

# other count regressions

*Chris Parrish*

*June 30, 2016*

## Contents

<b>multinomial</b>	<b>1</b>
data . . . . .	1
map . . . . .	1
<b>multinomial via Poisson</b>	<b>2</b>
<b>geometric</b>	<b>4</b>

other count regressions

reference: McElreath, Statistical Rethinking, chap 10, p.322

```
library(rethinking)
library(ggplot2)
```

## multinomial

### data

```
## R code 10.56
# simulate career choices among 500 individuals
N <- 500          # number of individuals
income <- 1:3    # expected income of each career
score <- 0.5*income # scores for each career, based on income
# next line converts scores to probabilities
p <- softmax(score[1],score[2],score[3])

# now simulate choice
# outcome career holds event type values, not counts
career <- rep(NA,N) # empty vector of choices for each individual
# sample chosen career for each individual
for ( i in 1:N ) career[i] <- sample( 1:3 , size=1 , prob=p )
```

### map

```
## R code 10.57
# fit the model, using dcategory and softmax link
m10.16 <- map(
  alist(
    career ~ dcategory( softmax(0,s2,s3) ),
    s2 <- b*2, # linear model for event type 2
    s3 <- b*3, # linear model for event type 3
```

```

      b ~ dnorm(0,5)
    ) ,
    data=list(career=career) )

## R code 10.58
N <- 100
# simulate family incomes for each individual
family_income <- runif(N)
# assign a unique coefficient for each type of event
b <- (1:-1)
career <- rep(NA,N) # empty vector of choices for each individual
for ( i in 1:N ) {
  score <- 0.5*(1:3) + b*family_income[i]
  p <- softmax(score[1],score[2],score[3])
  career[i] <- sample( 1:3 , size=1 , prob=p )
}

m10.17 <- map(
  alist(
    career ~ dategorical( softmax(0,s2,s3) ),
    s2 <- a2 + b2*family_income,
    s3 <- a3 + b3*family_income,
    c(a2,a3,b2,b3) ~ dnorm(0,5)
  ) ,
  data=list(career=career,family_income=family_income) )

```

## multinomial via Poisson

```

## R code 10.59
library(rethinking)
data(UCBadmit)
d <- UCBadmit

## R code 10.60
# binomial model of overall admission probability
m_binom <- map(
  alist(
    admit ~ dbinom(applications,p),
    logit(p) <- a,
    a ~ dnorm(0,100)
  ) ,
  data=d )

# Poisson model of overall admission rate and rejection rate
d$rej <- d$reject # 'reject' is a reserved word
m_pois <- map2stan(
  alist(
    admit ~ dpois(lambda1),
    rej ~ dpois(lambda2),
    log(lambda1) <- a1,
    log(lambda2) <- a2,
    c(a1,a2) ~ dnorm(0,100)
  ) ,
  data=d )

```

```
),
  data=d , chains=3 , cores=3 )

## Warning: Variable 'applicant.gender' contains dots '.'.
## Will attempt to remove dots internally.

## Warning in FUN(X[[i]], ...): data with name dept is not numeric and not
## used

## Warning in FUN(X[[i]], ...): data with name applicant_gender is not numeric
## and not used

## Warning in FUN(X[[i]], ...): data with name dept is not numeric and not
## used

## Warning in FUN(X[[i]], ...): data with name applicant_gender is not numeric
## and not used

##
## SAMPLING FOR MODEL 'admit ~ dpois(lambda1)' NOW (CHAIN 1).
## WARNING: No variance estimation is
##           performed for num_warmup < 20
##
##
## Chain 1, Iteration: 1 / 1 [100%] (Sampling)
## Elapsed Time: 4e-06 seconds (Warm-up)
##               3.1e-05 seconds (Sampling)
##               3.5e-05 seconds (Total)

## Computing WAIC

## Constructing posterior predictions

## [ 300 / 3000 ]
[ 600 / 3000 ]
[ 900 / 3000 ]
[ 1200 / 3000 ]
[ 1500 / 3000 ]
[ 1800 / 3000 ]
[ 2100 / 3000 ]
[ 2400 / 3000 ]
[ 2700 / 3000 ]
[ 3000 / 3000 ]

## R code 10.61
logistic(coef(m_binom))

##           a
## 0.3877741

## R code 10.62
k <- as.numeric(coef(m_pois))
exp(k[1])/(exp(k[1])+exp(k[2]))

## [1] 0.3873081
```

## geometric

```
## R code 10.63
# simulate
N <- 100
x <- runif(N)
y <- rgeom( N , prob=logistic( -1 + 2*x ) )

# estimate
m10.18 <- map(
  alist(
    y ~ dgeom( p ),
    logit(p) <- a + b*x,
    a ~ dnorm(0,10),
    b ~ dnorm(0,1)
  ),
  data=list(y=y,x=x) )
precis(m10.18)

##      Mean StdDev 5.5% 94.5%
## a -0.77  0.24 -1.16 -0.39
## b  1.57  0.44  0.87  2.27
```