4M24 Computational Statistics and Machine Learning
Coursework: High-Dimensional MCMC

Candidate Number: 5746G

January 15, 2021

Part 1- Simulation

A Gaussian Process (GP) defined on a two-dimensional domain $x_1, x_2 \in [0,1]$ with realisations defined as $\mathbf{u} \sim \mathcal{N}(0, C)$ where the elements of the covariance matrix $C_{i,j} = k(\mathbf{x}_i, \mathbf{x}_j)$ and k is a squared-exponential (SE) covariance function parametised by a length-scale l:

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

For this choice of covariance function, the covariance between variables is near unity when their corresponding inputs are close, and decreases as their separation in the input space increases. The GP prior is defined by the coordinates of the latent variables $\{\mathbf{x}_i\}_{i=1}^{N}$, which are placed on a $D \times D$ grid, and the observation model is defined as:

$$\mathbf{v} = G\mathbf{u} + \boldsymbol{\epsilon} \tag{2}$$

where the matrix G randomly subsamples the latent variable coordinate locations, and the observation noise $\boldsymbol{\epsilon} \sim \mathcal{N}(0, I)$.

Question (a)

Samples from the N dimensional GP prior $\mathbf{u} \sim \mathcal{N}(0, C)$ with zero mean and covariance matrix C were generated through use of a stable Cholesky decomposition. The Cholesky decomposition factorises the covariance matrix into a product of a lower triangular matrix and its transpose:

$$C = LL^T. \tag{3}$$

This decomposition can only be performed on strictly positive definite matrices. Hence, a small diagonal matrix was added to the covariance matrix to improve its numerical conditioning[1] and ensure it satisfies this constraint. The sample from $\mathbf{u}$ is then generated as:

$$\mathbf{u} = L\mathbf{z} \tag{4}$$

where $\mathbf{z} \sim \mathcal{N}(0, I)$. Thus, the covariance of $\mathbf{u}$ is:

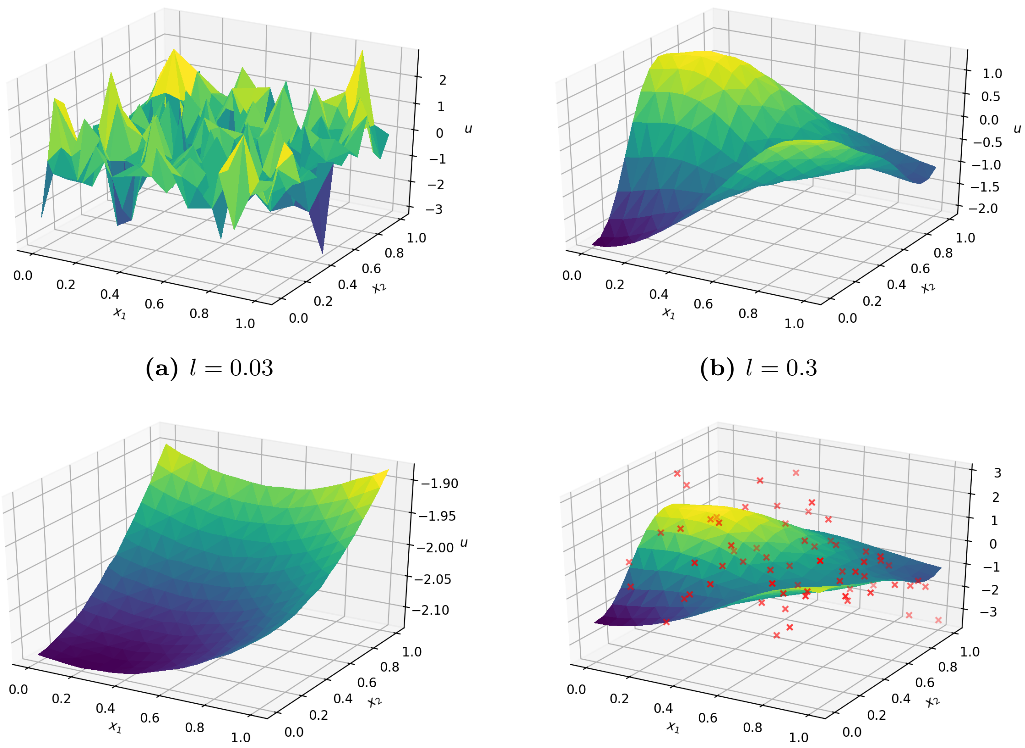
$$\mathbb{E}\left[\mathbf{u}\mathbf{u}^T\right] = \mathbb{E}\left[L\mathbf{z}\mathbf{z}^T L^T\right] = L\mathbb{E}\left[\mathbf{z}\mathbf{z}^T\right] L^T = LIL^T = C. \tag{5}$$

The specification of the hyperparameter l of the prior is significant, as it fixes the properties of the functions adopted for inference. Figure 1 plots the GP surface for three different length scales $l = \{0.03, 0.3, 3\}$ and 256 latent variables $\{\mathbf{x}_i\}_{i=1}^{256}$ placed on a 16×16 grid, with a subsample factor of 4.

The functions for all three length-scales are smooth and stationary. These are properties induced by the SE covariance function of the Gaussian process. As the covariance function is infinitely differentiable, the GP defined by this covariance function has convergence in mean square for derivatives of all order, and is therefore very smooth. Furthermore, the covariance function is isotropic, and so the two components of the latent variable $\mathbf{x}_i = [x_{i,1}, x_{i,2}]^T$ are treated with equal importance. Hence, the distance needed to move in the input space for the GP prior values to become uncorrelated is the same along either axis.

Short length-scales result in the covariance between variables decreasing at a fast rate as the separation between their corresponding inputs increases. Hence, the SE covariance functions vary quickly, giving rise

[1] [4] refers to this as adding "jitter" in the context of Markov chain Monte Carlo (MCMC) based inference; in his work the latent variables $\mathbf{u}$ are explicitly represented in the Markov chain, which makes addition of jitter difficult to avoid.



(a) $l = 0.03$

(b) $l = 0.3$

(c) $l = 3$

(d)

Figure 1: Panels (a)-(c) plot the GP surface for 3 different length-scales. Panel (d) plots the prior sample $\mathbf{u}$ (surface) generated for a characteristic length-scale $l = 0.3$ with observed data $\mathbf{v}$ (points) overlaid on top.

to GP priors which change rapidly between points close in the input space. Increasing the characteristic length-scale l results in the covariance between variables decreasing at a slower rate. Since the covariance function varies less quickly, the GP prior changes less rapidly, and so the GP surface is smoother.

Figure 1(d) plots the latent variable field $\mathbf{u}$ with the observations $\mathbf{v}$ overlaid on top. The effect of the additive Gaussian noise component $\boldsymbol{\epsilon}$ can clearly be seen; many of the observations deviate from the surface of the GP prior.

Question (b)

For a GP prior with probability density function $p(\mathbf{u}) = \mathcal{N}(0, C)$ the log-prior is given by:

$$\log(p(\mathbf{u})) = -\frac{1}{2}\left(\log(2\pi) + \log|C| + \mathbf{u}^T C^{-1}\mathbf{u}\right). \tag{6}$$

Since the matrix C is ill-conditioned, its determinant and inverse are computed efficiently by once again making use of the Cholesky decomposition $C = LL^T$:

$$|C| = |LL^T| = |L|^2; \quad C^{-1} = (LL^T)^{-1} = (L^T)^{-1}L^{-1}.$$

From the additive Gaussian noise observation model $\mathbf{v} = G\mathbf{u} + \boldsymbol{\epsilon}$, the likelihood is defined as the probability of the observed data set $\mathbf{v}$ given the configuration of the prior $\mathbf{u}$. The linear model equation is simply a vector change of variables from $\boldsymbol{\epsilon}$ to $\mathbf{x}$. Conditioning on $\mathbf{u}$, $G\mathbf{u}$ can be treated as a constant. Hence, the change of variables involves a transformation with unity Jacobian:

$$p(\mathbf{v} \mid \mathbf{u}) = p_{\boldsymbol{\epsilon}}(\boldsymbol{\epsilon})\big|_{\boldsymbol{\epsilon}=\mathbf{v}-G\mathbf{u}} = \mathcal{N}(\mathbf{v} - G\mathbf{u}; 0, I) = \mathcal{N}(\mathbf{v}; G\mathbf{u}, I). \tag{7}$$

The log-likelihood is then:

$$\log\left(p(\mathbf{v} \mid \mathbf{u})\right) = -\frac{1}{2}\left(\log(2\pi) + (\mathbf{v} - G\mathbf{u})^T(\mathbf{v} - G\mathbf{u})\right). \tag{8}$$

The posterior distribution μ is absolutely continuous with respect to a Gaussian dominating measure. Choosing this measure to be the prior μ_0, a centred Gaussian measure on $\mathcal{N}(0, C)$, the Radon–Nikodym derivative of the posterior distribution with respect to the prior Gaussian density is just a re-expression of Bayes' formula:

$$\frac{d\mu}{d\mu_0}(u) \propto \mathrm{L}(v \mid u) = \exp(-\Phi(v)) \tag{9}$$

for the likelihood L and real valued potential Φ.

Samples were drawn from the posterior distribution $p(\mathbf{u} \mid \mathbf{v})$ using two Markov chain Monte Carlo (MCMC) algorithms: Gaussian random walk Metropolis Hastings (GRW-MH) and preconditioned Crank-Nicolson (pCN). The two methods are very similar, with only a slight difference in the proposal steps. The GRW proposal is a centred random walk:

$$\mathbf{z}^{(k)} = \mathbf{u}^{(k)} + \beta \boldsymbol{\xi}^{(k)}, \tag{10}$$

where $\boldsymbol{\xi}^{(k)} \sim N(0, \mathcal{C})$, whereas the pCN proposal is an autoregressive (AR) process of order 1 [1]:

$$\mathbf{z}^{(k)} = (1 - \beta^2)^{1/2} \mathbf{u}^{(k)} + \beta \boldsymbol{\xi}^{(k)}, \tag{11}$$

where again $\boldsymbol{\xi}^{(k)} \sim N(0, \mathcal{C})$.

For both methods, 10000 samples were drawn using a fixed step-size parameter $\beta = 0.2$. Figure 2(a) plots the the mean of the inferred $\mathbf{u}$ alongside the observation data $\mathbf{v}$ for samples obtained from the GRW algorithm, and Figure 2(b) the absolute error field between the original $\mathbf{u}$ and the inferred $\mathbf{u}$. Figures 3(a) and 3(b) plot the same quantities, but for the samples generated from the pCN method.

Comparing Figures 2(b) and 3(b), the topology of the error fields is similar between the GRW and pCN methods, indicating similar levels of predictive performance. However, the execution time of the pCN algorithm was significantly lower than that of the GRW; the pCN method ran nearly ten times more iterations per second than the GRW method. Together, these two points suggest that the pCN method is the superior choice of MCMC algorithm for this inference problem.

The acceptance probability of the GRW method is given by:

$$a(u, z) = \min\{1, \exp(I(u) - I(z))\}, \tag{12}$$

where:

$$I(u) = \Phi(u) + \frac{1}{2} \left\| C^{-1/2} u \right\|^2. \tag{13}$$

Denoting the i-th eigenvalue of $C^{-1/2}$ by $1/\lambda_i$, the i-th coordinate of u is normally distributed with zero-mean and variance 1. Therefore, $\lambda_i^{-1} u_i \sim \mathcal{N}\left(0, \lambda_i^{-2} \lambda_i^2\right) = \mathcal{N}(0, 1)$. The expectation of the norm term is calculated as:

$$\mathbb{E}\left[\left\| C^{-1/2} u \right\|^2\right] = \mathbb{E}\left[\sum_{i=1}^{N} \left(\lambda_i^{-1} u_i\right)^2\right] = \sum_{i=1}^{N} \mathbb{E}\left[\left(\lambda_i^{-1} u_i\right)^2\right] = \sum_{i=1}^{N} 1, \tag{14}$$

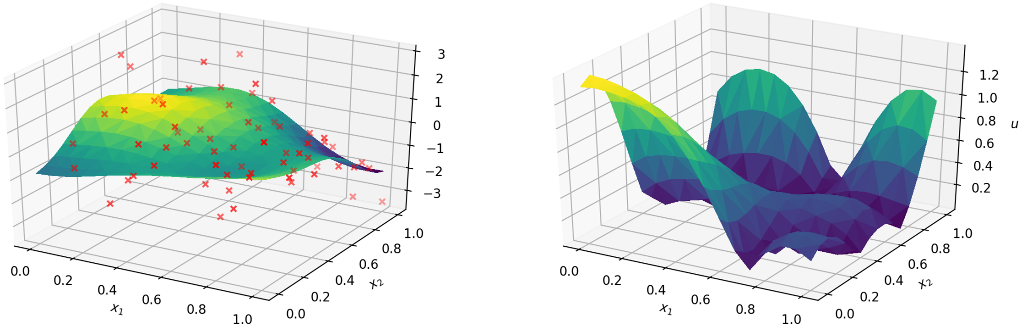
which diverges as the dimensionality D of the space increases. Since the quantity $I(u)$ is not well defined in an infinite dimensional space, the number of iterations required for this method to converge diverges with D, a phenomenon termed "the curse of dimensionality" [6].

For the alternative proposal adopted in the pCN method, if $u \sim N(0, C)$, then the proposal is also $z \sim N(0, C)$. Thus, the prior measure is preserved, and the acceptance probability is:

$$a(u, z) = \min\{1, \exp(\Phi(u) - \Phi(z))\}. \tag{15}$$

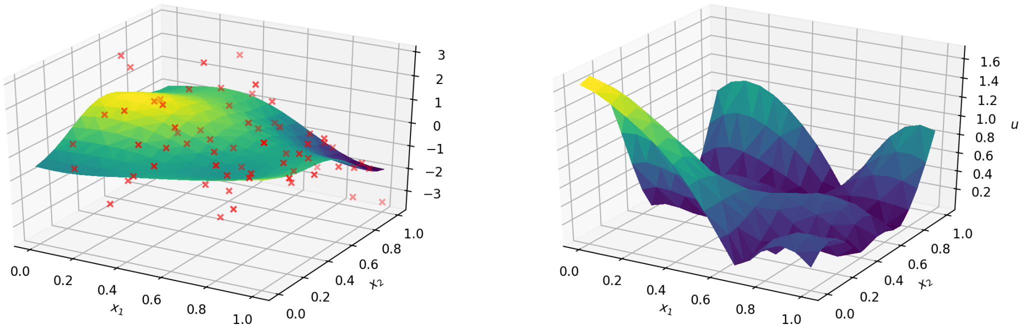
This is a simpler acceptance probability compared to the GRW expression given in eq. (12), and one that is well defined in an infinite dimensional space. The acceptance probability is defined entirely based on differences in log density, as is the case in finite dimensions for the standard GRW if the density is specified with respect to the Lebesgue measure. To this extent, the pCN proposal can be interpreted as the generalisation of the random walk proposal to the setting where the target measure is defined via a density taken with respect to a Gaussian measure [1].

Though there is only a small change between the GRW and pCN proposal steps, the reformulation of the acceptance probability to being well defined in an infinite dimensional space results in significant performance gains for the GP inference problem on the $D \times D$ discretised grid. Table 1 shows the acceptance rates of the GRW and pCN methods for low and high-dimensional latent variable spaces. 10000 samples were drawn from each algorithm using a fixed step-size parameter $\beta = 0.2$, and the dimensionality of the observed data $\mathbf{v}$ held constant. Increasing the dimensionality of the latent variable space from $D = 4$ to $D = 16$



(a) **(b)**

Figure 2: Panel (a) plots the mean of the inferred $\mathbf{u}$ (surface) obtained through sampling $p(\mathbf{u} \mid \mathbf{v})$ using the GRW algorithm, alongside the observed data $\mathbf{v}$ (points). Panel (b) plots the absolute error field between the original GP prior surface $\mathbf{u}$ and the $\mathbf{u}$ inferred from the GRW samples.



(a) **(b)**

Figure 3: Panel (a) plots the mean of the inferred $\mathbf{u}$ (surface) obtained through sampling $p(\mathbf{u} \mid \mathbf{v})$ using the pCN method, alongside the observed data $\mathbf{v}$ (points). Panel (b) plots the absolute error field between the original GP prior surface $\mathbf{u}$ and the $\mathbf{u}$ inferred from the pCN samples.

Table 1: GRW and pCN acceptance rates for a low and high-dimensional latent variable space.

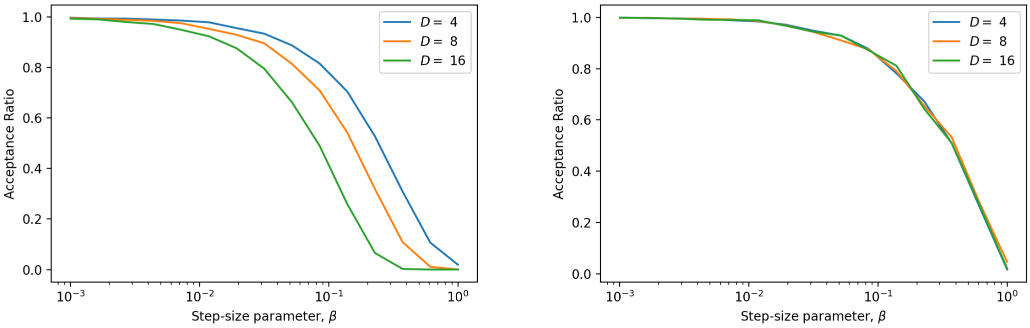
D	GRW	pCN
4	0.587	0.707
16	0.105	0.703

reduces the acceptance ratio of the GRW method by 82.1%, eluding to the fact that the standard GRW becomes arbitrarily slow under increasing dimensionality. On the contrary, there is only a 0.566% decrease in the average acceptance probability of the pCN method, demonstrating its ability to work effectively in high-dimensional spaces.

This point is further highlighted by Figures 4(a) and 4(b), which plot the acceptance ratio as a function of step-size parameter β with increasing dimensionality D for the GRW and pCN methods. For the GRW method, shown in Figure 4(a), the acceptance ratio curves shift to the left as the dimensionality of the grid is increased. The practical implication of this is that smaller proposal variances are necessary to yield the same acceptance probability as the grid-spacing is refined, and a greater number of steps are required to represent the function, thus demonstrating that the standard GRW method fairs poorly with increasing dimensionality.

On the other hand, for the pCN method shown in Figure 4(b), there is a limit on the acceptance ratio curves as the dimensionality of the space is increased. Hence, a fixed proposal variance is sufficient to obtain the same acceptance probability as the grid-spacing is refined. Since the same number of steps is required over all dimensions, this method is robust to increasing dimensionality.

The effect of the step-size parameter β is understood by examining its role in the proposal step. With β too small, the algorithm accepts proposed states often, but these changes in state are too small, so the algorithm does not explore the state space efficiently. In contrast, with β too big, larger jumps are proposed,



(a) GRW

(b) pCN

Figure 4: Acceptance ratio as a function of step-size parameter β for increasing dimensionality D. Panel (a) plots the curves obtained using the GRW method, and panel (b) for the pCN method.

but are often rejected since the proposal often has small probability density and so are often rejected.

Question (c)

The observation data $\mathbf{v}$ is passed through a *probit transform* to yield the vector $\mathbf{t}$, the i-th component of which is given by $t_i = \text{sign}(v_i)$. The likelihood for this component takes the form:

$$\begin{aligned} p(t_i = 1 \mid \mathbf{u}) &= p(v_i > 0 \mid \mathbf{u}) = p([G\mathbf{u} + \boldsymbol{\epsilon}]_i > 0 \mid \mathbf{u}) && (16)\\ &= p(\epsilon_i > -[G\mathbf{u}]_i \mid \mathbf{u}) = p(\epsilon_i < -[G\mathbf{u}]_i \mid \mathbf{u}) && (17)\\ &= \Phi([G\mathbf{u}]_i), && (18)\end{aligned}$$

where the cumulative probability function $\Phi(z) = \int_{-\infty}^{z} \mathcal{N}(x; 0, 1)dx$. By the independence property, the full form of the likelihood is derived as:

$$\begin{aligned} p(\mathbf{t} \mid \mathbf{u}) &= \prod_{i=1}^{N} p(t_i \mid \mathbf{u}) = \prod_{i=1}^{N} p(t_i = 1 \mid \mathbf{u})^{t_i} p(t_i = 0 \mid \mathbf{u})^{1-t_i} && (19)\\ &= \prod_{i=1}^{N} \Phi([G\mathbf{u}]_i)^{t_i} \left(1 - \Phi([G\mathbf{u}]_i)\right)^{1-t_i}. && (20)\end{aligned}$$

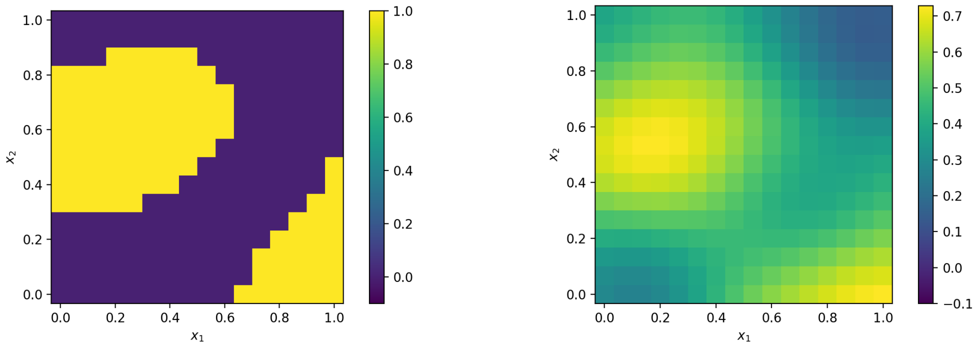
And the log-likelihood:

$$\log\left(p(\mathbf{t} \mid \mathbf{u})\right) = \sum_{i=1}^{N} \Big(t_i \log\left(\Phi([G\mathbf{u}]_i)\right) + (1 - t_i) \log\left(1 - \Phi([G\mathbf{u}]_i)\right)\Big). \quad (21)$$

Samples of the posterior distribution $p(\mathbf{u} \mid \mathbf{t})$ were generated using the pCN method, and the the true class assignments $\mathbf{t}_{\text{true}} = \text{probit}(\mathbf{u})$ predicted using a Monte-Carlo (MC) estimate of the posterior predictive distribution:

$$\begin{aligned} p\left(t_i^* = 1 \mid \mathbf{t}\right) &= \int p\left(t_i^* = 1, \mathbf{u} \mid \mathbf{t}\right) d\mathbf{u} = \int p\left(t_i^* = 1 \mid \mathbf{u}\right) p(\mathbf{u} \mid \mathbf{t}) d\mathbf{u} && (22)\\ &= \int \Phi(u_i) p(\mathbf{u} \mid \mathbf{t}) d\mathbf{u} = \mathbb{E}_{p(\mathbf{u}|\mathbf{t})}\left[\Phi(u_i)\right] && (23)\\ &\approx \frac{1}{n} \sum_{j=1}^{n} \Phi\left(u_i^{(j)}\right), && (24)\end{aligned}$$

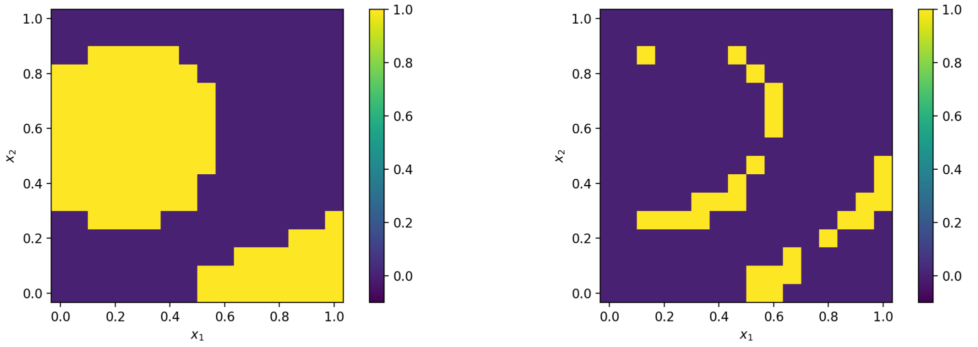
where $\mathbf{u}^{(j)} \sim p(\mathbf{u} \mid \mathbf{t})$, and n is the number of pCN samples. Figures 5(a) and 5(b) plot the true probit assignments of the original data and the predictions made using the MC estimate of the posterior predictive distribution. Comparing these two fields, the inferred field largely captures the distribution of the true field; there are two regions in the field of predictive assignments which are assigned high probability of being 1, matching that of the true field. The predictive uncertainty is lower in the centre of these regions, the predictive probabilities $p\left(t_i^* = 1 \mid \mathbf{t}\right)$ taking values close to 1. Moving away from the centres, the predictions reduce to near 0.5, indicating an increase in the predictive uncertainty. In the top right and bottom left corners, far from the regions assigned $t_i = 1$, the inferred field predicts $t_i^* = 0$ with high certainty.



(a)

(b)

Figure 5: Panel (a) plots the true probit assignments of $\mathbf{u}$ obtained by performing a probit transform of the original data. Panel (b) plots the predictions of the true class assignments found using a Monte-Carlo (MC) estimate of the predictive distribution.



(a)

(b)

Figure 6: Panel (a) plots the hard assignments obtained by thresholding the predicted class assignments in Figure 5(b). Panel (b) plots the absolute error field between the original probit assignments, shown in Figure 5(a), and the hard assignments of the predictive distribution.

Question (d)

Hard assignments were made by thresholding the predicted class assignments at $p(t^* = 1 \mid \mathbf{t}) = 0.5$. These assignments were compared to the true assignments to give the mean absolute error per cell $\mathcal{L}$:

$$\mathcal{L}(\mathbf{t}, \mathbf{t}^*) = \frac{1}{N} \sum_{n=1}^{N} |t_i - t_i^*|. \tag{25}$$

Figure 6(a) plots the hard assignments made by thresholding true class assignments predicted using the MC estimate of the posterior predictive distribution. Figure 6(b) plots the corresponding error field between this thresholded field and the original probit data. The absolute errors are zero for much of the domain. These regions align with where the true class assignments in Figure 5(b) were predicted with high certainty. Errors occur where the inferred field made predictions with low certainty, namely on the boundaries between areas where the true class assignments change from 1 to 0. For this inferred field, the mean absolute error per cell was 0.128.

Up until this point, the length-scale parameter was given to the model. In an alternative approach, the length-scale l was found through optimisation, by minimising the prediction error $\mathcal{L}$ with respect to l. The search domain was restricted to $l \in [0.01, 10]$, and a line-search performed. With no access to first or second-order derivative information, a gradient-based search was not possible. Instead, an interval reduction method was employed, in which the search interval was iteratively reduced by means of bracketing. The same proportion of solution spacing was maintained throughout the algorithm by imposing a constant reduction factor $\alpha = \frac{\sqrt{5}-1}{2}$ for the interval length. This value of α is the inverse of the golden ratio, giving this search-method its name of *the golden-section search* [2].

Figure 7 plots the trajectory of the solutions found for a golden-section search applied to the minimisation

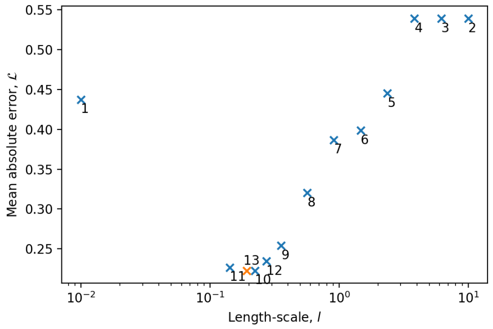


Figure 7: Trajectory of the solutions found for the minimisation of the prediction error $\mathcal{L}$ with respect to the length-scale l by a golden-section search. The first 12 solutions are shown in blue. The final, optimised length-scale $l = 0.192$ occurs at the 13-th solution, and is shown in orange.

of the prediction error with respect to l. The final, optimised length-scale is given by $l = 0.192$. One potential reason why this optimised length-scale differs from the true length-scale $l = 0.3$ used to generate the data is the fact that the prediction error is taken over the entire data set. The entire data set is constructed as the union of the subsampled data, which forms the training set, and any remaining data, which forms the test set. The model may have fit the noise in the training set (over-fit), leading to low training error but poor generalization performance. The generalisation error, found as the mean prediction error in the validation set, may therefore be a more appropriate loss function to minimise with respect to l, as it is indicative of generalisation performance.

A Bayesian approach provides an alternative means of finding the optimal length-scale l. This involves the computation of the probability of the model given the data, based on the marginal likelihood. The marginal likelihood Z is the normalising term in the posterior distribution $p(\mathbf{u} \mid \mathbf{t})$:

$$Z = p(\mathbf{t}) = \int p(\mathbf{u})p(\mathbf{t} \mid \mathbf{u})d\mathbf{u} \tag{26}$$

It is this integral over the parameter space which distinguishes the Bayesian scheme of inference from the optimisation approach: the marginal likelihood has the property that it automatically incorporates a trade-off between model fit and model complexity [5]. The marginal likelihood in eq. (26) can be maximised with respect to the hyperparameter l, a technique termed type II maximum likelihood (ML-II), to yield the optimal value of l. Due to the form of the probit likelihood, this integral is not analytically tractable, meaning an analytical approximation or MCMC method is required. Choices for analytical approximations include a Laplace approximation [7], which involves finding the mode of the posterior distribution and then fitting a Gaussian centred at that mode, or expectation propagation (EP) [3], in which the likelihood is approximate by a local likelihood approximation with parameters which are iteratively updated.

Part 2- Spatial Data

The dataset `cw_data.csv` contains a list of x, y coordinates of $400m^2$ cells dividing up the London borough of Lewisham, and the corresponding number of bike thefts for that area in 2015. From a subsample of this count data, the underlying field $\mathbf{u}$ is inferred on the original coordinates, and performance evaluated based on predictions of the original data.

Question (e)

To ensure positivity, the field $\mathbf{u}$ is mapped to $\boldsymbol{\theta} = \exp(\mathbf{u})$, and $\boldsymbol{\theta}$ used as the Poisson rate for each location. The overall likelihood of the data counts $\mathbf{c}$ is then:

$$p(\mathbf{c} \mid \boldsymbol{\theta}) = \prod_{i=1}^{N} f\left(c_i \mid \theta_i\right) = \prod_{i=1}^{N} \frac{e^{-\theta_i}\theta_i^{c_i}}{c_i!}. \tag{27}$$

The full Poisson log-likelihood is derived as:

$$\log\left(p(\mathbf{c} \mid \mathbf{u})\right) = \log\left(p(\mathbf{c} \mid \boldsymbol{\theta})\right) = \sum_{i=1}^{N} \log\left(f\left(c_i \mid \theta_i\right)\right) \tag{28}$$

$$= \sum_{i=1}^{N} \left(c_i \log \theta_i - \theta_i - \log(c_i!)\right) = \sum_{i=1}^{N} \left(c_i u_i - \exp(u_i) - \log(c_i!)\right). \tag{29}$$

Substituting this log-likelihood into the expression for the pCN acceptance probability in eq. (15), 10000 samples of the posterior $p(\mathbf{u} \mid \mathbf{c})$ were generated using the pCN method with a length-scale parameter of $l = 3$ and fixed step-size parameter $\beta = 0.2$.

Question (f)

The pCN samples drawn from the posterior distribution $p(\mathbf{u} \mid \mathbf{c})$ were used to form a MC estimate of the posterior predictive probability:

$$p\left(c_i^* = k \mid \mathbf{c}\right) = \int p\left(c_i^* = k, \mathbf{u} \mid \mathbf{c}\right) d\mathbf{u} = \int p\left(c_i^* = k \mid \mathbf{u}\right) p(\mathbf{u} \mid \mathbf{c}) d\mathbf{u} \tag{30}$$

$$= \int f\left(c_i^* = k \mid \boldsymbol{\theta}\right) p(\mathbf{u} \mid \mathbf{c}) d\mathbf{u} = \mathbb{E}_{p(\mathbf{u}\mid\mathbf{c})}\left[f\left(c_i^* = k \mid \boldsymbol{\theta}\right)\right] \tag{31}$$

$$\approx \frac{1}{n} \sum_{j=1}^{n} f\left({c_i}^* = k \mid \boldsymbol{\theta}^{(j)}\right) = \frac{1}{n} \sum_{j=1}^{n} \frac{e^{-\theta_i^{(j)}} {\theta_i^{(j)}}^k}{k!}, \tag{32}$$

where $\mathbf{u}^{(j)} \sim p(\mathbf{u} \mid \mathbf{c})$ and $\boldsymbol{\theta}^{(\boldsymbol{j})} = \exp(\mathbf{u}^{(j)})$. From the MC estimate, the inferred expected counts were approximated as:

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

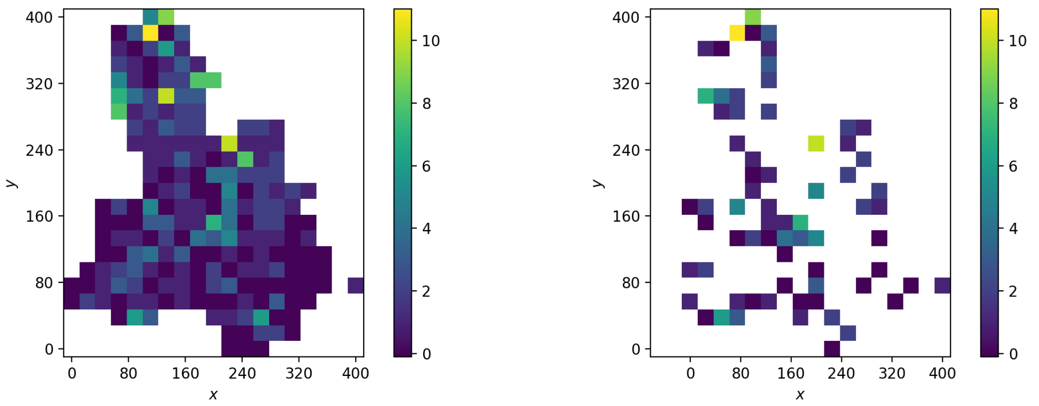
$$\approx \sum_{k=0}^{\infty} k \left(\frac{1}{n} \sum_{j=1}^{n} f\left({c_i}^* = k \mid \boldsymbol{\theta}^{(j)}\right)\right) = \frac{1}{n} \sum_{j=1}^{n} \sum_{k=0}^{\infty} k f\left({c_i}^* = k \mid \boldsymbol{\theta}^{(j)}\right) \tag{34}$$

$$= \frac{1}{n} \sum_{j=1}^{n} \mathbb{E}_{p\left(c_i^* \mid \mathbf{c}\right)}\left[f\left({c_i}^* \mid \boldsymbol{\theta}^{(j)}\right)\right] = \frac{1}{n} \sum_{j=1}^{n} \theta_i^{(j)} \tag{35}$$

Figures 8(a) and 8(b) plot the original count data and subsampled data. Figures 9(a) and 9(b) plot the inferred counts and absolute error between the predicted counts and true count data. There are large clusters where the inferred field predicts the bike count data to be very high or low. These clusters are joined by regions which take intermediate count values, producing a largely smoothly varying field. This in contrast to the true count data; whilst there are some regions where neighbouring cells take similarly high count values, a number of the cells with high count data are bordered by cells with lower counts. Consequently, a lower length-scale parameter is required, such that less information is leveraged between neighbouring cells when inferring the underlying count field.

The model was re-run using two extreme length-scale parameters. Figures 10 and 11 plot the inferred counts and absolute errors for low and large length scales of $l = 0.1$ and $l = 10$ respectively. Regions where neighbouring cells take similarly high count values are lost for the inferred field generated using a low length-scale, counts vary significantly between nearly all of the neighbouring cells. The opposite is true for the high length-scale, for which the distribution of inferred counts varies smoothly across the whole domain. This predicted field does not capture smaller deviations in count data, which can clearly be seen in the original field.

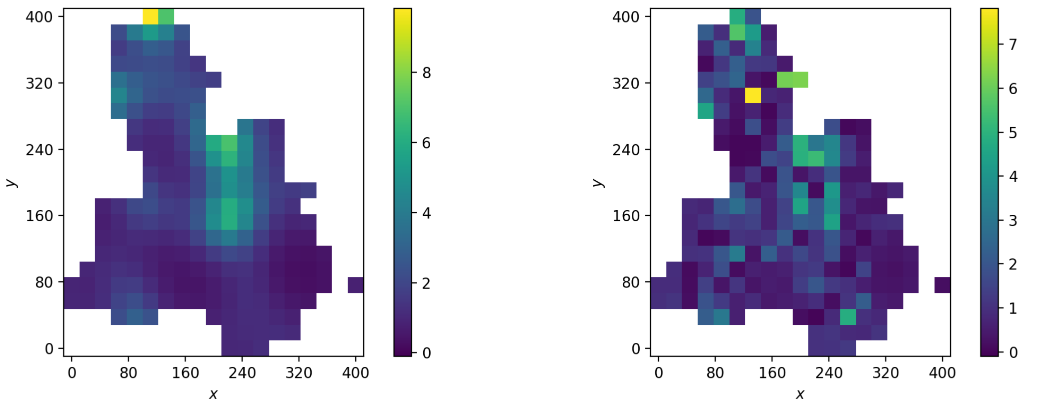
An appropriate length-scale value was found by minimising the mean absolute prediction error with respect to the length-scale parameter through a golden-section search. The trajectory of the solutions are shown in Fig. 12, with the optimised length-scale $l = 0.602$ occurring at the 13-th solution. As with the probit classification problem, the prediction error is formed as the union of the training error and validation error. To remove the impact of fitting to noise in the training set (over-fitting), the generalisation error could be minimised with respect to l. Likewise, the optimisation approach could be replaced with a Bayesian approach, in which the marginal likelihood is maximised with respect to l.



(a)

(b)

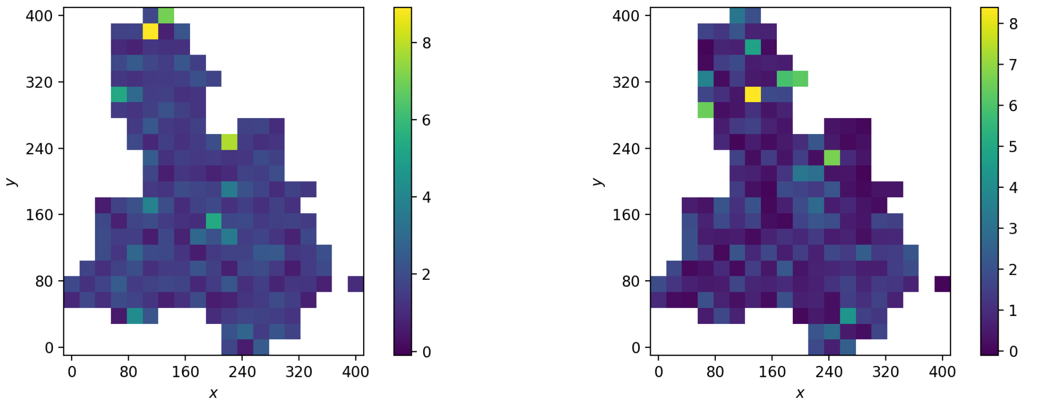
Figure 8: Panel (a) plots the original bike theft count data for data set given in `cw_data.csv`. Panel (b) plots the subsampled count data generated by transforming the original count data by a random observation matrix of the latent variable coordinate locations.



(a)

(b)

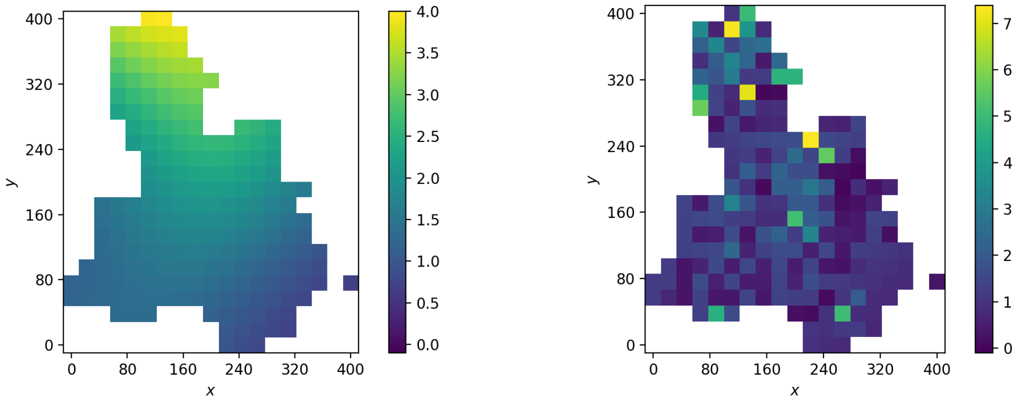
Figure 9: Panel (a) plots the inferred expected counts on the original coordinates for a length-scale parameter $l = 2$. Panel (b) plots the error field (absolute difference between the inferred and original counts) for each cell in this coordinate system.



(a)

(b)

Figure 10: Panel (a) plots the inferred expected counts on the original coordinates for a low length-scale parameter $l = 0.1$. Panel (b) plots the corresponding error field for each cell in this coordinate system.



(a) (b)

Figure 11: Panel (a) plots the inferred expected counts on the original coordinates for a high length-scale parameter $l = 10$. Panel (b) plots the corresponding error field for each cell in this coordinate system.

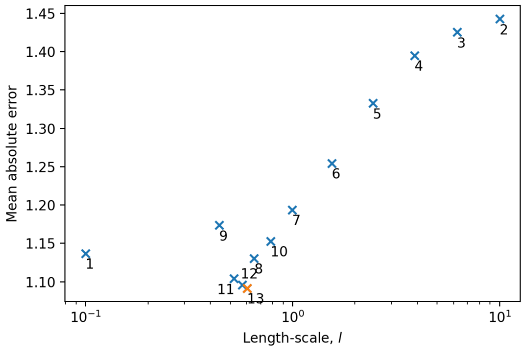


Figure 12: Trajectory of the solutions found for the minimisation of the prediction error with respect to the length-scale l by a golden-section search. The first 12 solutions are shown in blue. The final, optimised length-scale $l = 0.602$ occurs at the 13-th solution, and is shown in orange.

References

[1] S. L. Cotter, G. O. Roberts, A. M. Stuart, and D. White. MCMC Methods for Functions: Modifying Old Algorithms to Make Them Faster. *Statistical Science*, 28(3):424–446, Aug 2013.

[2] J. Kiefer. Sequential Minimax Search for a Maximum. *Proceedings of the American Mathematical Society*, 4(3):502–506, 1953.

[3] Thomas P. Minka and Rosalind Picard. *A Family of Algorithms for Approximate Bayesian Inference*. PhD thesis, Massachusetts Institute of Technology, USA, 2001.

[4] Radford M Neal. Regression and Classification Using Gaussian Process Priors. *Bayesian Statistics*, 6:475–501, 1998.

[5] Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning*. The MIT Press, 2006.

[6] C. R. Taylor. *Applications of Dynamic Programming to Agricultural Decision Problems*. Westview Press, Boulder, 1993.

[7] C. K. I. Williams and D. Barber. Bayesian classification with gaussian processes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(12):1342–1351, 1998.