Gamma-Poisson (Negative Binomial) Model

General Principles

To model the relationship between a count outcome variable and one or more independent variables with overdispersion , we can use the *Negative Binomial model*.

Considerations

> Caution
>
> - We have the same considerations as for the Poisson model.
> - Overdispersion is handled because the Negative Binomial model assumes that each Poisson count observation has its own rate. This is an additional parameter specified in the model (in the code, it is `log_days`).

Example

Below is an example code snippet demonstrating a Bayesian Gamma-Poisson model using the Bayesian Inference (BI) package. This example is based on McElreath (2018).

Python

```
from BI import bi
# Setup device ------------------------------------------------
m = bi(platform='cpu') # Import

# Import Data & Data Manipulation ------------------------------------------------
# Import
from importlib.resources import files
```

```python
data_path = files('BI.resources.data') / 'Sim dat Gamma poisson.csv'
m.data(data_path, sep=',')
m.data_to_model(['log_days', 'monastery', 'y']) # Send to model (convert to jax array)

# Define model ------------------------------------------------
def model(log_days, monastery, y):
    a = m.dist.normal(0, 1, name = 'a', shape=(1,))
    b = m.dist.normal(0, 1, name = 'b', shape=(1,))
    l = m.jnp.exp(log_days + a + b * monastery)
    m.poisson(rate = l, obs=y)
# Run MCMC ------------------------------------------------
m.fit(model) # Optimize model parameters through MCMC sampling

# Summary ------------------------------------------------
m.summary() # Get posterior distributions
```

R

```r
library(BI)

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

# Import data ------------------------------------------------
m$data(paste(system.file(package = "BI"),"/data/Sim dat Gamma poisson.csv", sep = ''), sep='
m$data_to_model(list('log_days', 'monastery', 'y' )) # Send to model (convert to jax array)

# Define model ------------------------------------------------
model <- function(log_days, monastery, y){
  # Parameter prior distributions
  alpha = bi.dist.normal(0, 1, name='alpha', shape=c(1))
  beta = bi.dist.normal(0, 1, name='beta', shape=c(1))
  l = jnp$exp(log_days + alpha + beta * monastery)
  # Likelihood
  m$poisson(rate=l, obs=y)
}

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

# Summary ------------------------------------------------
```

```
m$summary() # Get posterior distributions
```

Mathematical Details

Frequentist formulation

We model the relationship between the independent variable X and the count outcome variable Y using the following equation:

$$\log(\lambda_i) = \exp(\text{rates}_i + \alpha + \beta X_i)$$

Where:

- λ_i is the mean rate parameter of the negative binomial distribution (expected count) for observation i.
- $\log(\lambda_i)$ is the log of the mean rate parameter, ensuring it is positive for observation i.
- α is the intercept term.
- β is the regression coefficient.
- X_i is the value of the predictor variable for observation i.

Bayesian model

In the Bayesian formulation, we define each parameter with priors . We can express the Bayesian regression model accounting for prior distributions as follows:

$$Y_i \sim \text{Poisson}(\lambda_i)$$

$$\log(\lambda_i) = \text{rates}_i + \alpha + \beta X_i$$

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

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

Where:

- Y_i is the dependent variable for observation i.
- λ_i is the mean rate parameter of the Poisson distribution for observation i, assuming that each Poisson count observation has its own $rate_i$.
- $\log(\lambda_i)$ is the log of the mean rate parameter for observation i, ensuring it is positive.
- α is the intercept term.
- β is the regression coefficient.
- X_i is the value of the predictor variable for observation i.

Notes

> **Note**
>
> - We can apply multiple variables similarly as in chapter 2.
> - We can apply interaction terms similarly as in chapter 3.
> - We can apply categorical variables similarly as in chapter 4.

Reference(s)

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