Gamma-Poisson 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 Gamma-Poisson 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.

Python

```
from BI import bi, jnp

# Setup device ------------------------------------------------
m = bi(platform='cpu') # Import

# Import Data & Data Manipulation -----------------------------------------------
# Import
```

```python
from importlib.resources import files
data_path =  m.load.sim_gamma_poisson(only_path=True)
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,))
    phi = m.dist.exponential(1, name = 'phi', shape=(1,))
    mu = jnp.exp(log_days + a + b * monastery)
    Lambda =  m.dist.gamma(rate = mu*phi, concentration = phi, name = 'Lambda')
    m.dist.poisson(rate = Lambda, obs=y)
# Run MCMC ----------------------------------------------------
m.fit(model) # Optimize model parameters through MCMC sampling

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

```
jax.local_device_count 32
```

```
  0%|          | 0/1000 [00:00<?, ?it/s]warmup:   0%|          | 1/1000 [00:00<08:57,  1.86it
arviz - WARNING - Shape validation failed: input_shape: (1, 500), minimum_shape: (chains=2, d
```

	mean	sd	hdi_5.5%	hdi_94.5%	mcse_mean	mcse_sd	ess_bulk	ess_tail	r_hat
Lambda[0]	1.56	0.38	0.97	2.11	0.01	0.02	793.43	367.44	NaN
Lambda[1]	1.51	0.36	1.02	2.12	0.01	0.02	877.99	293.31	NaN
Lambda[2]	1.58	0.38	0.93	2.12	0.01	0.02	741.78	372.31	NaN
Lambda[3]	1.40	0.34	0.93	1.96	0.01	0.02	937.17	277.70	NaN
Lambda[4]	1.47	0.37	0.89	2.00	0.02	0.02	507.60	428.36	NaN
...	...	...	...	...	...	...	...	...	...
Lambda[3398]	3.53	0.77	2.43	4.67	0.03	0.04	914.23	399.13	NaN
Lambda[3399]	3.68	0.79	2.56	5.06	0.03	0.06	727.90	235.65	NaN
a[0]	-0.42	0.01	-0.45	-0.40	0.00	0.00	40.06	101.97	NaN
b[0]	-2.75	0.03	-2.81	-2.71	0.00	0.00	57.05	188.62	NaN
phi[0]	17.63	1.91	14.32	20.23	0.56	0.18	12.25	44.88	NaN

R

```
library(BayesianInference)

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

# Import data -------------------------------------------------
m$data(m$load$sim_gamma_poisson(only_path=T), sep = '')
m$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))
  phi = bi.dist.exponential(1, name='phi', shape=c(1))
  mu = jnp$exp(log_days + alpha + beta * monastery)
  Lambda =  bi.dist.gamma(rate = mu*phi, concentration = phi, name = 'Lambda')
  # Likelihood
  bi.dist.poisson(rate=Lambda, obs=y)
}
# Run MCMC ----------------------------------------------------
m$fit(model) # Optimize model parameters through MCMC sampling

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

Mathematical Details

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:

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

$$\lambda_i \sim \text{Gamma}(\mu_i \phi, \phi)$$

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

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

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

$$\phi \sim \text{Exponential}(1)$$

Where:

- Y_i is the dependent variable for observation i.
- λ_i is the rate parameter of the Poisson distribution for observation i, assuming that each Poisson count observation has its own $rate_i$.
- μ_i is the mean rate parameter.
-
- ϕ controls the level of overdispersion in the rates.
- 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)