

Lab – Modelos N-mixtos

Seguimiento de la Diversidad Biológica

José Jiménez
CSIC-IREC

Frecuentemente podemos estar más interesados en la abundancia que en la ocupación.

Incluso si estamos interesados en la ocupación, podemos modelizar la abundancia.

Por ejemplo, la ocupación no es útil si todos los sitios están ocupados.

La relación entre abundancia y una característica biológica puede resultar más patente que la relación entre ésta y la ocupación.

Los modelos binomiales N -mixtos son útiles si tenemos datos de conteos repetidos, en vez de datos de detección repetidos.

Frecuentemente podemos estar más interesados en la abundancia que en la ocupación.

Incluso si estamos interesados en la ocupación, podemos modelizar la abundancia.

Por ejemplo, la ocupación no es útil si todos los sitios están ocupados.

La relación entre abundancia y una característica biológica puede resultar más patente que la relación entre ésta y la ocupación.

Los modelos binomiales N -mixtos son útiles si tenemos datos de conteos repetidos, en vez de datos de detección repetidos.

Frecuentemente podemos estar más interesados en la abundancia que en la ocupación.

Incluso si estamos interesados en la ocupación, podemos modelizar la abundancia.

Por ejemplo, la ocupación no es útil si todos los sitios están ocupados.

La relación entre abundancia y una característica biológica puede resultar más patente que la relación entre ésta y la ocupación.

Los modelos binomiales N -mixtos son útiles si tenemos datos de conteos repetidos, en vez de datos de detección repetidos.

Frecuentemente podemos estar más interesados en la abundancia que en la ocupación.

Incluso si estamos interesados en la ocupación, podemos modelizar la abundancia.

Por ejemplo, la ocupación no es útil si todos los sitios están ocupados.

La relación entre abundancia y una característica biológica puede resultar más patente que la relación entre ésta y la ocupación.

Los modelos binomiales N -mixtos son útiles si tenemos datos de conteos repetidos, en vez de datos de detección repetidos.

Frecuentemente podemos estar más interesados en la abundancia que en la ocupación.

Incluso si estamos interesados en la ocupación, podemos modelizar la abundancia.

Por ejemplo, la ocupación no es útil si todos los sitios están ocupados.

La relación entre abundancia y una característica biológica puede resultar más patente que la relación entre ésta y la ocupación.

Los modelos binomiales N -mixtos son útiles si tenemos datos de conteos repetidos, en vez de datos de detección repetidos.

Modelos binomiales N -mixtos

Modelo de estado (con asunción de Poisson)

$$\begin{aligned}\log(\lambda_i) &= \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \dots \\ N_i &\sim \text{Poisson}(\lambda_i)\end{aligned}$$

Modelo de observación

$$\begin{aligned}\text{logit}(p_{ij}) &= \alpha_0 + \alpha_1 x_{i1} + \alpha_2 w_{ij} + \dots \\ y_{ij} &\sim \text{Binomial}(N_i, p_{ij})\end{aligned}$$

Definiciones

λ_i – Valor esperado de la abundancia en el sitio i

N_i – Valor encontrado de la abundancia en el sitio i

p_{ij} – Probabilidad de detectar un individuo en el sitio i en la ocasión j

y_{ij} – Datos de conteos

x_1 and x_2 – covariable de sitio

w – covariable de observación

Modelos binomiales N -mixtos

Modelo de estado (con asunción de Poisson)

$$\begin{aligned}\log(\lambda_i) &= \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \dots \\ N_i &\sim \text{Poisson}(\lambda_i)\end{aligned}$$

Modelo de observación

$$\begin{aligned}\text{logit}(p_{ij}) &= \alpha_0 + \alpha_1 x_{i1} + \alpha_2 w_{ij} + \dots \\ y_{ij} &\sim \text{Binomial}(N_i, p_{ij})\end{aligned}$$

Definiciones

λ_i – Valor esperado de la abundancia en el sitio i

N_i – Valor encontrado de la abundancia en el sitio i

p_{ij} – Probabilidad de detectar un individuo en el sitio i en la ocasión j

y_{ij} – Datos de conteos

x_1 and x_2 – covariable de sitio

w – covariable de observación

Modelos binomiales N -mixtos

Modelo de estado (con asunción de Poisson)

$$\begin{aligned}\log(\lambda_i) &= \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \dots \\ N_i &\sim \text{Poisson}(\lambda_i)\end{aligned}$$

Modelo de observación

$$\begin{aligned}\text{logit}(p_{ij}) &= \alpha_0 + \alpha_1 x_{i1} + \alpha_2 w_{ij} + \dots \\ y_{ij} &\sim \text{Binomial}(N_i, p_{ij})\end{aligned}$$

Definiciones

λ_i – Valor esperado de la abundancia en el sitio i

N_i – Valor encontrado de la abundancia en el sitio i

p_{ij} – Probabilidad de detectar **un individuo** en el sitio i en la ocasión j

y_{ij} – Datos de conteos

x_1 and x_2 – covariable de sitio

w – covariable de observación

Simulación – Sin covariables

Abundancia

```
> nSites <- 100
> nVisits <- 4
> set.seed(3439)  ## Para que sea reproducible
> lambda1 <- 2.6  ## Valor esperado de N
> N1 <- rpois(n=nSites, lambda=lambda1)
```

Probabilidad de detección y datos

```
> p1 <- 0.3
> y1 <- matrix(NA, nrow=nSites, ncol=nVisits)
> for(i in 1:nSites) {
+   y1[i,] <- rbinom(nVisits, size=N1[i], prob=p1)
+ }
```

Datos y abundancia latente

```
> cbind(y1, N1)[1:5,]
```

	N1				
[1,]	2	2	1	0	3
[2,]	0	0	0	0	1
[3,]	2	1	2	2	6
[4,]	0	0	0	0	1
[5,]	0	0	0	0	0

Simulación – Sin covariables

Abundancia

```
> nSites <- 100
> nVisits <- 4
> set.seed(3439)  ## Para que sea reproducible
> lambda1 <- 2.6  ## Valor esperado de N
> N1 <- rpois(n=nSites, lambda=lambda1)
```

Probabilidad de detección y datos

```
> p1 <- 0.3
> y1 <- matrix(NA, nrow=nSites, ncol=nVisits)
> for(i in 1:nSites) {
+   y1[i,] <- rbinom(nVisits, size=N1[i], prob=p1)
+ }
```

Datos y abundancia latente

```
> cbind(y1, N1)[1:5,]
```

	N1				
[1,]	2	2	1	0	3
[2,]	0	0	0	0	1
[3,]	2	1	2	2	6
[4,]	0	0	0	0	1
[5,]	0	0	0	0	0

Simulación – Sin covariables

Abundancia

```
> nSites <- 100
> nVisits <- 4
> set.seed(3439)  ## Para que sea reproducible
> lambda1 <- 2.6  ## Valor esperado de N
> N1 <- rpois(n=nSites, lambda=lambda1)
```

Probabilidad de detección y datos

```
> p1 <- 0.3
> y1 <- matrix(NA, nrow=nSites, ncol=nVisits)
> for(i in 1:nSites) {
+   y1[i,] <- rbinom(nVisits, size=N1[i], prob=p1)
+ }
```

Datos y abundancia latente

```
> cbind(y1, N1)[1:5,]
```

	N1				
[1,]	2	2	1	0	3
[2,]	0	0	0	0	1
[3,]	2	1	2	2	6
[4,]	0	0	0	0	1
[5,]	0	0	0	0	0

Simulación – Con covariables

Covariables

```
> forest <- factor(sample(c("Hardwood", "Mixed", "Pine"),  
+                         nSites, replace=TRUE))  
> forestMixed <- ifelse(forest=="Mixed", 1, 0)  
> forestPine <- ifelse(forest=="Pine", 1, 0)  
> temp <- matrix(rnorm(nSites*nVisits), nrow=nSites, ncol=nVisits)
```

Coeficientes, λ , y p

```
> beta0 <- 0; beta1 <- -1; beta2 <- 1  
> lambda2 <- exp(beta0 + beta1*forestMixed + beta2*forestPine)  
> alpha0 <- -2; alpha1 <- 1  
> p2 <- plogis(alpha0 + alpha1*temp)
```

Simulamos datos de abundancia y detección data

```
> N2 <- rpois(nSites, lambda=lambda2)          ## abundancia local  
> y2 <- matrix(NA, nrow=nSites, ncol=nVisits)  
> for(i in 1:nSites) {  
+   y2[i,] <- rbinom(nVisits, size=N2[i], prob=p2[i,])  
+ }
```

Datos simulados

Observaciones

```
> y2[1:20,]
```

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

Resumen

Detecciones en cada sitio

```
> # Conteos en cada sitio  
> maxCounts <- apply(y2, 1, max)  
> table(maxCounts)
```

```
maxCounts  
 0  1  2  3  
55 33  9  3
```

Proporción de sitios aparentemente ocupados

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

```
[1] 0.45
```

```
> library(unmarked)
```

Datos simulados

Observaciones

```
> y2[1:20,]
```

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

Resumen

Detecciones en cada sitio

```
> # Conteos en cada sitio  
> maxCounts <- apply(y2, 1, max)  
> table(maxCounts)
```

```
maxCounts  
 0  1  2  3  
55 33  9  3
```

Proporción de sitios aparentemente ocupados

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

```
[1] 0.45
```

```
> library(unmarked)
```

Datos simulados

Observaciones

```
> y2[1:20,]
```

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

Resumen

Detecciones en cada sitio

```
> # Conteos en cada sitio  
> maxCounts <- apply(y2, 1, max)  
> table(maxCounts)
```

```
maxCounts  
 0  1  2  3  
55 33  9  3
```

Proporción de sitios aparentemente ocupados

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

```
[1] 0.45
```

```
> library(unmarked)
```


Preparación de datos en ‘unmarked’

```
> umf <- unmarkedFramePCount(y=y2, siteCovs=data.frame(forest),
+                             obsCovs=list(temp=temp))
> summary(umf)

unmarkedFrame Object

100 sites
Maximum number of observations per site: 4
Mean number of observations per site: 4
Sites with at least one detection: 45

Tabulation of y observations:
  0  1  2  3
332 52 13  3

Site-level covariates:
      forest
Hardwood:31
Mixed   :32
Pine    :37

Observation-level covariates:
      temp
Min.    : -3.229012
1st Qu.: -0.603218
Median : -0.019813
Mean    : -0.001029
3rd Qu.:  0.663820
Max.    :  3.001200
```

Preparación de datos en ‘unmarked’

```
> umf <- unmarkedFramePCount(y=y2, siteCovs=data.frame(forest),
+                             obsCovs=list(temp=temp))
> summary(umf)

unmarkedFrame Object

100 sites
Maximum number of observations per site: 4
Mean number of observations per site: 4
Sites with at least one detection: 45

Tabulation of y observations:
  0  1  2  3
332 52 13  3

Site-level covariates:
      forest
Hardwood:31
Mixed   :32
Pine    :37

Observation-level covariates:
      temp
Min.    :-3.229012
1st Qu. :-0.603218
Median  :-0.019813
Mean    :-0.001029
3rd Qu. : 0.663820
Max.    : 3.001200
```

Ejecutamos el modelo

`pcount` es similar a `occu`, pero hay nuevos argumentos. K debe ser un entero mucho más alto que el valor más alto local de abundancia.

```
> fm <- pcount(~temp ~forest, umf, K=100)
> fm
```

Call:

```
pcount(formula = ~temp ~ forest, data = umf, K = 100)
```

Abundance:

	Estimate	SE	z	P(> z)
(Intercept)	-0.326	0.366	-0.891	0.372706
forestMixed	-0.326	0.461	-0.706	0.480137
forestPine	1.207	0.336	3.595	0.000324

Detection:

	Estimate	SE	z	P(> z)
(Intercept)	-1.96	0.313	-6.26	3.79e-10
temp	1.18	0.205	5.73	9.83e-09

AIC: 373.6761

Comparamos los valores reales de los parámetros:

```
> c(beta0=beta0, beta1=beta1, beta2=beta2); c(alpha0=alpha0, alpha1=alpha1)
```

```
beta0 beta1 beta2
    0    -1     1
```

```
alpha0 alpha1
   -2     1
```

Ejecutamos el modelo

`pcount` es similar a `occu`, pero hay nuevos argumentos. K debe ser un entero mucho más alto que el valor más alto local de abundancia.

```
> fm <- pcount(~temp ~forest, umf, K=100)
> fm
```

Call:

```
pcount(formula = ~temp ~ forest, data = umf, K = 100)
```

Abundance:

	Estimate	SE	z	P(> z)
(Intercept)	-0.326	0.366	-0.891	0.372706
forestMixed	-0.326	0.461	-0.706	0.480137
forestPine	1.207	0.336	3.595	0.000324

Detection:

	Estimate	SE	z	P(> z)
(Intercept)	-1.96	0.313	-6.26	3.79e-10
temp	1.18	0.205	5.73	9.83e-09

AIC: 373.6761

Comparamos los valores reales de los parámetros:

```
> c(beta0=beta0, beta1=beta1, beta2=beta2); c(alpha0=alpha0, alpha1=alpha1)
```

```
beta0 beta1 beta2
    0    -1     1
```

```
alpha0 alpha1
   -2     1
```

Veamos si K es suficientemente alto

Las estimas no deben cambiar al incrementar K .

```
> round(coef(fm), digits=4)
      lam(Int) lam(forestMixed) lam(forestPine)      p(Int)      p(temp)
      -0.3260      -0.3258      1.2073      -1.9615      1.1760
```

Parece correcto:

```
> fm.test <- pcount(~temp ~forest, umf, K=150)
> round(coef(fm.test), digits=4)
      lam(Int) lam(forestMixed) lam(forestPine)      p(Int)      p(temp)
      -0.3260      -0.3258      1.2073      -1.9615      1.1760
```

Si las estimas cambian, hay que incrementar K hasta que se estabilicen.

Veamos si K es suficientemente alto

Las estimas no deben cambiar al incrementar K .

```
> round(coef(fm), digits=4)
      lam(Int) lam(forestMixed) lam(forestPine)      p(Int)      p(temp)
      -0.3260      -0.3258      1.2073      -1.9615      1.1760
```

Parece correcto:

```
> fm.test <- pcount(~temp ~forest, umf, K=150)
> round(coef(fm.test), digits=4)
      lam(Int) lam(forestMixed) lam(forestPine)      p(Int)      p(temp)
      -0.3260      -0.3258      1.2073      -1.9615      1.1760
```

Si las estimas cambian, hay que incrementar K hasta que se estabilicen.

Veamos si K es suficientemente alto

Las estimas no deben cambiar al incrementar K .

```
> round(coef(fm), digits=4)
      lam(Int) lam(forestMixed) lam(forestPine)      p(Int)      p(temp)
      -0.3260      -0.3258      1.2073      -1.9615      1.1760
```

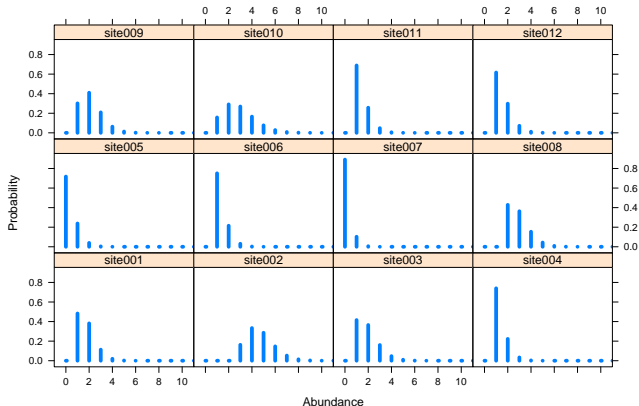
Parece correcto:

```
> fm.test <- pcount(~temp ~forest, umf, K=150)
> round(coef(fm.test), digits=4)
      lam(Int) lam(forestMixed) lam(forestPine)      p(Int)      p(temp)
      -0.3260      -0.3258      1.2073      -1.9615      1.1760
```

Si las estimas cambian, hay que incrementar K hasta que se estabilicen.

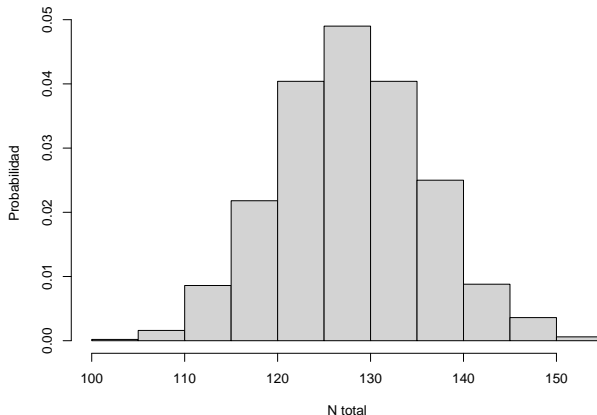
Estima empírica bayesiana – Abundancia por sitios

```
> re <- ranef(fm)  
> plot(re, layout=c(4,3), subset=site%in%1:12, xlim=c(-1, 11), lwd=5)
```



Abundancia total en la región muestreada

```
> N.total.post <- predict(re, func=sum, nsim=1000)
> hist(N.total.post, freq=FALSE, main="", xlab="N total",
+      ylab="Probabilidad")
```



Predicción en 'unmarked'

Creamos un `data.frame` con las covariable de predicción.

```
> pred.data <- data.frame(forest=c("Hardwood", "Mixed", "Pine"),  
+                           temp=0)
```

Obtenemos predicciones de λ para cada fila de datos de predicción.

```
> lambda.pred <- predict(fm, newdata=pred.data,  
+                          type='state', append=TRUE)
```

Vemos las predicciones de λ

```
> print(head(lambda.pred), digits=2)
```

	Predicted	SE	lower	upper	forest	temp
1	0.72	0.26	0.35	1.5	Hardwood	0
2	0.52	0.22	0.23	1.2	Mixed	0
3	2.41	0.64	1.44	4.0	Pine	0

Predicción en 'unmarked'

Creamos un `data.frame` con las covariable de predicción.

```
> pred.data <- data.frame(forest=c("Hardwood", "Mixed", "Pine"),  
+                           temp=0)
```

Obtenemos predicciones de λ para cada fila de datos de predicción.

```
> lambda.pred <- predict(fm, newdata=pred.data,  
+                          type='state', append=TRUE)
```

Vemos las predicciones de λ

```
> print(head(lambda.pred), digits=2)
```

	Predicted	SE	lower	upper	forest	temp
1	0.72	0.26	0.35	1.5	Hardwood	0
2	0.52	0.22	0.23	1.2	Mixed	0
3	2.41	0.64	1.44	4.0	Pine	0

Predicción en 'unmarked'

Creamos un `data.frame` con las covariable de predicción.

```
> pred.data <- data.frame(forest=c("Hardwood", "Mixed", "Pine"),  
+                           temp=0)
```

Obtenemos predicciones de λ para cada fila de datos de predicción.

```
> lambda.pred <- predict(fm, newdata=pred.data,  
+                           type='state', append=TRUE)
```

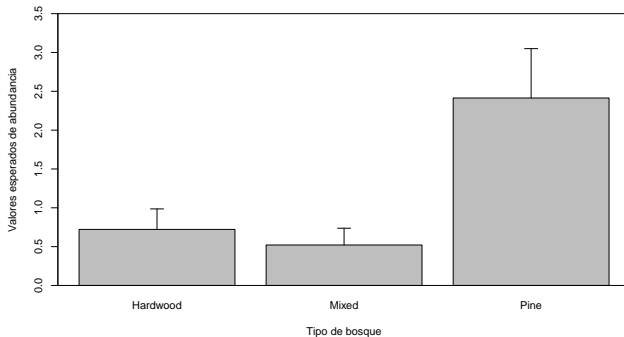
Vemos las predicciones de λ

```
> print(head(lambda.pred), digits=2)
```

	Predicted	SE	lower	upper	forest	temp
1	0.72	0.26	0.35	1.5	Hardwood	0
2	0.52	0.22	0.23	1.2	Mixed	0
3	2.41	0.64	1.44	4.0	Pine	0

Predicción en 'unmarked'

```
> bpx <- barplot(lambda.pred$Predicted,  
+               ylab="Valores esperados de abundancia",  
+               ylim=c(0,3.5), names=lambda.pred$forest,  
+               xlab="Tipo de bosque"); box()  
> arrows(bpx, lambda.pred$Predicted, bpx,  
+       lambda.pred$Predicted+lambda.pred$SE,  
+       angle=90, length=0.1)
```



Modelo en BUGS

```
model {  
  
  # Priors para los coeficientes de abundancia  
  beta0 ~ dnorm(0, 0.5)  # variance=1/0.5  
  beta1 ~ dnorm(0, 0.5)  
  beta2 ~ dnorm(0, 0.5)  
  # Priors para los coeficientes de deteccion  
  alpha0 ~ dnorm(0, 0.5)  
  alpha1 ~ dnorm(0, 0.5)  
  
  for(i in 1:nSites) {  
    log(lambda[i]) <- beta0 + beta1*forestMixed[i] + beta2*forestPine[i]  
    N[i] ~ dpois(lambda[i])          # Abundancia local latente  
    for(j in 1:nOccasions) {  
      logit(p[i,j]) <- alpha0 + alpha1*temp[i,j]  
      y[i,j] ~ dbin(p[i,j], N[i])    # Data  
    }  
  }  
  totalAbundance <- sum(N)  
}
```