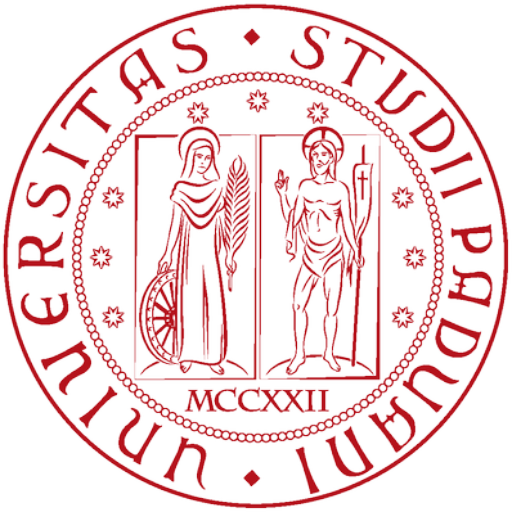
University of Padova

School of Engineering
Department of Information Engineering



Homework for

Estimation and Filtering - Markov Chain Monte Carlo

by

Maximillian Michael Ulrich Pries - 2144162

Submitted to

Prof. Dr. Gianluigi Pillonetto

Padova, July 24, 2025

1 Model and Motivation

This homework aimed to choose a dynamic system model and a related parameter estimation problem solve it using the Markov Chain Monte Carlo (MCMC) method. In our case, an estimation of the parameters of a SEIR compartment model for the spread of SARS-COV-2 in India has been studied. This particular case stands out due to the rapid spread of the disease and showcases which parameters play major roles in slowing the spread, thus saving lives. Compartment models describe disease spread by dividing the population in compartments and modeling the transitions between them. The SEIR model divides the whole population into susceptible (S), exposed (E), infected (I), and recovered (R), and only allows movement through them in this order. Neglecting birth and death dynamics, the model equations are:

$$\begin{aligned} \tfrac{dS}{dt} &= -\beta SI, \\ \tfrac{dE}{dt} &= \beta SI - \alpha E, \\ \tfrac{dI}{dt} &= \alpha E - \gamma I, \\ \tfrac{dR}{dt} &= \gamma I \end{aligned} \tag{1.1}$$

Where α is the incubation rate in 1/day, β is the effective contact rate in 1/day/person and γ is the recovery rate in 1/day. In real settings, all of these are positive. Typical measurements are the daily reported new infections, which can be described by αE with added noise. The daily acquisition of new data leaves enough time to resort to the MCMC method for parameter estimation, rather than relying on online estimation methods. Based on the study by [1], the true parameter values for the spread of COVID-19 in India in June 2020 were estimated as:

$$\alpha_{true} = 0.071 \qquad \beta_{true} = 0.476 \qquad \gamma_{true} = 0.286$$

From this, we define the true parameter vector as $\mathbf{x}_{true} = [\alpha_{true}, \quad \beta_{true}, \quad \gamma_{true}]^T$. For this study, the total population was set to $N_p = 1000$ individuals to facilitate computational efficiency. The initial conditions were established to reflect the early stages of an outbreak with:

$$S_0 = N_p - E_0 = 999 \qquad E_0 = N_p \cdot 0.1\% = 1 \qquad I_0 = 0 \qquad R_0 = 0$$

The dynamics of this model were simulated over a time span of 20 days. This range captures most of the transient and makes the assumption of neglecting the birth and death dynamics reasonable. From the simulations, a set of synthetic measurements was generated, and the MCMC method was applied to estimate the true parameter vector from the model, the measurements, and a prior distribution of the parameters.

2 Measurements Model, Likelihood and Posterior

Before implementing the Markov Chain, a set of noisy measurements and the posterior probability density function (up to a constant scaling factor) for the model parameters after seeing the data have to be derived. For the measurement model, the number of daily new infections with noise from a gamma distribution was chosen. The measurements y_t were generated by drawing samples of a gamma distribution with mean equal to the true value obtained from simulation and a chosen shape parameter k. The gamma distribution has more probability mass lower than the mean, but also allows higher measurements, e.g., due to previously undetected or unreported infections.

$$y_t \sim \text{Gamma}\left(k, \frac{\alpha E_t(x_{true})}{k}\right) \tag{2.1}$$

Where $E_t(x_{true})$ is taken from a simulation of the model with the true parameters. From this follows the likelihood of seeing the measurement of a specific day, and - assuming independent noise for each sample generation - the likelihood of seeing the whole dataset y:

$$\begin{aligned} p(y_t|X) &= \text{Gamma}\left(k, \frac{\alpha E_t(X)}{k}\right) && (2.2)\\ &= \frac{1}{\left(\frac{\alpha E_t(X)}{k}\right)^k \Gamma(k)}\, y_t^{k-1} \exp\left(-\frac{y_t k}{\alpha E_t(X)}\right) && (2.3)\\ &\propto y_t^{k-1} \exp\left(-\frac{y_t k}{\alpha E_t(X)}\right) && (2.4) \end{aligned}$$

$$\begin{aligned} p(y|X) &= \prod_t p(y_t|X) && (2.5)\\ &\propto \prod_t y_t^{k-1} \exp\left(-\frac{y_t k}{\alpha E_t(X)}\right) && (2.6) \end{aligned}$$

Where $X \in \mathbb{R}^3$ describes the model parameters as a random vector. To compute the posterior, a prior on the parameters is needed. The same was chosen for all parameters as a normal distribution with mean $\mu_i = 0.5$ and variance $\sigma_i^2 = 1$. To prevent physically infeasible parameters, the prior is set to zero for values ≤ 0. Omitting a rescaling, it follows:

$$p(X) \propto \begin{cases} 0 & \text{if } X_i \leq 0 \text{ for any } i = 1..3 \\ \exp\left(-\frac{1}{2}(X-\mu)^T(X-\mu)\right) & \text{otherwise} \end{cases} \tag{2.7}$$

With this, reminding that y is treated as known, it follows for the posterior:

$$p(X|y) \propto \begin{cases} 0 & \text{if } X_i \leq 0 \text{ for any } i = 1..3 \\ \prod_t \exp\left(-\frac{y_t k}{\alpha E_t(X)}\right) \cdot \exp\left(-\frac{1}{2}(X-\mu)^T(X-\mu)\right) & \text{otherwise} \end{cases} \tag{2.8}$$

3 MCMC Implementation

To obtain the posterior mean (the minimum variance estimate), a Markov Chain with the posterior as invariant density equal to the posterior is used to generate (correlated) samples from the posterior, and Monte Carlo integration is used to compute the mean. The following describes the principle of the used algorithm, practical implementations, and the obtained results.

3.1 MCMC using the Metropolis-Hastings Algorithm

The Markov Chain is implemented using the Metropolis-Hastings algorithm with Gaussian random walk proposals of variance Σ_c. The algorithm generates the samples consecutively, starting from an initial guess for the parameter vector x_0. Each step can be broken down into the following tasks:

- evaluate the posterior $p(X = x_i|y)$
- propose a new sample c using the random walk: $c_i \sim \mathcal{N}(x_i, \Sigma_c)$
- evaluate the posterior $p(X = c_i|y)$
- compute the acceptance probability a of the proposal: $a_i = \min\left(1, \frac{p(X=c_i|y)}{p(X=x_i|y)}\right)$
- generate a sample u from a uniform distribution $u_i \sim \mathcal{U}_{[0,1]}$
- if $u_i < a_i$, accept the sample and set $x_{i+1} = c_i$, else set $x_{i+1} = x_i$
- store the sample x_{i+1}

After the algorithm has performed a given number of N iterations, the first b samples are discarded as *burn in*, and the posterior mean can be computed using Monte Carlo integration:

$$\hat{x} = \frac{1}{N}\sum_{i=b}^{N} x_i \tag{3.1}$$

3.2 Practical Implementation

The actual implementation differs slightly from the method described above. Instead of evaluating the posterior, the logarithmic posterior is evaluated. This brings several numerical and computational advantages, such as preventing underflow due to multiplying large numbers of probabilities, avoiding calculating exponentials, and computing sums and differences instead of products and fractions. From Equation 2.8, the log posterior can be evaluated up to a constant offset C:

$$\log p(X|y) = \begin{cases} -\infty & \text{if } X_i \leq 0 \text{ for any } i = 1..3 \\ -\sum\limits_t \frac{y_t k}{\alpha E_t(X)} - \frac{1}{2}(X-\mu)^T(X-\mu) + C & \text{otherwise} \end{cases} \tag{3.2}$$

Due to the monotony of the logarithm, inequality relations remain. The acceptance rule for new proposals thus becomes:

$$\log u_i < \min\left(0, \log p(X = c|y) - \log(X = x|y)\right) \qquad u_i \sim \mathcal{U}_{[0,1]} \tag{3.3}$$

The offset can thus be disregarded in the calculation. To prevent the problem of evaluating the logarithm at values ≤ 0, the case of proposals having a non-positive part gets filtered out by an if-statement before the evaluation of the log posterior.
Considering the chosen non-linear dynamic model, the posterior can not be evaluated in closed form due to the $E_t(X)$ term. Therefore, the model has to be simulated with the current parameters and the current proposal before each evaluation of the log posterior. To avoid repeatedly simulating the model with the same parameters, the simulation is performed with the initial parameters before initiating the Markov Chain process. In the loop, the model is simulated with the proposed parameter vector. Depending on the acceptance, the value of the posterior resulting from the proposal or the previous parameters is used for the acceptance check. This way, only one simulation and posterior evaluation have to be performed at each chain iteration.
At each iteration, an acceptance counter is updated to keep track of the acceptance rate. The variance of the Gaussian random walk is tuned to achieve an acceptance rate of roughly $30 - 40$ %.
All fixed parameters whose value was not mentioned above were implemented to be a user choice.

4 Results

In the following, the obtained results are presented and discussed. For reproducibility, the provided results were obtained with the MATLAB random number generator initialized to the default (`rng("default")`) (Note: this was not used during tuning). The user choice parameters were chosen as follows:

$$\begin{aligned}
\text{initial guess: } & x_0 = \begin{bmatrix}\alpha_0 = 0.2 & \beta_0 = 0.4 & \gamma_0 = 0.4\end{bmatrix}^T \\
\text{random walk covariance: } & \Sigma_c = \text{diag}\{10^{-4},\ 3 \cdot 10^{-3},\ 5 \cdot 10^{-4}\} \\
\text{chain length: } & N = 10000 \\
\text{burn in: } & b = 2500 \\
\text{shape factor for noise: } & k = 20
\end{aligned}$$

The measurements generated from the model and the true new daily infections are depicted in Figure 4.1 and the resulting posterior for the parameters in Figure 4.2. It can be seen, that there is practically no correlation between γ and the other parameters, and that there is a correlation for small values of α and β.

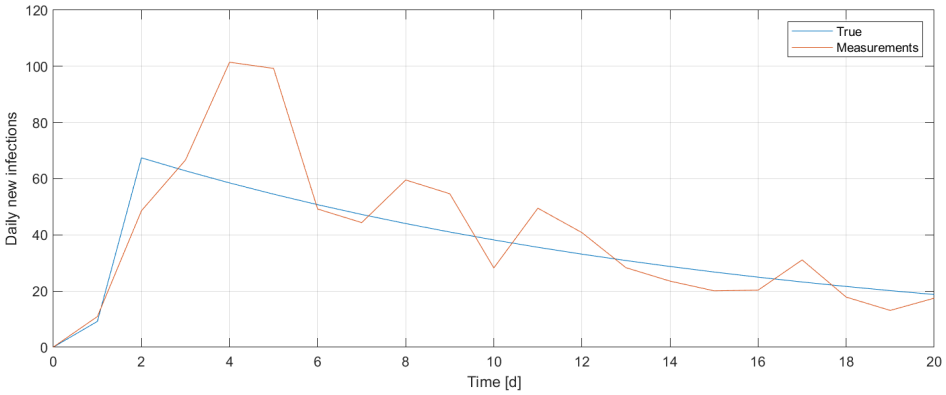


Figure 4.1: Synthetic measurements of new daily infections.

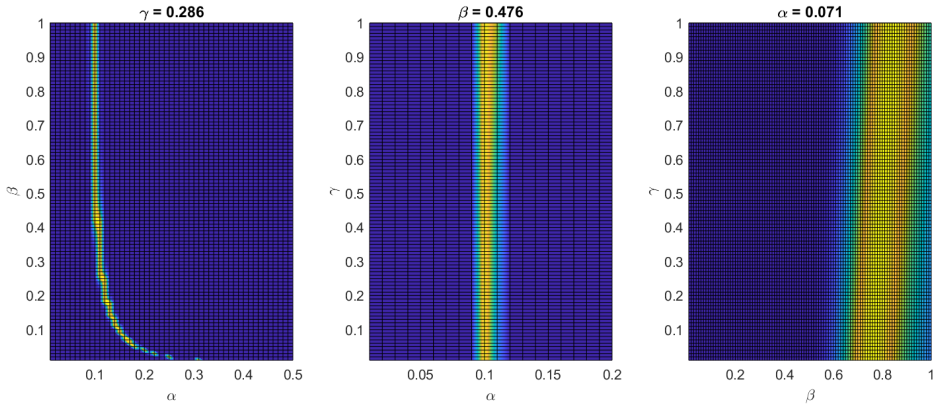


Figure 4.2: True posterior resulting from the synthetic measurements (one parameter fixed).

Figure 4.3 shows the generated samples from the Markov Chain. The discarded samples of the burn-in phase are depicted in gray, the starting point in red, and the true values in green. Even though the true value does not lie in the point cloud, it can be seen, that most samples lie in the region where most probability mass of the posterior is concentrated. The corresponding traceplots are depicted in Figure 4.4 and show good convergence of α and β, while γ diverges, as expected from the posterior. The final estimated parameter vector is:

$$\hat{\boldsymbol{x}} = [0.105 \quad 0.443 \quad 0.634]^T \tag{4.1}$$

With relative deviations from the true states of:

$$\Delta\boldsymbol{x} = [48.4\% \quad 6.97\% \quad 121.6\%]^T \tag{4.2}$$

The predicted dynamics with these parameters versus the true dynamics can be seen in Figure 4.5.

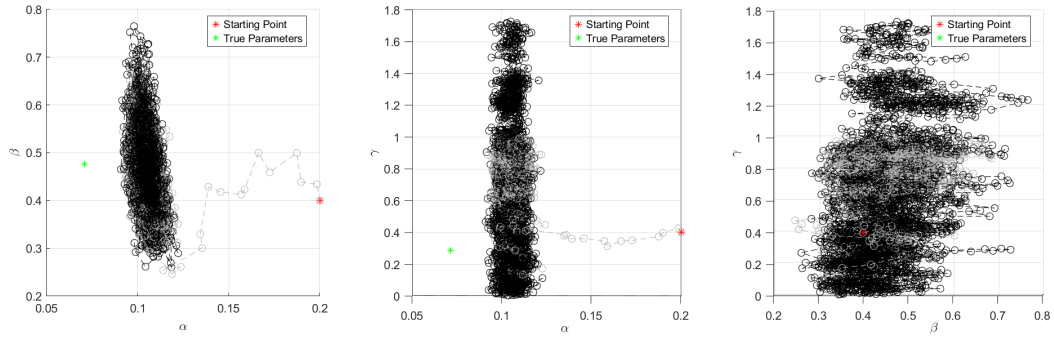


Figure 4.3: Samples generated by the Markov Chain from different views.

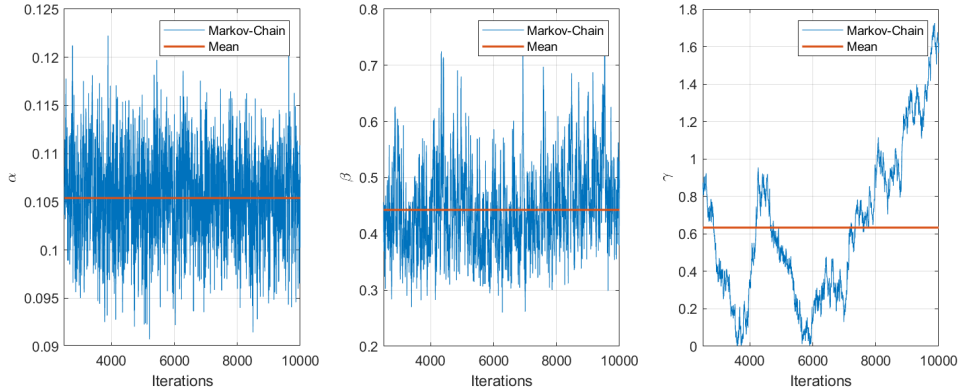


Figure 4.4: Traceplots of the parameters from the Markov Chain.

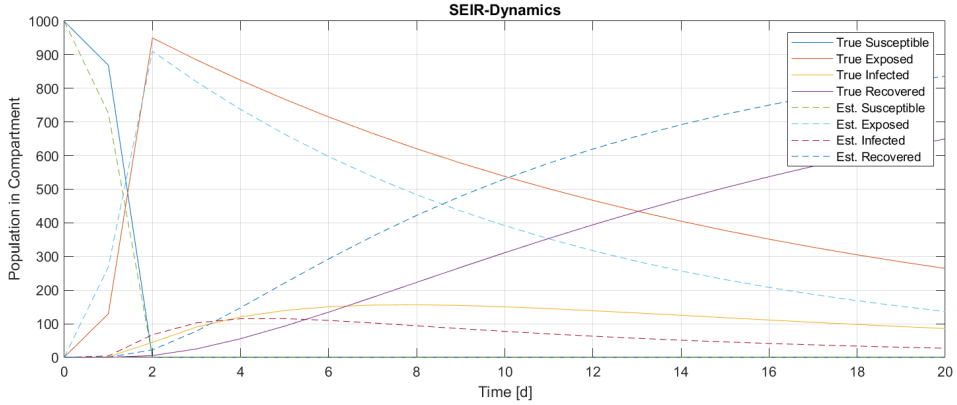


Figure 4.5: Predicted dynamics versus true dynamics.

References

[1] S. Paul, A. Mahata, U. Ghosh, *et al.*, "Study of seir epidemic model and scenario analysis of covid-19 pandemic", *Ecological Genetics and Genomics*, vol. 19, 2021. DOI: https://doi.org/10.1016/j.egg.2021.100087.