Odds and ends

ISTA 410 / INFO 510: Bayesian Modeling and Inference

U. of Arizona School of Information

November 17, 2021

Outline

Last week:

- Linear dynamical systems and the Kalman filter
- Hidden Markov models

Today:

- Zero-inflated models
- A parametric functional model

Zero-inflated models

Example

Data file: `fish.csv`

Contains:

- Records of visits to national parks
- Each row is a visit
- Recorded:
    - Number of adults and children in the group
    - How long the group stayed in the park (hours)
    - Whether the group had a camper
    - Whether the group had live bait for fishing
    - Number of fish caught

A potential modeling problem

Suppose we want to assess whether live bait leads to more fish caught.

- Try a Poisson GLM
- Reasonable to assume that the number of people in the group influences fish caught, so add adults and children in as covariates
- Time-in-park should be accounted for using an offset

Offset in the Poisson model

Poisson RV models a count of events in a fixed time interval

- Assumes constant rate of independently occurring events, λ events/unit time
- Linear model:

$$\log \lambda_i = \alpha + \beta_A A_i + \beta_C C_i + \beta_B B_i$$

- Take $\lambda = \frac{\mu_i}{\tau_i}$, where τ_i is time-in-park

$$\log \mu_i = \log \tau_i + \alpha + \beta_A A_i + \beta_C C_i + \beta_B B_i$$

- So we just need to add $\log \tau_i$ to our model equation

Offset in the Poisson model

```
rate = pm.math.exp(np.log(fish.hours.values)
                   + alpha
                   + b_bait * fish.livebait
                   + b_adults * fish.persons
                   + b_children * fish.child)
```

- On the outcome scale:

$$\mu_i = \tau_i \exp(\alpha + \beta_A A_i + \beta_C C_i + \beta_B B_i)$$

- so, for otherwise identical visits the expected fish caught is proportional to the time spent

Full model

So the full model is:

$$
\begin{aligned}
y_i &\sim \mathrm{Poisson}(\mu) \\
\mu &= \log \tau_i + \alpha + \beta_A A_i + \beta_C C_i + \beta_B B_i \\
\alpha &\sim \mathrm{Normal}(0, 0.5) \\
\beta_{\cdot} &\sim \mathrm{Normal}(0.5, 0.5)
\end{aligned}
$$

Let's write the code and sample it.

Full model code

```
with pm.Model() as offset_fish_model:
    alpha = pm.Normal('alpha', 0, 0.5)
    b_adults = pm.Normal('b_adults', 0.5, 0.5)
    b_children = pm.Normal('b_children', 0.5, 0.5)
    b_bait = pm.Normal('b_bait', 0.5, 0.5)

    rate = pm.math.exp(np.log(fish.hours.values)
                       + alpha
                       + b_bait * fish.livebait
                       + b_adults * fish.persons
                       + b_children * fish.child)

    fish_caught = pm.Poisson('fish_caught', rate, observed = fish.fish_caught)
```

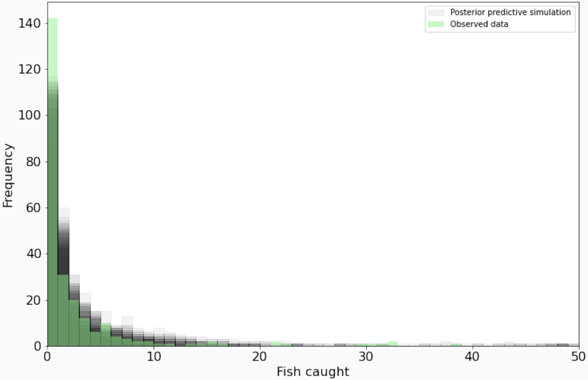
Some results

After uneventful sampling:

	mean	sd	hdi_3%	hdi_97%
alpha	-2.798	0.130	-3.049	-2.560
b_adults	0.685	0.032	0.622	0.741
b_children	0.453	0.081	0.300	0.608
b_bait	0.428	0.120	0.199	0.649

Posterior predictive check

A routine posterior predictive check:



What's going on

- The data set is missing a key variable: whether the group attempted fishing
- Some groups visit the park to hike, camp, etc. but do not attempt to catch fish – obviously, these are guaranteed zeros
- Zero-inflated Poisson:
    - Mixture model: mix zeros with Poisson RVs
    - Two parameters: p, λ
    - With probability p, the random variable is a $\mathrm{Poisson}(\lambda)$ RV
    - With probability $1 - p$, the random variable is 0

As a DAG

New model

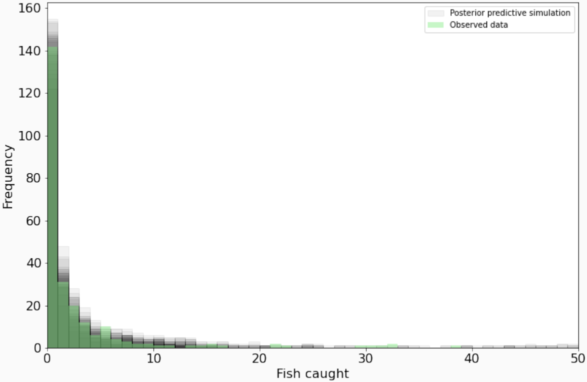
$$
\begin{aligned}
y_i &\sim \mathrm{ZIPoisson}(\mu, p) \\
\mu &= \log \tau_i + \alpha + \beta_A A_i + \beta_C C_i + \beta_B B_i \\
\alpha &\sim \mathrm{Normal}(0, 0.25) \\
\beta. &\sim \mathrm{Normal}(0.5, 0.5) \\
p &\sim \mathrm{Beta}(3, 2)
\end{aligned}
$$

Some results

	mean	sd	hdi_3%	hdi_97%
alpha	-2.074	0.149	-2.359	-1.813
b_adults	0.524	0.034	0.460	0.587
b_children	0.526	0.084	0.376	0.683
b_bait	0.342	0.133	0.089	0.597
fishing_p	0.720	0.045	0.636	0.806

Rerun the PPC

With zero-inflation:



Differences

- The new model is more trustworthy, based on the posterior predictive check
- New model shows a weaker live bait effect – why?

Differences

- The new model is more trustworthy, based on the posterior predictive check
- New model shows a weaker live bait effect – why?
- Without zero-inflation, live bait is a proxy variable for fishing

Parametric functional model

Cracking nuts

- Panda nuts – a kind of nut primates like to eat
- Chimpanzees crack them open using tools

2

- Want to study/model chimpanzees learning the skill
- Have observations of nut-opening "sessions"
  - Age, sex, tool used, time spent, nuts opened

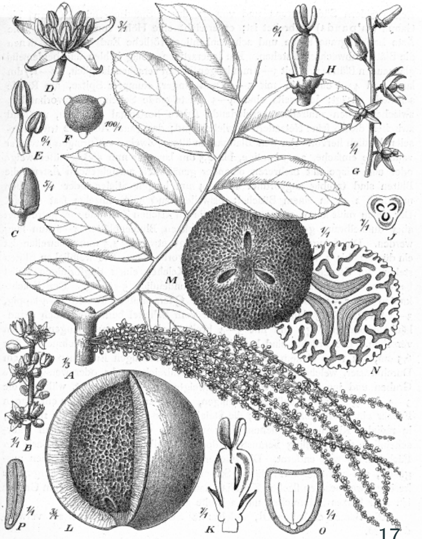


Fig. 317. Panda oleosa Pierre. *A* Zweig mit männlichen Blütenständen, *B* Stücke des Blütenstandes, vergr.; *C* Knospe; *D* ♂ Blüte geöffnet; *E* ein äußeres und ein inneres Staubblatt; *F* Pollen; *G* Stück eines weiblichen Blütenstandes; *H* Kelch und Pistill der ♀ Blüte; *J* Quer-

A naive GLM

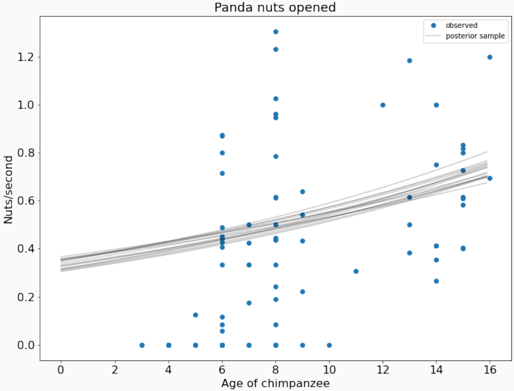
As a chimp gets older, it gets larger/stronger and thus is more able to open nuts.

We can attempt to model this with a naive Poisson GLM:

$$
\begin{aligned}
y_i &\sim \mathrm{Poisson}(\mu) \\
\log \mu_i &= \log \tau_i + \alpha + \beta_A A_i \\
\alpha &\sim \mathrm{Normal}(0, 0.1) \\
\beta_A &\sim \mathrm{Normal}(0.1, 0.05)
\end{aligned}
$$

Result

After some uneventful sampling:



Result

- Fit appears mediocre, and there is a clear problem at 0
- Model predicts a baby chimpanzee will take about 3 seconds to crack a nut
- Probably, baby chimps cannot crack any nuts at all!

Problem: GLM not constrained to pass through 0

Better: use a little scientific intuition to design the model

Deriving a crude model

We'll derive a very simple and not very good model from an ODE – but still better than the GLM!

In animals that grow to a stable adult size, growth rate is proportional to the growth remaining.

Mathematically,

$$\frac{dM}{dt} = k(M_{max} - M(t))$$

As a result,

$$M(t) = M_{max}(1 - \exp(-kt))$$

Thinking about the effect of mass

Imagine that strength S is proportional to mass M:

- $S = \beta M$
- Higher strength helps with nut opening in multiple ways, so we expect nut opening rate to be proportional to S^θ for some θ:

$$\mu = \alpha(\beta M_{max}(1 - \exp(-kt)))^\theta$$

- A lot of parameters, but we can scale some of them out

Thinking about the effect of mass

Starting from:

$$\mu = \alpha(\beta M_{max}(1 - \exp(-kt)))^{\theta}$$

Mass measurement scale is arbitrary, so we set $M_{max} = 1$:

$$\mu = \alpha\beta^{\theta}(1 - \exp(-kt))^{\theta}$$

The factor in front is overparameterized; set $\phi = \alpha\beta^{\theta}$:

$$\mu = \phi(1 - \exp(-kt))^{\theta}$$

Writing down the model

Now we can write down a model:

$$y_i \sim \mathrm{Poisson}(\mu_i)$$
$$\mu_i = \tau_i \phi (1 - \exp(-kt))^\theta$$

Priors should take into account some reasonable biological and physical assumptions:

- Most importantly, the growth rate k should have the growth flattening off around 12 years (when chimpanzees reach adult mass)
- The prior for ϕ should have a mean near the maximum nut opening rate (maybe around one nut/second?)

Writing down the model

$$
\begin{aligned}
y_i &\sim \mathrm{Poisson}(\mu_i) \\
\mu_i &= \tau_i \phi (1 - \exp(-kt))^{\theta} \\
\phi &\sim \mathrm{LogNormal}(\log(1), 0.1) \\
k &\sim \mathrm{LogNormal}(\log(2), 0.25) \\
\theta &\sim \mathrm{LogNormal}(\log(5), 0.25)
\end{aligned}
$$

Log-normal:

- Constrained to be positive and zero-avoiding

The model code

Model code is little different from any other model:

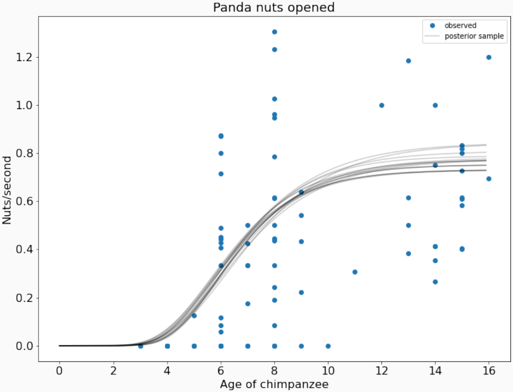
```
with pm.Model() as model:
    # Priors
    phi = pm.Lognormal('phi', np.log(1), 0.1)
    k = pm.Lognormal('k', np.log(2), 0.25)
    theta = pm.Lognormal('theta', np.log(5), 0.25)

    # Model equation
    rate = (data.seconds.values
            * phi
            * (1 - pm.math.exp(-k*data.age.values))
            ** theta)

    # Likelihood
    y_ = pm.Poisson('y', rate, observed = data.nuts_opened)
    trace = pm.sample()
```

Results

With the new model:



ODE models in general

- This model was built on an ODE – but a pretty trivial one
- More complex ODEs: solve numerically
- Section 16.4 of *Rethinking*: Lotka-Volterra equations
- ODE module in PyMC3 seems a bit rusty

Summary

Today:

- Simple models can be improved by more carefully considering the data generating process
  - Mixture models for when there are two processes combined
  - Parameterized functional models (e.g. from ODEs) when the linear model does not make sense

Next week:

- Modeling missing data
- Grab bag topics