Binomial Model (Logistic Regression)

General Principles

To model the relationship between a binary dependent variable—e.g., success/failure, yes/no, or 1/0—and one or more independent variables, we can use a *Binomial model*.

Considerations

Caution

- We have the same considerations as for Regression for continuous variable.
- The first link function is the *logit*. The *logit* link function in the Bayesian binomial model converts the linear combination of predictor variables into probabilities, making it suitable for modeling binary outcomes. It helps to estimate the relationship between predictors and the probability of success, ensuring results fall within the bounds of the binomial distribution $\in [0, 1]$.

Example

Below is an example code snippet that demonstrates Bayesian binomial regression using the Bayesian Inference (BI) package. The data consist of one binary dependent variable (*pulled_left*), which represents which side individuals pulled. The goal is to evaluate the probability of pulling the left side.

Python

```python
from BI import bi

# setup platform------------------------------------------------
m = bi(platform='cpu')
# import data ------------------------------------------------
# Import
from importlib.resources import files
data_path = files('BI.resources.data') / 'chimpanzees.csv'
m.data(data_path, sep=';')
m.data_to_model(['pulled_left'])

# Define model ------------------------------------------------
def model(pulled_left):
    a = m.dist.normal( 0, 10, shape=(1,), name = 'a')
    m.binomial(logits=a[0], obs=pulled_left)

# Run sampler ------------------------------------------------
m.fit(model, num_samples=500)

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

R

```r
library(BI)

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

# import data ------------------------------------------------
m$data(paste(system.file(package = "BI"),"/data/chimpanzees.csv", sep = ''), sep=';')
m$data_to_model(list('pulled_left')) # Send to model (convert to jax array)

# Define model ------------------------------------------------
model <- function(pulled_left){
  # Parameters priors distributions
  alpha = bi.dist.normal( 0, 10, name = 'alpha')
  # Likelihood
  m$binomial(logits = alpha, obs=pulled_left)
}
```

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

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

Mathematical Details

Frequentist formulation

We model the relationship between the independent variable (X_i) and the binary dependent variable (Y_i) using the following equation:

$$logit(Y_i) = \alpha + \beta X_i$$

Where:

- Y_i is the probability of success (i.e., the probability of the binary outcome being 1) for observation i.
- α is the intercept term.
- β is the regression coefficient.
- X_i is the value of the independent variable for observation i.
- $logit(Y_i)$ is the log-odds of success, calculated as the log of the odds ratio of success. Through this link function, the relationship between the independent variables and the log-odds of success is modeled linearly, allowing us to interpret the effect of each independent variable on the log-odds of success for observation i.

Bayesian formulation

priors . We can express the Bayesian regression model accounting for prior distributions as follows:

$$Y_i \sim Binomial(n = 1, p)$$

$$logit(p) \sim \alpha + \beta X_i$$

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

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

Where:

- Y_i is the probability of success (i.e., the probability of the binary outcome being 1) for observation i.
- $n = 1$ represents the number of trials in the binomial distribution (binary outcome).
- β and α are the prior distributions for the regression coefficient and intercept, respectively.
- *logit* is the log-odds of success, calculated as the log of the odds ratio of success. Through this link function, the relationship between the independent variables and the log-odds of success is modeled linearly, allowing us to interpret the effect of each independent variable on the log-odds of success for observation i.

Notes

Note

- We can apply multiple variables similarly to chapter 2.
- We can apply interaction terms similarly to chapter 3.
- We can apply categorical variables similarly to chapter 4.
- Below is an example code snippet demonstrating a Bayesian binomial model for multiple categorical variables using the Bayesian Inference (BI) package. The data consist of one binary dependent variable (*pulled_left*), which represents which side individuals pulled, and three independent variables (*actor*, *side*, *cond*). The goal is to evaluate, for each individual, the probability of pulling the left side, accounting for whether the individual is left-handed or right-handed, as well as the different conditions.

```python
from BI import bi
m = bi(platform='cpu')
m.data('../resources/data/chimpanzees.csv', sep=';')
m.df['treatment'] =  m.df.prosoc_left + 2 * m.df.condition
m.df['actor'] = m.df['actor'] - 1

m.data_to_model(['actor', 'treatment', 'pulled_left'])

def model(actor, treatment, pulled_left):
    a = m.dist.normal(0, 1.5, shape = (7,), name='a')
    b = m.dist.normal(0, 0.5, shape = (4,), name='b')
    p = a[actor] + b[treatment]
    m.lk("y", m.dist.binomial(1, logits=p), obs=pulled_left)

# Run sampler ------------------------------------------------
m.fit(model)
# Diagnostic ------------------------------------------------
m.summary()
```

Reference(s)

McElreath (2018)

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