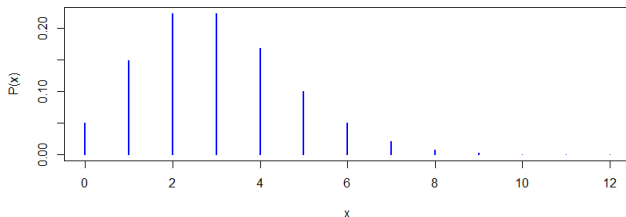


Modelos lineales generalizados



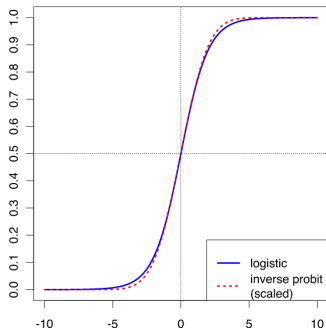
Mike Wiper

Departamento de Estadística

Universidad Carlos III de Madrid

Grado en Estadística y Empresa

Objetivo



Ilustrar como implementar los modelos lineales generalizados desde el enfoque bayesiano.

Modelos lineales generalizados

Con un modelo lineal generalizado, se intenta generalizar el concepto del modelo lineal a familias de distribuciones más allá del normal.

Un modelo lineal generalizado se define con:

- 1 Un componente aleatoria, especificando la variable de respuesta, y su distribución de probabilidad.
- 2 Un vector de parámetros β y un matrix del modelo X determinando el predictor lineal de cada variable $X\beta$.
- 3 Una función vínculo que define la relación entre $E[y] = \mu_i$ y $X\beta$:

$$g(\mu_i) = X\beta.$$

Funciones vínculo

- En el caso de datos normales, se tiene $E[y] = X\beta$.
- Para datos binarios $y = \text{"sí" o "no"}$, se puede emplear una función logística:

$$\log \frac{P(Y = \text{"sí"})}{1 - P(Y = \text{"sí"})} = \sum_{j=1}^k \beta_j x_j$$

o una función probit:

$$P(Y = \text{"sí"}) = \Phi \left(\sum_{j=1}^k \beta_j x_j \right)$$

- Para datos Poisson o exponenciales se suele emplear el vínculo logarítmico:

$$Y|\lambda \sim \text{Poisson}(\lambda) \quad \log \lambda = \sum_{j=1}^k \beta_j x_j.$$

Inferencia clásica

Existen dos métodos tipo de estimación:

- 1 Estimación de máxima verosimilitud
En contraste al modelo lineal de siempre, típicamente, soluciones analíticas no son disponibles y se emplean algoritmos como Newton Raphson para encontrar el máximo.
- 2 Mínimos cuadrados ponderados

En ambos casos, igual al caso del modelo lineal, se suele estimar la varianza del modelo con

$$\frac{1}{n - k} \sum_{i=1}^n (y_i - \hat{\mu}_i)^2.$$

Ejemplo 1: datos binarios

Datos de UCLA sobre solicitudes de admisión de candidatos al grado.

```
df <- read.csv("binary.csv")
str(df)
summary(df)
xtabs(~ admit +rank , data=df)
```

```
admit: 0 1 1 1 0 1 1 0 1 0 ...
gre : 380 660 800 640 520 760 560 400 540 700 ...
gpa : 3.61 3.67 4 3.19 2.93 3 2.98 3.08 3.39 3.92 ...
rank : 3 3 1 4 4 2 1 2 3 2 ...
```

Queremos modelizar la relación entre la probabilidad de admisión y las demás variables.

Ejemplo 1

Podemos usar `glm` en R para ajustar el modelo , pero necesitamos convertir la variable “rank” en un factor.

```
logit <- glm(admit~gre+gpa+
  as.factor(rank),
  data=df, family=binomial)
summary(logit)
```

```
Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept) -3.989979    1.139951  -3.500 0.000465 ***
gre           0.002264    0.001094   2.070 0.038465 *
gpa           0.804038    0.331819   2.423 0.015388 *
rank2        -0.675443    0.316490  -2.134 0.032829 *
rank3        -1.340204    0.345306  -3.881 0.000104 ***
rank4        -1.551464    0.417832  -3.713 0.000205 ***
```

Además podemos hacer predicciones.

```
x <- data.frame(gre=790,
  gpa=3.8, rank=as.factor(1))
predict(logit, x)
```

Estimamos la probabilidad de admisión para este candidato como 85 %.

Ejemplo 1

También podemos hacer el ajuste con el vínculo probit.

```
probit <- glm(admit~gre+gpa+
  as.factor(rank), data=df,
  family=binomial(link="probit"))
summary(probit)
predict(probit, x)
```

Los coeficientes cambian y la predicción es bastante distinta al anterior.

¡El tipo de vínculo que aplicamos es bastante influyente!

Ejemplo II

Se quiere modelizar el número de veces que se rompe la lana (breaks) en una fábrica de ropa en función del tipo (A o B) y de la tensión (L, M, H).

```
help(warpbreaks)
data <- warpbreaks
```

El modelo es

$$\begin{aligned} \text{breaks} &\sim \text{Poisson}(\lambda) \\ \log \lambda &= \beta_0 + \beta_B \delta_B + \beta_M \delta_M + \beta_H \delta_H \end{aligned}$$

donde los β son parámetros y los δ son indicadores, por ejemplo $\delta_B = 1$ si el tipo de lana es B y 0 si no.

Ejemplo II: ajuste y predicción frecuentista

```
poisson.model <- glm(breaks ~ wool + tension, data,
  family = poisson(link = "log"))
summary(poisson.model)
nuevosdatos <- data.frame(wool=as.factor("A"),
  tension=as.factor("H"))
pred <- predict(poisson.model, newdata = nuevosdatos,
  type = "response")
pred
```

Inferencia bayesiana

- En general la inferencia no es conjugada.
- Varias opciones para a priori de los regresores: normal, Cauchy, t, ...
- Implementación a través del MCMC.
- Fácil implementación en R a través del MCMCpack o R2OpenBUGS.

Una ventaja es que se puede calcular la distribución predictiva de una observación nueva además de la predicción puntual.

Ejemplo 1

Usamos `mcmcpack` para correr el modelo en R.

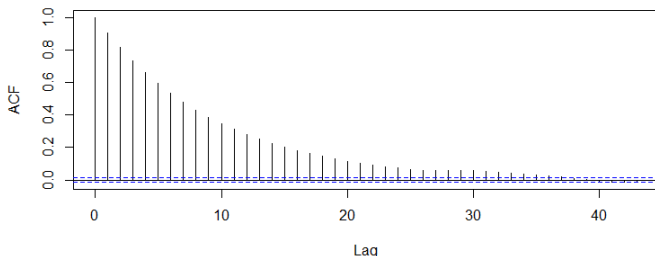
```
library(MCMCpack)
burn <- 10000
mcmc <- 20000
logitmcmc <- MCMClogit(admit~gre+gpa+as.factor(rank),
  data=df, burnin = burn, mcmc = mcmc, b0=0, B0=0.00001)
```

Ejemplo I: convergencia

Antes de analizar los resultados, miramos la convergencia del algoritmo.

```
intercepto <- logitmcmc[,1]
ts.plot(intercepto)
ts.plot(cumsum(intercepto)/c(1:mcmc))
acf(intercepto)
```

Series intercepto



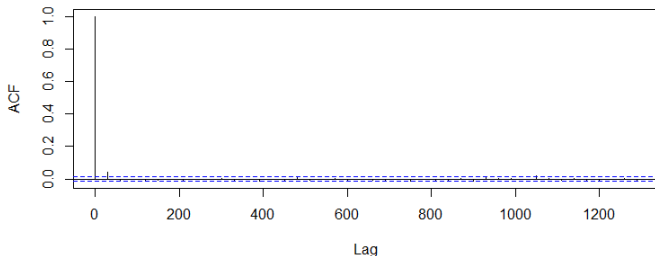
¡Hay mucha autocorrelación!

Ejemplo I: convergencia con thinning

Repetimos el ejercicio pero sólo usando 1 en cada 30 de los valores simulados.

```
thinning <- 30
burn <- burn*thinning
mcmc <- mcmc*thinning
logitmcmc <- MCMClogit(admit~gre+gpa+as.factor(rank),
  data=df, burnin = burn, mcmc = mcmc,b0=0,B0=0.00001,
  thin=thinning)
intercepto <- logitmcmc[,1]
```

Series intercepto



Ejemplo I: distribuciones a posteriori

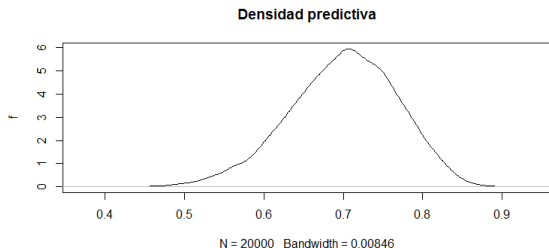
```
summary(logitmcmc)
densplot(logitmcmc)
```

| | Mean | SD | Naive SE | Time-series SE |
|------------------|-----------|----------|-----------|----------------|
| (Intercept) | -4.082008 | 1.145214 | 8.098e-03 | 8.459e-03 |
| gre | 0.002304 | 0.001109 | 7.838e-06 | 8.188e-06 |
| gpa | 0.823800 | 0.334418 | 2.365e-03 | 2.481e-03 |
| as.factor(rank)2 | -0.680844 | 0.318212 | 2.250e-03 | 2.303e-03 |
| as.factor(rank)3 | -1.364232 | 0.350929 | 2.481e-03 | 2.604e-03 |
| as.factor(rank)4 | -1.584110 | 0.425180 | 3.006e-03 | 3.154e-03 |

Las medias a posteriori de los parámetros son bastante cercanas a las estimaciones clásicas.

Ejemplo I: predicción

```
intercepto <- logitmcmc[,1]
betagre <- logitmcmc[,2]
betagpa <- logitmcmc[,3]
betarank <- matrix(rep(0, mcmc*4), nrow=mcmc)
betarank[,2:4] <- logitmcmc[4:6]
predmcmc <- 1/(1+exp(-(intercepto+betagre*790+betagpa*3.8)))
plot(density(predmcmc))
mean(predmcmc)
```



La probabilidad media de “sí” es 0,7, algo diferente a la predicción clásica.

Ejemplo I: vínculo probit

```
probitmcmc <- MCMCprobit(admit~gre+gpa+as.factor(rank), data=df,
  burnin = burnin, mcmc = mcmc, b0=0, B0=0.00001, thin=thinning)
summary(probitmcmc)
densplot(probitmcmc)
intercepto <- probitmcmc[,1]
betagre <- probitmcmc[,2]
betagpa <- probitmcmc[,3]
betarank <- matrix(rep(0, mcmc*4), nrow=mcmc)
betarank[,2:4] <- probitmcmc[4:6]
predmcmc <- pnorm(intercepto+betagre*790+betagpa*3.8)
plot(density(predmcmc), main="Densidad predictiva", ylab="f")
mean(predmcmc)
```

Las estimaciones de los parámetros son parecidas a las estimaciones frecuentistas. No obstante, la media predictiva es casi igual a la del modelo logístico.

Ejemplo II

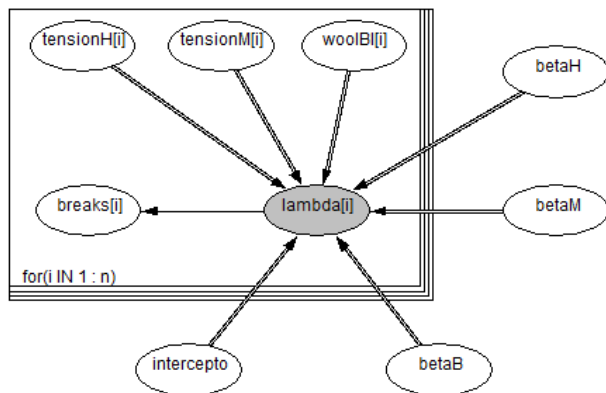
Para correr el modelo en OpenBUGS, es necesario tener las variables independientes de forma que toman el valor 0 o 1, dependiente de la categoría del dato.

```
y <- data$breaks
n <- length(y)
mean(y)
woolB <- as.numeric(data$wool)-1
tensiont <- as.numeric(data$tension)-1
tensionM <- tensiont
tensionM[tensionM!=1] <- 0
tensionH <- tensiont
tensionH[tensionH!=2] <- 0
```

Ejemplo II: Doodle de OpenBUGS

La representación gráfica del modelo es la siguiente:

name: lambda[i] type: logical link: log
value: intercepto+betaB*woolBI[i]+betaM*tensionM[i]+betaH*tensionH[i]



Ejemplo II: Implementación a través de R

Incluimos el código BUGS hecho a través del doodle en una función en R:

```
library(R2OpenBUGS)
m1g <- function(){
  for (i in 1:n) {
    y[i] ~ dpois(lambda[i])
    log(lambda[i]) <- intercepto+betaB*woolB[i]+
      betaM*tensionM[i]+betaH*tensionH[i]
  }
  intercepto ~ dnorm(0,0.0001)
  betaB ~ dnorm(0,0.0001)
  betaM ~ dnorm(0,0.0001)
  betaH ~ dnorm(0,0.0001)
}
```

Ejemplo II: Implementación a través de R

Utilizamos un `list` en R para definir los datos y un `list` dentro de una función para los valores iniciales.

```
mlgdata <- list(n = n, y = y, woolB = woolB, tensionM = tensionM,
  tensionH = tensionH)
mlginits <- function() {
  list(intercepto = 3.7, betaB = 0, betaM = 0, betaH = 0)
}
```

Llamamos al OpenBUGS con el comando `bugs`.

```
mlgout <- bugs(data = mlgdata, inits = mlginits, parameters.to.save = c("intercepto", "betaB",
  "betaM", "betaH"), model.file = mlg, n.chains = 1, n.iter = 10000)
mlgout
```

Ejemplo II: Análisis de convergencia

Podemos utilizar el paquete coda para analizar los resultados pero es fácil extraer los datos:

```
intercepto <- mlgout$sims.list$intercepto
plot(intercepto, type='l', ylab='intercepto')
plot(cumsum(intercepto)/c(1:length(intercepto)), type='l',
      ylab='E[intercepto]')
acf(intercepto)
```

EN este caso, la convergencia de la cadena está bien y no hay autocorrelación.

Ejemplo II: Distribuciones a posteriori y predicción

```
plot(density(intercepto), ylab='f', xlab='intercepto',  
     main="Densidad a posteriori del intercepto")  
betaH <- mlgout$sims.list$betaH  
lambda <- exp(intercepto+betaH)  
mean(lambda)  
ypred <- rpois(length(lambda), lambda)  
ynum <- table(ypred)  
barplot(ynum/sum(ynum), xlab='y', ylab='P(y)')  
quantile(ypred, c(0.025, 0.5, 0.975))
```

Vemos alguna diferencia entre la predicción media bayesiana y frecuentista.

Resumen y siguiente sesión

En esta sesión, hemos visto como implementar la inferencia bayesiana para modelos lineales generalizados en R a través del `mcmcpack` o `R2openBUGS`.

| Bayes factor BF_{12} | | | Interpretation |
|------------------------|---|-------|--------------------------------|
| | > | 100 | Extreme evidence for M_1 |
| 30 | - | 100 | Very Strong evidence for M_1 |
| 10 | - | 30 | Strong evidence for M_1 |
| 3 | - | 10 | Moderate evidence for M_1 |
| 1 | - | 3 | Anecdotal evidence for M_1 |
| | 1 | | No evidence |
| 1/3 | - | 1 | Anecdotal evidence for M_2 |
| 1/10 | - | 1/3 | Moderate evidence for M_2 |
| 1/30 | - | 1/10 | Strong evidence for M_2 |
| 1/100 | - | 1/30 | Very Strong evidence for M_2 |
| | < | 1/100 | Extreme evidence for M_2 |

En las siguientes sesiones, ilustramos como hacer contrastes de hipótesis y seleccionar modelos mediante técnicas bayesianas.