CUED - Engineering Tripos Part IIB 2021-2022 **Module Coursework**

Module	4M24	Title of report	Coursework: High Dimensional MCMC
Date submitted:	19/01/2022	Assessment for this module is 100% / ☐ 25% coursework of which this assignment forms 100 %	

UNDERGRADUATE STUDENTS ONLY		POST GRADUATE STUDENTS ONLY			
Candidate number:	5588C	Name:		College:	

Feedback to the student

☐ **See also comments in the text**

		Very good	**Good**	Needs improvmt
CONTENT	**Completeness, quantity of content:** Has the report covered all aspects of the lab? Has the analysis been carried out thoroughly?			
	Correctness, quality of content Is the data correct? Is the analysis of the data correct? Are the conclusions correct?			
	Depth of understanding, quality of discussion Does the report show a good technical understanding? Have all the relevant conclusions been drawn?			
	Comments:			
PRESENTATION	**Attention to detail, typesetting and typographical errors** Is the report free of typographical errors? Are the figures/tables/references presented professionally?			
	Comments:			

Indicative grades are not provided for the FINAL piece of coursework in a module

Assessment (circle one or two grades)	A*	A	B	C	D
Indicative grade guideline	>75%	65-75%	55-65%	40-55%	<40%
Penalty for lateness:	*20% of maximum achievable marks per week or part week that the work is late.*				

Marker: Date:

1 Exercise (a)

Figures (1) and (2) show samples drawn from a Gaussian Process prior $\mathcal{N}(0, C)$, where C is the covariance matrix induced by an isometric squared exponential covariance function evaluated on a uniformly spaced $D \times D$ grid on $[0, 1]^2$. Larger length scales show smoother prior surfaces, whereas smaller length scales show more ragged, complex surfaces. This corresponds to the fact that small length scales cause the covariance between distant points to go to 0 quicker, and hence any point is only dependent on nearby points, allowing for more abrupt changes in the value of the latent variable $\mathbf{u}$.

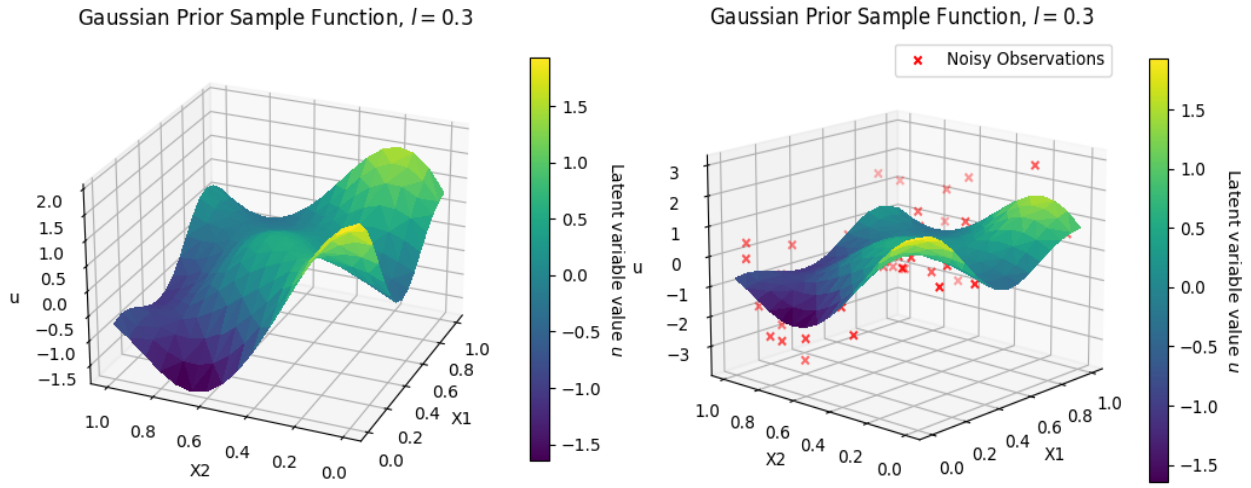

Figure 1: *GP Prior with* $\{l, D\} = \{0.3, 16\}$.

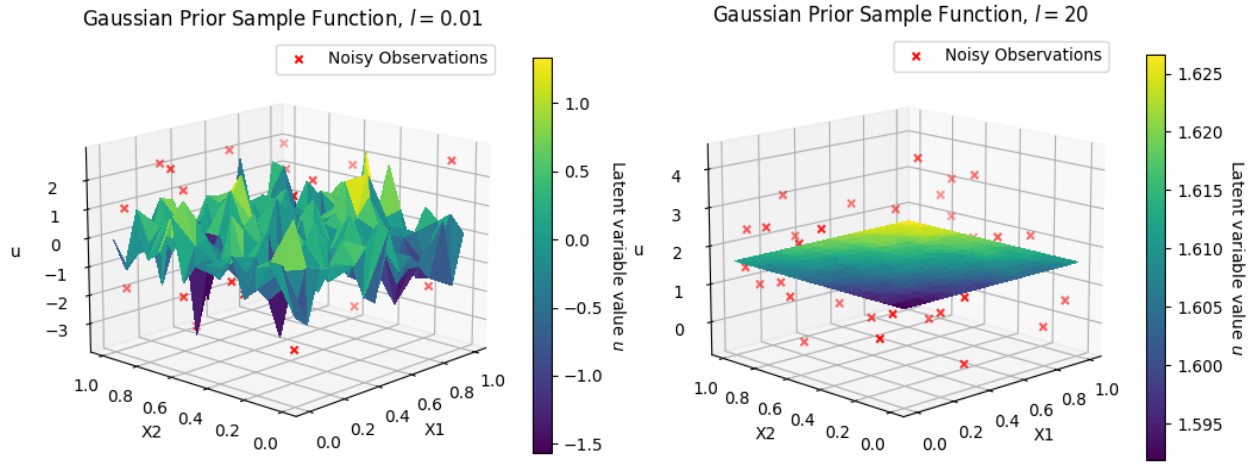

Figure 2: *GP Priors with Noisy Observations overlaid for* $l = 0.01, 20$*, left and right, respectively.*

The choice of prior is important because it models the a priori knowledge of the dependency between latent variables, and it influences the samples drawn from the target posterior measure. In the extremes, a very large length scale acts as a uniform prior, as shown with $l = 20$ in Figure (2), meaning the samples contain similar entries and are drawn from a target measure which is a scaled version of the likelihood. On the other hand, an extremely small length scale means the model can fit the data better since each sample $\mathbf{u}$ can have very different valued entries, but this can come at the cost of overfitting to the observed data.

2 Exercise (b)

Figures (3) and (4) show the mean and error fields of sampled latent variables, $\mathbf{u}$, from the target posterior, $p(\mathbf{u}|\mathbf{v})$, using the Gaussian Random Walk Metropolis Hastings (GRWMH) and the pre-conditioned Crank-Nicholson (pCN) algorithms, respectively. Both error fields have similar colour contrast, and their similar shape reflects the idea that both algorithms explore the same target density, so, if enough iterations have passed for the algorithms to converge, both algorithms should draw samples from the same posterior. Further, both pCN and GRWMH converge independent of the choice of initial sample, although the convergence rate is longer for poorly-chosen initial states, i.e., states chosen far away from the region of high probability mass of the target measure. This is behaviour is expected of all MCMC algorithms, as the algorithms should end up drawing samples from the target measure irrespective of where they are started from.

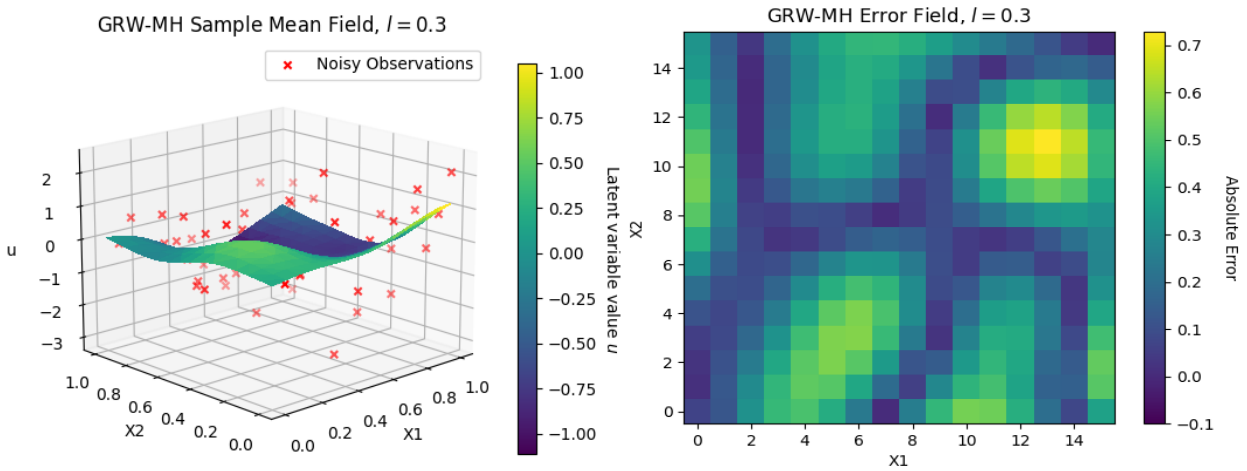

Figure 3: *GRWMH Mean Field and Error field, left and right, respectively.*

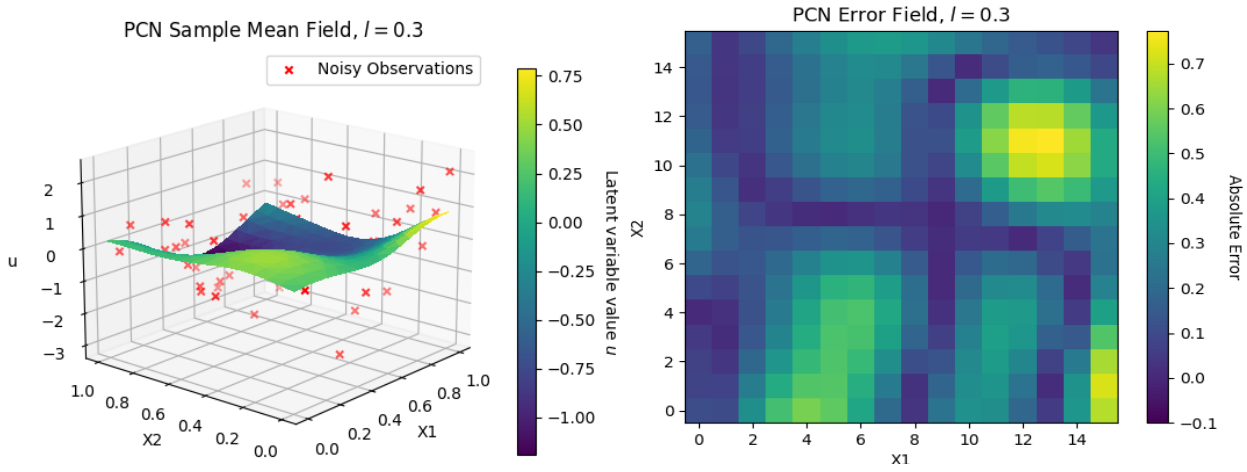

Figure 4: *Preconditioned Crank Nicholson Mean Field and Error field, left and right, respectively.*

Algorithm	Acceptance Prob.	Mean Absolute Error
GRWMH	0.0793	0.2735
PCN	0.467	0.2468

Table 1: *Acceptance Probability and Mean Errors for GRWMH and PCN;* $\{n, l, \beta, D\} = \{10000, 0.3, 0.2, 16\}$

While the error fields have similar shape, Table (1) shows that the error using the pCN algorithm is 9.76% lower, which can be attributed to the low acceptance probability of GRWMH. Table (1) shows the GRWMH algorithm accepts less than 8% of proposed new samples, while pCN accepts 46.7%. Thus, for the same number of iterations, pCN explores the target space much quicker than GRWMH, and the drawn samples are more representative of the target density. However, as Figure (5) shows, by increasing the iteration time the algorithms are run for, the error of both algorithms converges, as the state space is fully explored by both.

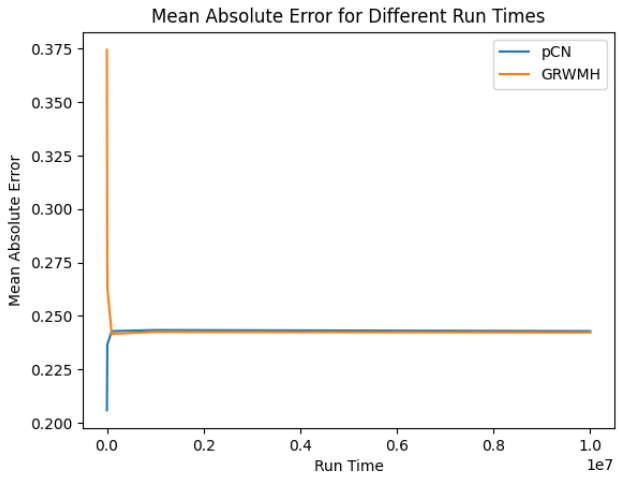

Figure 5: *Mean Absolute Error of pCN and GRWMH algorithms for varying run-times.*

Figure (6) shows that the acceptance probability is a non-increasing function of the step-size, where samples are accepted with probability 1 with a step size of 0.0. Large step sizes correspond to proposing large jumps, which are accepted with low probability, as the ratio of the target measures is low. On the other extreme, small step sizes correspond to proposing small jumps, which are accepted with high probability, but the jumps are so small the state-space isn't explored efficiently. A step size of 0 suggests the algorithms are not exploring the state-space at all, but rather keeping the initial sample, $\mathbf{u_0}$, meaning the samples obtained are all the same and are not representative of the target density.

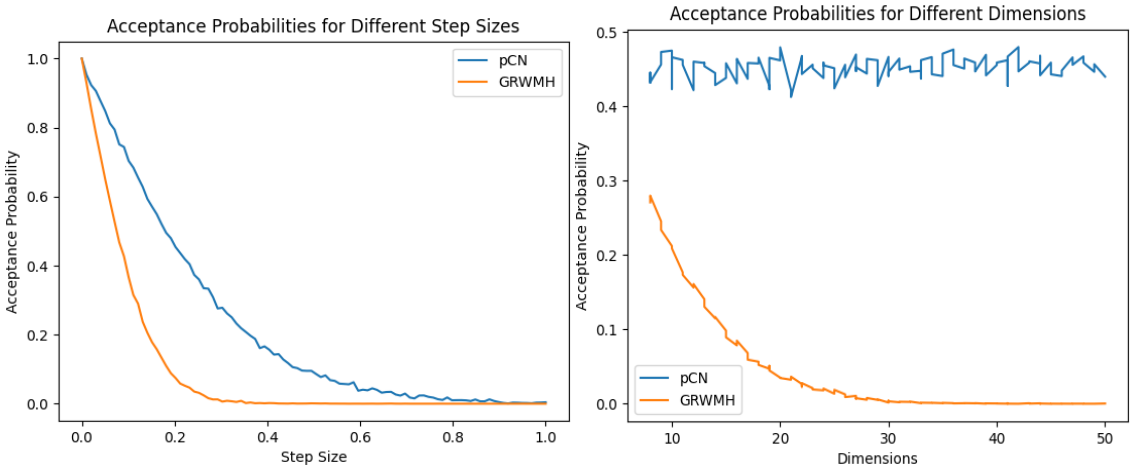

Figure 6: *Acceptance Probabilities for varying step size and dimensionality, left and right, respectively.*

By fixing the dimensions of the observation space, M, and varying the latent space dimension, N, Figure (6) also shows the acceptance probability tends to 0 for the Gaussian Random Walk, reflecting the fact that the GRWMH algorithm has a poorly defined acceptance probability in infinite dimensions because it contains the norm $\left|C^{-\frac{1}{2}}\mathbf{u}\right|$ which is undefined. On the other hand, in this application, the pCN algorithm has a constant average acceptance probability of 0.461 as latent dimension space increases, reflecting the fact that the pCN acceptance probability is dependent only on the ratio of likelihoods $p(\mathbf{v}|\mathbf{u}) = \mathcal{N}(\mathbf{Gu}|\mathbf{I})$, and is therefore independent of N.

3 Exercise (c)

In probit classification, each datapoint, u_i, is assigned to a particular class, $t_i \in \{0, 1\}$, with probability $p(t_i = 1|\mathbf{u}) = \phi(u_i)$, where ϕ is the standard normal CDF function. Given a vector of classes $\mathbf{t} = \{t_i\}_{i=1}^N$, and of the latent states, $\mathbf{u}$, the full probit log-likelihood is given below, where $\mathbb{1}$ is the indicator function and N is the number of datapoints.

$$\begin{aligned}
\log\left[p(\mathbf{t}|\mathbf{u})\right] &= \log\left[\prod_{j=1}^{N} p(t_j|\mathbf{u})\right] \\
&= \sum_{j=1}^{N} \log\left[p(t_j = 1|\mathbf{u})^{\mathbb{1}(t_j=1)} p(t_j = 0|\mathbf{u})^{\mathbb{1}(t_j=0)}\right] \\
&= \sum_{j=1}^{N} t_j \log\left[\phi(u_j)\right] + (1 - t_j)\log\left[1 - \phi(u_j)\right] \qquad (1)
\end{aligned}$$

To find the predictive probability assigned to new t_j^*, $p(t_j^* = 1|\mathbf{t})$, a Monte Carlo estimate of the predictive distribution can be computed by drawing samples from the target posterior distribution, $p(\mathbf{u}|\mathbf{t}) \propto p(\mathbf{t}|\mathbf{u})p(\mathbf{u})$, where $p(\mathbf{u}) = \mathcal{N}(0, C)$, as before. The Monte Carlo estimate, given M samples $\mathbf{u}^{(i)} \sim p(\mathbf{u}|\mathbf{t})$, is given below:

$$p(t_j^* = 1|\mathbf{t}) \approx \frac{1}{M}\sum_{i=1}^{M} p(t_j^* = 1|\mathbf{u}^{(i)}) = \frac{1}{M}\sum_{i=1}^{M} \phi(u_j^{(i)}) \qquad (2)$$

The full predictive distribution is therefore:

$$p(\mathbf{t}^*|\mathbf{t}) \approx \frac{1}{M}\sum_{i=1}^{M}\phi(\mathbf{u}^{*(i)}) \tag{3}$$

Figure (7) shows the true class assignments for the entire latent space, where yellow indicates class 1 if $u_j > 0, u_j \in \mathbf{u}$, and purple indicates class 0 otherwise. It is clear the predictive distribution matches reasonably well the true assignments, where yellow regions correspond to true assignments of class 1. However, the classifier's confidence on its predictions is lower near the decision boundaries, where, for example, both the top left and bottom right hand corners have probabilities close to 0.5, rather than being confidently class 1, as shown in the true assignment plot. Nonetheless, this behaviour is expected, or even desirable. Any observed datapoint near the boundary could belong to one class or the other if subject to small observation noise, so a lower probability avoids data overfitting by reducing the confidence on its prediction.

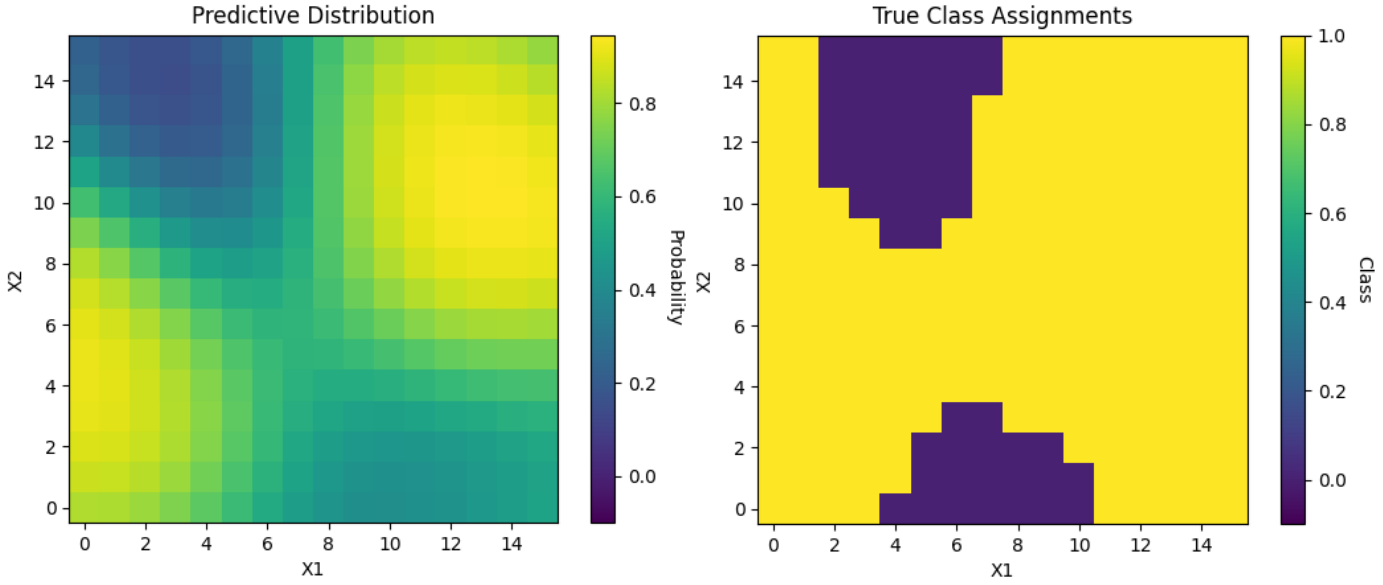

Figure 7: *Class predictive probabilities, left, and true classes, right for* $l = 0.3$.

4 Exercise (d)

Using the Monte Carlo estimate for the predictive probabilities in Exercise (3), hard class assignments are achieved by setting $t_i = 1 \iff p(t_i|\mathbf{t}) \geq 0.5$, and Figure (8) shows the classification results of the pCN samples. The right-hand plot also shows the absolute error field between the hard and the true class assignments. As mentioned in Exercise (3), the predictive distribution predicts the true class well, with a mean prediction error of 0.0793. However, different values for the model's length scale or the number of drawn posterior samples will cause the prediction error to vary, as, given a finite number of samples, the Monte Carlo estimate for the predictive density is only an approximation to the distribution.

Figure 8: *From left to right: hard classification, true classification, and absolute error between both.*

Although it is known the data was generated with a length scale $l = 0.3$, it is unrealistic to assume this hyperparameter is known a priori, so it is useful to investigate whether length-scale inference is possible. Figure (9) shows that an 100-point 1D grid search between $0.01 \leq l \leq 10$ is able to correctly infer the order of magnitude of the true length-scale generating the data, as the optimal length-scale is 0.266. Clearly, extreme length scales yield large prediction errors, as large lengths do not capture variation in the latent field quickly enough, and small lengths treat each entry in $\mathbf{u}$ as independent, failing to account for covariances in the true data. It is important to note that the optimisation procedure cannot obtain the exact length scale since the predictive estimate in Equation (3) is only an approximation to the true predictive distribution, and the posterior samples are only draws from the posterior $p(\mathbf{u}|\mathbf{t})$, rather than the actual data set.

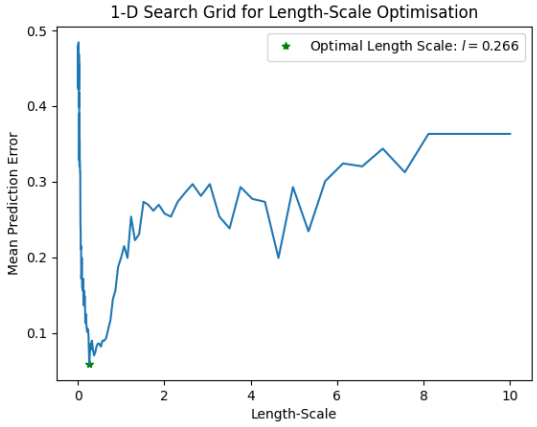

Figure 9: *1D Grid Search for optimal length-scale.*

5 Exercise (e)

This next two exercises look at predicting bike theft counts from the Lewisham Borough in London. To do this, theft counts, $\mathbf{c} = \{c_i\}_{i=1}^N$, are assumed to follow a Poisson distribution, with the mean rate modelled as $\boldsymbol{\theta} = e^{\mathbf{u}}$, where it is assumed $\mathbf{u} \sim \mathcal{N}(0, C)$ a priori. Given that the one-to-one mapping $\mathbf{u} \to \boldsymbol{\theta}$ is deterministic, the full log-likelihood is:

$$\log\left[p(\mathbf{c}|\mathbf{u})\right] = \log\left[p(\mathbf{c}|\boldsymbol{\theta})\right] = \log\left[\prod_{i=1}^{N} p(c_i|\theta_i)\right] = \sum_{i=1}^{N} \log\left(\frac{\theta_i^{c_i} e^{-\theta_i}}{c_i!}\right)$$

Hence:

$$\log\left[p(\mathbf{c}|\mathbf{u})\right] = \sum_{i=1}^{N} -\theta_i + c_i \log\left(\theta_i\right) - \log(c_i!) \tag{4}$$

The full log-likelihood above in Equation (4) can be combined with the prior on the field $\mathbf{u} \sim \mathcal{N}(0, C)$ to sample from the posterior $p(\mathbf{u}|\mathbf{c}) \propto p(\mathbf{u})p(\mathbf{c}|\mathbf{u})$. However, for the purpose of efficient posterior sampling using the pCN algorithm, the factorial term can be ignored, since the pCN acceptance probability reduces to the difference of the likelihoods in Equation (4), and $c_i!$ is constant for all proposed $\mathbf{u}$.

6 Exercise (f)

Figure (10) shows the true bike theft counts in the Lewisham Borough, with higher counts denoted by a yellow color. Subsampled counts are shown on the right, corresponding to observing 3 times less data points than those in the dataset. The use of subsampled data as observations allows to reduce the computational cost of evaluating the log-likelihood in Equation (4), while also reducing overfitting by only training on a subset of the dataset. It is also useful to note that the true count data shows that the north of the borough has more cells with high theft counts than the south, though, on average, the theft count per cell is low.

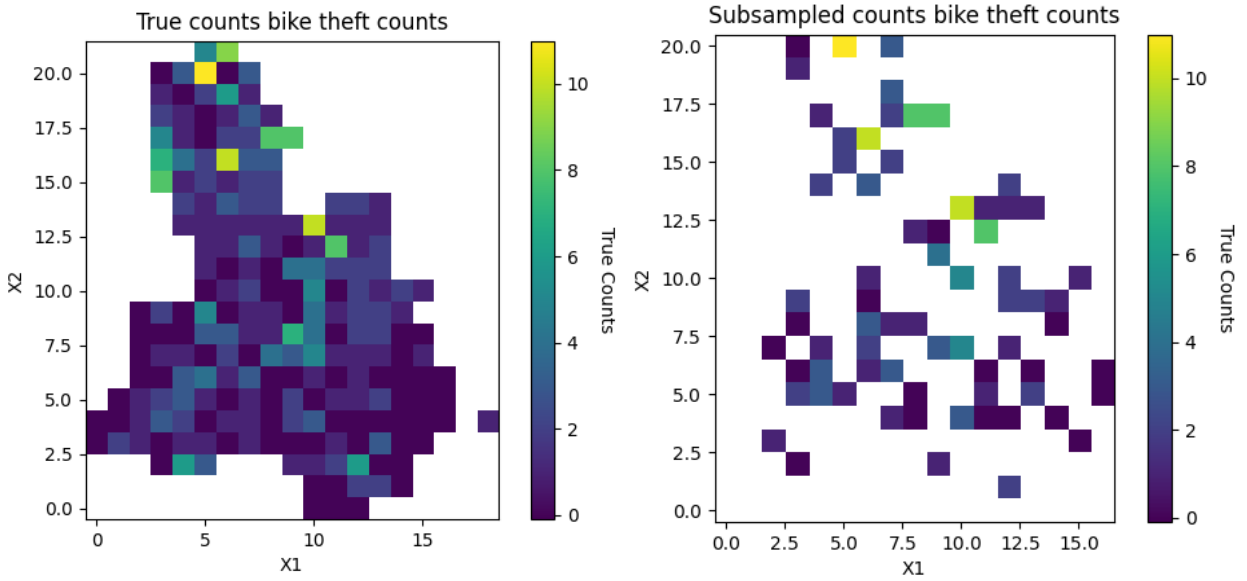

Figure 10: *True and Subsampled Theft Counts.*

Drawing samples from the posterior over rates, or, equivalently, from the posterior over the latent field,

$p(\mathbf{u}|\mathbf{c})$, it is possible to infer the expected theft counts in the borough with:

$$\mathbb{E}\left[c^*\right] = \sum_{k=0}^{\infty} k p(c^* = k|\mathbf{c}) \tag{5}$$

where $p(c^* = k|\mathbf{c}) = \int p(c^* = k|\mathbf{u})p(\mathbf{u}|\mathbf{c})d\mathbf{u}$ is the predictive distribution. Using the Monte Carlo estimate for the predictive distribution, we obtain a MC estimate for the expected counts, where $\mathbf{u}^{(i)} \sim p(\mathbf{u}|\mathbf{c})$:

$$\mathbb{E}\left[c^*\right] \approx \sum_{k=0}^{\infty} k \frac{1}{N} \sum_{i=1}^{N} p(c^* = k|\mathbf{u}^{(i)}) = \frac{1}{N} \sum_{i=1}^{N} \left[\sum_{k=0}^{\infty} k p(c^* = k|\mathbf{u}^{(i)})\right] = \frac{1}{N} \sum_{i=1}^{N} \mathbb{E}_{p(c^*|\mathbf{u}^{(i)})}[c^*]$$

Hence, using the fact $\mathbb{E}_{p(c^*|\mathbf{u}^{(i)})}[c^*] = \mathbb{E}_{p(c^*|\boldsymbol{\theta}^{*(i)})}[c^*] = \boldsymbol{\theta}^{*(i)}$:

$$\mathbb{E}\left[c^*\right] \approx \frac{1}{N} \sum_{i=1}^{N} \boldsymbol{\theta}^{*(i)} \tag{6}$$

Figure (11) below shows the inferred expected counts using the Monte Carlo estimate in Equation (6) for different length scales. The smaller length scale implies smaller covariance between distant points in the map, which corresponds to bike theft rates influenced by nearby areas in the borough only. This is what is expected, as areas with high thefts are more likely to be surrounded by areas of high thefts, and this dependence doesn't occur on the larger scale. However, extremely small length scales cause the model to treat theft counts as independent even within the same cell, which is unrealistic, as the true counts in Figure (10) show cells with higher thefts are surrounded by cells with higher-than-average thefts.

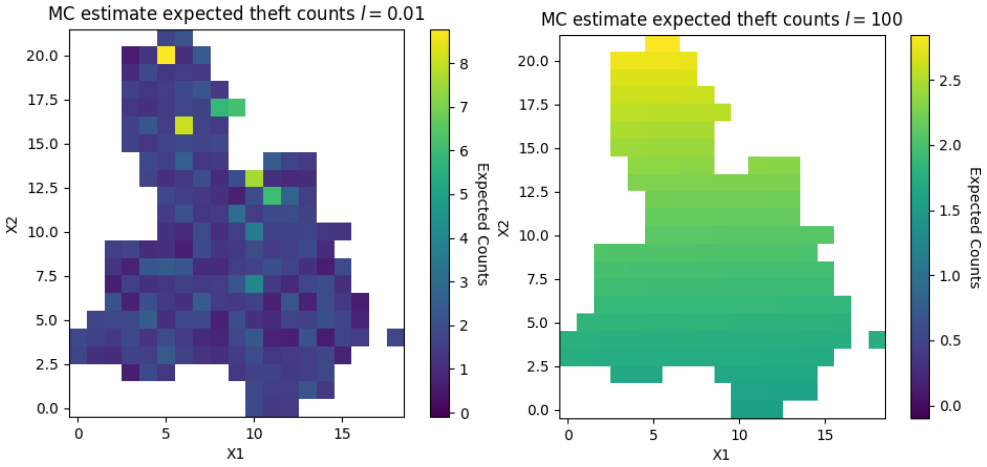

Figure 11: *Inferred Expected Counts for* $l = 0.01, 100$*, left and right, respectively.*

Larger length scales, on the other hand, imply that thefts in one area are influenced by far-away areas. This is reflected by the $l = 100$ plot, which shows a smoother color gradient showing greater spatial dependency between theft rates. This model captures the trend of the north having higher theft counts than the south, but it fails to account for differences at a more granular level. It also under-predicts the peak true theft count, from 11.0 to 2.69. Therefore, both extreme length-scales are unsuitable for the current application, though Table (2) and Figure (12) show that the mean error for larger length scales is larger than for smaller length scales.

	$l = 0.01$	$l = 100$	$l = 0.0014$	$l = 0.63504$
Mean Error	1.1226	1.5568	1.0750	1.1118

Table 2: *Mean Absolute Error for different lengths.*

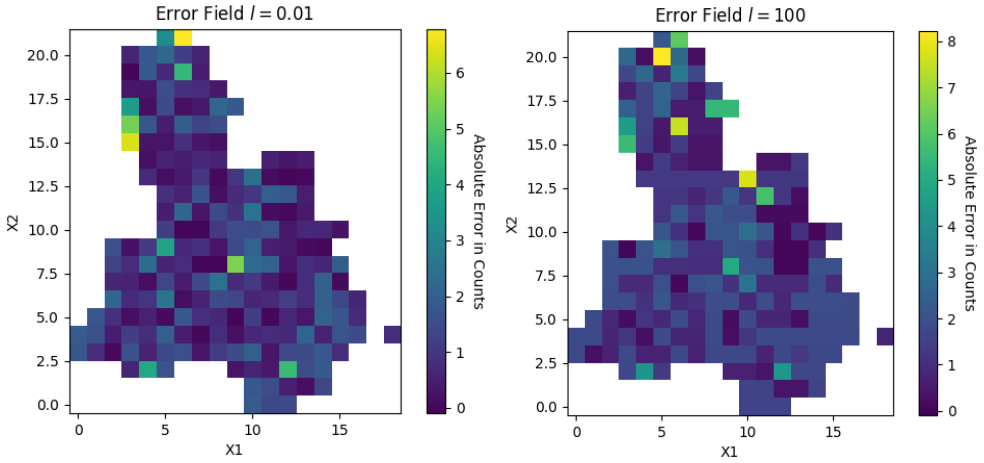

Figure 12: *Absolute Error in Counts for* $l = 0.01, 100$*, left and right, respectively.*

By perform a 1-D grid search, Figure (13) shows lower length scales attain lower mean prediction errors than larger length scales, reflecting the intuition that theft counts are only locally dependent, rather than globally. However, choosing a very small length scale, as suggested by the optimisation procedure, risks overfitting and losing the ability to generalise predictions outside of the dataset provided, as $l = \mathcal{O}(10^{-3})$ suggests theft counts are dependent on a very small surrounding area. This prevents the model from correctly predicting counts in situations with different dynamics to those of this particular borough in that particular year. Therefore, the optimal length scale should be the one which maintains a low prediction error, but allows for better generalisation. Based on the 1-D grid search, $l = 0.63504$ is the best option, with an absolute error only 3.43% higher than the optimal, according to Table (2).

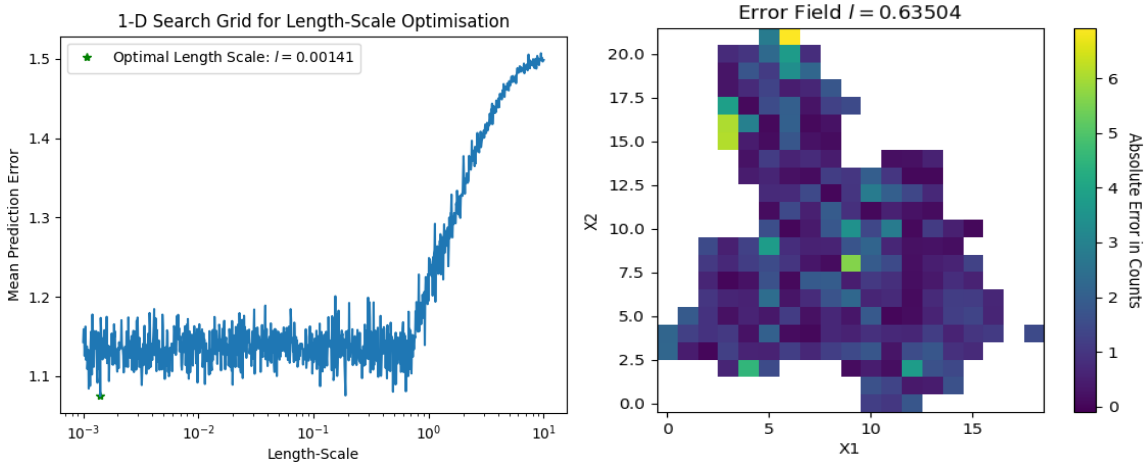

Figure 13: *1-D Grid Search for Optimal Length Scale, left, and error field for best length scale, right.*