Multinomial Model

General Principles

To model the relationship between a categorical outcome variable with more than two categories and one or more independent variables, we can use a *Multinomial* model.

Considerations

Caution

- We have the same considerations as for Regression for continuous variable.
- One way to interpret a multinomial model is to consider that we need to build $K-1$ linear models, where K is the number of categories. Once we get the linear prediction for each category, we can convert these predictions to probabilities by building a simplex . To do this, we convert the regression outputs using the softmax function (see the "jax.nn.softmax" line in the code).
- The intercept α captures the difference in the log-odds of the outcome categories; thus, different categories need different intercepts.
- On the other hand, as we assume that the effect of each predictor on the outcome is consistent across all categories, the regression coefficients β are shared across categories.
- The relationship between the predictor variables and the log-odds of each category is modeled linearly, allowing us to interpret the effect of each predictor on the log-odds of each category.

Example

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

Python

```python
from BI import bi
import jax.numpy as jnp
import pandas as pd
import jax
# Setup device ------------------------------------------------
m = bi('cpu')

# Import Data & Data Manipulation ------------------------------------------------
# Import
from importlib.resources import files
data_path = files('BI.resources.data') / 'Sim data multinomial.csv'
m.data(data_path, sep=',')

# Define model ------------------------------------------------
def model(career, income):
    a = m.dist.normal(0, 1, shape=(2,), name = 'a')
    b = m.dist.halfnormal(0.5, shape=(1,), name = 'b')
    s_1 = a[0] + b * income[0]
    s_2 = a[1] + b * income[1]
    s_3 = [0] #pivot
    p = jax.nn.softmax(jnp.stack([s_1[0], s_2[0], s_3[0]]))
    m.dist.categorical(probs=p, obs=career)

# Run sampler ------------------------------------------------
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 data multinomial.csv", sep = ''), sep=',
keys <- c("income", "career")
```

```r
income = unique(m$df$income)
income = income[order(income)]
values <- list(jnp$array(as.integer(income)),jnp$array( as.integer(m$df$career)))
m$data_on_model = py_dict(keys, values, convert = TRUE)

# Define model ------------------------------------------------
model <- function(income, career){
  # Parameter prior distributions
  alpha = bi.dist.normal(0, 1, name='alpha', shape = c(2))
  beta = bi.dist.halfnormal(0.5, name='beta')

  s_1 = alpha[0] + beta * income[0]
  s_2 = alpha[1] + beta * income[1]
  s_3 = 0 # reference category

  p = jax$nn$softmax(jnp$stack(list(s_1, s_2, s_3)))

  # Likelihood
  m$categorical(probs=p[career], obs=career)
}

# 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 predictor variables (X1, X2, ..., Xn) and the categorical outcome variable (Y_i) using the following equation:

$$logit(p_{ik}) = \alpha_k + \beta X_i$$

Where:

- p_{ik} is the probability of the -th observation being in category .
- α_k is the intercept for category .

- β is the regression coefficients common to all categories.
- X_i is the vector of independent variables for the -th observation.
- A reference category is often chosen to simplify the model.

Bayesian model

In Bayesian multinomial modeling, the likelihood function of the data is specified using a multinomial distribution. The multinomial distribution models the counts of outcomes falling into different categories. For an outcome variable with categories, the multinomial likelihood function is:

$$Multinomial(y|\theta) = \frac{N!}{\prod_{k=1}^{K} y_k!} \prod_{k=1}^{K} \theta_k^{y_k}$$

Where:

- $y = (y_1, y_2, ..., y_K)$ represents the counts of observations in each of the categories.
- N is the total number of observations or trials.
- $\theta = (\theta_1, \theta_2, ..., \theta_K)$ is a simplex of category probabilities, with θ_k representing the probability of category .
- $\frac{N!}{\prod_{k=1}^{K} y_k!}$ is the multinomial coefficient that accounts for the number of ways to arrange the observations into the categories. This coefficient ensures that the likelihood function properly accounts for the permutations of the counts across different categories.

Reference(s)

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