Multivariate Linear Regression

General Principles

To study relationships between multiple continuous independent variables (e.g., the effect of weight and age on height), we can use a multiple regression approach. Essentially, we extend Linear Regression for continuous variable by adding a regression coefficient β_x for each continuous variable (e.g., β_{weight} and β_{age}).

Considerations

> ℹ Note
>
> - We have the same considerations as for the Regression for continuous variable.
> - The model interpretation of the regression coefficients β_x is considered for fixed values of the other independent variable(s)' regression coefficients—i.e., for a given age, β_{weight} represents the expected change in the dependent variable (height) for each one-unit increase in weight, holding all other variables (e.g., age) constant.

Example

Below is example code demonstrating Bayesian multiple linear regression using the Bayesian Inference (BI) package. Data consist of three continuous variables (*height*, *weight*, *age*), and the goal is to estimate the effect of *weight* and *age* on *height*. This example is based on McElreath (2018).

Python

```python
from BI import bi

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

# Import Data & Data Manipulation ------------------------------------------------
from importlib.resources import files
# Import
data_path = files('BI.resources.data') / 'Howell1.csv'
m.data(data_path, sep=';')
m.df = m.df[m.df.age > 18] # Subset data to adults
m.scale(['weight', 'age']) # Normalize

# Define model ------------------------------------------------
def model(height, weight, age):
    # Parameter prior distributions
    alpha = m.dist.normal(0, 0.5, name = 'alpha')
    beta1 = m.dist.normal(0, 0.5, name = 'beta1')
    beta2 = m.dist.normal(0, 0.5, name = 'beta2')
    sigma = m.dist.uniform(0, 50, name = 'sigma')
    # Likelihood
    m.dist.normal(alpha + beta1 * weight + beta2 * age, sigma, obs = height)

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

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

```
jax.local_device_count 16
```

```
  0%|          | 0/1000 [00:00<?, ?it/s]warmup:   0%|          | 1/1000 [00:01<23:20,  1.40s/
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
alpha	5.20	0.49	4.46	6.06	0.02	0.02	469.94	349.45	NaN
beta1	0.20	0.51	-0.60	1.03	0.02	0.03	570.81	264.55	NaN
beta2	-0.02	0.49	-0.89	0.69	0.02	0.02	576.00	338.15	NaN
sigma	49.98	0.02	49.96	50.00	0.00	0.00	579.62	271.13	NaN

R

```r
library(BI)
m=importbi(platform='cpu')

# Import Data & Data Manipulation -------------------------------------------------
m$data(paste(system.file(package = "BI"),"/data/Howell1.csv", sep = ''), sep=';')# Import
m$df = m$df[m$df$age > 18,] # Subset data to adults
m$scale(list('weight', 'age')) # Normalize
m$data_to_model(list('weight', 'height', 'age')) # Send to model (convert to jax array)

# Define model -------------------------------------------------
model <- function(height, weight, age){
  # Parameter prior distributions
  alpha = bi.dist.normal(0, 0.5, name = 'a')
  beta1 = bi.dist.normal(0, 0.5, name = 'b1')
  beta2 = bi.dist.normal(0, 0.5, name = 'b2')
  sigma = bi.dist.uniform(0, 50, name = 's')
  # Likelihood
  m$normal(alpha + beta1 * weight + beta2 * age, sigma, obs=height)
}

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

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

Caution

For R users, if you create the regression coefficient in a single call:

```r
betas = bi.dist.normal(0, 0.5, name = 'regression_coefficients', shape = (2,))
```

you need to index them starting by 0:

```r
 m$normal(alpha + betas[0] * weight + betas[1] * age, sigma, obs=height)
```

Mathematical Details

Frequentist formulation

We model the relationship between the independent variables $(X_{1i}, X_{2i}, ..., X_{[K,i]})$ and the dependent variable Y using the following equation:

$$Y_i = \alpha + \beta_1 X_{[1,i]} + \beta_2 X_{[2,i]} + ... + \beta_n X_{[K,i]} + \epsilon_i$$

Where:

- Y_i is the dependent variable for observation i.
- α is the intercept term.
- $X_{[1,i]}$, $X_{[2,i]}$, ..., $X_{[K,i]}$ are the values of the independent variables for observation i.
- β_1, β_2, ..., β_K are the regression coefficients.
- ϵ_i is the error term for observation i, and the vector of the error terms, ϵ, are assumed to be independent and identically distributed.

Bayesian formulation

In the Bayesian formulation, we define each parameter with priors . We can express the Bayesian model as follows:

$$Y_i \sim \text{Normal}(\alpha + \sum_{k=1}^{K} \beta_k X_{[K,i]}, \sigma)$$

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

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

$$\sigma \sim \text{Uniform}(0, 50)$$

Where:

- Y_i is the dependent variable for observation i.
- α is the intercept term, which in this case has a unit-normal prior.

- β_k are slope coefficients for the K distinct independent variables, which also have unit-normal priors.
- $X_{[1,i]}$, $X_{[2,i]}$, ..., $X_{[K,i]}$ are the values of the independent variables for observation i.
- σ is a standard deviation parameter, which here has a Uniform prior that constrains it to be positive.

Reference(s)

McElreath, Richard. 2018. *Statistical Rethinking: A Bayesian course with examples in R and Stan.* Chapman; Hall/CRC.