(1 | psi[m])
                            + bernoulli_lpmf(0 | p[m]));
    else                // detected
      target += bernoulli_lpmf(1 | psi[m])
                + bernoulli_lpmf(Y[m] | p[m]);
  }
}
```

generated quantities ブロック

```
generated quantities {
  array[M] int<lower=0, upper=1> z;
  int<lower=0, upper=M> Nocc;
  for (m in 1:M)
    if (sum(Y[m]) > 0) {  // detected
      z[m] = 1;
    } else {              // not detected
      real lp1 = bernoulli_lpmf(0 | psi[m]);
      real lp2 = bernoulli_lpmf(1 | psi[m])
                 + bernoulli_lpmf(Y[m] | p[m]);
      z[m] = bernoulli_rng(exp(lp2) / (exp(lp1) + exp(lp2)));
    }
  Nocc = sum(z);
}
```

Stan によるあてはめ

```
model <- cmdstan_model(file.path("models", "occ.stan"))
fit3 <- model$sample(data = list(M = occ_data$M,
                                 J = occ_data$J,
                                 Y = occ_data$y,
                                 Elev = occ_data$elev,
                                 Forest = occ_data$forest,
                                 Wind = occ_data$wind),
                     seed = 1,
                     chains = 4, parallel_chains = 4,
                     iter_sampling = 1000,
                     iter_warmup = 1000,
                     output_dir = output_dir)
```

結果

```
# A tibble: 7 x 10
  variable    mean  median    sd   mad       q5     q95  rhat ess_bulk ess_tail
  <chr>      <dbl>   <dbl> <dbl> <dbl>    <dbl>   <dbl> <dbl>    <dbl>    <dbl>
1 beta[1]    0.503   0.478 0.329 0.318   0.0109   1.08   1.00    2654.    2438.
2 beta[2]   -2.38   -2.34  0.605 0.568  -3.42    -1.47   1.00    3039.    2521.
3 beta[3]    2.83    2.77  0.591 0.559   1.94     3.87   1.00    2982.    1739.
4 beta[4]   -0.883  -0.879 0.196 0.197  -1.21    -0.570  1.00    3020.    3005.
5 beta[5]   -0.839  -0.838 0.341 0.335  -1.42    -0.287  1.00    3017.    2413.
6 beta[6]   -2.97   -2.95  0.341 0.342  -3.54    -2.42   1.00    4289.    3087.
7 Nocc     118.    118     6.77  5.93  108      130      1.00    2622.    2709.
```

（二項）N混合モデル

（二項）N 混合モデル

あるサイトでの、ある生物の個体数 N を推定する

- 潜在状態: サイト m における真の個体数 N_m
  - N_m ~ Poisson(λ_m)
- 観測過程: j 回目の観測では、そのうち Y_{mj} 個体を、発見確率 p_{mj} で発見する
  - Y_{mj} ~ Binomial(N_m, p_{mj})

偽陽性の誤差はないとする

模擬データの作成

```
nmix_data <- AHMbook::simNmix(
  nsites = 267,         # サイト数
  nvisits = 3,          # 各サイトでの観察回数
  mean.lam = 2,         # 平均個体数
  mean.p = 0.6,         # 平均発見確率
  beta2.lam = 1,        # 個体数についての共変量 2 の係数
  beta.p.survey = -2,   # 発見確率についての観測共変量の係数
  show.plots = FALSE, verbose = FALSE)
```

データの確認

真の個体数の合計 ($\sum_{m=1}^{M} N_m$)

```
> nmix_data$Ntotal
```

```
[1] 859
```

各サイト での最大観測数の合計

```
> sum(apply(nmix_data$C, 1, max))
```

```
[1] 763
```

BUGS コード

```
model {
  for (m in 1:M) {
    log(lambda[m]) <- beta[1] + beta[2] * Xsite[m]
    N[m] ~ dpois(lambda[m])
    for (j in 1:J) {
      logit(p[m, j]) <- beta[3] + beta[4] * Xsurvey[m, j]
      Y[m, j] ~ dbin(p[m, j], N[m])
    }
  }
  for (i in 1:4) {
    beta[i] ~ dnorm(0, 1.0e-4)
  }
  Ntotal = sum(N)
}
```

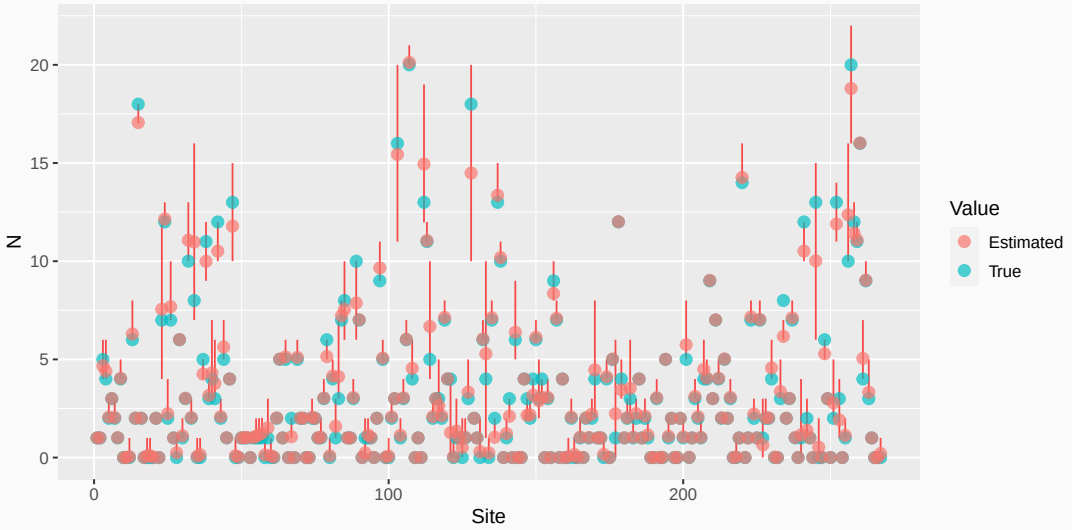
あてはめ

```
model <- jags.model(nmix_bugs,
                    data = list(M = nmix_data$nsites,
                                J = nmix_data$nvisits,
                                Y = nmix_data$C,
                                Xsite = nmix_data$site.cov[, 2],
                                Xsurvey = nmix_data$survey.cov),
                    inits = list(N = apply(nmix_data$C, 1, max)),
                    n.chains = 3, n.adapt = 1000)
update(model, n.iter = 1000)
fit4 <- coda.samples(model, variable = c("beta", "N", "Ntotal"),
                     n.iter = 1000)
```

結果

	Mean	SD	Naive SE	Time-series SE
Ntotal	861.0506667	13.21555838	0.2412819811	0.554390808
beta[1]	0.7162637	0.05028378	0.0009180520	0.002219351
beta[2]	0.9525292	0.03842966	0.0007016263	0.001645437
beta[3]	0.4845376	0.09169699	0.0016741504	0.004277889
beta[4]	-2.0610536	0.09069491	0.0016558550	0.003648820

各サイトでの N の真値と推定値（事後平均と 95%信用区間）



NIMBLE モデル

```
code <- nimbleCode({
  for (m in 1:M) {
    log(lambda[m]) <- beta[1] + beta[2] * Xsite[m]
    N[m] ~ dpois(lambda[m])
    for (j in 1:J) {
      logit(p[m, j]) <- beta[3] + beta[4] * Xsurvey[m, j]
      Y[m, j] ~ dbinom(size = N[m], prob = p[m, j])
    }
  }
  for (i in 1:4) {
    beta[i] ~ dnorm(mean = 0, sd = 1.0e+2)
  }
  Ntotal <- sum(N[])
})
```

NIMBLE によるあてはめ

```
M <- nmix_data$nsites
J <- nmix_data$nvisits
const <- list(M = M, J = J)
data <- list(Y = nmix_data$C, Xsite = nmix_data$site.cov[, 2],
             Xsurvey = nmix_data$survey.cov)
init <- function() {
  list(beta = runif(4, -2, 2), N = apply(nmix_data$C, 1, max))
}
fit5 <- nimbleMCMC(code, constants = const, data = data, inits = init,
                   monitors = c("beta", "N", "Ntotal"),
                   niter = 2000, nburnin = 1000, nchain = 3,
                   summary = TRUE)
```

結果

	Mean	Median	St.Dev.	95%CI_low	95%CI_upp
Ntotal	860.9533333	860.0000000	12.62691445	837.0000000	886.0000000
beta[1]	0.7196754	0.7201693	0.05241679	0.6103157	0.8161716
beta[2]	0.9488411	0.9457027	0.04076814	0.8771912	1.0306003
beta[3]	0.4821721	0.4820897	0.08531192	0.3227855	0.6485387
beta[4]	-2.0621422	-2.0646993	0.08480341	-2.2279864	-1.8959172

Stan

離散パラメータの N を消去

- 各サイトについて

$$L(\lambda, \boldsymbol{p} \mid \boldsymbol{Y}) = \sum_{k=\max(\boldsymbol{Y})}^{\infty} \left(\text{Pois}(k \mid \lambda) \prod_{j=1}^{J} \text{Binom}(Y_j \mid k, p_j) \right)$$

Stan のモデル

```
data {
  int<lower=0> M;
  int<lower=0> J;
  array[M, J] int<lower=0> Y;
  vector[M] Xsite;
  matrix[M, J] Xsurvey;
  int<lower=0> K;  // upper limit of N for calculation
}

parameters {
  array[4] real beta;
}
```

```
transformed parameters {
  matrix[M, K + 1] lp;  // (k - 1) = 0, 1, 2,..., K
  for (m in 1:M) {
    real log_lambda = beta[1] + beta[2] * Xsite[m];
    vector[J] logit_p = beta[3] + beta[4] * Xsurvey[m]';
    int y_max = max(Y[m]);
    for (k in 1:y_max)
      lp[m, k] = negative_infinity();
    for (k in y_max:K)
      lp[m, k + 1] = poisson_log_lpmf(k | log_lambda)
                     + binomial_logit_lpmf(Y[m] | k, logit_p);
  }
}
```

```
model {
  for (m in 1:M)
    target += log_sum_exp(lp[m]);
}

generated quantities {
  array[M] int N;
  int Ntotal;
  for (m in 1:M)
    N[m] = categorical_rng(softmax(lp[m]')) - 1;
  Ntotal = sum(N);
}
```

Stan によるあてはめ

```
model <- cmdstan_model(file.path("models", "nmix.stan"))
fit6 <- model$sample(data = list(M = nmix_data$nsites,
                                 J = nmix_data$nvisits,
                                 Y = nmix_data$C,
                                 K = 100 + max(nmix_data$C),
                                 Xsite = nmix_data$site.cov[, 2],
                                 Xsurvey = nmix_data$survey.cov),
                     seed = 1, chains = 4, parallel_chains = 4,
                     iter_sampling = 1000, iter_warmup = 1000,
                     output_dir = "output_dir")
```

結果

```
# A tibble: 5 x 10
  variable    mean  median      sd     mad      q5     q95  rhat ess_b~1 ess_t~2
  <chr>      <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl> <dbl>   <dbl>   <dbl>
1 Ntotal   862.    862     13.0    13.3    841     885      1.00   3280.   3637.
2 beta[1]    0.719   0.718  0.0537  0.0537   0.631   0.807  1.00   2538.   2666.
3 beta[2]    0.949   0.949  0.0404  0.0417   0.882   1.02   1.00   2571.   2795.
4 beta[3]    0.477   0.475  0.0892  0.0898   0.331   0.624  1.00   2642.   2500.
5 beta[4]   -2.06   -2.06   0.0902  0.0908  -2.21   -1.91   1.00   2757.   2813.
# ... with abbreviated variable names 1: ess_bulk, 2: ess_tail
```

Stan での N 混合モデルの高速化

- 連鎖内での並列化
  - `reduce_sum` の使用
    - https://mc-stan.org/users/documentation/case-studies/reduce_sum_tutorial.html
- アルゴリズムの見直し
  - 多変量ポアソン分布によるあてはめ
    - 2 変量ポアソン分布の実装例: https://github.com/stan-dev/example-models/blob/master/BPA/Ch.12/Nmix1.stan

おわりに

おわりに

- 階層モデリング
  - 潜在状態と観測過程を明示的にモデリングする
    - 潜在状態のばらつきと、観測誤差を分離
- ソフトウェアの利用
  - JAGS, NIMBLE, Stan
  - MCMC によるベイズ推定
  - Stan では、離散パラメータの消去に一工夫がいる

今回の資料

今回の発表に使用したコードは Github で公開しています。

https://github.com/ito4303/jfssa2022