Universidad Externado

Bayesian Inference Tutorial: Modeling Pizza Preferences in Italy and the USA

Winter School Montevideo

Montevideo, Uruguay

Objective

This workshop introduces Bayesian inference using real-world scenarios related to pizza preferences. We will compare posterior distributions from different models using PyMC and ArviZ, exploring how the choice of prior and likelihood affects inference.

Scenario 1: Who orders Hawaiian Pizza more often? (Beta-Binomial)

Context: A survey was conducted in Italy and the USA. Each country surveyed 10 people asking whether they would order a Hawaiian pizza. Results:

- Italy: 2 out of 10 said yes.
- USA: 7 out of 10 said yes.

We model the probability θ of ordering a Hawaiian pizza using a **Beta-Binomial** model:

$$\theta \sim \text{Beta}(\alpha, \beta) \quad \text{and} \quad y \sim \text{Binomial}(n, \theta)$$

Scenario 2: Average Hawaiian Pizza Sales (Gamma-Poisson)

Context: Daily Hawaiian pizza sales for one week were recorded:

- Italy: [1, 2, 1, 0, 1, 2, 1]
- USA: [5, 6, 4, 7, 5, 6, 5]

We model the average number of pizzas sold per day using a **Gamma-Poisson** model:

$$\lambda \sim \text{Gamma}(\alpha, \beta) \quad \text{and} \quad y_i \sim \text{Poisson}(\lambda)$$

Scenario 3: Comparing Pizza Flavor Preferences (Normal)

Context: We track daily sales of three pizza flavors in each country:

- **Italy**
  - Hawaiian: [1, 2, 1, 0, 1, 2, 1]
  - Margherita: [10, 12, 11, 13, 10, 11, 12]
  - Pepperoni: [5, 4, 6, 5, 5, 4, 6]
- **USA**
  - Hawaiian: [5, 6, 4, 7, 5, 6, 5]
  - Margherita: [6, 5, 7, 6, 6, 5, 7]
  - Pepperoni: [8, 9, 8, 9, 8, 10, 9]

We model the mean preference using a **Normal model**:

$$\mu \sim \text{Normal}(0, 20) \quad \text{and} \quad y_i \sim \text{Normal}(\mu, \sigma) \quad \text{with} \quad \sigma \sim \text{HalfNormal}(5)$$

Activities

1. Estimate and visualize the posterior for each parameter.
2. Compare the posteriors between Italy and USA.
3. Interpret differences in uncertainty and central tendency.
4. Justify your prior choices.
5. Modify the prior and observe how posterior inference changes.
6. Implement the model using PyMC.
7. Run 4 MCMC chains with 5000 posterior samples each, using 2000 tuning steps.
8. Plot the trace using ArviZ.
9. Track the posterior distributions and highlight convergence.
10. Use the full power of your CPU: configure PyMC to run chains in parallel using all available cores .
11. Visualize the model graph using $pm.model\_to\_graphviz(model)$.

Tools Required

- Python 3.11+
- Packages: `pymc`, `arviz`, `matplotlib`, `numpy`
- Jupyter Notebook or any IDE (Colab, VS Code)

Extension Ideas

- Introduce outlier detection using Student-t likelihood.
- Animate the chain behavior or add a prior predictive check.