



Automatic Monte Carlo Methods for Bayesian Inference

Noah Grudowski

Applied Mathematics, Illinois Institute of Technology

Bayesian Statistics

- Model parameters are random
- **Prior** distribution reflects beliefs about the parameters
- Sampling yields a likelihood function
- **Posterior** density is the product of the prior density and the likelihood
- Estimates involve integrals, e.g.

$$\hat{\beta}_j = \frac{\int_{\mathbb{R}^{d+1}} b_j L(b) \pi(b) db}{\int_{\mathbb{R}^{d+1}} L(b) \pi(b) db} =: \frac{\mu_j}{\mu}, \quad j = 0, \dots, d$$

Automatic Quasi-Monte Carlo Cubature

- A commonly used estimator for Bayesian inference is
- $\hat{\beta}_j = \frac{\int_{\mathbb{R}^{d+1}} b_j L(b) \pi(b) db}{\int_{\mathbb{R}^{d+1}} L(b) \pi(b) db} =: \frac{\mu_j}{\mu}$, for $j = 0, 1, 2, \dots, d$ where $\hat{\beta}_j$ is a component of $\hat{\beta}$ (the unknown parameters), $L(b)$ is the likelihood function given the observed data, and $\pi(b)$ is some prior distribution.
- The problem with this estimator is that this ratio of integrals cannot be calculated analytically
- To approximate this ratio of integrals, we use the quasi-Monte Carlo cubature rule. This is given by the formula

$$\int_{[0,1]^{d+1}} f(x) dx \approx \frac{1}{n} \sum_{i=0}^{n-1} f(x_i)$$

Problem Formulation

- We explored the problem of using Bayesian inference for logistic regression, meaning that each t_i is an independent, identically distributed Bernoulli variable.

$$t_i \sim \text{Ber} \left(\frac{\exp(\beta_0 + \sum_{j=1}^d \beta_j s_{ij})}{1 + \exp(\beta_0 + \sum_{j=1}^d \beta_j s_{ij})} \right), \text{ for } i = 1, 2, \dots, M$$

- For simplicity, we will choose the standard normal distribution with respect to our parameters $\hat{\beta}_j$ to be our prior distribution, or $\pi(\beta) = \frac{\exp(-\frac{1}{2}\beta^T \beta)}{\sqrt{(2\pi)^{d+1}}}$

- Given the above t_i , we arrive at the following likelihood function: $L(\beta) = \prod_{i=1}^M \left(\frac{\exp(\beta_0 + \sum_{j=1}^d \beta_j s_{ij})}{1 + \exp(\beta_0 + \sum_{j=1}^d \beta_j s_{ij})} \right)^{t_i} \left(1 - \frac{\exp(\beta_0 + \sum_{j=1}^d \beta_j s_{ij})}{1 + \exp(\beta_0 + \sum_{j=1}^d \beta_j s_{ij})} \right)^{1-t_i}$

- Using this information, μ and $\hat{\mu}$ are defined as the following integrals: $\mu = \int_{\mathbb{R}^{d+1}} L(b) \pi(b) db$, $\mu_j = \int_{\mathbb{R}^{d+1}} b_j L(b) \pi(b) db$

Choice of Density

- Since we are not able to calculate the ratio $\frac{\mu_j}{\mu}$ analytically, we rewrite the formulas as such:

$$\mu = \int_{\mathbb{R}^{d+1}} L(b) \pi(b) db = \int_{\mathbb{R}^{d+1}} \frac{L(b) \pi(b)}{\rho(b)} \rho(b) db,$$

where $\rho(b)$ is some density. This allows us to estimate the values of the ratios of integrals by sampling from this $\rho(b)$.

- We have the following choices for $\rho(b)$:
 - 1) $\rho = \pi$
 - 2) $\rho = \rho_{\text{MLE}} = \text{Gaussian approximation to the likelihood}$
 - 3) $\rho \propto \pi \cdot \rho_{\text{MLE}}$

New MATLAB classes

- We have developed a **function** `BayesianFunction` in order to execute Bayesian inference for logistic regression using GAIL (Choi Et al, 2015) (quasi-)Monte Carlo methods.
- `BayesianFunction(...)` **automatically** estimates the unknown parameters $\hat{\beta}$ within some desired tolerance level.
- The function allows the user to input the number of x-Data values, the type of x-Data to be used, the number of iterations to be run, the absolute tolerance desired, and which of the three density choices they would like to use (can choose more than one).
- We specifically tested three different types of x-Data: uniformly distributed, uniform random, and normal random.

Evenly Spaced x-Data

Uniform Random x-Data

Normal Random x-Data

Discussion

- Add some
- Concluding
- Remarks here

References

-
-
-
-