JAGS non linear regression

Duncan Golicher

12/19/2018

d<-read.csv("/home/aqm/course/data/Hollings.csv")
plot(d,pch=21,bg=2)



## Jags linear

library(rjags)

## Loading required package: coda

## Linked to JAGS 4.2.0

## Loaded modules: basemod,bugs

library(ggplot2)
library(dplyr)

##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
## filter, lag

## The following objects are masked from 'package:base':
##
## intersect, setdiff, setequal, union

reg\_mod<-"
model{

#Likelihood
 for( i in 1:n)
 {
 y[i]~dnorm(mu[i], tau)
 mu[i]<-b0+b1\*x[i]
 }

#priors
b0~dnorm(0,.01)
b1~dnorm(0,.01)
tau<-pow(sd, -2)
sd~dunif(0,100)

}"

dat<-list(n=length(d$Resource),x=d$Resource,y=d$Consumption)

mod1 <- jags.model(textConnection(reg\_mod), data = dat, n.chains = 3)

## Compiling model graph
## Resolving undeclared variables
## Allocating nodes
## Graph information:
## Observed stochastic nodes: 30
## Unobserved stochastic nodes: 3
## Total graph size: 133
##
## Initializing model

update(mod1, 1000)

mod1\_sim <- coda.samples(model = mod1, variable.names = c("b0","b1"), n.iter = 5000)
tidybayes:::tidy\_draws(mod1\_sim)

## # A tibble: 15,000 x 5
## .chain .iteration .draw b0 b1
## <int> <int> <int> <dbl> <dbl>
## 1 1 1 1 14.1 0.0101
## 2 1 2 2 14.2 0.00911
## 3 1 3 3 14.2 0.0104
## 4 1 4 4 14.2 0.0102
## 5 1 5 5 14.2 0.00990
## 6 1 6 6 14.1 0.0104
## 7 1 7 7 14.0 0.00971
## 8 1 8 8 14.5 0.00887
## 9 1 9 9 14.7 0.00793
## 10 1 10 10 14.3 0.00944
## # ... with 14,990 more rows

summary(mod1\_sim)

##
## Iterations = 2001:7000
## Thinning interval = 1
## Number of chains = 3
## Sample size per chain = 5000
##
## 1. Empirical mean and standard deviation for each variable,
## plus standard error of the mean:
##
## Mean SD Naive SE Time-series SE
## b0 13.84704 0.378297 3.089e-03 9.779e-03
## b1 0.01088 0.001168 9.537e-06 3.006e-05
##
## 2. Quantiles for each variable:
##
## 2.5% 25% 50% 75% 97.5%
## b0 13.115255 13.60240 13.84360 14.09025 14.60279
## b1 0.008541 0.01012 0.01088 0.01164 0.01315

confint(lm(Consumption~Resource,data=d))

## 2.5 % 97.5 %
## (Intercept) 13.102428065 14.59200222
## Resource 0.008568358 0.01318626

hyp\_mod<-"
model{

#Likelihood
 for( i in 1:n)
 {
 y[i]~dnorm(mu[i], tau)
 mu[i]<-b0\*x[i]/(b1+x[i])
 }

#priors
b0~dnorm(0,.01)
b1~dnorm(0,.01)
tau<-pow(sd, -2)
sd~dunif(0,100)

}"

mod1 <- jags.model(textConnection(hyp\_mod), data = dat, n.chains = 3)

## Compiling model graph
## Resolving undeclared variables
## Allocating nodes
## Graph information:
## Observed stochastic nodes: 30
## Unobserved stochastic nodes: 3
## Total graph size: 163
##
## Initializing model

update(mod1, 1000)

mod1\_sim <- coda.samples(model = mod1, variable.names = c("b0","b1"), n.iter = 5000)

summary(mod1\_sim)

##
## Iterations = 2001:7000
## Thinning interval = 1
## Number of chains = 3
## Sample size per chain = 5000
##
## 1. Empirical mean and standard deviation for each variable,
## plus standard error of the mean:
##
## Mean SD Naive SE Time-series SE
## b0 19.79 0.189 0.001543 0.004894
## b1 36.41 2.415 0.019717 0.064631
##
## 2. Quantiles for each variable:
##
## 2.5% 25% 50% 75% 97.5%
## b0 19.42 19.67 19.80 19.92 20.16
## b1 31.56 34.82 36.44 38.04 41.02

nlmod<-nls(Consumption~s\*Resource/(F+Resource),data = d,start = list( F = 20,s=20))
confint(nlmod)

## Waiting for profiling to be done...

## 2.5% 97.5%
## F 33.97300 43.66697
## s 19.58576 20.32930

## Buntings

d<-read.csv("/home/aqm/course/data/buntings.csv")
plot(rate~density,data=d,pch=21,bg=2)



Trying to write the Bayesian model.

Use Gamma errors to avoid negative rates. Set gammas on both paprameters as priors. Use the rep-paramaterisation of gamma in terms of mean and standard deviation instead of shape and scale for clarity.

## Priors

priors<-"
model{
#priors
a~dgamma( ((a\_mu^2)/a\_sig), (a\_mu/a\_sig))
H~dgamma( ((H\_mu^2)/H\_sig), (H\_mu/H\_sig))
sig~dunif(0.0001,1)

}"

dat<-list(a\_mu=0.00117,a\_sig= 0.0000053,H\_mu=2.64950,H\_sig=0.28727)
mod1 <- jags.model(textConnection(priors), data = dat, n.chains = 5)

## Compiling model graph
## Resolving undeclared variables
## Allocating nodes
## Graph information:
## Observed stochastic nodes: 0
## Unobserved stochastic nodes: 3
## Total graph size: 17
##
## Initializing model

update(mod1, 10000)

priors\_mod <- coda.samples(model = mod1, variable.names = c("a","H"), n.iter = 50000)

plot(priors\_mod)



hol\_mod<-"
model{

#Likelihood
 for( i in 1:n)
 {

 y[i] ~dgamma( ((mu[i]^2)/sig), (mu[i]/sig))
 mu[i]<- a\*x[i]/(1+a\*x[i]\*H)

 }

#priors
a~dgamma( ((a\_mu^2)/a\_sig), (a\_mu/a\_sig))
H~dgamma( ((H\_mu^2)/H\_sig), (H\_mu/H\_sig))

sig~dunif(0.0001,1)

}"

The model still does not really work, even with informative priors derived from the quantile regression.

dat<-list(n=length(d$Resource),x=d$density,y=d$rate,a\_mu=0.00117,a\_sig= 0.0000053,H\_mu=2.64950,H\_sig=0.28727)
mod1 <- jags.model(textConnection(hol\_mod), data = dat, n.chains = 5)

## Compiling model graph
## Resolving undeclared variables
## Allocating nodes
## Graph information:
## Observed stochastic nodes: 0
## Unobserved stochastic nodes: 3
## Total graph size: 310
##
## Initializing model

update(mod1, 10000)

mod1\_sim <- coda.samples(model = mod1, variable.names = c("a","H"), n.iter = 50000)

summary(mod1\_sim)

##
## Iterations = 10001:60000
## Thinning interval = 1
## Number of chains = 5
## Sample size per chain = 50000
##
## 1. Empirical mean and standard deviation for each variable,
## plus standard error of the mean:
##
## Mean SD Naive SE Time-series SE
## H 2.648451 0.535223 1.070e-03 1.068e-03
## a 0.001173 0.002306 4.612e-06 4.612e-06
##
## 2. Quantiles for each variable:
##
## 2.5% 25% 50% 75% 97.5%
## H 1.704e+00 2.270e+00 2.6135104 2.987838 3.795427
## a 1.901e-09 1.411e-05 0.0002188 0.001246 0.007996

plot(mod1\_sim)



HDmod<-nls(rate~a\*density/(1+a\*density\*H),data = d,start = list(a =0.001,H=2))
confint(HDmod)

## Waiting for profiling to be done...

## 2.5% 97.5%
## a 0.002593086 0.006694939
## H 1.713495694 1.976978655