Beta-Binomial Model

General Principles

To model the relationship between a binary outcome variable representing success counts and one or more independent variables with overdispersion , we can use the *Beta-Binomial model*.

Considerations

Caution

- We have the same considerations as for Binomial regression.
- A Beta-Binomial model assumes that each binomial count observation has its own probability of success. The model estimates the distribution of probabilities of success across cases, instead of a single probability of success.
- A Beta distribution has two parameters: the rates for each probability and a shape parameter . influences how probabilities are distributed between 0 and 1. Specifically, it consists of two parameters, α and β, which determine the concentration of probability around 0 and 1.
  - If both are equal to or greater than 1, the distribution is bell-shaped and centered around 0.5.
  - If $\alpha > \beta$, the distribution is skewed toward 1, and if $\beta > \alpha$, it is skewed toward 0. Thus, the shape parameters γ and η provide flexibility in modeling various types of prior beliefs about probabilities.

Example

Below is an example code snippet demonstrating Bayesian Beta-Binomial regression using the Bayesian Inference (BI) package. The data consist of:

1) One binary dependent variable (admit), which represents candidates' admission status.
2) One independent categorical variable representing individuals' gender (gid).
3) Additionally, we have the number of applications (applications) per gender, which will be used to account for independent rates.

The goal is to evaluate whether the probability of admission is different between genders, while accounting for differences in the number of applications between genders.

Python

```python
from BI import bi

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

# Import Data & Data Manipulation ------------------------------------------------
# Import
from importlib.resources import files
data_path = files('BI.resources.data') / 'UCBadmit.csv'
m.data(data_path, sep=';')
m.df["gid"] = (m.df["applicant.gender"] != "male").astype(int)

# Define model ------------------------------------------------
def model(gid, applications, admit):
    phi = m.dist.exponential(1,  name = 'phi')
    alpha = m.dist.normal( 0., 1.5, shape=(2,), name = 'alpha')
    theta =  phi + 2
    pbar = jax.nn.sigmoid(alpha[gid])
    concentration1 = pbar*theta
    concentration0 = (1 - pbar) * theta

    m.dist.betabinomial(total_count = applications, concentration1 = concentration1, concent

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

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

R

```r
library(BI)

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

# import data ------------------------------------------------
m$data(paste(system.file(package = "BI"),"/data/UCBadmit.csv", sep = ''), sep=';')
m$df["gid"] = as.integer(ifelse(m$df["applicant.gender"] == "male", 0, 1)) # Manipulate
m$data_to_model(list('gid', 'applications', 'admit' )) # Send to model (convert to jax array)

# Define model ------------------------------------------------
model <- function(gid, applications, admit){
  # Parameter prior distributions
  phi = bi.dist.exponential(1, name = 'phi',shape=c(1))
  alpha = bi.dist.normal(0., 1.5, shape= c(2), name='alpha')
  t = phi + 2
  pbar = jax$nn$sigmoid(alpha[gid])
  gamma = pbar * t
  eta = (1 - pbar) * t
  # Likelihood
  m$betabinomial(total_count=applications, concentration1=gamma, concentration0=eta, obs=admi
}

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

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

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:

$$Y_i \sim BetaBinomial(n_i, \gamma_i, \eta_i)$$

$$\gamma_i = \overline{\rho}\tau$$

$$\eta_i = (1 - \overline{\rho})\tau$$

$$\overline{\rho} = logit(\alpha_i)$$

$$\tau = \phi + 2$$

$$\alpha \sim Normal(0, 1)$$

$$\phi \sim Exponential(1)$$

Where:

- Y_i is the count of successes for the i-th observation, which follows a beta-binomial distribution with n_i trials.
- γ_i represents the concentration parameter for the number of successes, derived from the probability of success and scaled by τ.
- η_i represents the concentration parameter for failures, derived from the probability of failure $(1 - \overline{\rho})$ and also scaled by τ.
- $\overline{\rho}$ is the probability of success for the i-th observation. The logit function transforms the linear predictor (which can take any real value) into a probability value between 0 and 1.
- τ is derived from and is used as a scaling factor for the shape parameters and .
- is a vector of parameters, each representing the effect of group i on the success probability.
- is a random variable following an exponential distribution with a rate of 1.

Reference(s)

McElreath (2018)

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