

# Nonlinear models

Bayesian statistics 8 – dynamic and nonlinear model fitting

Frédéric Barraquand (CNRS, IMB)

07/12/2021

# Things that we learned the last time

- logistic regression with  $y_i \sim \mathcal{B}(n, p_i)$  with  
 $\text{logit}(p_i) = \log\left(\frac{p_i}{1-p_i}\right) = a + bx_i$  or  $\gamma(x_i - \mu_x)$  which is equivalent to  
 $p_i = \text{logistic}(a + bx_i) = \frac{\exp(a+bx_i)}{1+\exp(a+bx_i)} = \frac{1}{1+\exp(-(a+bx_i))}$

# Things that we learned the last time

- logistic regression with  $y_i \sim \mathcal{B}(n, p_i)$  with  
 $\text{logit}(p_i) = \log\left(\frac{p_i}{1-p_i}\right) = a + bx_i$  or  $\gamma(x_i - \mu_x)$  which is equivalent to  
$$p_i = \text{logistic}(a + bx_i) = \frac{\exp(a+bx_i)}{1+\exp(a+bx_i)} = \frac{1}{1+\exp(-(a+bx_i))}$$
- “logistic ANOVA”  $p_i = \text{logistic}(\alpha + \beta_1 x_{1i} + \beta_2 x_{2i} + \beta_3 x_{3i})$ . Here the logistic function maps  $(-\infty, +\infty) \rightarrow [0, 1]$ .

# Things that we learned the last time

- logistic regression with  $y_i \sim \mathcal{B}(n, p_i)$  with  
 $\text{logit}(p_i) = \log\left(\frac{p_i}{1-p_i}\right) = a + bx_i$  or  $\gamma(x_i - \mu_x)$  which is equivalent to  
$$p_i = \text{logistic}(a + bx_i) = \frac{\exp(a + bx_i)}{1 + \exp(a + bx_i)} = \frac{1}{1 + \exp(-(a + bx_i))}$$
- “logistic ANOVA”  $p_i = \text{logistic}(\alpha + \beta_1 x_{1i} + \beta_2 x_{2i} + \beta_3 x_{3i})$ . Here the logistic function maps  $(-\infty, +\infty) \rightarrow [0, 1]$ .
- We can also want to use the logistic function to *model a known curve*,  
 $y_i = f(x_i) + \epsilon_i$  or  $y_i = f(t_i) + \epsilon_i$

# Organism growth basics: von Bertalanffy

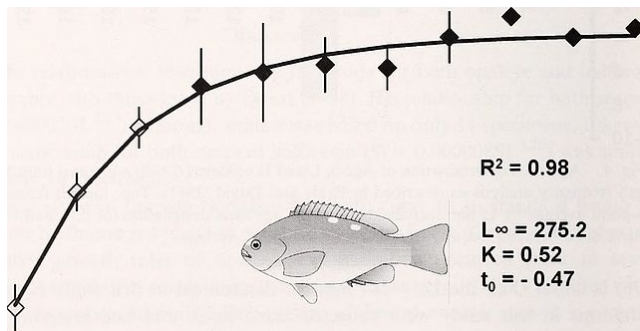


Figure 1: Von Bertalanffy growth curve fit to *Girella nigricans*

$$L(t) = L_{\infty}(1 - \exp(-k(t_i - t_0))) + \epsilon_i$$

## Connection to dynamics

$$\frac{dL}{dt} = k(L_{\infty} - L)$$

## Connection to dynamics

$$\frac{dL}{dt} = k(L_{\infty} - L)$$

We'll do for practical another example of organismal growth, Gompertz growth (Winsor PNAS 1932).

Dynamics

$$\frac{d \ln(L)}{dt} = k(\ln(L_{\infty}) - \ln(L))$$

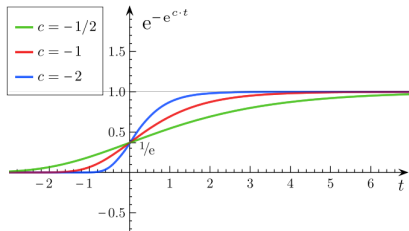
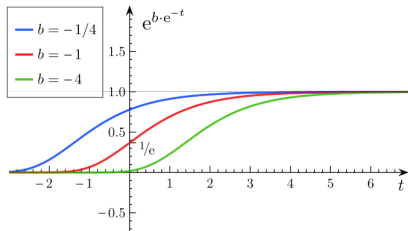
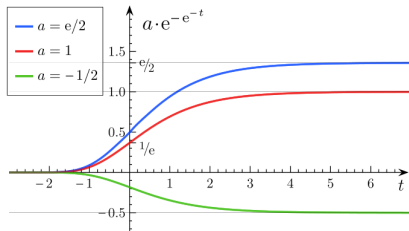
Solution

$$L(t) = ae^{-e^{b-ct}}$$

where  $a = L_{\infty}e^1$ ,  $b = kt_0$  and  $c = k$ .

Trick: note  $I = \ln(L)$ , solve von Bertalanffy for  $I$ , transform back.

# The Gompertz growth curve is more logistic-like





# Mathematical cousins of von Bertalanffy: modelling *saturation*

Monod function (microbiology) aka Michaelis-Menten (chemistry) aka Holling type II (ecology) aka ...

$$f(x) = \frac{ax}{b+x}$$

## Another example of connection to dynamics: logistic population growth

$N$  = population size (microbes, humans, wild boars, plants, . . .)

$$\frac{dN}{dt} = rN \left( 1 - \frac{N}{K} \right)$$

And the solution is. . .

## Another example of connection to dynamics: logistic population growth

$N$  = population size (microbes, humans, wild boars, plants, ...)

$$\frac{dN}{dt} = rN \left(1 - \frac{N}{K}\right)$$

And the solution is...

$$N(t) = \frac{N_0 e^{rt}}{1 + N_0(e^{rt} - 1)/K}$$

If we use  $t = 1, 2, 3, \dots$

$$N_1 = \frac{N_0 e^r}{1 + N_0(e^r - 1)/K}, N_2 = \frac{N_1 e^r}{1 + N_1(e^r - 1)/K}, \dots$$

aka Beverton-Holt model.

# Two kinds of “noise” or stochasticity

## Observational noise

$$N_t = \frac{N_t e^r}{1 + N_t(e^r - 1)/K} + \epsilon_t, \quad \epsilon_t \sim \mathcal{N}(0, \sigma^2) \text{ i.i.d}$$

- The effect of  $\epsilon_t$  does not accumulate.  $\epsilon_1$  does not affect  $N_8$ .

# Two kinds of “noise” or stochasticity

## Observational noise

$$N_t = \frac{N_t e^r}{1 + N_t(e^r - 1)/K} + \epsilon_t, \quad \epsilon_t \sim \mathcal{N}(0, \sigma^2) \text{ i.i.d}$$

- The effect of  $\epsilon_t$  does not accumulate.  $\epsilon_1$  does not affect  $N_8$ .

## Process noise

$$N_t = \frac{N_t e^{r+\epsilon_t}}{1 + N_t(e^r - 1)/K}, \quad \epsilon_t \sim \mathcal{N}(0, \sigma^2) \text{ i.i.d}$$

- The effect of  $\epsilon_t$  does accumulate.  $\epsilon_1$  *does* affect  $N_8$ .

# Two kinds of “noise” or stochasticity

## Observational noise

$$N_t = \frac{N_t e^r}{1 + N_t(e^r - 1)/K} + \epsilon_t, \quad \epsilon_t \sim \mathcal{N}(0, \sigma^2) \text{ i.i.d}$$

- The effect of  $\epsilon_t$  does not accumulate.  $\epsilon_1$  does not affect  $N_8$ .

## Process noise

$$N_t = \frac{N_t e^{r+\epsilon_t}}{1 + N_t(e^r - 1)/K}, \quad \epsilon_t \sim \mathcal{N}(0, \sigma^2) \text{ i.i.d}$$

- The effect of  $\epsilon_t$  does accumulate.  $\epsilon_1$  *does* affect  $N_8$ .
- Creates *autocorrelation* between  $N_t$  values

# Two kinds of “noise” or stochasticity

## Observational noise

$$N_t = \frac{N_t e^r}{1 + N_t(e^r - 1)/K} + \epsilon_t, \epsilon_t \sim \mathcal{N}(0, \sigma^2) \text{ i.i.d}$$

- The effect of  $\epsilon_t$  does not accumulate.  $\epsilon_1$  does not affect  $N_8$ .

## Process noise

$$N_t = \frac{N_t e^{r+\epsilon_t}}{1 + N_t(e^r - 1)/K}, \epsilon_t \sim \mathcal{N}(0, \sigma^2) \text{ i.i.d}$$

- The effect of  $\epsilon_t$  does accumulate.  $\epsilon_1$  *does* affect  $N_8$ .
- Creates *autocorrelation* between  $N_t$  values
- Of course in real life you can have both (and sometimes it is hard to distinguish between the two)

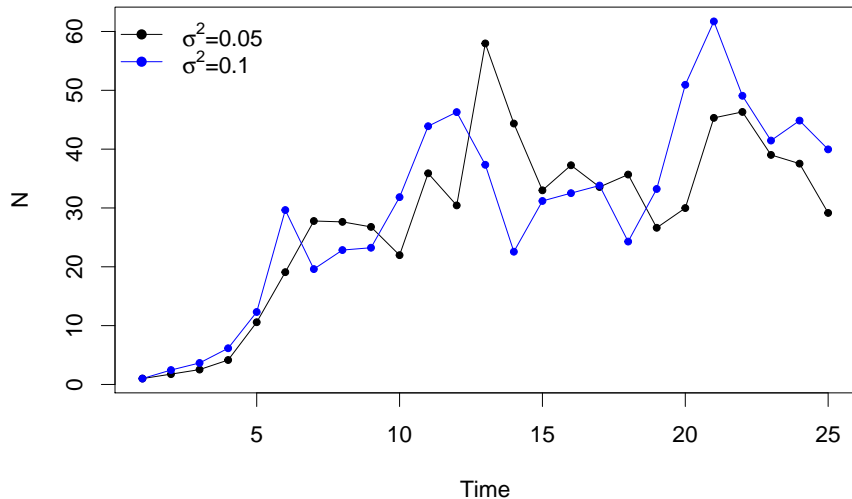
# Fitting the logistic model in discrete-time with process noise I

(you can fit the observational noise model with the solution  $N(t)$  – you can't fit the process noise model solution, you have to fit the *dynamics*)

```
r=0.5
alpha=0.02
tmax=25
R=exp(r)
K=(exp(r)-1)/alpha
N_BH=N_BH1=rep(NA,tmax)
N_BH[1]=N_BH1[1]=1
for (t in 1:(tmax-1)){N_BH[t+1] =
  (exp(r+rnorm(1,0,sqrt(0.05))))*N_BH[t]/(1+alpha*N_BH[t])}
for (t in 1:(tmax-1)){N_BH1[t+1] =
  (exp(r+rnorm(1,0,sqrt(0.1))))*N_BH1[t]/(1+alpha*N_BH1[t])}
###
par(pch=20,cex=1.5)
plot(1:tmax,N_BH,type="o",ylim=range(c(N_BH,N_BH1)),xlab="Time",ylab="N")
lines(1:tmax,N_BH1,type="o",col="blue")
legend("topleft",c(expression(paste(sigma^"2", "=0.05", sep="")),
  expression(paste(sigma^"2", "=0.1", sep=""))),
  col=c("black", "blue"),lty=1,pch=16,bty="n")
```



# Fitting the logistic model in discrete-time with process noise II



# Let's fit the model

```
logistic.data <- list(logN = log(N_BH), tmax=tmax)

cat(file="logistic.growth.txt", "
model {
  r ~ dnorm(2, 0.01) ## prior on r
  alpha ~ dlnorm(1, 0.01) ## prior on alpha
  K <-(exp(r)-1)/alpha

  sigma ~ dunif(0.01,2)
  tau<-pow(sigma,-2)

  logN[1] ~ dnorm(0,1)
  N[1] <-exp(logN[1])

  #Likelihood
  for (t in 1:(tmax-1)){
    logNpred[t] <- logN[t]+ r - log(1 + alpha*N[t])
    logN[t+1] ~ dnorm(logNpred[t],tau)
    N[t+1] <- exp(logN[t+1])
  }
}
")
```

# Running the model I

```
# Inits function
inits <- function(){list(r = rnorm(1, 0, 1),
                        alpha = rlnorm(1,0,1))}

# Parameters to estimate
params <- c("r", "alpha", "K", "sigma")

# MCMC settings
nc <- 3 ; ni <- 2000 ; nb <- 1000 ; nt <- 2

# Call JAGS, check convergence and summarize posteriors
out <- jags(logistic.data, inits, params, "logistic.growth.txt", n.thin = nt,
           n.chains = nc, n.burnin = nb, n.iter = ni)
```

```
## Compiling model graph
##   Resolving undeclared variables
##   Allocating nodes
## Graph information:
##   Observed stochastic nodes: 25
##   Unobserved stochastic nodes: 3
##   Total graph size: 183
##
## Initializing model
```

## Running the model II

```
print(out, dig = 3)      # Bayesian analysis

## Inference for Bugs model at "logistic.growth.txt", fit using jags,
## 3 chains, each with 2000 iterations (first 1000 discarded), n.thin = 2
## n.sims = 1500 iterations saved
##
```

	mu.vect	sd.vect	2.5%	25%	50%	75%	97.5%	Rhat	n.eff
## K	36.144	5.224	28.599	32.810	35.348	38.596	48.108	1.006	440
## alpha	0.028	0.009	0.012	0.022	0.028	0.034	0.048	1.042	56
## r	0.683	0.140	0.399	0.591	0.684	0.775	0.950	1.041	55
## sigma	0.245	0.040	0.179	0.218	0.242	0.268	0.338	1.000	1500
## deviance	0.842	2.927	-2.685	-1.258	0.095	2.263	8.236	1.001	1500

```
##
## For each parameter, n.eff is a crude measure of effective sample size,
## and Rhat is the potential scale reduction factor (at convergence, Rhat=1).
##
## DIC info (using the rule,  $pD = \text{var}(\text{deviance})/2$ )
##  $pD = 4.3$  and  $DIC = 5.1$ 
## DIC is an estimate of expected predictive error (lower deviance is better).
```

# Showing traceplots

```
S<-ggs(as.mcmc(out)) #R2jags  
S<-filter(S,Parameter != "deviance")  
ggs_traceplot(S)
```



# Showing correlations ( $r, K$ ) and ( $r, \alpha$ )

`ggs_pairs(S)`

