Zero-Inflated Models

General Principles

Zero-Inflated Regression models are used when the outcome variable is a count variable with an excess of zero counts. These models combine a count model (e.g., Poisson or Negative Binomial) with a separate model for predicting the probability of excess zeros.

Example

Below is an example code snippet demonstrating Bayesian Zero-Inflated Poisson regression using the Bayesian Inference (BI) package. The data represent the production of books in a monastery (y), which is affected by the number of days that individuals work, as well as the number of days individuals drink. This example is based on McElreath (2018).

Python

```
from BI import bi
from jax.scipy.special import expit
# Setup device ------------------------------------------------
m = bi('cpu')

# Simulated data------------------------------------------------
prob_drink = 0.2  # 20% of days
rate_work = 1     # average 1 manuscript per day

# Sample one year of production
N = 365
drink = m.dist.binomial(1, prob_drink, shape = (N,), sample = True)
y = (1 - drink) * m.dist.poisson(rate_work, shape = (N,), sample = True)

# Setup device------------------------------------------------
```

```python
m = bi(platform='cpu')
m.data_on_model = dict(
    y=jnp.array(y)
)

# Define model ------------------------------------------------
def model(y):
    al = dist.normal(1, 0.5, name='al')
    ap = dist.normal(-1.5, 1, name='ap')
    p = expit(ap)
    lambda_ = jnp.exp(al)
    m.zero_inflated_poisson(p, lambda_, obs=y)

# Run MCMC ------------------------------------------------
m.fit(model)

# Summary ------------------------------------------------
m.summary()
```

R

```r
library(BI)

# setup platform------------------------------------------------
m=importbi(platform='cpu')

# Simulate data ------------------------------------------------
prob_drink = 0.2  # 20% of days
rate_work = 1     # average 1 manuscript per day
# sample one year of production
N = as.integer(365)
drink = bi.dist.binomial(total_count = as.integer(1), probs = prob_drink, shape = c(N), sampl
y = (1 - drink) *  bi.dist.poisson(rate_work, shape = c(N), sample = T)
data = list()
data$y = y
m$data_on_model = data

# Define model ------------------------------------------------
model <- function(y){
  al = bi.dist.normal(1, 0.5, name='al', shape=c(1))
  ap = bi.dist.normal(-1, 1, name='ap', shape=c(1))
```

```
  p = jax$scipy$special$expit(ap)
  lambda_ = jnp$exp(al)
  m$zeroinflatedpoisson(p, lambda_, obs=y)
}

# Run MCMC ------------------------------------------------
m$fit(model) # Optimize model parameters through MCMC sampling

# Summary ------------------------------------------------
m$summary() # Get posterior distribution
```

Mathematical Details

We model the relationship between the independent variable X and the count outcome variable Y. The model assumes that the data-generating process has two states: a “zero” state that generates only zero counts, and a “Poisson” state that generates counts from a Poisson distribution.

The model has two main components:

$$Y_i \sim \text{ZIPoisson}(\pi_i, \lambda_i) \sim \begin{cases} \pi + (1-\pi_i) \times \text{Poisson}(0|\lambda_i) & \text{if } y = 0 \\ (1-\pi_i) \times \text{Poisson}(y|\lambda_i) & \text{if } y > 0 \end{cases}$$

1. $\pi + (1-\pi_i) \times \text{Poisson}(0|\lambda_i)$ A logistic regression model to predict the probability, π_i, of an observation being in the “zero” state (an excess zero). Where, the probability of getting a zero (y=0) is the sum of two chances: the probability of a structural zero (π) plus the probability of not being a structural zero (1-π) and then getting a zero from the Poisson distribution anyway.
2. $(1-\pi_i) \times \text{Poisson}(y|\lambda_i)$ A Poisson regression model to predict the expected count, λ_i, for observations in the “Poisson” state. Where, the probability of getting a non-zero count (y>0) is simply the probability of not being a structural zero (1-π) multiplied by the probability of the Poisson distribution giving you that specific count.

The parameters π_i and λ_i are modeled as functions of the independent variable X_i:

$$\text{logit}(\pi_i) = \alpha_\pi + \beta_\pi X_i$$

$$\log(\lambda_i) = \alpha_\lambda + \beta_\lambda X_i$$

We can assign prior distributions to the model parameters with weakly informative Normal priors:

$$\alpha_\pi \sim \text{Normal}(0, 1)$$

$$\beta_\pi \sim \text{Normal}(0, 1)$$

$$\alpha_\lambda \sim \text{Normal}(0, 1)$$

$$\beta_\lambda \sim \text{Normal}(0, 1)$$

Reference(s)

McElreath, Richard. 2018. *Statistical Rethinking: A Bayesian Course with Examples in r and Stan.* Chapman; Hall/CRC.