Lab – Modelos de ocupación

Seguimiento de la Diversidad Biológica
Estima de parámetros demográficos

José Jiménez
CSIC-IREC

Para qué usar estos modelos

Frecuentemente utilizamos regresiones logísticas entre la presencia de una especie y características ecológicas, pero los datos que obtenemos están sujetos a error.

Concretamente, los datos de fauna suelen están mediatizados por los falsos negativos (las especies están presentes, pero no se detectan).

Ignorar esta fuente de errores puede tener como consecuencia inferencias equivocadas.

- La ocupación será infraestimada
- Las asociaciones de hábitat obtenidas en las regresiones pueden ser incorrectas

Para qué usar estos modelos

Frecuentemente utilizamos regresiones logísticas entre la presencia de una especie y características ecológicas, pero los datos que obtenemos están sujetos a error.

Concretamente, los datos de fauna suelen están mediatizados por los falsos negativos (las especies están presentes, pero no se detectan).

Ignorar esta fuente de errores puede tener como consecuencia inferencias equivocadas.

- La ocupación será infraestimada
- Las asociaciones de hábitat obtenidas en las regresiones pueden ser incorrectas

Para qué usar estos modelos

Frecuentemente utilizamos regresiones logísticas entre la presencia de una especie y características ecológicas, pero los datos que obtenemos están sujetos a error.

Concretamente, los datos de fauna suelen están mediatizados por los falsos negativos (las especies están presentes, pero no se detectan).

Ignorar esta fuente de errores puede tener como consecuencia inferencias equivocadas.

- La ocupación será infraestimada
- Las asociaciones de hábitat obtenidas en las regresiones pueden ser incorrectas

Considerando la probabilidad de detección

El modelo de proceso de estado -o ecológico- es el mismo que en la regresión logística:

$$\text{logit}(\psi_i) = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots$$
$$z_i \sim \text{Bern}(\psi_i)$$

Modelo para el proceso de observación, condicional en el proceso de estado:

$$\text{logit}(p_{ij}) = \alpha_0 + \alpha_1 x_{ij1} + \alpha_2 x_{ij2} + \cdots$$
$$y_{ij} \sim \text{Bern}(z_i \times p_{ij})$$

Definiciones

ψ_i – probabilidad de que la especie esté en el sitio i

z_i – variable binaria de presencia/ausencia en el sitio i

p_{ij} – probabilidad de detectar la especie en el sitio i en la ocasión j

y_{ij} – dato binario de detección/no-detección

Considerando la probabilidad de detección

El modelo de proceso de estado -o ecológico- es el mismo que en la regresión logística:

$$\mathrm{logit}(\psi_i) = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots$$
$$z_i \sim \mathrm{Bern}(\psi_i)$$

Modelo para el proceso de observación, condicional en el proceso de estado:

$$\mathrm{logit}(p_{ij}) = \alpha_0 + \alpha_1 x_{ij1} + \alpha_2 x_{ij2} + \cdots$$
$$y_{ij} \sim \mathrm{Bern}(z_i \times p_{ij})$$

Definiciones
ψ_i – probabilidad de que la especie esté en el sitio i
z_i – variable binaria de presencia/ausencia en el sitio i
p_{ij} – probabilidad de detectar la especie en el sitio i en la ocasión j
y_{ij} – dato binario de detección/no-detección

Considerando la probabilidad de detección

El modelo de proceso de estado -o ecológico- es el mismo que en la regresión logística:

$$\text{logit}(\psi_i) = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots$$
$$z_i \sim \text{Bern}(\psi_i)$$

Modelo para el proceso de observación, condicional en el proceso de estado:

$$\text{logit}(p_{ij}) = \alpha_0 + \alpha_1 x_{ij1} + \alpha_2 x_{ij2} + \cdots$$
$$y_{ij} \sim \text{Bern}(z_i \times p_{ij})$$

Definiciones
ψ_i – probabilidad de que la especie esté en el sitio i
z_i – variable binaria de presencia/ausencia en el sitio i
p_{ij} – probabilidad de detectar la especie en el sitio i en la ocasión j
y_{ij} – dato binario de detección/no-detección

Simulación

Caso más simple (sin covariables):

1. Especificamos los parámetros y dimensiones

```
> psi <- 0.5         ## Probabilidad de presencia
> p <- 0.2           ## Probabililidad de detección
> nSites <- 20
> nVisits <- 4
```

2. Ahora simulamos presencias-ausencias (proceso de estado)

```
> set.seed(3439)    ## Reproducible
> z <- rbinom(nSites, size=1, psi) ## presencia/ausencia
```

3. Simulamos el proceso de observación

```
> y <- matrix(NA, nrow=nSites, ncol=nVisits)
> for(i in 1:nSites) {
+     y[i,] <- rbinom(nVisits, size=1, prob=z[i]*p)
+ }
```

Simulación

Caso más simple (sin covariables):

1. Especificamos los parámetros y dimensiones

```
> psi <- 0.5          ## Probabilidad de presencia
> p <- 0.2            ## Probabililidad de detección
> nSites <- 20
> nVisits <- 4
```

2. Ahora simulamos presencias-ausencias (proceso de estado)

```
> set.seed(3439)    ## Reproducible
> z <- rbinom(nSites, size=1, psi) ## presencia/ausencia
```

3. Simulamos el proceso de observación

```
> y <- matrix(NA, nrow=nSites, ncol=nVisits)
> for(i in 1:nSites) {
+     y[i,] <- rbinom(nVisits, size=1, prob=z[i]*p)
+ }
```

Simulación

Caso más simple (sin covariables):

1. Especificamos los parámetros y dimensiones

```
> psi <- 0.5         ## Probabilidad de presencia
> p <- 0.2           ## Probabililidad de detección
> nSites <- 20
> nVisits <- 4
```

2. Ahora simulamos presencias-ausencias (proceso de estado)

```
> set.seed(3439)    ## Reproducible
> z <- rbinom(nSites, size=1, psi) ## presencia/ausencia
```

3. Simulamos el proceso de observación

```
> y <- matrix(NA, nrow=nSites, ncol=nVisits)
> for(i in 1:nSites) {
+     y[i,] <- rbinom(nVisits, size=1, prob=z[i]*p)
+ }
```

Datos simulados

Observaciones

```
> y
      [,1] [,2] [,3] [,4]
 [1,]    0    0    0    0
 [2,]    0    0    0    0
 [3,]    1    0    0    0
 [4,]    0    0    0    0
 [5,]    0    0    0    0
 [6,]    0    0    0    0
 [7,]    0    1    1    0
 [8,]    0    0    0    0
 [9,]    0    0    0    0
[10,]    0    1    0    1
[11,]    0    0    0    0
[12,]    0    0    0    0
[13,]    0    0    0    0
[14,]    0    0    0    0
[15,]    0    0    0    0
[16,]    0    1    1    0
[17,]    0    0    0    0
[18,]    1    0    1    0
[19,]    0    0    0    0
[20,]    0    1    0    0
```

Resumen

Detecciones en cada sitio

```
> siteDets <- rowSums(y) # Dets por sitio
> table(siteDets)        # Frecuencia

siteDets
 0  1  2
14  2  4
```

Proporción de sitios realmente ocupados

```
> naiveOccupancy <- sum(siteDets>0)/nSites
> naiveOccupancy

[1] 0.3
```

Datos simulados

Observaciones

```
> y
      [,1] [,2] [,3] [,4]
 [1,]    0    0    0    0
 [2,]    0    0    0    0
 [3,]    1    0    0    0
 [4,]    0    0    0    0
 [5,]    0    0    0    0
 [6,]    0    0    0    0
 [7,]    0    1    1    0
 [8,]    0    0    0    0
 [9,]    0    0    0    0
[10,]    0    1    0    1
[11,]    0    0    0    0
[12,]    0    0    0    0
[13,]    0    0    0    0
[14,]    0    0    0    0
[15,]    0    0    0    0
[16,]    0    1    1    0
[17,]    0    0    0    0
[18,]    1    0    1    0
[19,]    0    0    0    0
[20,]    0    1    0    0
```

Resumen

Detecciones en cada sitio

```
> siteDets <- rowSums(y) # Dets por sitio
> table(siteDets)        # Frecuencia

siteDets
 0  1  2
14  2  4
```

Proporción de sitios realmente ocupados

```
> naiveOccupancy <- sum(siteDets>0)/nSites
> naiveOccupancy

[1] 0.3
```

Datos simulados

Observaciones

```
> y
      [,1] [,2] [,3] [,4]
 [1,]    0    0    0    0
 [2,]    0    0    0    0
 [3,]    1    0    0    0
 [4,]    0    0    0    0
 [5,]    0    0    0    0
 [6,]    0    0    0    0
 [7,]    0    1    1    0
 [8,]    0    0    0    0
 [9,]    0    0    0    0
[10,]    0    1    0    1
[11,]    0    0    0    0
[12,]    0    0    0    0
[13,]    0    0    0    0
[14,]    0    0    0    0
[15,]    0    0    0    0
[16,]    0    1    1    0
[17,]    0    0    0    0
[18,]    1    0    1    0
[19,]    0    0    0    0
[20,]    0    1    0    0
```

Resumen

Detecciones en cada sitio

```
> siteDets <- rowSums(y) # Dets por sitio
> table(siteDets)        # Frecuencia

siteDets
 0  1  2
14  2  4
```

Proporción de sitios realmente ocupados

```
> naiveOccupancy <- sum(siteDets>0)/nSites
> naiveOccupancy

[1] 0.3
```

Librería ‘unmarked’ en R

A diferencia de las funciones estándar en R, como `lm` y `glm`, que requieren que los datos estén en `data.frames`, los datos en ‘unmarked’ tienen que estar formateados en un objeto `unmarkedFrame`.

```
> library(unmarked)
> umf <- unmarkedFrameOccu(y=y)
> summary(umf)
```

```
unmarkedFrame Object

20 sites
Maximum number of observations per site: 4
Mean number of observations per site: 4
Sites with at least one detection: 6

Tabulation of y observations:
 0  1
70 10
```

Librería 'unmarked' en R

Ajustamos el modelo de ocupación

```
> fm <- occu(~1 ~1, umf)
> summary(fm)
```

```
Call:
occu(formula = ~1 ~ 1, data = umf)

Occupancy (logit-scale):
 Estimate    SE      z P(>|z|)
    -0.52 0.613 -0.849   0.396

Detection (logit-scale):
 Estimate    SE     z P(>|z|)
   -0.684 0.543 -1.26   0.208

AIC: 59.11882
Number of sites: 20
optim convergence code: 0
optim iterations: 12
Bootstrap iterations: 0
```

Estimas obtenidas usando máxima verosimilitud (MLE)

Librería 'unmarked' en R

Estima de ocupación ($\hat{\psi}$)

```
> backTransform(fm, type="state")

Backtransformed linear combination(s) of Occupancy estimate(s)

 Estimate    SE LinComb (Intercept)
    0.373 0.143   -0.52           1

Transformation: logistic
```

Probabilidad de detección estimada ($\hat{p}$)

```
> backTransform(fm, type="det")

Backtransformed linear combination(s) of Detection estimate(s)

 Estimate    SE LinComb (Intercept)
    0.335 0.121  -0.684           1

Transformation: logistic
```

Librería 'unmarked' en R

Estima de ocupación ($\hat{\psi}$)

```
> backTransform(fm, type="state")
Backtransformed linear combination(s) of Occupancy estimate(s)

 Estimate    SE LinComb (Intercept)
    0.373 0.143   -0.52           1

Transformation: logistic
```

Probabilidad de detección estimada ($\hat{p}$)

```
> backTransform(fm, type="det")
Backtransformed linear combination(s) of Detection estimate(s)

 Estimate    SE LinComb (Intercept)
    0.335 0.121  -0.684           1

Transformation: logistic
```

Dos tipos de estima de ocupación

1. Ocupación no condicional $\psi = \Pr(z = 1)$
   - Podriamos pensar en el $\hat{\psi}$ que hemos calculado antes como una predicción que puede ser aplicado a cualquier nuevo sitio o período.
2. Ocupación condicional $\psi^* = \Pr(z_i = 1|y_i)$
   - La estima de ocupación condicional se aplica solo a los sitios de muestreo, en el momento de muestreo.
   - En 'unmarked', la probabilidad condicional se calcula usando métodos bayesianos empíricos, con las funciones `ranef` y `bup`. En JAGS usaremos la inferencia completa bayesiana.
   - La estima de la ocupación condicional se usa para responder cuestiones como:
     - ¿Cuál de estos sitios estaban ocupados?
     - ¿Cuantos sitios estaban ocupados?

```
> ## Probs posterior Pr(z_i=1 | y_i)
> z.post <- ranef(fm)
> ## Extraemos medias posteriores
> psi.conditional <-  bup(z.post, stat="mean")
```

Dos tipos de estima de ocupación

1. Ocupación no condicional $\psi = \Pr(z = 1)$
   - Podriamos pensar en el $\hat{\psi}$ que hemos calculado antes como una predicción que puede ser aplicado a cualquier nuevo sitio o período.
2. Ocupación condicional $\psi^* = \Pr(z_i = 1|y_i)$
   - La estima de ocupación condicional se aplica solo a los sitios de muestreo, en el momento de muestreo.
   - En 'unmarked', la probabilidad condicional se calcula usando métodos bayesianos empíricos, con las funciones `ranef` y `bup`. En JAGS usaremos la inferencia completa bayesiana.
   - La estima de la ocupación condicional se usa para responder cuestiones como:
     - ¿Cuál de estos sitios estaban ocupados?
     - ¿Cuantos sitios estaban ocupados?

```
> ## Probs posterior Pr(z_i=1 | y_i)
> z.post <- ranef(fm)
> ## Extraemos medias posteriores
> psi.conditional <-  bup(z.post, stat="mean")
```

Dos tipos de estima de ocupación

1. Ocupación no condicional $\psi = \Pr(z = 1)$
   - Podriamos pensar en el $\hat{\psi}$ que hemos calculado antes como una predicción que puede ser aplicado a cualquier nuevo sitio o período.
2. Ocupación condicional $\psi^* = \Pr(z_i = 1|y_i)$
   - La estima de ocupación condicional se aplica solo a los sitios de muestreo, en el momento de muestreo.
   - En 'unmarked', la probabilidad condicional se calcula usando métodos bayesianos empíricos, con las funciones `ranef` y `bup`. En JAGS usaremos la inferencia completa bayesiana.
   - La estima de la ocupación condicional se usa para responder cuestiones como:
     - ¿Cuál de estos sitios estaban ocupados?
     - ¿Cuantos sitios estaban ocupados?

```
> ## Probs posterior Pr(z_i=1 | y_i)
> z.post <- ranef(fm)
> ## Extraemos medias posteriores
> psi.conditional <-  bup(z.post, stat="mean")
```

Dos tipos de estima de ocupación

1. Ocupación no condicional $\psi = \Pr(z = 1)$
   - Podriamos pensar en el $\hat{\psi}$ que hemos calculado antes como una predicción que puede ser aplicado a cualquier nuevo sitio o período.
2. Ocupación condicional $\psi^* = \Pr(z_i = 1|y_i)$
   - La estima de ocupación condicional se aplica solo a los sitios de muestreo, en el momento de muestreo.
   - En 'unmarked', la probabilidad condicional se calcula usando métodos bayesianos empíricos, con las funciones `ranef` y `bup`. En JAGS usaremos la inferencia completa bayesiana.
   - La estima de la ocupación condicional se usa para responder cuestiones como:
     - ¿Cuál de estos sitios estaban ocupados?
     - ¿Cuantos sitios estaban ocupados?

```
> ## Probs posterior Pr(z_i=1 | y_i)
> z.post <- ranef(fm)
> ## Extraemos medias posteriores
> psi.conditional <-  bup(z.post, stat="mean")
```

Dos tipos de estima de ocupación

1. Ocupación no condicional $\psi = \Pr(z = 1)$
   - Podriamos pensar en el $\hat{\psi}$ que hemos calculado antes como una predicción que puede ser aplicado a cualquier nuevo sitio o período.
2. Ocupación condicional $\psi^* = \Pr(z_i = 1 | y_i)$
   - La estima de ocupación condicional se aplica solo a los sitios de muestreo, en el momento de muestreo.
   - En 'unmarked', la probabilidad condicional se calcula usando métodos bayesianos empíricos, con las funciones `ranef` y `bup`. En JAGS usaremos la inferencia completa bayesiana.
   - La estima de la ocupación condicional se usa para responder cuestiones como:
     - ¿Cuál de estos sitios estaban ocupados?
     - ¿Cuantos sitios estaban ocupados?

```
> ## Probs posterior Pr(z_i=1 | y_i)
> z.post <- ranef(fm)
> ## Extraemos medias posteriores
> psi.conditional <-  bup(z.post, stat="mean")
```

Estimas de ocupación

```
> round(data.frame(y=y, psi.unconditional=predict(fm, type="state")[,1],
+                   psi.conditional), 3)
   y.1 y.2 y.3 y.4 psi.unconditional psi.conditional
1    0   0   0   0             0.373           0.104
2    0   0   0   0             0.373           0.104
3    1   0   0   0             0.373           1.000
4    0   0   0   0             0.373           0.104
5    0   0   0   0             0.373           0.104
6    0   0   0   0             0.373           0.104
7    0   1   1   0             0.373           1.000
8    0   0   0   0             0.373           0.104
9    0   0   0   0             0.373           0.104
10   0   1   0   1             0.373           1.000
11   0   0   0   0             0.373           0.104
12   0   0   0   0             0.373           0.104
13   0   0   0   0             0.373           0.104
14   0   0   0   0             0.373           0.104
15   0   0   0   0             0.373           0.104
16   0   1   1   0             0.373           1.000
17   0   0   0   0             0.373           0.104
18   1   0   1   0             0.373           1.000
19   0   0   0   0             0.373           0.104
20   0   1   0   0             0.373           1.000
```

Métodos bayesianos

Los métodos bayesianos incorporan distribuciones *a priori* a las inferencias

A veces tenemos información previa, y es fácil usarla

Si no tenemos información previa, podemos usar información vaga y obtener inferencias similares a los métodos de máxima verosimilitud

En última instancia queremos obtener la distribución posterior de los parámetros del modelo. Usaremos el MCMC para obtener muestras de esta distribución.

Métodos bayesianos

Los métodos bayesianos incorporan distribuciones *a priori* a las inferencias

A veces tenemos información previa, y es fácil usarla

Si no tenemos información previa, podemos usar información vaga y obtener inferencias similares a los métodos de máxima verosimilitud

En última instancia queremos obtener la distribución posterior de los parámetros del modelo. Usaremos el MCMC para obtener muestras de esta distribución.

Métodos bayesianos

Los métodos bayesianos incorporan distribuciones *a priori* a las inferencias

A veces tenemos información previa, y es fácil usarla

Si no tenemos información previa, podemos usar información vaga y obtener inferencias similares a los métodos de máxima verosimilitud

En última instancia queremos obtener la distribución posterior de los parámetros del modelo. Usaremos el MCMC para obtener muestras de esta distribución.

Métodos bayesianos

Los métodos bayesianos incorporan distribuciones *a priori* a las inferencias

A veces tenemos información previa, y es fácil usarla

Si no tenemos información previa, podemos usar información vaga y obtener inferencias similares a los métodos de máxima verosimilitud

En última instancia queremos obtener la distribución posterior de los parámetros del modelo. Usaremos el MCMC para obtener muestras de esta distribución.

JAGS y NIMBLE

Usaremos dos aproximaciones de trabajo para la inferencia bayesiana con BUGS; JAGS con la librería 'jagsUI', y Nimble.

Los pasos básicos de un análisis bayesiano son:

1. Escribir el modelo
2. Poner los datos en formato lista
3. Crear una función para generar valores de inicio
4. Especificar los parámetros que queremos monitorizar
5. Usar MCMC para extraer las muestras posteriores
6. Usar las muestras posteriores para inferencia y predicción

JAGS y NIMBLE

Usaremos dos aproximaciones de trabajo para la inferencia bayesiana con BUGS; JAGS con la librería 'jagsUI', y Nimble.

Los pasos básicos de un análisis bayesiano son:

1. Escribir el modelo
2. Poner los datos en formato lista
3. Crear una función para generar valores de inicio
4. Especificar los parámetros que queremos monitorizar
5. Usar MCMC para extraer las muestras posteriores
6. Usar las muestras posteriores para inferencia y predicción

Modelo en BUGS

```
model {

# Prior para el parametro de ocupacion
psi ~ dunif(0,1)

# Prior para el parametro de deteccion
p ~ dunif(0,1)

for(i in 1:nSites) {
  # Presencia/ausencia (proceso de estado o ecologico)
  z[i] ~ dbern(psi)
  for(j in 1:nOccasions) {
    # Modelo para los datos (proceso de observacion)
    y[i,j] ~ dbern(z[i]*p)
  }
}

sitesOccupied <- sum(z)

}
```