

Forecasting Emerging Pandemic Risks Through Random-Walk Metropolis–Hastings Monte Carlo Simulations

November 22, 2025

Andressa Garcia Fator^{1,†}, Anacleto Silva de Souza^{1,2,†}, and Cristiane Rodrigues Guzzo^{1,*}

¹ Department of Microbiology, Institute of Biomedical Sciences, University of São Paulo, São Paulo, Brazil

² Faculty of Science and Engineering, University of Groningen, Nijenborgh 4, 9747 AG, Groningen, Netherlands

[†] These authors contributed equally.

* E-mail: crisguzzo@usp.br, crisguzzo@gmail.com

Abstract

This theoretical framework presents a Bayesian approach for pandemic risk forecasting using Random-Walk Metropolis–Hastings Monte Carlo simulations. The methodology combines logistic regression with adaptive Markov Chain Monte Carlo sampling to provide robust uncertainty quantification in emerging pathogen risk assessment.

1 Theoretical Development

1.1 Bayesian Logistic Regression Framework

The probabilistic foundation for pandemic risk classification begins with the likelihood function for binary outcomes. For each observation i with features $\mathbf{x}_i \in R^p$ and pandemic status $y_i \in \{0, 1\}$, we have:

$$p(y_i|\mathbf{x}_i, \boldsymbol{\beta}) = \text{Bernoulli}(y_i|\sigma(\mathbf{x}_i^T \boldsymbol{\beta})) \quad (1)$$

where the logistic sigmoid function $\sigma(z) = (1 + \exp(-z))^{-1}$ ensures probabilistic outputs between 0 and 1. The complete data likelihood for N independent observations is:

$$p(\mathbf{y}|\mathbf{X}, \boldsymbol{\beta}) = \prod_{i=1}^N [\sigma(\mathbf{x}_i^T \boldsymbol{\beta})]^{y_i} [1 - \sigma(\mathbf{x}_i^T \boldsymbol{\beta})]^{1-y_i} \quad (2)$$

1.2 Prior Specification and Posterior Derivation

We specify independent Gaussian priors for each parameter to regularize the solution:

$$p(\beta_j) = \mathcal{N}(\beta_j|\mu_0, \tau_0^{-1}), \quad j = 1, \dots, p \quad (3)$$

with $\mu_0 = 0$ and $\tau_0 = 1$ defining weakly informative priors. The joint prior distribution is:

$$p(\boldsymbol{\beta}) = \prod_{j=1}^p p(\beta_j) = \left(\frac{\tau_0}{2\pi}\right)^{p/2} \exp\left(-\frac{\tau_0}{2} \sum_{j=1}^p (\beta_j - \mu_0)^2\right) \quad (4)$$

Applying Bayes' theorem, the posterior distribution combines likelihood and prior:

$$p(\boldsymbol{\beta}|\mathbf{X}, \mathbf{y}) \propto p(\mathbf{y}|\mathbf{X}, \boldsymbol{\beta}) \cdot p(\boldsymbol{\beta}) \quad (5)$$

The log-posterior, more suitable for computational implementation, becomes:

$$\log p(\boldsymbol{\beta}|\mathbf{X}, \mathbf{y}) = \sum_{i=1}^N [y_i \mathbf{x}_i^T \boldsymbol{\beta} - \log(1 + \exp(\mathbf{x}_i^T \boldsymbol{\beta}))] \quad (6)$$

$$- \frac{\tau_0}{2} \sum_{j=1}^p (\beta_j - \mu_0)^2 + \text{constant} \quad (7)$$

1.3 Random-Walk Metropolis–Hastings Algorithm

The intractability of the posterior normalization constant necessitates Markov Chain Monte Carlo methods. The Random-Walk Metropolis–Hastings (RWMH) algorithm generates samples from the posterior through an iterative accept-reject mechanism.

Algorithm 1 Random-Walk Metropolis–Hastings for Bayesian Logistic Regression

```

1: Input: Data  $\mathbf{X}$ ,  $\mathbf{y}$ ; prior parameters  $\mu_0, \tau_0$ ; iterations  $T$ 
2: Output: Posterior samples  $\{\boldsymbol{\beta}^{(1)}, \dots, \boldsymbol{\beta}^{(T)}\}$ 
3: Initialize  $\boldsymbol{\beta}^{(0)} \leftarrow \mathbf{0}$ 
4: Initialize proposal variances  $\sigma_j^2$  for  $j = 1, \dots, p$ 
5: for  $t = 1$  to  $T$  do
6:   for  $j = 1$  to  $p$  do
7:     Propose new value:  $\beta_j^* \sim \mathcal{N}(\beta_j^{(t-1)}, \sigma_j^2)$ 
8:     Create proposal vector:  $\boldsymbol{\beta}^* = (\beta_1^{(t)}, \dots, \beta_j^*, \dots, \beta_p^{(t)})$ 
9:     Compute log-posterior ratio:
10:     $\Delta = \log p(\boldsymbol{\beta}^* | \mathbf{X}, \mathbf{y}) - \log p(\boldsymbol{\beta}^{(t-1)} | \mathbf{X}, \mathbf{y})$ 
11:    Compute acceptance probability:  $\alpha = \min(1, \exp(\Delta))$ 
12:    Sample  $u \sim \text{Uniform}(0, 1)$ 
13:    if  $u < \alpha$  then
14:       $\beta_j^{(t)} \leftarrow \beta_j^*$  ▷ Accept proposal
15:    else
16:       $\beta_j^{(t)} \leftarrow \beta_j^{(t-1)}$  ▷ Reject proposal
17:    end if
18:   end for
19: end for

```

1.4 Detailed Algorithmic Components

1.4.1 Log-Posterior Calculation

The key computational step involves evaluating the log-posterior ratio Δ . For numerical stability, we implement:

$$\log p(\boldsymbol{\beta}|\mathbf{X}, \mathbf{y}) = \sum_{i=1}^N [y_i z_i - \log(1 + \exp(z_i))] \quad (8)$$

$$- \frac{\tau_0}{2} \sum_{j=1}^p (\beta_j - \mu_0)^2 \quad (9)$$

where $z_i = \mathbf{x}_i^T \boldsymbol{\beta}$, with clipping $z_i \in [-250, 250]$ to prevent numerical overflow.

1.4.2 Adaptive Proposal Mechanism

The proposal variance σ_j^2 adapts dynamically based on empirical acceptance rates:

$$\sigma_j^{(t+1)} = \begin{cases} 0.9 \cdot \sigma_j^{(t)} & \text{if } \hat{\alpha}_j < 0.15 \\ 1.1 \cdot \sigma_j^{(t)} & \text{if } \hat{\alpha}_j > 0.35 \\ \sigma_j^{(t)} & \text{otherwise} \end{cases} \quad (10)$$

where $\hat{\alpha}_j$ is the acceptance rate for parameter j over a moving window of iterations.

1.4.3 Multiple Chain Implementation

We employ K independent chains with dispersed initializations:

$$\boldsymbol{\beta}_k^{(0)} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_k), \quad k = 1, \dots, K \quad (11)$$

with $\boldsymbol{\Sigma}_k$ chosen to ensure exploration of different posterior modes.

1.5 Convergence Diagnostics

The Gelman–Rubin statistic \hat{R} assesses convergence across multiple chains:

$$W = \frac{1}{K} \sum_{k=1}^K s_k^2 \quad (\text{within-chain variance}) \quad (12)$$

$$B = \frac{N}{K-1} \sum_{k=1}^K (\bar{\beta}_k - \bar{\beta})^2 \quad (\text{between-chain variance}) \quad (13)$$

$$\hat{V} = \frac{N-1}{N} W + \frac{1}{N} B \quad (\text{marginal posterior variance}) \quad (14)$$

$$\hat{R} = \sqrt{\frac{\hat{V}}{W}} \quad (15)$$

where $\hat{R} \approx 1$ indicates convergence, and values > 1.1 suggest inadequate mixing.

1.6 Posterior Inference

After discarding burn-in samples B , we obtain the posterior sample:

$$\mathcal{S} = \{\boldsymbol{\beta}^{(B+1)}, \dots, \boldsymbol{\beta}^{(T)}\} \quad (16)$$

Point estimates derive from posterior summaries:

$$\hat{\boldsymbol{\beta}}_{\text{mean}} = \frac{1}{|\mathcal{S}|} \sum_{\boldsymbol{\beta} \in \mathcal{S}} \boldsymbol{\beta} \quad (17)$$

$$\hat{\boldsymbol{\beta}}_{\text{median}} = \text{median}(\mathcal{S}) \quad (18)$$

$$\text{HPD}_{1-\alpha}(\beta_j) = [L, U] \quad \text{where} \quad \int_L^U p(\beta_j | \mathbf{X}, \mathbf{y}) d\beta_j = 1 - \alpha \quad (19)$$

The Highest Posterior Density (HPD) interval provides the shortest interval containing $(1 - \alpha) \times 100\%$ of posterior probability.

1.7 Prediction and Uncertainty Propagation

For new data \mathbf{x}^* , the posterior predictive distribution integrates over parameter uncertainty:

$$p(y^* = 1 | \mathbf{x}^*, \mathbf{X}, \mathbf{y}) = \int \sigma(\mathbf{x}^{*T} \boldsymbol{\beta}) p(\boldsymbol{\beta} | \mathbf{X}, \mathbf{y}) d\boldsymbol{\beta} \quad (20)$$

Approximated via Monte Carlo integration:

$$\hat{p}(y^* = 1 | \mathbf{x}^*, \mathbf{X}, \mathbf{y}) \approx \frac{1}{|\mathcal{S}|} \sum_{\boldsymbol{\beta} \in \mathcal{S}} \sigma(\mathbf{x}^{*T} \boldsymbol{\beta}) \quad (21)$$

This Bayesian approach naturally propagates parameter uncertainty into predictive probabilities, providing calibrated uncertainty estimates for pandemic risk assessments.