4M24: High-Dimensional MCMC for Gaussian Processes

Author: 5641G

Introduction

This report investigates Markov Chain Monte Carlo (MCMC) methods for high-dimensional latent variable models, focusing on Gaussian Processes (GPs). The study compares the Gaussian Random Walk Metropolis-Hastings (GRW-MH) and the preconditioned Crank-Nicolson (pCN) algorithms for sampling latent variables in a 2D spatial domain.

Part I - Simulation

A Samples from GP

We define a 2D grid of size $D \times D$ on a 2D domain $\mathbf{x} \in [0,1] \times [0,1]$. Latent variables are defined as $\mathbf{u} \sim \mathcal{N}(0, \mathbf{K})$, where $(\mathbf{K})_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$ is a Gaussian covariance function parametrised by a length-scale ℓ.

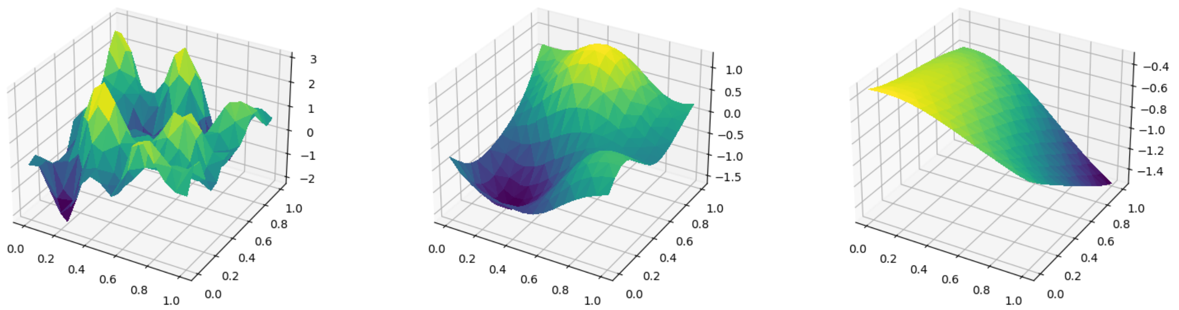
$$k(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2l^2}\right). \tag{1}$$

The data $\mathbf{v} \in \mathbb{R}^M$ is a noisy observation of a subset of the latent variables, given by:

$$\mathbf{v} = \mathbf{G}\mathbf{u} + \boldsymbol{\epsilon}, \quad \mathbf{G} \in \{0,1\}^{M \times N}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(0, \mathbf{I}), \tag{2}$$

where $\mathbf{G}$ is a sparse selection matrix ensuring that each latent variable component is either sampled once or not at all. M is the total number of observed data points and $M = N/subsample\_factor$.

Using $D = 16$, a subsample factor of 4, and three different length-scales $\ell = 0.1, 0.3, 0.9$, we generated the latent field $\mathbf{u}$, as shown in Figure 1.



(a) $\ell = 0.1$ (high-frequency) (b) $\ell = 0.3$ (moderate smooth) (c) $\ell = 0.9$ (very smooth)

Figure 1: Effect of length-scale ℓ on the latent variable surface.

The length-scale ℓ controls the smoothness of the inferred Gaussian process surface. Larger values of ℓ increase the covariance between points, leading to a smoother latent function. When $\ell = 0.1$, the surface exhibits high-frequency variations, whereas $\ell = 0.3$ results in a moderately smoother trend, as shown in Figures 1a and 1b, respectively. In contrast, a much larger length-scale of $\ell = 0.9$ leads to an even smoother and more slowly varying surface (Figure 1c).

To further illustrate the relationship between the prior sample and observed data, Figure 2 overlays the observed data points on the sampled latent field for $\ell = 0.3$. The red crosses indicate the observed noisy values $\mathbf{v}$, while the background represents the inferred latent function.

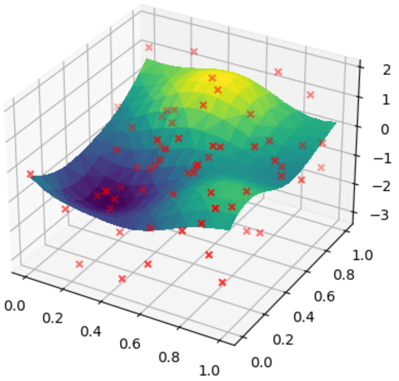


Figure 2: Prior sample $\mathbf{u}$ and observed data overlay for $\ell = 0.3$.

B.1 Derivation of Log-Prior and Log-Likelihood

Log-Prior The latent variables are drawn from a zero-mean Gaussian prior:

$$\pi^0(\mathbf{u}) = \mathcal{N}(\mathbf{0}, \mathbf{K}). \tag{3}$$

The corresponding log-prior is:

$$\ln \pi^0(\mathbf{u}) = -\frac{1}{2}\mathbf{u}^\top \mathbf{K}^{-1}\mathbf{u} - \frac{1}{2}\ln|\mathbf{K}| - \frac{N}{2}\ln(2\pi) \tag{4}$$

Log-Likelihood Observations follow the Gaussian noise model:

$$\mathbf{v} = \mathbf{G}\mathbf{u} + \boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}). \tag{5}$$

The likelihood function is given by:

$$p(\mathbf{v}|\mathbf{u}) = \mathcal{N}(\mathbf{G}\mathbf{u}, \mathbf{I}). \tag{6}$$

Taking the logarithm:

$$\ln p(\mathbf{v}|\mathbf{u}) = -\frac{1}{2}(\mathbf{v} - \mathbf{G}\mathbf{u})^\top(\mathbf{v} - \mathbf{G}\mathbf{u}) - \frac{M}{2}\ln(2\pi) \tag{7}$$

B.2 Gaussian Random Walk Metropolis-Hastings (GRW-MH)

The Gaussian Random Walk Metropolis-Hastings (GRW-MH) algorithm and the Preconditioned Crank-Nicolson (pCN) algorithm are two Markov Chain Monte Carlo (MCMC) methods used for sampling from the posterior distribution $p(\mathbf{u}|\mathbf{v})$. Given observations $\mathbf{v}$, both methods generate a sequence of samples $\{\mathbf{u}^{(i)}\}$ from the posterior. Note that every $\mathbf{u}^{(i)}$ ia a $N = D^2$ dimentional vector.

The difference between the two methods lies in the proposal mechanism: GRW-MH is based on a Gaussian random walk, while pCN avoids random walk behavior by maintaining prior structure. Here we present **GRW-MH** method first.

Proposal Distribution A new candidate $\mathbf{u}^*$ is proposed using a Gaussian step:

$$\mathbf{u}^* = \mathbf{u}^{(i-1)} + \beta\boldsymbol{\xi}, \quad \boldsymbol{\xi} \sim \mathcal{N}(\mathbf{0}, \mathbf{K}). \tag{8}$$

where $\mathbf{K}$ is the prior covariance matrix and β controls the step size.

Acceptance Criterion In general Metropolis-Hastings, a candidate sample $\mathbf{u}^*$ is accepted with probability:

$$\alpha = \min\left(1, \frac{p(\mathbf{u}^*|\mathbf{v})}{p(\mathbf{u}^{(i-1)}|\mathbf{v})} \frac{q(\mathbf{u}^{(i-1)}|\mathbf{u}^*)}{q(\mathbf{u}^*|\mathbf{u}^{(i-1)})}\right), \tag{9}$$

where $q(\cdot|\cdot)$ is the proposal distribution. For symmetric proposals, such as GRW-MH proposals, we have $\frac{q(\mathbf{u}^{(i-1)}|\mathbf{u}^*)}{q(\mathbf{u}^*|\mathbf{u}^{(i-1)})} = 1$. Thus, the acceptance probability simplifies to:

$$\alpha = \min\left(1, \frac{p(\mathbf{u}^*|\mathbf{v})}{p(\mathbf{u}^{(i-1)}|\mathbf{v})}\right) = \min\left(1, \frac{p(\mathbf{v}|\mathbf{u}^*)}{p(\mathbf{v}|\mathbf{u}^{(i-1)})} \frac{\pi^0(\mathbf{u}^*)}{\pi^0(\mathbf{u}^{(i-1)})}\right) \tag{10}$$

B.3 Preconditioned Crank-Nicolson (pCN)

Proposal Distribution The pCN algorithm generates a new candidate $\mathbf{u}^*$ using the following non-symmetric proposal mechanism:

$$\mathbf{u}^* = \sqrt{1-\beta^2}\mathbf{u}^{(i-1)} + \beta\boldsymbol{\xi}, \quad \boldsymbol{\xi} \sim \mathcal{N}(\mathbf{0}, \mathbf{K}). \tag{11}$$

where $\mathbf{K}$ is the prior covariance matrix and β is a step-size parameter that controls how much of the previous sample $\mathbf{u}^{(i-1)}$ is retained.

Acceptance Criterion Unlike the GRW-MH algorithm, the pCN method employs an asymmetric proposal distribution, meaning that we cannot assume $q(\mathbf{u}^{(i-1)}|\mathbf{u}^*) = q(\mathbf{u}^*|\mathbf{u}^{(i-1)})$. Specifically, the forward and reverse proposal densities are given by:

$$\begin{aligned} q(\mathbf{u}^*|\mathbf{u}^{(i-1)}) &= \mathcal{N}\left(\mathbf{u}^* \middle| \sqrt{1-\beta^2}\mathbf{u}^{(i-1)}, \beta^2\mathbf{K}\right) & (12)\\ &\propto \exp\left(-\frac{1}{2}\left(\frac{\mathbf{u}^* - \sqrt{1-\beta^2}\mathbf{u}^{(i-1)}}{\beta}\right)^\top \mathbf{K}^{-1}\left(\frac{\mathbf{u}^* - \sqrt{1-\beta^2}\mathbf{u}^{(i-1)}}{\beta}\right)\right), & (13)\end{aligned}$$

$$\begin{aligned} q(\mathbf{u}^{(i-1)}|\mathbf{u}^*) &= \mathcal{N}\left(\mathbf{u}^{(i-1)} \middle| \sqrt{1-\beta^2}\mathbf{u}^*, \beta^2\mathbf{K}\right) & (14)\\ &\propto \exp\left(-\frac{1}{2}\left(\frac{\mathbf{u}^{(i-1)} - \sqrt{1-\beta^2}\mathbf{u}^*}{\beta}\right)^\top \mathbf{K}^{-1}\left(\frac{\mathbf{u}^{(i-1)} - \sqrt{1-\beta^2}\mathbf{u}^*}{\beta}\right)\right). & (15)\end{aligned}$$

Thus, their ratio simplifies as follows:

$$\frac{q(\mathbf{u}^{(i-1)}|\mathbf{u}^*)}{q(\mathbf{u}^*|\mathbf{u}^{(i-1)})} = \exp\left(-\frac{1}{2}\mathbf{u}^{(i-1)\top}\mathbf{K}^{-1}\mathbf{u}^{(i-1)}\right) \Big/ \exp\left(-\frac{1}{2}\mathbf{u}^{*\top}\mathbf{K}^{-1}\mathbf{u}^*\right) = \frac{\pi^0(\mathbf{u}^{(i-1)})}{\pi^0(\mathbf{u}^*)}. \tag{16}$$

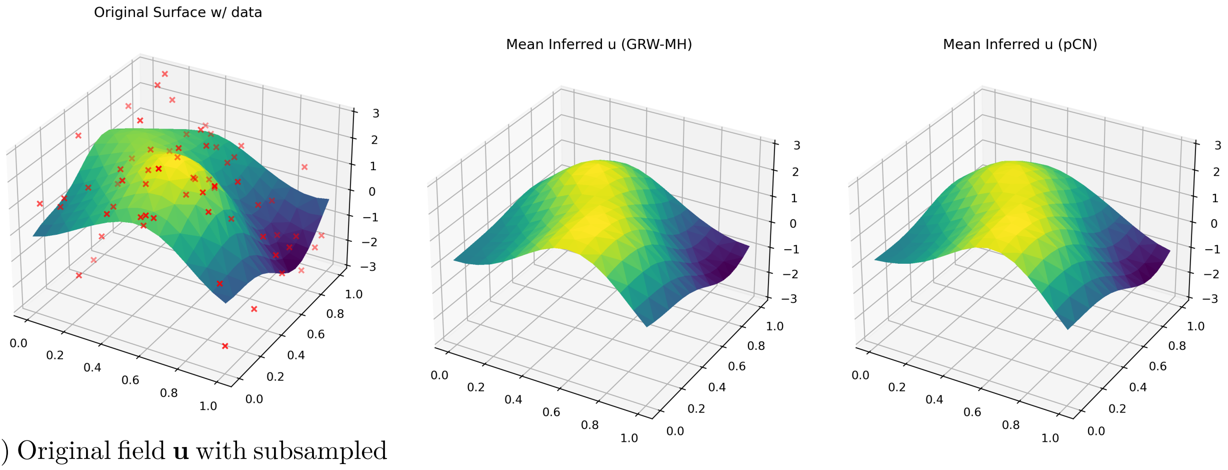
Substituting Equation 16 into the standard Metropolis-Hastings acceptance criterion (Equation 9),we obtain a simplified acceptance criterion that **depends solely on the likelihood**:

$$\alpha = \min\left(1, \frac{p(\mathbf{v}|\mathbf{u}^*)}{p(\mathbf{v}|\mathbf{u}^{(i-1)})}\right). \tag{17}$$

B.4 Inference Using MCMC Methods

In this experiment, we infer $\mathbf{u}$ from observations $\mathbf{v}$ using GRW-MH and pCN. We generate posterior samples from $p(\mathbf{u}|\mathbf{v})$ with parameters $n = 10,000$ and $\beta = 0.2$, then compute and analyze the mean inferred $\hat{\mathbf{u}}$ and the absolute error fields $|\hat{\mathbf{u}} - \mathbf{u}|$.

The original latent field $\mathbf{u}$ and subsampled observations $\mathbf{v}$ are shown in Figure 3a. The mean inferred $\hat{\mathbf{u}}$ from the posterior samples are shown in Figure 3b and Figure 3c. The absolute error fields are plotted in Fig. 4a and Fig. 4b.

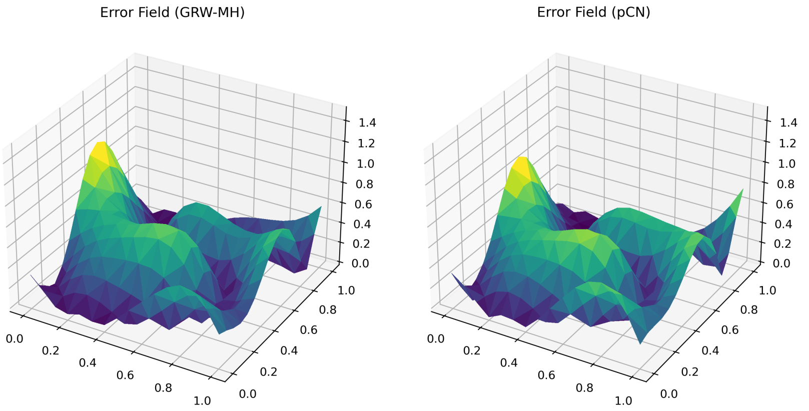


(a) Original field $\mathbf{u}$ with subsampled observations $\mathbf{v}$.

(b) Inferred $\hat{\mathbf{u}}$ using GRW-MH.

(c) Inferred $\hat{\mathbf{u}}$ using pCN.

Figure 3: Comparison of the original field $\mathbf{u}$ and mean inferred $\hat{\mathbf{u}}$.



(a) GRW-MH.

(b) pCN.

Figure 4: Comparison of absolute error fields for different algorithms.

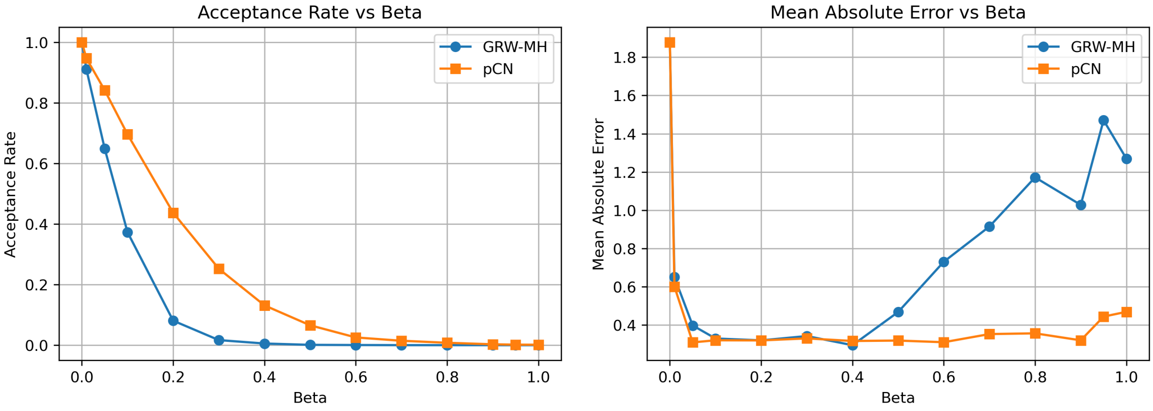
The mean absolute error for GRW-MH is 0.348, while for pCN it is 0.329. In terms of inference results and error field distributions (Figure 3 and 4, both algorithms exhibit similar performance, with no significant differences in the shapes, patterns, or magnitudes of the estimated fields and error distributions. This similarity arises because the problem under consideration has relatively low dimensionality. Given sufficient iterations, both methods successfully converge to the target distribution and achieve comparable results.

The primary advantage of pCN over GRW-MH becomes apparent in convergence speed in high dimensions, significantly affected by acceptance rate. This is illustrated in Table 1, which compares the acceptance rates for both methods at $D = 4$ and $D = 16$. As dimensionality increases, the acceptance rate of GRW-MH declines rapidly, approaching zero. This decline occurs because the random walk proposals increasingly deviate from high-probability regions, reducing their likelihood of acceptance. This leads to a dramatic increase in the number of iterations required for meaningful sampling, making the method computationally impractical. In contrast, pCN maintains a relatively stable acceptance rate, allowing for more efficient exploration of the parameter space. his property makes pCN a more robust choice for high-dimensional Bayesian sampling tasks.

	$D = 4$	$D = 16$
GRW-MH	66%	8%
pCN	84%	44%

Table 1: Acceptance Rate for $D = 4$ and $D = 16$.

Effect of β on Acceptance Rate Figures 5a and 5b shows how varying β from 0 to 1 affects the acceptance rates and mean errors for $D = 16$. The acceptance rate for both GRW-MH and pCN follows a similar trend, decreasing from nearly 100% to 0% as β increases. However, the declining rate of GRW-MH is much faster than pCN. In the case of GRW-MH, the acceptance rate drops to nearly zero at $\beta \approx 0.4$, whereas for pCN, it remains higher for a broader range of β values, only reaching zero at $\beta \approx 0.7$. This discrepancy arises from the distinct mechanisms by which new samples are proposed in each algorithm.



(a) Acceptance Rate

(b) Mean Absolute Error

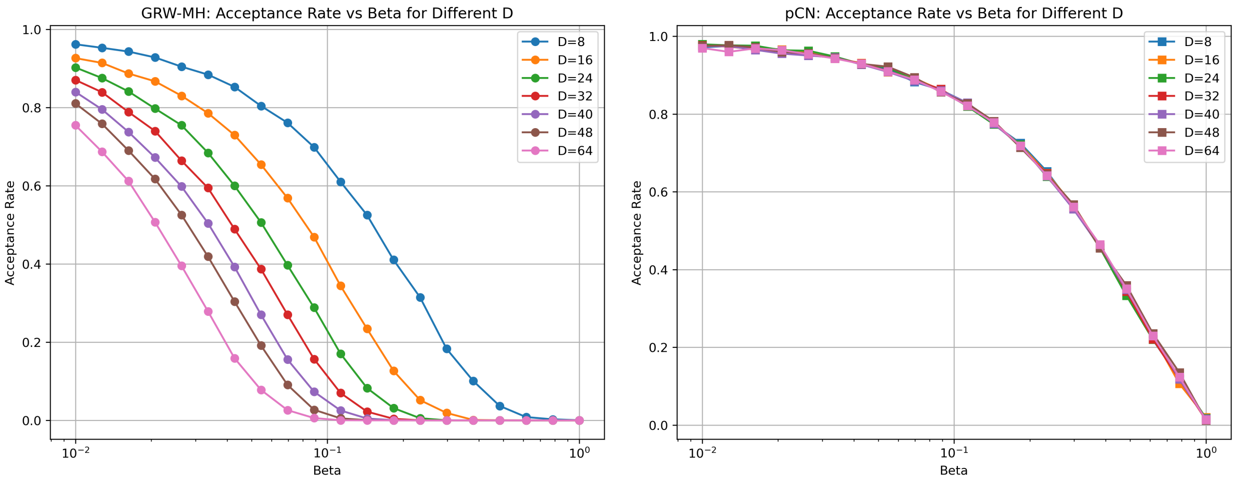
Figure 5: Acceptance Rate and Mean Absolute Error vs. β for GRW-MH and pCN.

When β is close to 1, the acceptance probability for the basic GRW algorithm approaches zero, causing the algorithm to move extremely slowly through the parameter space. The primary reason for this behavior lies in the prior term appearing in the acceptance probability (Equation 10). Specifically, the ratio $\frac{\pi^0(\mathbf{u}^*)}{\pi^0(\mathbf{u}^{(i-1)})}$ tends to be very small for large β, as the new proposal $\mathbf{u}^*$ deviates significantly from the current state $\mathbf{u}^{(i-1)}$. This results in an overall acceptance rate that approaches zero, severely limiting the efficiency of the algorithm.

In contrast, the pCN algorithm effectively eliminates the influence of the prior term in the acceptance probability (Equation 17). Instead of relying on the ratio of posterior densities, the acceptance probability in pCN depends only on the ratio of likelihoods. As a result, the acceptance probability does not tend to zero, even for large β. This fundamental difference allows pCN to maintain stable performance across different dimensions. Theoretically, its efficiency does not degrade with increasing dimensionality, enabling effective exploration of the parameter space even in very high or infinite-dimensional settings.

B.5 Extension Quesiton

To analyze the effect of mesh refinement, we perform MCMC sampling using both pCN and GRW-MH methods on a fixed subsampled data with refined mesh. Figure 6 illustrate the acceptance rate as a function of β for both methods



(a) GRW-MH

(b) pCN

Figure 6: Comparison of acceptance rates vs β for GRW-MH and pCN as mesh is refined.

As the mesh is refined, the acceptance probability curves of GRW-MH shift to the left (see Figure 6a), indicating that smaller values of β are needed to maintain a consistent acceptance probability. With fixed β, the acceptance rate diminishes toward zero as the mesh is refined, due to the proposal step deviating significantly from high-probability regions. In contrast, for pCN (see Figure 6b), the acceptance probability curves remain stable across different levels of mesh refinement. This suggests that, as the resolution of the random field model increases, a fixed β suffices to achieve the same acceptance probability, making pCN substantially faster than GRW-MH in high-resolution settings.

C.1 Log-Probit-Likelihood Derivation

Now the model is extended to work on a probit classification problem.

$$v_i = [\mathbf{G}\mathbf{u}]_i + \epsilon_i, \quad \epsilon_i \sim \mathcal{N}(0, 1). \tag{18}$$

The observed classification variable t_i is defined as:

$$t_i = \begin{cases} 0, & \text{if } v_i \leq 0 \\ 1, & \text{otherwise} \end{cases} \tag{19}$$

The probability of t_i given $\mathbf{u}$ follows:

$$p(t_i = 1|\mathbf{u}) = \Phi([\mathbf{G}\mathbf{u}]_i), \quad p(t_i = 0|\mathbf{u}) = 1 - \Phi([\mathbf{G}\mathbf{u}]_i). \tag{20}$$

Assuming independence across observations, the full likelihood is:

$$p(\mathbf{t}|\mathbf{u}) = \prod_{i=1}^{M} \Phi([\mathbf{G}\mathbf{u}]_i)^{t_i} (1 - \Phi([\mathbf{G}\mathbf{u}]_i))^{1-t_i}. \tag{21}$$

Taking the logarithm,

$$\log p(\mathbf{t}|\mathbf{u}) = \sum_{i=1}^{M} \left(t_i \log \Phi([\mathbf{G}\mathbf{u}]_i) + (1 - t_i) \log(1 - \Phi([\mathbf{G}\mathbf{u}]_i))\right). \tag{22}$$

C.2 Predictive Distribution Approximation

We aim to compute the predictive probability of a new class assignment t_i^* given the observed class data $\mathbf{t}$, which is given by:

$$p(t_i^* = 1|\mathbf{t}) = \int p(t_i^* = 1|\mathbf{u})p(\mathbf{u}|\mathbf{t})d\mathbf{u}. \tag{23}$$

We can approximate the integral over the posterior distribution $p(\mathbf{u}|\mathbf{t})$ using Monte Carlo sampling, where $\{\mathbf{u}^{(k)}\}_{k=1}^{K}$ are samples drawn from the posterior $p(\mathbf{u}|\mathbf{t})$ and K is the total number of samples.

$$\int p(t_i^* = 1|\mathbf{u})p(\mathbf{u}|\mathbf{t})d\mathbf{u} \approx \frac{1}{K}\sum_{k=1}^{K} p(t_i^* = 1|\mathbf{u}^{(k)}) \tag{24}$$

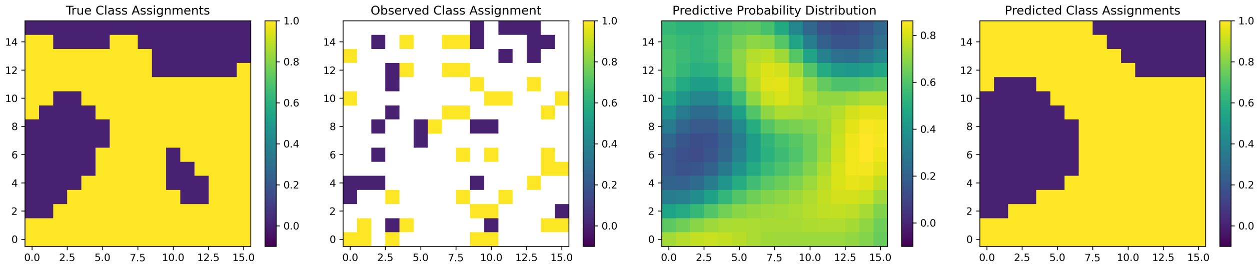
Given the true underlying $\mathbf{u}$, the probability of class assignment is:

$$p(t_i^* = 1|\mathbf{u}) = \Phi(u_i), \tag{25}$$

Thus, our Monte Carlo estimate of the predictive probability is:

$$p(t_i^* = 1|\mathbf{t}) = \int \Phi(u_i)p(\mathbf{u}|\mathbf{t})d\mathbf{u} \approx \frac{1}{K}\sum_{k=1}^{K} \Phi(u_i^{(k)}), \tag{26}$$

Figures 7a and 7b display the true class assignments and the subsampled observation data, respectively. The predictive distribution, approximated using Equation 26, is visualized in Figure 7c. This distribution generally aligns with the true class assignments, although some regions exhibit inaccuracies due to noise and the effects of subsampling. By thresholding the predictive distribution at 0.5, the predicted class assignments are obtained, as shown in Figure 7d. The mean prediction error in this example is 0.164, indicating that 16.4% of the points are misclassified. The predicted classes effectively capture the overall structure of the true class assignments, although some edge regions and small clusters of points are inaccurately represented.



(a) True assignments $\mathbf{t}_{\text{true}}$ (b) Subsampled $\mathbf{t}_{\text{data}}$ (c) Predictive probability (d) Thresholded predicted

Figure 7: Classification results from observed data to predicted classes.

D.1 Grid Search for Length-Scale Optimization

A grid search was performed to estimate the true length-scale ℓ by minimizing the classification error. Figure 8 illustrates the mean prediction error as a function of the tested length-scale ℓ, averaged over three independent experiments. The dashed vertical line indicates the true length-scale ($\ell_{\text{true}} = 0.3$).

The inferred length-scale (0.45) is reasonably close to the true value (0.3). An excessively large ℓ would fail to capture the fine-scale variations in the data, leading to oversmoothing, while an excessively small ℓ would result in poor interpolation in regions lacking data. Thus, the estimated length-scale represents a reasonable trade-off between capturing local variability and ensuring smooth generalization.

Since the dataset is generated using the same type of model, its complexity is well-matched to the inference model. Theoretically, this suggests that the exact length-scale should be identifiable, provided sufficient data and an appropriate estimation procedure. The primary source of inference deviation is the stochastic variability in the model.

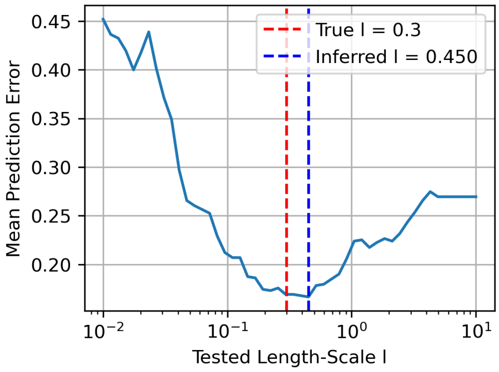


Figure 8: Mean prediction error as a function of the tested ℓ, averaged over three experiments. The dashed line indicates the true length-scale value ($\ell_{\text{true}} = 0.3$).

Part II - Spatial Data

In this section, we focus on modeling spatially distributed count data, specifically predicting the number of bike thefts in Lewisham borough using a Poisson likelihood-based approach. Our objective is to infer the underlying latent field $\mathbf{u}$ that governs the observed counts using pCN method and evaluate the model's predictive performance.

E.1 Derivation of Poisson Log-Likelihood

We begin with the Poisson likelihood model. The observed count data $\mathbf{c} = [c_1, c_2, \ldots, c_M]$ follows a Poisson distribution parameterized by $\boldsymbol{\theta}$, where:

$$\theta_i = \exp([\mathbf{G}\mathbf{u}]_i). \tag{27}$$

Thus, the likelihood function for the observed data $\mathbf{c}$ given $\mathbf{u}$ is:

$$p(\mathbf{c}|\mathbf{u}) = \prod_{i=1}^{M} f(c_i|\theta_i), \tag{28}$$

where each term follows a Poisson probability mass function:

$$f(c_i|\theta_i) = \frac{e^{-\theta_i}\theta_i^{c_i}}{c_i!}. \tag{29}$$

Taking the logarithm of likelihood $p(\mathbf{c}|\mathbf{u})$:

$$\begin{aligned} \log p(\mathbf{c}|\mathbf{u}) &= \sum_{i=1}^{M} \log f(c_i|\theta_i) = \sum_{i=1}^{M} \left[\log\left(\frac{e^{-\theta_i}\theta_i^{c_i}}{c_i!}\right)\right] & (30) \\ &= \sum_{i=1}^{M} \left[-\theta_i + c_i \log\theta_i - \log c_i!\right]. & (31) \end{aligned}$$

Substituting $\theta_i = \exp([\mathbf{G}\mathbf{u}]_i)$:

$$\log p(\mathbf{c}|\mathbf{u}) = \sum_{i=1}^{M} \left[c_i[\mathbf{G}\mathbf{u}]_i - \exp([\mathbf{G}\mathbf{u}]_i) - \log c_i!\right]. \tag{32}$$

This represents the log-likelihood of the observed count data $\mathbf{c}$ given the latent field $\mathbf{u}$.

F.1 Counts Approximation using MC

We now show how to use MC estimate to approx expectation of the predictive distribution of bike theft counts at a test location $\mathbf{x}^*$. The inferred expected counts are given by:

$$\mathbb{E}_{p(c^*|\mathbf{c})}[c^*] = \sum_{k=0}^{\infty} k p(c^* = k|\mathbf{c}) \tag{33}$$

The predictive distribution in Equation 33 can be approximated by MC estimate. Given samples $\{u^{*(t)}\}_{t=1}^{T}$ from posterior distribution $p(\mathbf{u}|\mathbf{c})$,

$$p(c^* = k|\mathbf{c}) = \int p(c^* = k|\mathbf{u})p(\mathbf{u}|\mathbf{c})d\mathbf{u} \approx \frac{1}{T}\sum_{t=1}^{T} p(c^* = k|\mathbf{u}^{(t)}) = \frac{1}{T}\sum_{t=1}^{T} f\left(c^*|\theta^{*(t)}\right) \tag{34}$$

Since $f(c^*|\theta^*)$ is Poisson distributed:

$$\mathbb{E}_{f(c^*|\theta^*)}[c^*] = \sum_{k=0}^{\infty} k \times f(c^* = k|\theta^*) = \theta^* \tag{35}$$

Thus, using the expectation of a Poisson-distributed random variable, we approximate:

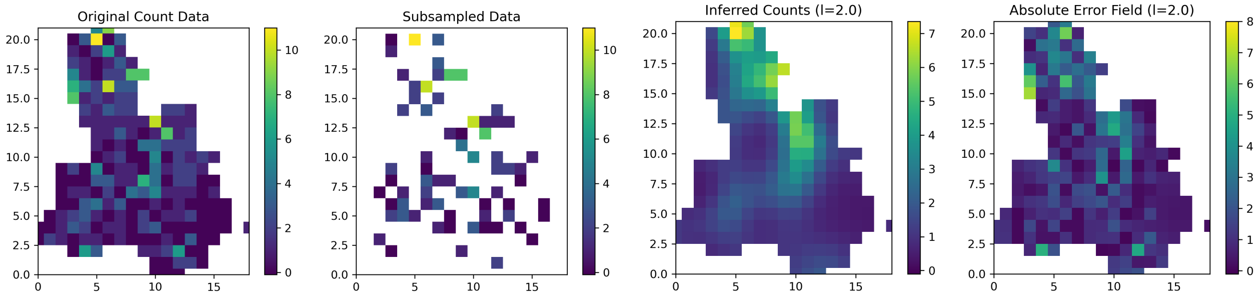
$$\mathbb{E}_{p(c^*|\mathbf{c})}[c^*] = \sum_{k=0}^{\infty} k p(c^* = k|\mathbf{c}) \approx \sum_{k=0}^{\infty} k \times \frac{1}{T}\sum_{t=1}^{T} f\left(c^* = k|\theta^{*(t)}\right) \tag{36}$$

$$= \frac{1}{T}\sum_{t=1}^{T}\sum_{k=0}^{\infty} k \times f\left(c^* = k|\theta^{*(t)}\right) \tag{37}$$

$$= \frac{1}{T}\sum_{t=1}^{T} \theta^{*(t)} \;=\; \frac{1}{T}\sum_{t=1}^{T} e^{u^{*(t)}} \tag{38}$$

F.2 Inference of Expected Counts

In this analysis, we infer expected counts from a subsampled dataset and compare them to the original bike theft data (Figure 9). We investigate the effect of the length-scale parameter ℓ on the inferred counts by visualizing the results for different values of ℓ. Additionally, we perform a grid search over ℓ to determine a reasonable length-scale value. The parameters used in the primary inference are $\ell = 2,\ \beta = 0.2,\ T = 10^4,$ subsampling factor $= 3$. The mean absolute error is 1.33.

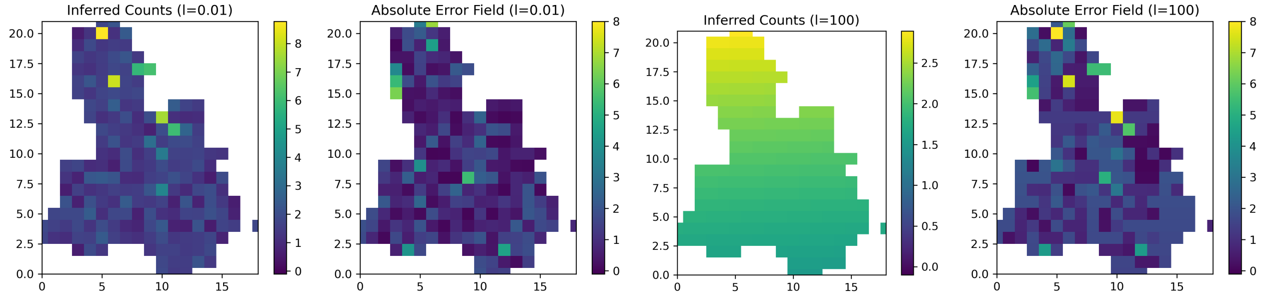


(a) Original Count Data (b) Subsampled Data (c) Inferred Counts (d) Error Field.

Figure 9: Comparison of original count data, subsampled data, inferred counts, and error field.

F.3 Impact of Length-scales

To understand the effect of different length-scales, we visualize the inferred counts and error fields for extreme values $\ell = 0.01$ and $\ell = 100$. The results are shown in Figure 10.



(a) Inferred Counts. $\ell = 0.01$

(b) Error Field. $\ell = 0.01$. MAE=1.12

(c) Inferred Counts. $\ell = 100$

(d) Error Field. $\ell = 100$, MAE=1.54

Figure 10: Comparison of inferred counts and error fields for extreme length-scales.

- $\ell = 0.01$ yields a smaller mean error compared to both $\ell = 2$ and $\ell = 100$, but the inferred field is highly noisy, suggesting overfitting to the observations.
- $\ell = 2$ provides a reasonable degree of smoothing while retaining local features.
- $\ell = 100$ leads to excessive smoothing, removing almost all local variations, resulting in a very slow north-south variation.

To further investigate the effect of ℓ, we conducted a grid search to evaluate the absolute mean error as a function of ℓ. The results are plotted in Figure 11.

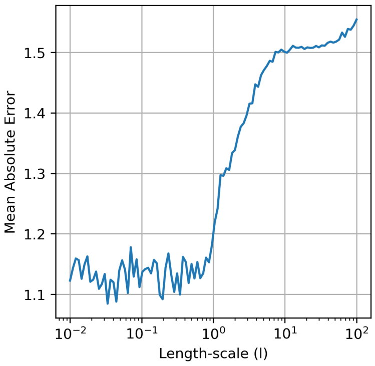


Figure 11: Absolute Mean Error vs. Length-scale ℓ

For small values of ℓ ($\ell < 1$), the error initially decreases and reaches its minimum around $\ell = 1$. However, excessively small ℓ values ($\ell \ll 1$) lead to overfitting, capturing high-frequency variations that do not generalize well. The error stabilizes in the range of 1.1 to 1.2 before continuing to rise. Conversely, for $\ell > 1$, the error increases as ℓ grows. Larger length-scales excessively smooth the inferred field, suppressing meaningful variations in the data and leading to higher prediction errors.

Based on these observations, a length-scale of $\ell \approx 1$ provides a good balance, minimizing error while avoiding the pitfalls of overfitting at small ℓ and oversmoothing at large ℓ .

Conclusion

This report compared GRW-MH and pCN for high-dimensional Gaussian Process inference, demonstrating pCN's superior scalability due to its stable acceptance rate. We explored the impact of length-scale ℓ on inference accuracy, finding that an optimal ℓ balances overfitting and oversmoothing. For spatial count data, we derived the Poisson log-likelihood and applied pCN for inference, confirming its efficiency for large-scale Bayesian modeling. Our findings highlight pCN's robustness in high-dimensional inference tasks.