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

Module	4E6	Title of report	High-Dimensional MCMC

Date submitted: 05/01/2022	Assessment for this module is ☑ 100% / □ 25% coursework of which this assignment forms 100

UNDERGRADUATE STUDENTS ONLY		**POST GRADUATE STUDENTS ONLY**				
Candidate number:	5747A	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:

Engineering Tripos Part IIB Student Candidate No. 5747A

4M24 Coursework: High-Dimensional MCMC

1 Problem Definition

In this coursework, two MCMC algorithms - the Gaussian Random Walk Metropolis-Hastings (GRW) and preconditioned Crank-Nicholson (pCN) methods - are applied to a high-dimensional problem, and their performance is compared. The latent variables of a 2D Gaussian Process (GP) posterior are sampled and applied to modelling the spatial distribution of bike thefts in London (Fig ??).

Part Ia - Simulation Regression

Initially, the algorithms are compared by sampling a set of points from a GP prior $\boldsymbol{u} \sim \mathcal{N}(0, C^{(L)})$, where $C^{(l)}_{i,j} = k^{(l)}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp\left(-\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2/2l^2\right)$ and coordinates $\boldsymbol{x} = (x_1, x_2) \in [0,1]^2$ lie on a uniform $D \times D$ mesh with $D^2 = N$, then attempting to infer the original $\boldsymbol{u}$. A random subset of $\boldsymbol{u}$'s components are observed with noise, with an $M \times N$ matrix G of ones and zeros choosing the observation positions:

$$\boldsymbol{v} = G\boldsymbol{u} + \boldsymbol{\epsilon} \in \mathbb{R}^M; \qquad \epsilon_i \sim N(0, \sigma_n^2 I)$$

Both algorithms are based on the Metropolis-Hastings method. They differ in their proposal distributions $q(\boldsymbol{u}^{(k)}) = \boldsymbol{u}'$ and acceptance probabilities $\alpha(\boldsymbol{u}', \boldsymbol{u}^{(k)})$:

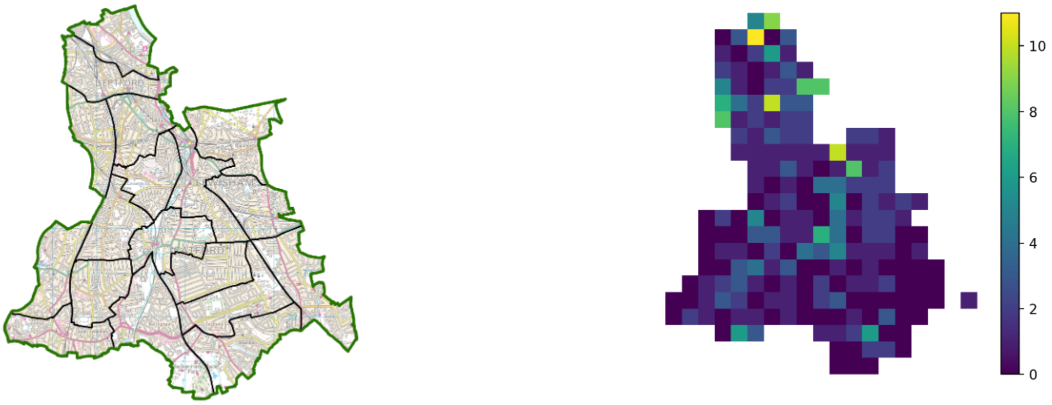
$$\begin{aligned}
\text{GRW:} \quad & q(\boldsymbol{u}^{(k)}) = \boldsymbol{u}' = \boldsymbol{u}^{(k)} + \beta\boldsymbol{\xi} \\
& \alpha(\boldsymbol{u}', \boldsymbol{u}^{(k)}) = \min\left\{\frac{p(\boldsymbol{v}|\boldsymbol{u}')p(\boldsymbol{u}')}{p(\boldsymbol{v}|\boldsymbol{u}^{(k)})p(\boldsymbol{u}^{(k)})}, 1\right\} \\
\text{pCN:} \quad & q(\boldsymbol{u}^{(k)}) = \boldsymbol{u}' = \sqrt{1-\beta^2}\boldsymbol{u}^{(k)} + \beta\boldsymbol{\xi} \\
& \alpha(\boldsymbol{u}', \boldsymbol{u}^{(k)}) = \min\left\{\frac{p(\boldsymbol{v}|\boldsymbol{u}')}{p(\boldsymbol{v}|\boldsymbol{u}^{(k)})}, 1\right\}
\end{aligned}$$

where β is a step size hyperparameter and $\boldsymbol{\xi} \sim \mathcal{N}(0, C^{(l)})$ for some guessed length scale hyperparameter l. The acceptance probabilities are computed in the log space for numerical stability, and since only ratios are considered, constant factors can be discarded, leading to the simple likelihood and prior expressions:

$$\log p(\boldsymbol{v}|\boldsymbol{u}) = -\frac{1}{2\sigma_n^2}\sum_{m=1}^{M} \|\boldsymbol{v} - G\boldsymbol{u}\|^2 + \text{const.}$$

$$\log p(\boldsymbol{u}) = -\frac{1}{2}\boldsymbol{u}^T C^{-1}\boldsymbol{u} + \text{const.}$$

The reconstructed field $\hat{\boldsymbol{u}}$ is simply the Monte Carlo expectation $\mathbb{E}_{\boldsymbol{u}|\boldsymbol{v}}[\boldsymbol{u}] = \int \boldsymbol{u}p(\boldsymbol{u}|\boldsymbol{v})d\boldsymbol{u} \approx \frac{1}{N_s}\sum_{i=1}^{N_s} \boldsymbol{u}^{(i)}$ over the samples, since they are distributed as $p(\boldsymbol{u}|\boldsymbol{v})$.



(a) Map of Lewisham Borough

(b) Bike theft dataset

Figure 1

Part Ib - Simulation Classification

This regression is then extended to a classification problem where the true length scale L used to generate the sample $\boldsymbol{u}$ is withheld, in order to better represent the true bicycle theft data (which is reported in discrete counts). The probit transform $f : \boldsymbol{u} \rightarrow \boldsymbol{t}_{true}$ is taken, defined by $[t_{true}]_m = \mathbb{I}_{u_m > 0}$, while the observed values are similarly mapped as $\boldsymbol{v} \rightarrow \boldsymbol{t}$. This means that

$$\begin{aligned} p(t_m = 1|\boldsymbol{u}) &= \Phi([G\boldsymbol{u}]_m/\sigma_n) \\ p(t_m = 0|\boldsymbol{u}) &= 1 - \Phi([G\boldsymbol{u}]_m/\sigma_n) \end{aligned}$$

Hereafter, only the pCN method is used due to its superior high-dimensional performance. Therefore only the sample likelihood is required:

$$p(\boldsymbol{t}|\boldsymbol{u}) = \prod_{m=1}^{M} p(t_m = 1|\boldsymbol{u})^{t_m} p(t_m = 0|\boldsymbol{u})^{1-t_m}$$

This expression can be expanded to its full form by substituting the t_m conditionals. The logarithm is then again taken for practical computation, resulting in a more numerically stable sum. A posterior predictive distribution can be constructed with a Monte Carlo estimate using the posterior samples $\{\boldsymbol{u}^{(j)}\}_{j=1}^{N_s}$ as

$$\begin{aligned} p(t_n^* = 1|\boldsymbol{t}) &= \int p(t_n^* = 1|\boldsymbol{u}) p(\boldsymbol{u}|\boldsymbol{t}) d\boldsymbol{u} \\ &= \mathbb{E}_{\boldsymbol{u}|\boldsymbol{t}}\left[p(t_n^* = 1|\boldsymbol{u})\right] \\ &\approx \frac{1}{N_s} \sum_{i=1}^{N_s} p(t_n^* = 1|\boldsymbol{u}^{(i)}) \end{aligned}$$

Hard assignments are made by thresholding these values at 0.5. Creating a good predictive distribution relies on having a good prior model for $\boldsymbol{u}$, which includes knowing an appropriate length scale $l \approx L$. As a result, significant effort is dedicated to estimating the true length scale L via optimisation over l by comparing prediction errors.

Part II - Spatial Data

The GP prior is next applied to regression of bike theft rates in Lewisham Borough (Fig 1b). The dataset contains a list of x_1, x_2 locations of $400m^2$ cells and the corresponding numbers of bike thefts $\boldsymbol{c}$ in those cells in 2015, though this is rescaled to a 1×1 grid to aid comparison to Part I. This data is subsampled and the theft rate $\boldsymbol{\theta}$ is inferred over the original field, using a Poisson model for theft incidences:

$$p(\boldsymbol{c}|\boldsymbol{\theta}) = \prod_{m=1}^{M} \frac{e^{\theta_m} \theta_m^{c_m}}{c_m!}$$

The field $\boldsymbol{\theta}$ is obtained by taking $\theta_m = e^{[G\boldsymbol{u}]_m}$ to ensure positivity. The log-likelihood required for pCN is then:

$$\begin{aligned} \log p(\boldsymbol{c}|\boldsymbol{\theta}) &= \sum_{m=1}^{M} \big(c_m \log \theta_m - \theta_m - \log c_m!\big) \\ &= \sum_{m=1}^{M} \Big(c_m \log\, [G\boldsymbol{u}]_m - e^{[G\boldsymbol{u}]_m}\Big) + \text{const.} \end{aligned}$$

The prior implicitly created over $p(\boldsymbol{\theta})$ is nontrivial, but its marginals θ_m can be easily found by considering a change-of-variable transformation from their corresponding u_n:

$$\begin{aligned} p(\theta_m) = f_{\theta_m}(\theta_m) &= F'_{\theta_m}(\theta_m) \\ &= f_{u_n}(\ln \theta_m)/\theta_m \\ &= \frac{1}{\theta_m \sqrt{2\pi}} e^{(\ln \theta_m)^2/2} \\ &\sim \text{Lognormal}(0, 1) \end{aligned}$$

In order to make predictions and evaluate model error, posterior Monte Carlo estimate of the rate field must be formulated. For a Poisson distribution, the expectation of the count is equal to the rate parameter, and this expectation can be found at location n as:

$$\begin{aligned} \mathbb{E}_{c_n^*|\boldsymbol{c}}[c_n^*] &= \sum_{k=0}^{\infty} k p(c_n^* = k|\boldsymbol{c}) \\ &= \sum_{k=0}^{\infty} k \int p(c_n^* = k|\theta_n^*) p(\theta_n^*|\boldsymbol{c})\ d\theta_n^* \\ &= \int \left(\sum_{k=0}^{\infty} k p(c_n^* = k|\theta_n^*) \right) p(\theta_n^*|\boldsymbol{c})\ d\theta_n^* \\ &= \int \big(\mathbb{E}_{c_n^*|\theta_n^*}[c_n^*]\big) p(\theta_n^*|\boldsymbol{c})\ d\theta_n^* \\ &= \int \theta_n^* p(\theta_n^*|\boldsymbol{c})\ d\theta_n^* = \mathbb{E}_{\theta_n^*|\boldsymbol{c}}[\theta_n^*] \\ &= \mathbb{E}_{\theta_n^*|\boldsymbol{c}}[e^{u_n^*}] \approx \frac{1}{N_s} \sum_{i=1}^{N_s} e^{{u_n^*}^{(i)}} \end{aligned}$$

These predictions are compared to the data in order to attempt inference of the true length scale of variance of bike thefts in Lewisham. If an appropriate l can be found that gives a good model, this could be used in several applications - for example, interpolating bike theft risk over space, or obtaining predictive bike theft probabilities in unobserved locations.

2 Analysis

Part Ia - Simulation Regression

Visualising prior samples

Initially, data was subsampled by a factor of 4 with $D = 16$ and $\sigma_n = 1$. This value of D is approximately representative of the bike theft data, which has 207 ($= 14.39^2$) datapoints bounded by a grid of size 18×21.

The typical vertical scale of drawn functions are the square roots of the marginal u_i variances, which are C_{ii} for all i, so $\text{Var}(u_i) = 1$. For observations the additive noise increases this to $\text{Var}(v_i) = 1 + \sigma_n^2 = 2$.

The correlation length L of the GP prior determines the typical length scale of variations of functions drawn from the GP (Fig 2). This is because when the L_2 distance between vectors $\boldsymbol{x}_i$, $\boldsymbol{x}_j$ decreases relative to L, $k^{(L)}(\boldsymbol{x}_i, \boldsymbol{x}_j)$ increases, approaching 1 as the grid vectors coalesce. This constrains the posterior probability of the function values at neighbouring points to strongly favour being close to one another.

Since the data is evaluated at grid points within the 1×1 square, L increasing beyond 1 leads to very flat functions, approaching a constant for $L \gg 1$. Decreasing L below $\frac{1}{D}$ leads to rapid fluctuations even between neighbouring grid points, and as $L \to 0$ evaluations approach uncorrelated Gaussian noise (distributed as $\mathcal{N}(0, 1)$, since $k^{(L)}(\boldsymbol{x}_i, \boldsymbol{x}_j) \to \delta_{ij}$). This means that the most interesting functions occur approximately in the range $L \in [\frac{1}{2D}, 2]$. For L below this range the model has very little predictive power, and for L above it the model is too inflexible to make useful predictions.

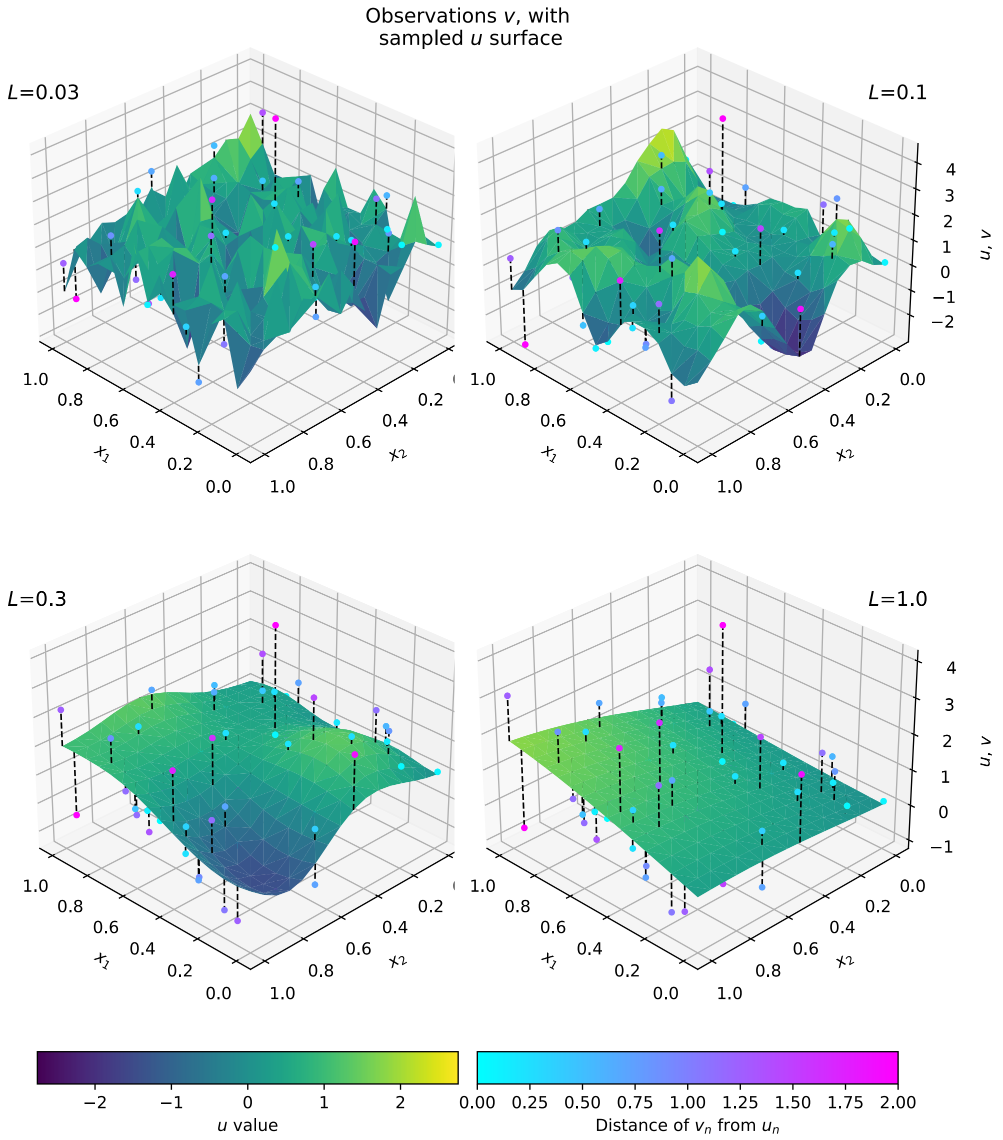


Figure 2: GP prior samples and observations for various L. Seed $= 0$, 4x subsampling, $D = 16$, $\sigma_n = 1$.

Comparing GRW with pCN

For the same hyperparameters and number of iterations, pCN appears to be a much more effective algorithm than GRW: in Fig 3, GRW is evidently giving an inferior Monte-Carlo estimate.

Exploring the acceptance rates for the two techniques reveals the problem: with GRW, almost no samples are accepted for $D = 16$, $\beta = 0.2$ - and for a fixed number of observations, the problem only gets worse with increasing D (Fig 4a). This is because the acceptance probability includes as a factor the ratio $p(\boldsymbol{u}')/p(\boldsymbol{u}^{(k)})$ at the proposed and current positions, and these two probability measures become singular with respect to one another as $D \to \infty$. The mechanism by which the acceptance rate reduces is that the prior covariance $C^{(L)}$ does not converge to a trace class operator as the dimension increases: instead, $|{C^{(L)}}^{-1/2}\boldsymbol{u}| \to \infty$. Therefore, the acceptance probability $\to 0$ if the step would reduce the prior pdf term at all, and GRW grinds to a halt when it eventually finds itself in such a position.

As Fig 4 suggests, decreasing β improves the rejection rate problem. This is because as $\beta \to 0$, $\boldsymbol{u}' \to \boldsymbol{u}^{(k)}$, so $\alpha \to 1$. However, the Markov chain then mixes more slowly. It takes much longer to burn in to a high-probability region, and successive samples will become highly correlated, so the effective number of independent samples reduces and the chain has to be run for much longer in order to get reliable Monte Carlo estimates (Figs 5).

It should also be noted that the apparent convergence of the acceptance rate to 0.5 in Fig 4a for large D and small β does not contradict the previous argument. The reason for this convergence is that while β is very small with respect to the prior landscape's length scale of variations, it is not small enough with respect to its *local gradient* (which diverges as D increases) to accept every step. Since at small β the landscape will be locally linear, the isotropic Gaussian proposed step will be equally likely increase or decrease the prior by an enormous factor, sending $p(\boldsymbol{u}')/p(\boldsymbol{u}^{(k)})$ to 0 or ∞ (and so α to 0 or 1), making acceptance and rejection equally probable.

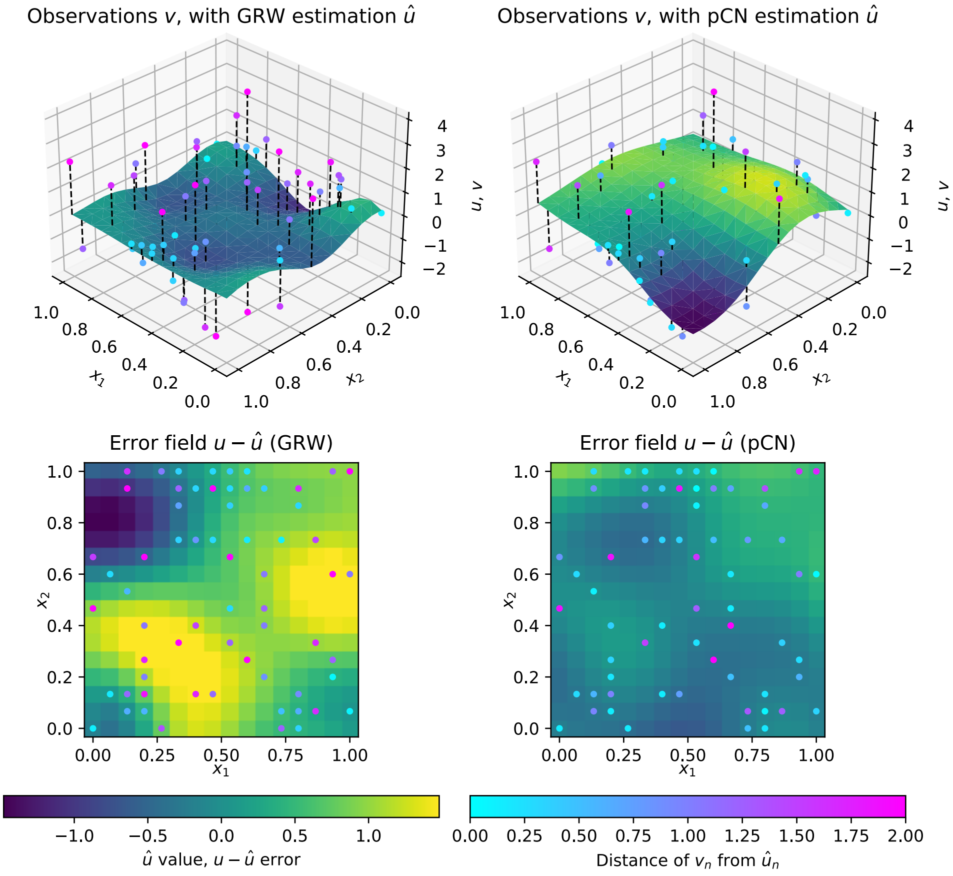


Figure 3: Visualisation of inferred $\hat{\boldsymbol{u}}$ with observations and comparison to $\boldsymbol{u}$ for both algorithms. Seed= 0, 4x subsampling, $D = 16$, $\sigma_n = 1$, $\beta = 0.2$, 10x thinning.

Meanwhile, pCN is highly robust to the process of increasing D for fixed data (known as 'mesh refinement'), maintaining a high acceptance rate for β up to $\beta \sim 0.3$ independent of D (Fig 4b). This is because the proposal step is tuned to ensure that the two probability measures in the numerator and denominator of α are equivalent in the limit of high D (rather than singular), and so the acceptance ratio remains well-defined - no longer containing the prior ratio term.

It is clear that GRW performs poorly for large D. However, it remains to convincingly show that pCN *does* converge in a reasonable time, and contrast it to GRW for the approximate relevant dimension, $D = 16$.

Verifying Convergence

First, an appropriate step size should be found for the chain to mix well. One way to diagnose poor mixing is with the Autocorrelation Function or ACF (Fig 5a). A rough way to obtain an effective correlation length is taking the index offset at which the ACF first descends below 0.5. Fig 5b shows this is optimised by using pCN, and setting $\beta \sim [0.1, 0.4]$. In this region, it is clear that a thinning factor of $\sim$10 will speed computations without losing much information.

To help check convergence of the estimated $\hat{\boldsymbol{u}}$, its Root Mean Squared Error (RMSE) can be plotted versus iteration until either it stabilises to a sufficient degree or an unacceptable compute time has elapsed. If only the second half of samples are used for each estimate, then both the inherent Monte Carlo estimation variance and the bias due to starting in a low-probability region should decrease with iteration. Because all inferences in this report are informed by this analysis, the first half of samples are discarded for every estimate. Fig 6a,c show that pCN significantly outperforms GRW, and that 20,000 iterations is a good cutoff point for all 'interesting' length scales $L \in [\frac{1}{2D}, 2]$ ($\beta = 0.2$). The poor scores and straight segments in the GRW lines are due to long runs of rejections - Fig 6b illustrates this bad mixing by starting runs with the same $\boldsymbol{u}$ but different $\boldsymbol{u}^{(0)}$.

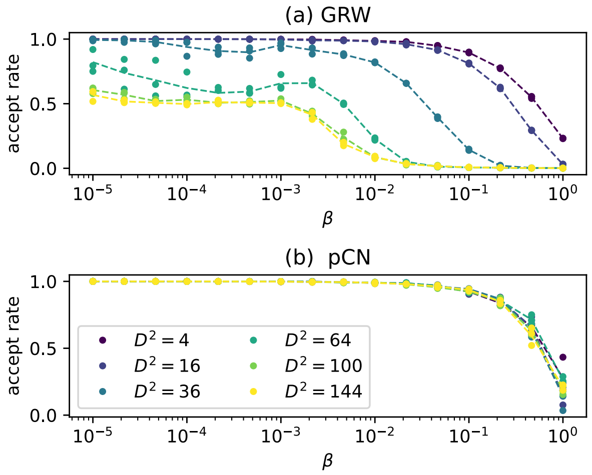


Figure 4: Mean acceptance rate for various mesh spacings and step sizes across seeds $\{0, ..., 9\}$, compared between algorithms. Scatter point are values for individual runs. 10,000 iterations, 4x subsampling, $L = 0.3$, $\sigma_n = 1$.

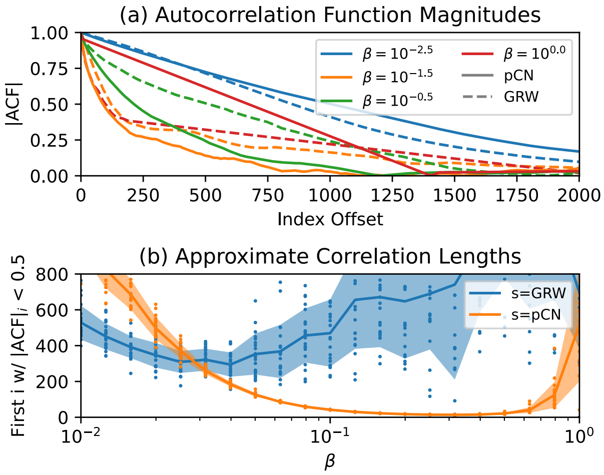


Figure 5: Sample autocorrelations for various β compared between algorithms. Scatter values in (b) are computed across seeds $\{0, ...19\}$, solid lines are means, shaded regions are ±standard deviations. 20,000 iterations, 4x subsampling, $L = 0.3$, $\sigma_n = 1$, $D = 16$.

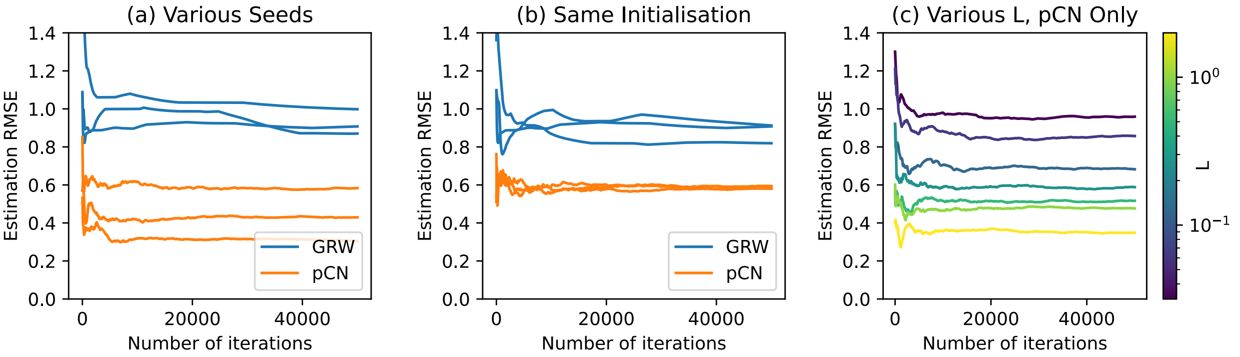


Figure 6: Convergence of estimation RMSEs, compared between algorithms and over L. 50,000 iterations, 4x subsampling, L=0.3, σ_n=1, D=16, β=0.2 and 10x thinning. (a,b) use seeds $\in \{0, 1, 2\}$, (c) uses seed = 0.

Part Ib - Simulation Classification

A discrete version of the previous task can be formulated by applying a probit transform to $\boldsymbol{u}$ and the observed $\boldsymbol{v}$.

Fig 7 shows predictions made for various guessed GP length scales given $D = 16$ and 4x subsampling, given the same prior sample and observations. The figure demonstrates the importance of knowing the correct length scale L when running the MCMC: predictions that use guessed l values differing by only half an order of magnitude give predictions that hardly resemble the original data. This raises the important question: Can the true length scale be inferred by minimising prediction error?

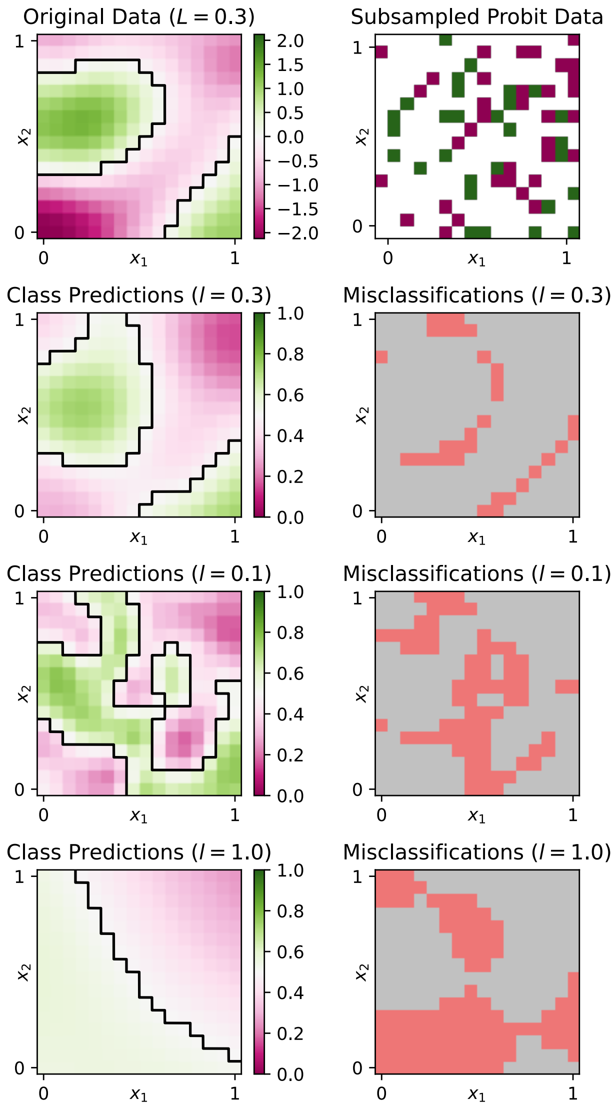


Figure 7: Predicted classifications and their performances for various l. 20,000 iterations, 4x subsampling, $L = 0.3$, $\sigma_n = 1$, seed = 0, $D = 16$, $\beta = 0.4$ 10x thinning.

Computational Constraints

The length scale reconstruction problem can be explored by scanning L through the full range of 'interesting' scales, $L \in (2^{-5}, 2^{-4}, ..., 2^1)$, and for each L finding the optimal $l \in (2^{-7}, 2^{-6}, ..., 2^3)$, denoted $\hat{L}$.

The posterior being explored is different to that in Part Ia, changing appropriate MCMC step sizes and run lengths. Fig 8 implies that $\beta \sim 0.4$ is acceptable for all l and L considered, and that although autocorrelation and convergence time are constant for fixed L, they both deteriorate as L decreases, with significantly more than $50,000$ iterations required to have good convergence at $L = 2^{-7}$.

For generalisable conclusions, results must be taken over a number of seeds to obtain a representative range of $\boldsymbol{u}$ samples. Computation time then poses a significant issue if a full 7×11 (L, l) scan is made. Running $10,000$ samples takes ~15-20s on the device used ($D = 16$), so setting N_{its} to e.g. $100,000$ would take ~3hrs per seed. A tradeoff must therefore be made between early termination for small L, and the number of $\boldsymbol{u}$ surfaces examined. The chosen parameters were $N_{its} = 10000$, $N_{seeds} = 16$, under the assumption that this would still give reasonable Monte Carlo estimates for smaller L.

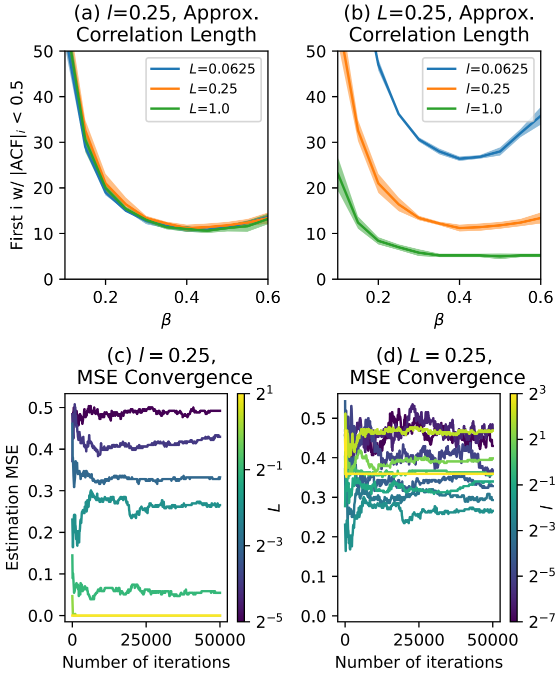


Figure 8: Correlation lengths (a,b) and MSE convergence (c,d) for line scans in (L, l) space, centered at $(0.25, 0.25)$. An intuitive explanation for why ACF only depends on l is that - assuming similar acceptance probabilities - MCMC steps depend only on $\boldsymbol{\zeta} \sim \mathcal{N}(0, C^{(l)})$. For (a,b), solid line is mean over 5 seeds, shaded region is ±1 standard deviation. 10,000 iterations and seeds $\in \{0, ..., 4\}$ used in (a,b); seed = 1, $\beta = 0.4$ and 10x thinning used in (c,d); 4x subsampling, $\sigma_n = 1$ and $D = 16$ used in both.

Visualisation of Results

Scans of the Mean Squared Error (MSE) between predictions $\boldsymbol{t}$ and $\boldsymbol{t}_{true}$ over trial length scales l for $D = 16$ are plotted in Fig 9[1]. Since assignments are to the classes (0, 1), errors are always either 0 or ± 1, so MSE is equivalent to the overall misclassification rate, and is constrained to the range $[0, 1]$.

The typical performance at different L scales can also be visualised by binning these assignments of $\hat{L}$ over many seeds for each L and constructing a 'confusion' heatmap based on these counts, as shown in Fig 10. Clearly the minimum score for a particular $\boldsymbol{u}$ and L does not always correspond to the correct length scale, but is often close, especially for the smaller L values. This is in apparent contradiction to fears about convergence.

Performance Discussion

For very small l, the predictions consistently perform slightly better than 50% classification accuracy. This is because for one quarter of samples, the noisy observations give some information about the true classes, while away from these samples, each grid point's predictions are effectively mutually uncorrelated guesses (and so pull the MSE towards 0.5).

The average performance then improves slightly with increasing l as local correlations kick in to reflect the $\boldsymbol{u}$ field's local structure and reject noise, up to approximately $l = L$. Past this point performance generally deteriorates for a given seed, and prediction variance between seeds increases - this is because there is more opportunity to get the landscape significantly wrong with an inflexible model, while it is still possible for a particular draw of $\boldsymbol{u}$ to vary slowly and be broadly compatible with large length scales. There are two exceptions to this general behaviour.

One exception is for very small L: performance simply deteriorates with increasing l to around 50% accuracy with approximately constant variance between seeds. This is because for any large L, each decision region contains a large number of almost independently drawn true classifications, and so the result is roughly a sum of $D^2/2$ uniform binomial draws, which approaches $\mathcal{N}(\frac{1}{2}, \sigma^2)$ where $\sigma = \frac{1}{16\sqrt{2}} \approx 0.044$ - this matches the value in 9.

Meanwhile, for very large L, a wide range of larger length scales perform equally well (often perfectly). This is because there are many noisy observations relative to the low flexibility of the models in this regime, and because these inflexible models produce very similar $\boldsymbol{u}$ surfaces - often simply two classes separated by a nearly straight dividing line. Therefore it is easy to recover the original $\boldsymbol{u}$, but hard to precisely infer L.

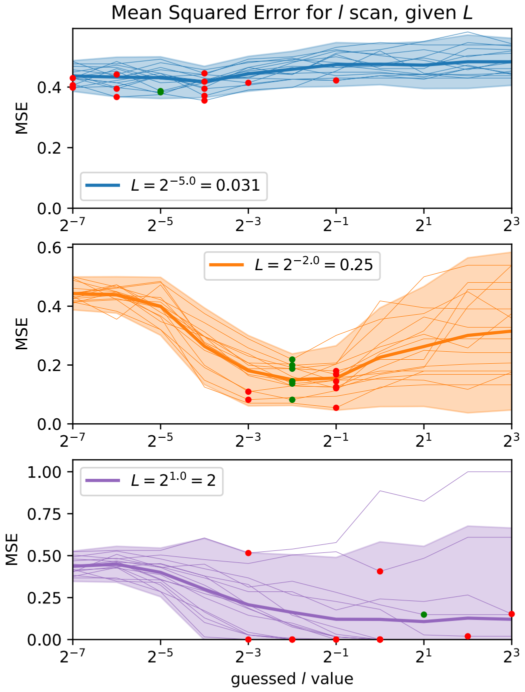


Figure 9: Range of MSEs of inferred classifications, obtained by scanning over l for various fixed L. Solid line is mean over 16 seeds $\in \{0, ..., 15\}$, shaded region is ± 1 standard deviation. 50,000 iterations, 4x subsampling, $\sigma_n = 1$, $\beta = 0.4$, 10x thinning.

Sources of Error

Inability to Differentiate Scale for Large L

As explained above, for large L, a broad range of l is highly effective, making length scale inference difficult. Two things may improve this: either further increasing the subsampling factor to 8x, so that the performance does not 'saturate' with a perfect score, or simply observing a larger region of data such as a 4×4 grid. Comparing Figs 10a and 10b shows that the former only gives minor gains, if any. The latter is mathematically equivalent to keeping the 1×1 grid and reducing L by a factor of 4, so the heatmap would be effectively shifted 2 rows up. This is, of course, not possible in practice if the observations are fixed.

Lack of MCMC Convergence

This has been discussed previously and becomes more severe for smaller L. If the inferred length scale turns out to be small, increasing the number of samples in a smaller search space can remove this effect while satisfying computational constraints. However, Fig 10c indicates although there is a performance gain, it is extremely mild and not computationally cost-efficient unless the sampling run is a one-off calculation.

[1] Note that directly optimising over the likelihoods $p(\boldsymbol{u}|\boldsymbol{t}, l)$ is not possible because the expression is non-analytic (hence the use of MCMC in the first place!).

Misleading Observations

Subsampling the data and adding noise can sometimes simply mean that the optimal length scale is 'misled' to a most probable surface that does not recover $\boldsymbol{u}$, and another length scale can better reject the noise. In fact, sometimes a sampled $\boldsymbol{u}$ from a prior with a a particular length scale L is simply more likely under a different prior with scale L'. These effects are unavoidable for a given data set. However, the probability of error due to subsampling and noise can be decreased respectively by either performing less subsampling, or increasing D so that more observations exist to better represent the true $\boldsymbol{u}$. Both of these lead to significant improvements in performance, and combining them would likely give highly effective length scale inference up to $L \approx 2^{-1}$ (Fig 10d,e).

Overfitting to Noise

The discussion so far has implied that subsampling serves no purpose at all. However, that is only because cases so far have assumed that the true classifications $\boldsymbol{t}_{true}$ are known. In practical application of this model, such as for the Lewisham bike theft data, there may instead only be an observation available. This scenario can be simulated by taking $\boldsymbol{t}' = \text{probit}(\boldsymbol{u}') = \text{probit}(\boldsymbol{u} + \boldsymbol{\epsilon}')$, with $\epsilon_i' \sim \epsilon_i \sim \mathcal{N}(0, 1)$. Length scale inference then becomes more difficult because small l values will always fit best to the noisier, faster-varying $\boldsymbol{u}'$ field (Fig 10f). Subsampling by taking $\boldsymbol{v} = G\boldsymbol{u}'$ is then extremely helpful, as it highlights the increased predictive power of the model for l by *testing* it on the withheld data (Fig 10g-i), heavily penalising poor models with very small l.

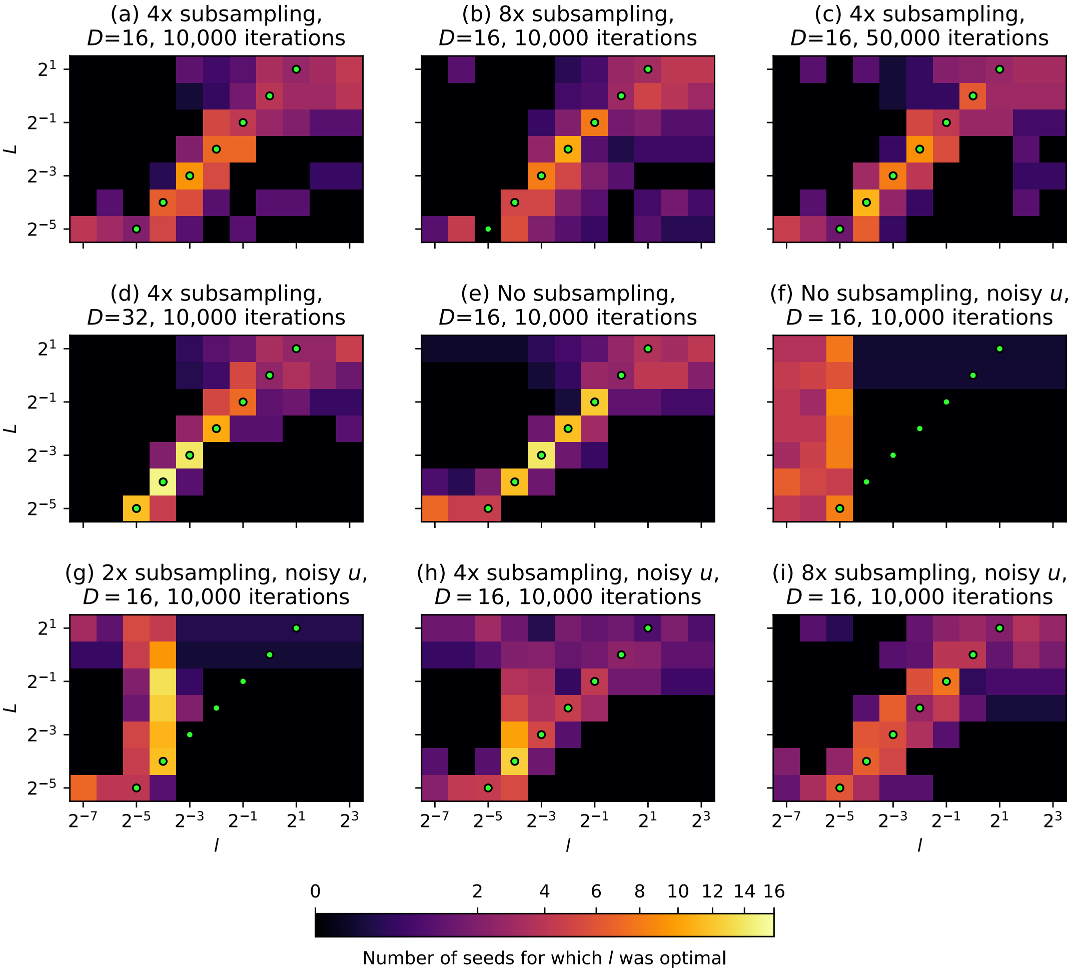


Figure 10: Confusion heatmaps for length scale reconstruction, constructed using the method described in the text. Note that the colormap is nonlinear (following a square root power law): this is so that cell values can be easily differentiated in all plots. The correct values of l for each row are highlighted with a green dot. Plots (a-e) use $\boldsymbol{t}_{true}$ as a reference for calculating scores, while plots (f-i) use $\boldsymbol{t}'$ derived from a noisy version, $\boldsymbol{u}'$. All plots use $\sigma_n = 1$, $\beta = 0.4$, seeds $\in \{0, ..., 15\}$, 10x thinning.

Part II - Spatial Data

For this inference problem, the available data is modelled as a realisation of an assumed latent Poisson rate field $\boldsymbol{\theta}_{true}$, and the goal is to infer this field and its length scale L.

Applying the Poisson model with the pCN algorithm to the (subsampled) Lewisham data at various length-scales results in Fig 12. Convergence is assessed by taking the RMSE between the predicted count expectations and the observed counts, and autocorrelation lengths imply that a reasonable step size is $\beta = 0.16$ (Fig 11), with around 100,000 iterations required at the shortest length scales for satisfactory convergence.

Since the original data is a *sample* and not the underlying rates, assessment of model fit can be aided by drawing samples using the inferred rates and comparing them in various ways: for example, directly visualised (Fig 12), by assembling an overall histogram of drawn counts and compared to PMFs implied by inferred fields (Fig 13), or by comparing the RMSE between an inferred rate field and samples from it to the RMSE between the data it claims to be responsible for to diagnose possible overfitting (Fig 14).

It is immediately clear from the error bars of Fig 13 and visually from Fig 12 that very large length scales do not model the data well, smoothing the rate field too much so that extreme low and high counts are underrepresented and middling counts are too frequent.

Very small length scales, on the other hand, face the issues described at the end of Part Ib. When subsampling is performed, posterior marginals rapidly 'relax' to $\theta_n \sim \text{Lognormal}(0,1)$ away from data, and even at observed coordinates, the inference essentially reduces to N independent 1D inference problems with a single observation. This results in low confidence posteriors that - despite using Bayes' rule - overfit to the data by ignoring local correlations, and are also biased to the prior, which (though reasonable) is somewhat arbitrarily chosen. For greater l, the model can potentially take advantage of local correlations to gain a higher number of *effective* observations.

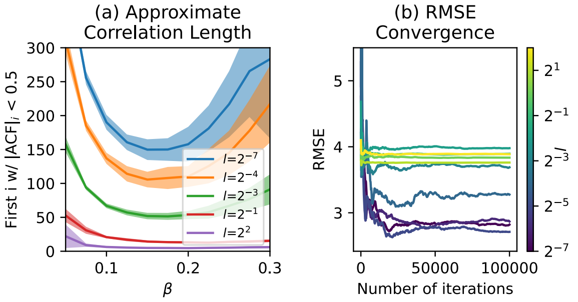


Figure 11: Autocorrelation lengths and RMSE convergence over l. Solid line in (a) is mean over seeds $\{0,...,4\}$, shaded region is ± 1 standard deviation. 50,000 iterations used in (a), $\beta = 0.16$ used in (b), 3x subsampling used in both.

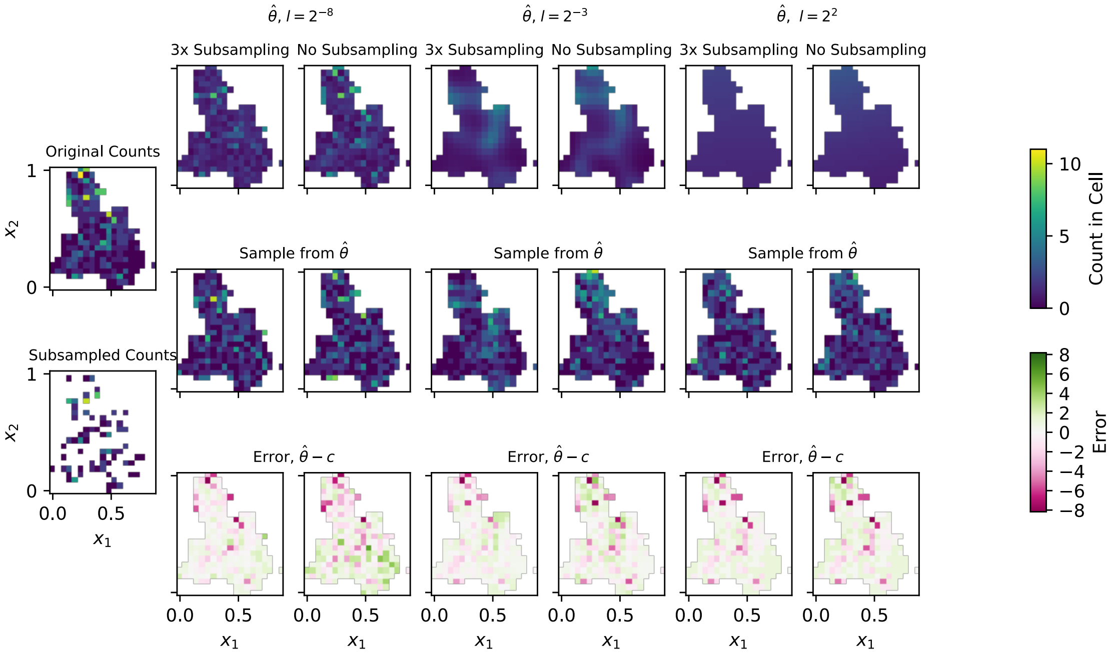


Figure 12: Visualisation of inferred rate fields $\hat{\boldsymbol{\theta}}$ and comparison to observed counts $\boldsymbol{c}$ by sampling from these fields and by calculating per-cell errors $\hat{\boldsymbol{\theta}} - \boldsymbol{c}$. Seed = 0, 3x subsampling, $\beta = 0.16$.

Inferring Length Scale

Looking at the overall RMSEs and the RMSEs at test (withheld) locations in Fig 14 is inconclusive for inferring l - both are very flat and noisy.

Useful values of L can be constrained somewhat. The point at which RMSE at training (observed) locations crosses below the inherent noise floor of the typical RMSE between a rate field and its samples can be considered a lower bound on L of approximately 2^{-3} if a good predictive model is desired[2]. Additionally, the fact that at large l the test and training RMSE are not near-identical excludes the possibility of the true L being approximately 2^0 or above, as this has been seen to strongly constrain field values at neighbouring locations.

1-dimensional heatmaps generated for each subsample factor in the same manner as Fig 10 do not narrow down the range of L much further, although the 4x subsampling heatmap weakly suggests a value of $L \approx 2^{-1} = 0.5$. One possible way to refine this prediction further would be to simulate fields using different L to generate 2-dimensional heatmaps as in 10, then attempt to determine which row the heatmaps in 14 correspond most closely to. However, experience from the classification problem suggests that more data (either a greater area or more refined mesh) is required.

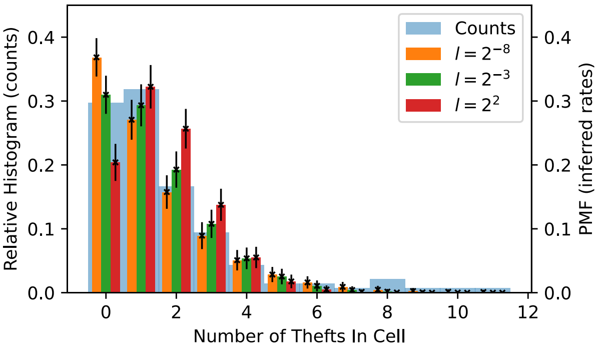


Figure 13: Histogram over counts c_n, compared to histograms constructed by sampling from $\hat{\boldsymbol{\theta}}$ and calculating mean (bar height) and standard devation (black error bars). Seed = 0, 3x subsampling, $\beta = 0.16$.

Unfortunately, even the lower bound of 2^{-3} is unable to reproduce the isolated high counts seen in the data (Fig 12). This implies that either the true scale is smaller and the model is not useful at this resolution, or that the Poisson model is inappropriate and an overdispersed cousin should be used. Ignoring these outliers, $L = 2^{-3}$ (1.05km in real terms) appears a reasonable length scale to use for the model, but predictions should be regarded with scepticism.

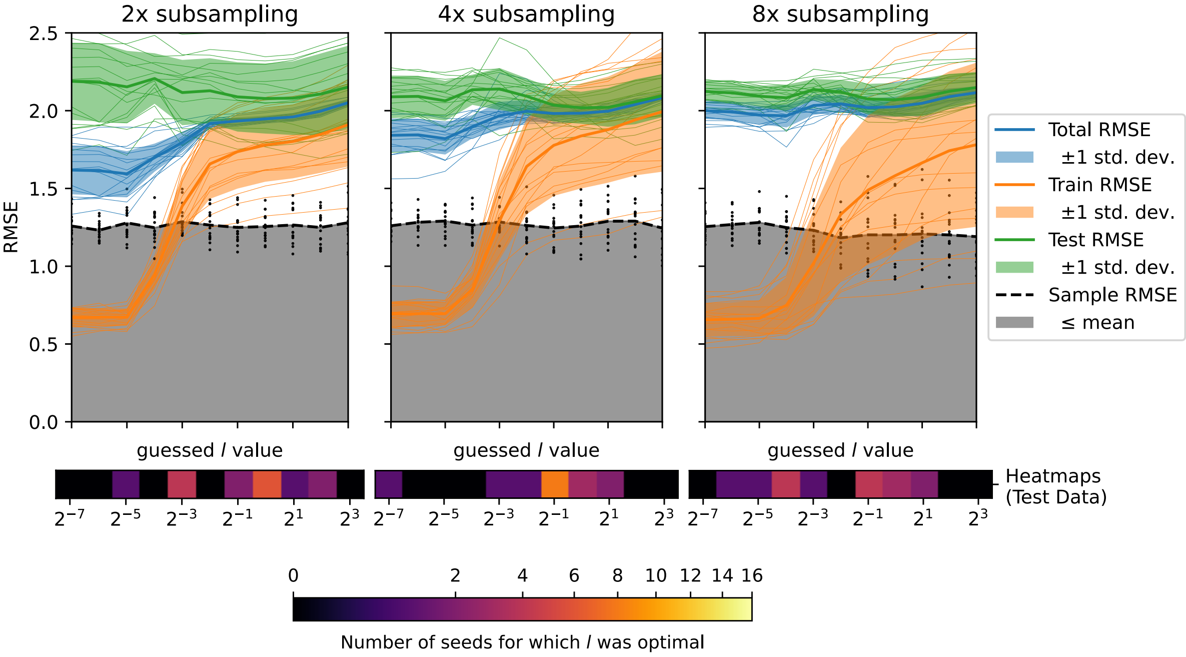


Figure 14: Total, training, and test RMSEs between inferred $\hat{\boldsymbol{\theta}}$ and $\boldsymbol{c}$, plotted alongside RMSEs between samples drawn from each rate field $\hat{\boldsymbol{\theta}}$ and the rates themselves, for various subsampling factors, scanned over l. Heatmaps are generated as in Fig 10, but only considering test locations. Seeds $\in \{0, ..., 15\}$, $\beta = 0.16$.

[2] Note that this does, however, not exclude the possibility that the *true* value of L is below 2^{-3}.