

NONPARAMETRIC BAYESIAN DENSITY MODELING WITH GAUSSIAN PROCESSES

BY RYAN P. ADAMS*, IAIN MURRAY AND DAVID J.C. MACKAY

University of Toronto and University of Cambridge

We present the *Gaussian process density sampler* (GPDS), an exchangeable generative model for use in nonparametric Bayesian density estimation. Samples drawn from the GPDS are consistent with exact, independent samples from a distribution defined by a density that is a transformation of a function drawn from a Gaussian process prior. Our formulation allows us to infer an unknown density from data using Markov chain Monte Carlo, which gives samples from the posterior distribution over density functions and from the predictive distribution on data space. We describe two such MCMC methods. Both methods also allow inference of the hyperparameters of the Gaussian process.

1. Introduction. We propose a method for incorporating a Gaussian process into a prior on probability density functions. While such constructions have been proposed before [Leonard, 1978, Thorburn, 1986, Lenk, 1988, 1991, Csató, 2002, Tokdar and Ghosh, 2007, Tokdar, 2007], ours is the first that allows a procedure for drawing exact and exchangeable data samples from a density drawn from the prior. We call this prior and the associated procedure the *Gaussian process density sampler* (GPDS). Given data, this generative prior allows us to perform inference of the unnormalised density. We present two Markov chain Monte Carlo (MCMC) algorithms for performing this inference, one based on exchange sampling [Murray et al., 2006] and the other based on inferring the latent generative history. In both cases we are also able to infer the parameters governing the covariance kernel, and draw samples from the predictive distribution on data space.

Bayesian nonparametric inference is appealing because it allows models to include an arbitrary number of parameters, without requiring expensive dimensionality-altering computations for inference. The most popular tool for nonparametric Bayesian modeling of an unknown probability measure is the Dirichlet process [Ferguson, 1973] and related constructions (e.g., Pitman and Yor [1997] and Ishwaran and James [2001]). Samples from the

*Supported by the Canadian Institute for Advanced Research.

AMS 2000 subject classifications: Primary 62G07, 62G07; secondary 62F15

Keywords and phrases: Bayesian nonparametrics, Gaussian process, density estimation

Dirichlet process, however, are discrete distributions with probability one. For many inference problems, we wish to model probabilities on continuous spaces and in such problems our prior beliefs are often best captured by a distribution over probability density functions.

To fill the gap between nonparametric priors on discrete distributions and nonparametric priors on continuous densities, the Dirichlet process is frequently used to add a countably-infinite number of parameters into a continuous model. The most popular example is the infinite mixture of parametric distributions [Escobar and West, 1995], another example is kernel convolution [Lo, 1984]. The Dirichlet diffusion tree [Neal, 2001, 2003] and Pólya trees [Lavine, 1992, 1994] provide more direct nonparametric priors on distributions and, in contrast to the Dirichlet process, can produce densities. All of these priors are based on beliefs of an underlying structure, either a clustering or tree-based hierarchy.

Prior beliefs about a distribution over data are sometimes best expressed directly in terms of the probability density function — its continuity, support and smoothness properties, for example. There is a rich literature on incorporating prior beliefs about functions into nonparametric Bayesian regression models, using splines, neural networks and stochastic processes (e.g., DiMatteo et al. [2001], MacKay [1992], and O’Hagan [1978]). However, priors on general functions have largely resisted application to density estimation, due to the requirements that probability density functions be nonnegative and integrate to one. This work introduces the first fully-nonparametric Bayesian kernel method for density estimation that does not require a finite-dimensional approximation to perform inference.

2. The Gaussian process density sampler prior. The GPDS provides a probability distribution on a space \mathcal{X} , which we call the *data space*. In many problems, \mathcal{X} is the D -dimensional real space \mathbb{R}^D .

We first place a Gaussian process prior over a scalar function $g(\mathbf{x}) : \mathcal{X} \rightarrow \mathbb{R}$. This means that the prior distribution over any discrete set of function values, $\{g(\mathbf{x}_n)\}_{n=1}^N$, is a multivariate normal distribution. These distributions can be consistently defined with a positive definite covariance function $C(\cdot, \cdot) : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ and a mean function $m(\cdot) : \mathcal{X} \rightarrow \mathbb{R}$. The mean and covariance functions are parameterised by *hyperparameters* θ . For a more detailed review of Gaussian processes see, e.g., Rasmussen and Williams [2006].

We construct a map from the function $g(\mathbf{x})$ to a probability density func-

tion¹ $f(\mathbf{x})$ via

$$(2.1) \quad f(\mathbf{x}) = \frac{1}{\mathcal{Z}_\pi[\mathbf{g}]} \Phi(g(\mathbf{x})) \pi(\mathbf{x} | \psi)$$

where $\pi(\mathbf{x} | \psi)$ is a parametric *base density* that corresponds to an arbitrary base probability measure on \mathcal{X} , with hyperparameters ψ . The function $\Phi(\cdot) : \mathbb{R} \rightarrow (0, 1)$ is a positive function with upper bound 1. We use the bold notation \mathbf{g} to refer to the function $g(\mathbf{x})$ compactly as a vector of (infinite) length on which it is possible to perform inference. The normalisation constant $\mathcal{Z}_\pi[\mathbf{g}]$ is a functional of $g(\mathbf{x})$:

$$(2.2) \quad \mathcal{Z}_\pi[\mathbf{g}] = \int dx' \Phi(g(x')) \pi(x' | \psi).$$

We include the subscript π to indicate implicit dependence on the density $\pi(\mathbf{x})$. Through the map defined by Equation 2.1, a Gaussian process provides a prior distribution over normalised probability density functions on \mathcal{X} . Figure 1 shows several realisations of densities from this prior, along with sample data.

Although we only require that the function $\Phi(\cdot)$ be positive and bounded, it is convenient for inference if it is a bijective map between \mathbb{R} and $(0, 1)$. If $\Phi(\cdot)$ is bijective then each function that maps \mathcal{X} to $(0, 1)$ corresponds to a unique realisation $g(\mathbf{x})$ from the Gaussian process. Sigmoids, such as the cumulative normal distribution function and the logistic function, are bijective functions with this domain and range. We take $\Phi(\cdot)$ to be the logistic function, i.e., $\Phi(z) = 1/(1 + \exp(-z))$.

3. Generating data from the prior. We can use rejection sampling to simulate samples from a common density drawn from the the prior described in Section 2. A rejection sampler requires a proposal density that upper bounds the unnormalised density of interest. In this case, the proposal density is $\pi(\mathbf{x} | \psi)$ and the unnormalised density of interest is $\Phi(g(\mathbf{x})) \pi(\mathbf{x} | \psi)$. We assume that it is possible to draw samples directly from $\pi(\mathbf{x} | \psi)$.

If $g(\mathbf{x})$ were known, rejection sampling would proceed as follows: first generate proposals $\{\tilde{\mathbf{x}}_r\}$ from the base density $\pi(\mathbf{x} | \psi)$. The proposal $\tilde{\mathbf{x}}_r$ would be accepted if a variate u_r drawn uniformly from $(0, 1)$ was less than $\Phi(g(\tilde{\mathbf{x}}_r))$. These samples would be exact in the sense that they were not biased by the starting state of a finite Markov chain. However, in the

¹We will use the word “density,” according to the idea that \mathcal{X} is \mathbb{R}^D . However, this construction would provide a distribution over probability mass functions for countable \mathcal{X} .

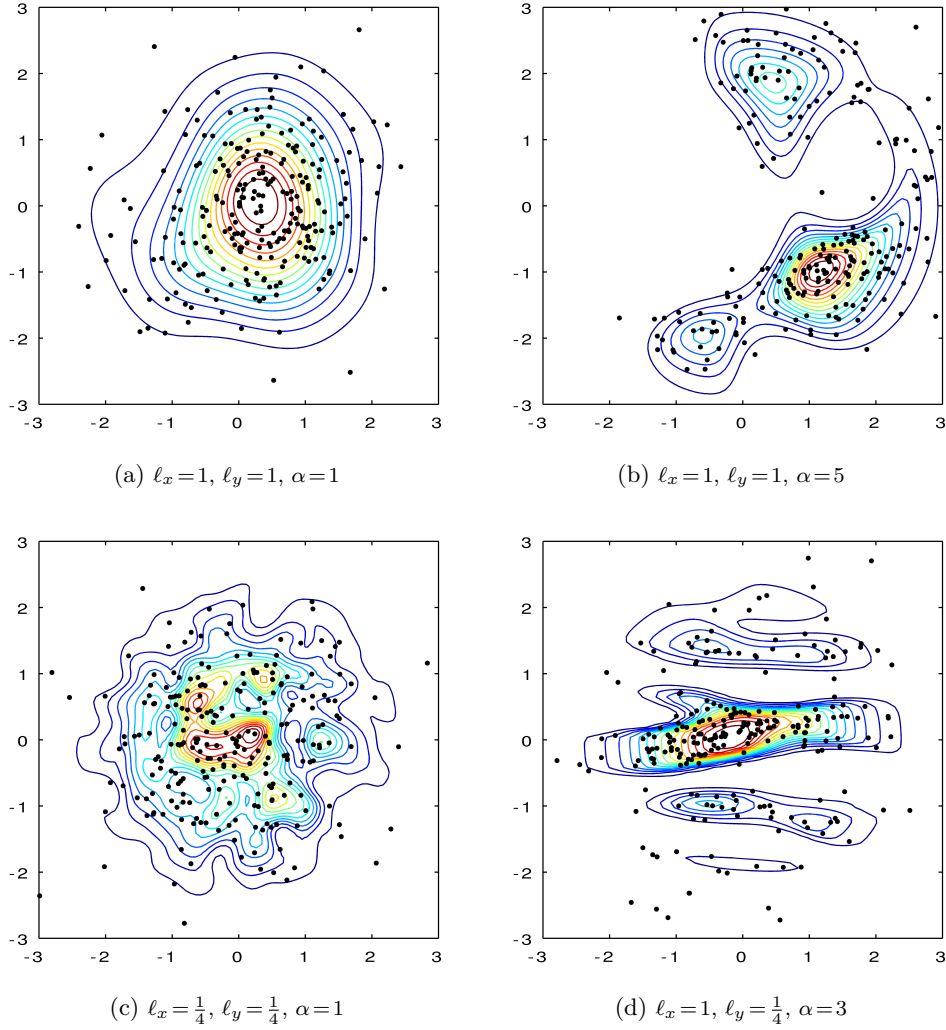
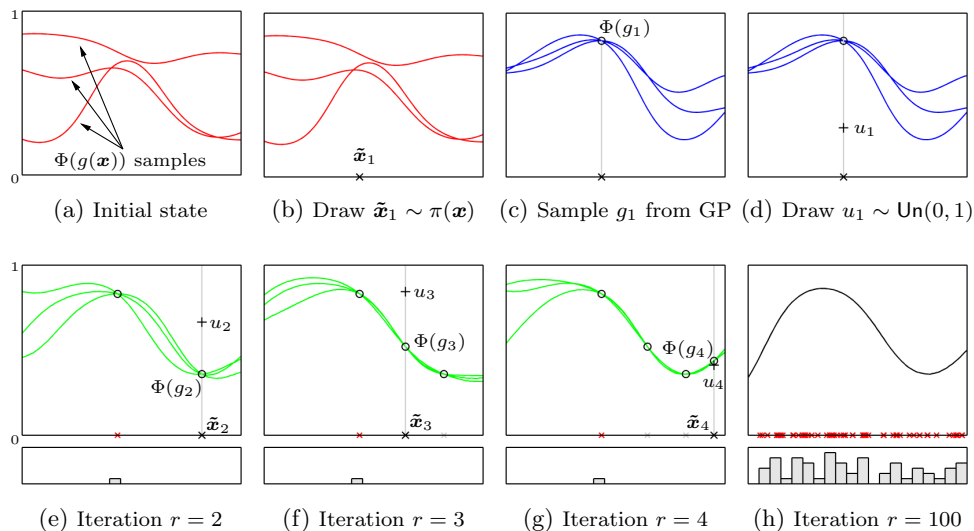


Fig 1: Four samples from the GPDS prior are shown, with 250 data samples. The contour lines show the approximate unnormalized densities. In each case the base density is the zero-mean circular Gaussian with unit variance. The mean function is set to zero. The covariance function is the squared exponential: $C(\mathbf{x}, \mathbf{x}') = \alpha^2 \exp(-\frac{1}{2} \sum_d (x_d - x'_d)^2 / \ell_d^2)$, with parameters varied as labeled in each subplot. $\Phi(\cdot)$ is the logistic function in these plots.



GPDS, $g(\mathbf{x})$ is not known: it is a random function drawn from a Gaussian process prior. We can nevertheless use rejection sampling by “discovering” $g(\mathbf{x})$ as we proceed at just the places we need to know it, by sampling from the prior distribution of the latent function. As the values of $g(\mathbf{x})$ evaluated at the $\{\tilde{\mathbf{x}}_r\}$ are consistent with a single draw of the whole function, the samples are exact. This type of retrospective sampling trick has been used in a variety of MCMC algorithms for infinite-dimensional models [Beskos et al. \[2006\]](#), [Papaspiliopoulos and Roberts \[2008\]](#). Figure 3 shows the generative procedure graphically.

In practice, we generate the samples sequentially, as in Algorithm 3.1, so that we may be assured of having as many accepted samples as we require. In each loop, a proposal is drawn from the base density $\pi(\mathbf{x} | \psi)$ and the function $g(\mathbf{x})$ is sampled from the Gaussian process at this proposed coordinate, conditional on all the function values already sampled. We will call these data the *conditioning set* for the function $g(\mathbf{x})$ and will denote the conditioning inputs as \mathbf{X} and the conditioning function values as \mathbf{G} . After the function is sampled, a variate is drawn uniformly from $(0,1)$ and compared to the Φ -squashed function at the proposal location. If the uniform variate falls below $\Phi(g(\mathbf{x}))$ then we accept the proposal, otherwise we reject. The proposals and their function values are added into the conditioning set regardless of whether that proposal was accepted or rejected. The loop repeats until we have as many acceptances as are required.

The sequential procedure is infinitely exchangeable; the probability of the

Algorithm 3.1 Generate N exact samples from a density drawn from the prior

Inputs:

- Number of samples to draw N
- Gaussian process covariance function $C(\mathbf{x}, \mathbf{x}'; \theta)$
- Base density $\pi(\mathbf{x} | \psi)$

Outputs:

- N samples $\mathcal{D} = \{\mathbf{x}_n\}_{n=1}^N$ from a random density drawn from the prior.

1: $\mathbf{X} \leftarrow \emptyset, \mathbf{G} \leftarrow \emptyset$ ▷ Initially the conditioning sets are empty.
2: $\mathcal{D} \leftarrow \emptyset$ ▷ Initialize the set to be returned.
3: $r \leftarrow 0$ ▷ Count the number of proposals.
4: **repeat**
5: $\tilde{\mathbf{x}}_r \sim \pi(\mathbf{x} | \psi)$ ▷ Draw a proposal.
6: $g(\tilde{\mathbf{x}}_r) \sim \mathcal{GP}(g | \tilde{\mathbf{x}}_r, \mathbf{X}, \mathbf{G}, \theta)$ ▷ Sample from the GP at the proposal.
7: $u_r \sim \text{Un}(0, 1)$ ▷ Draw uniformly on $(0, 1)$.
8: **if** $u_r < \Phi(g(\tilde{\mathbf{x}}_r))$ **then** ▷ Rejection sampling acceptance rule.
9: $\mathcal{D} \leftarrow \mathcal{D} \cup \tilde{\mathbf{x}}_r$ ▷ Store the proposal.
10: **end if**
11: $\mathbf{X} \leftarrow \mathbf{X} \cup \tilde{\mathbf{x}}_r, \mathbf{G} \leftarrow \mathbf{G} \cup g(\tilde{\mathbf{x}}_r)$ ▷ Update the conditioning sets, even on rejections.
12: $r \leftarrow r + 1$
13: **until** $|\mathcal{D}| = N$ ▷ Loop until N samples are accepted.
14: **return** \mathcal{D}

data is the same under reordering. First, the base density draws are i.i.d.. Second, conditioned on the proposals from the base density, the Gaussian process is a simple multivariate Gaussian distribution, which is exchangeable in its components. Finally, conditioned on the draw from the Gaussian process, the acceptance/rejection steps are independent Bernoulli samples, and the overall procedure is exchangeable. This property ensures that the sequential procedure generates data from the same distribution as the simultaneous procedure described above. More broadly, exchangeable priors are useful in Bayesian modeling because we may consider the data conditionally independent, given the latent density.

4. Inference. We now consider the problem of inference with the GPDS. We observe N data $\mathcal{D} = \{\mathbf{x}_n\}_{n=1}^N$ that we model as having been drawn independently from an unknown density $f(\mathbf{x})$. We place the GPDS prior of Section 2 on $f(\mathbf{x})$. The posterior on \mathbf{g} is given by Bayes' theorem:

$$(4.1) \quad p(\mathbf{g} | \mathcal{D}, \theta) = \frac{p(\mathbf{g} | \theta) (\mathcal{Z}_\pi[\mathbf{g}])^{-N} \prod_{n=1}^N \Phi(g(\mathbf{x}_n)) \pi(\mathbf{x}_n | \psi)}{\int d\mathbf{g}' p(\mathbf{g}' | \theta) (\mathcal{Z}_\pi[\mathbf{g}'])^{-N} \prod_{n=1}^N \Phi(g'(\mathbf{x}_n)) \pi(\mathbf{x}_n | \psi)}.$$

Even with Markov chain Monte Carlo, inference in this model is difficult. Evaluating the posterior requires computing two difficult integrals, the denominator and the normalisation constant $\mathcal{Z}_\pi[\mathbf{g}]$. It is common for the marginal likelihood in the denominator of the posterior to be intractable;

MCMC methods such as Metropolis–Hastings are well-suited for this situation. Posteriors such as Equation 4.1 with difficult sums in both the numerator and denominator are called *doubly-intractable*. Doubly-intractable posterior distributions appear most frequently when performing inference in undirected graphical models, where the partition function can be difficult to evaluate [Møller et al., 2006, Murray et al., 2006].

To see the difficulty concretely, consider a naïve Metropolis–Hastings Markov chain on \mathbf{g} , with proposal density $q(\hat{\mathbf{g}} \leftarrow \mathbf{g})$:

$$(4.2) \quad a_{\text{naïve}} = \frac{q(\mathbf{g} \leftarrow \hat{\mathbf{g}}) p(\hat{\mathbf{g}} | \theta)}{q(\hat{\mathbf{g}} \leftarrow \mathbf{g}) p(\mathbf{g} | \theta)} \left(\frac{\mathcal{Z}_\pi[\hat{\mathbf{g}}]}{\mathcal{Z}_\pi[\mathbf{g}]} \right)^N \prod_{n=1}^N \frac{\Phi(\hat{\mathbf{g}}(\mathbf{x}_n)) \pi(\mathbf{x}_n | \psi)}{\Phi(\mathbf{g}(\mathbf{x}_n)) \pi(\mathbf{x}_n | \psi)}.$$

The functions \mathbf{g} and $\hat{\mathbf{g}}$ are infinite-dimensional objects, which cannot be evaluated everywhere in practice. In other contexts, such as Gaussian process classification, it is possible to construct the proposal density such that the acceptance ratio only depends on the functions at $\{\mathbf{x}_n\}$. Here, the intractable ratio of normalising constants makes it impossible to evaluate the acceptance ratio without knowing the \mathbf{g} and $\hat{\mathbf{g}}$ everywhere. We present two Markov chain Monte Carlo algorithms that sidestep this difficulty. The equilibrium distribution in both cases is the posterior in Equation 4.1, and both algorithms take advantage of the exact data generation procedure described in Section 3.

4.1. *Exchange sampling.* Exchange sampling [Murray et al., 2006, Murray, 2007] is a variant of the Metropolis–Hastings method that enables sampling from doubly-intractable posterior distributions, subject to the requirement that exact samples can be generated from the model. The procedure is a simpler alternative to the auxiliary variable method of Møller et al. [2006]. Exchange sampling introduces additional state into the Markov chain that is chosen so that the intractable constants cancel out of the Metropolis–Hastings acceptance ratio. Murray et al. [2006] used exchange sampling to infer the coupling parameters of Ising models where exact data could be generated via coupling from the past [Propp and Wilson, 1996]. In the GPDS we generate exact samples via the rejection method of Section 3.

Initially, we apply exchange sampling to the posterior on \mathbf{g} using the Gaussian process prior as the proposal distribution, i.e., $q(\hat{\mathbf{g}} \leftarrow \mathbf{g}) = p(\hat{\mathbf{g}} | \theta)$. The joint distribution over the data \mathcal{D} , the current Markov state \mathbf{g} and the proposal $\hat{\mathbf{g}}$ is augmented with N “fantasy data” $\mathcal{W} = \{\mathbf{w}_n\}_{n=1}^N$. These fantasy data live on the same space \mathcal{X} as the true data, but are drawn from the distribution implied by the proposal $\hat{\mathbf{g}}$. The augmented joint distribution

is

$$(4.3) \quad p(\mathbf{g}, \mathcal{D}, \hat{\mathbf{g}}, \mathcal{W} \mid \theta, \psi) = p(\mathbf{g} \mid \theta) p(\{\mathbf{x}_n\}_{n=1}^N \mid \mathbf{g}, \psi) p(\hat{\mathbf{g}} \mid \theta) p(\{\mathbf{w}_n\}_{n=1}^N \mid \hat{\mathbf{g}}, \psi).$$

Given the current state \mathbf{g} , we jointly propose $\hat{\mathbf{g}}$ and \mathcal{W} by using Algorithm 3.1. This algorithm simultaneously draws $\hat{\mathbf{g}}$ from the prior and generates the N fantasy data \mathcal{W} . We then propose swapping \mathbf{g} with $\hat{\mathbf{g}}$. The acceptance ratio of the swap proposal is the ratio of the joint density in Equation 4.3 under each setting:

$$(4.4) \quad \begin{aligned} a_{\text{exch}} &= \frac{p(\hat{\mathbf{g}} \mid \theta) p(\{\mathbf{x}_n\}_{n=1}^N \mid \hat{\mathbf{g}}, \psi) p(\mathbf{g} \mid \theta) p(\{\mathbf{w}_n\}_{n=1}^N \mid \mathbf{g}, \psi)}{p(\mathbf{g} \mid \theta) p(\{\mathbf{x}_n\}_{n=1}^N \mid \mathbf{g}, \psi) p(\hat{\mathbf{g}} \mid \theta) p(\{\mathbf{w}_n\}_{n=1}^N \mid \hat{\mathbf{g}}, \psi)} \\ &= \frac{\cancel{\mathcal{Z}_\pi[\mathbf{g}]^N} \cancel{\mathcal{Z}_\pi[\hat{\mathbf{g}}]^N} \prod_{n=1}^N \Phi(\hat{\mathbf{g}}(\mathbf{x}_n)) \cancel{\pi(\mathbf{x}_n \mid \psi)} \prod_{n=1}^N \Phi(g(\mathbf{w}_n)) \cancel{\pi(\mathbf{w}_n \mid \psi)}}{\cancel{\mathcal{Z}_\pi[\hat{\mathbf{g}}]^N} \cancel{\mathcal{Z}_\pi[\mathbf{g}]^N} \prod_{n=1}^N \Phi(g(\mathbf{x}_n)) \cancel{\pi(\mathbf{x}_n \mid \psi)} \prod_{n=1}^N \Phi(\hat{\mathbf{g}}(\mathbf{w}_n)) \cancel{\pi(\mathbf{w}_n \mid \psi)}} \\ &= \prod_{n=1}^N \frac{\Phi(\hat{\mathbf{g}}(\mathbf{x}_n)) \Phi(g(\mathbf{w}_n))}{\Phi(g(\mathbf{x}_n)) \Phi(\hat{\mathbf{g}}(\mathbf{w}_n))}. \end{aligned}$$

The normalisation constants cancel out, and the functions $g(\mathbf{x})$ and $\hat{\mathbf{g}}(\mathbf{x})$ need only be sampled from the Gaussian process at a finite number of locations.

Algorithm 4.1 shows the exchange sampling inference procedure for the GPDS. Some amount of bookkeeping is required for this procedure to be valid. Specifically, once something is learned about a particular function $g(\mathbf{x})$, i.e., sampled from the Gaussian process, it cannot be forgotten until that $g(\mathbf{x})$ is discarded. For example, when fantasy data is generated from $\hat{\mathbf{g}}(\mathbf{x})$, as in steps 5 to 15, even if $\tilde{\mathbf{x}}$ is rejected in step 11, the $(\tilde{\mathbf{x}}, g(\tilde{\mathbf{x}}))$ pair must be stored in the conditioning set (step 14). If the proposed $\hat{\mathbf{g}}(\mathbf{x})$ is ultimately rejected by step 19, only then can the conditioning set for $\hat{\mathbf{g}}(\mathbf{x})$ be discarded. Similarly, when the current Markov state $g(\mathbf{x})$ is sampled from the Gaussian process at the fantasy data in step 16, this information must be kept if the proposal is rejected (step 22). Thus step 22 expands the Markov state with every rejection, as information about the current $g(\mathbf{x})$ accumulates. When the proposal $\hat{\mathbf{g}}(\mathbf{x})$ is accepted, the Markov state reduces in size, as fewer points will typically have been sampled from $\hat{\mathbf{g}}(\mathbf{x})$. An example sequence of rejections and an acceptance is illustrated in Figure 2.

4.1.1. *Improving the acceptance rate.* For clarity, we introduced the algorithm with $q(\hat{\mathbf{g}} \leftarrow \mathbf{g}) = p(\hat{\mathbf{g}} \mid \theta)$, but this proposal is a poor choice in practice.

Algorithm 4.1 Simulate R steps of an exchange sampling Markov chain on $p(\mathbf{g} \mid \mathcal{D})$

Inputs:

- Number of MCMC iterations R
- Observed data $\mathcal{D} = \{\mathbf{x}_n\}_{n=1}^N$
- Gaussian process covariance function $C(\mathbf{x}, \mathbf{x}'; \theta)$
- Base density $\pi(\mathbf{x} \mid \psi)$

Outputs:

- R conditioning sets of function inputs and outputs $\{\mathbf{X}^{(r)}, \mathbf{G}^{(r)}\}_{r=1}^R$

```

1:  $\{g(\mathbf{x}_n)\}_{n=1}^N \sim \mathcal{GP}(g \mid \mathcal{D}, \theta)$  ▷ Initialise the function at the data.
2:  $\mathbf{X}^{(1)} \leftarrow \{\mathbf{x}_n\}_{n=1}^N, \mathbf{G}^{(1)} \leftarrow \{g(\mathbf{x}_n)\}_{n=1}^N$  ▷ Initialise conditioning sets.
3: for  $r \leftarrow 1 \dots R$  do ▷ Take  $R$  exchange sampling steps.
4:    $\{\hat{g}(\mathbf{x}_n)\}_{n=1}^N \sim \mathcal{GP}(g \mid \mathcal{D}, \theta)$  ▷ Draw a new function at the data.
5:    $\hat{\mathbf{X}} \leftarrow \{\mathbf{x}_n\}_{n=1}^N, \hat{\mathbf{G}} \leftarrow \{\hat{g}(\mathbf{x}_n)\}_{n=1}^N$  ▷ Initialise proposal conditioning sets.
6:    $\mathcal{W} \leftarrow \emptyset$  ▷ Initialise empty fantasy data set.
7:   repeat ▷ Run the rejection sampling loop.
8:      $\tilde{\mathbf{w}} \sim \pi(\mathbf{x} \mid \psi)$  ▷ Draw a proposal from the base density.
9:      $\hat{g}(\tilde{\mathbf{w}}) \sim \mathcal{GP}(\hat{g} \mid \tilde{\mathbf{w}}, \hat{\mathbf{X}}, \hat{\mathbf{G}}, \theta)$  ▷ Draw the function value at the proposal.
10:     $u_{\text{fant}} \sim \text{Un}(0, 1)$  ▷ Draw a uniform random variate on  $(0, 1)$ .
11:    if  $u_{\text{fant}} < \Phi(\hat{g}(\tilde{\mathbf{w}}))$  then ▷ Rejection sampling acceptance rule.
12:       $\mathcal{W} \leftarrow \mathcal{W} \cup \tilde{\mathbf{w}}$  ▷ Keep the fantasy.
13:    end if
14:     $\hat{\mathbf{X}} \leftarrow \hat{\mathbf{X}} \cup \tilde{\mathbf{w}}, \hat{\mathbf{G}} \leftarrow \hat{\mathbf{G}} \cup \hat{g}(\tilde{\mathbf{w}})$  ▷ Add proposals to the conditioning sets.
15:  until  $|\mathcal{W}| = N$  ▷ Loop until  $N$  fantasies are accepted.
16:   $\{g(\mathbf{w}_n)\}_{n=1}^N \sim \mathcal{GP}(g \mid \mathcal{W}, \mathbf{X}^{(r)}, \mathbf{G}^{(r)})$  ▷ Sample the current func. at the fantasies.
17:   $a_{\text{exch}} \leftarrow \prod_{n=1}^N \frac{\Phi(\hat{g}(\mathbf{x}_n)) \Phi(g(\mathbf{w}_n))}{\Phi(g(\mathbf{x}_n)) \Phi(\hat{g}(\mathbf{w}_n))}$  ▷ Calculate the acceptance ratio.
18:   $u_{\text{mh}} \sim \text{Un}(0, 1)$  ▷ Draw a uniform random variate on  $(0, 1)$ .
19:  if  $u_{\text{mh}} < a_{\text{exch}}$  then ▷ Apply the Metropolis–Hastings acceptance rule.
20:     $\mathbf{X}^{(r+1)} \leftarrow \hat{\mathbf{X}}, \mathbf{G}^{(r+1)} \leftarrow \hat{\mathbf{G}}$  ▷ Keep the new function data.
21:  else
22:     $\mathbf{X}^{(r+1)} \leftarrow \mathbf{X}^{(r)} \cup \{\mathbf{w}_n\}_{n=1}^N$  ▷ Add the fantasy evaluations to the current state.
23:     $\mathbf{G}^{(r+1)} \leftarrow \mathbf{G}^{(r)} \cup \{g(\mathbf{w}_n)\}_{n=1}^N$ 
24:  end if
25: end for
26: return  $\{\mathbf{X}^{(r)}, \mathbf{G}^{(r)}\}_{r=1}^R$ 

```

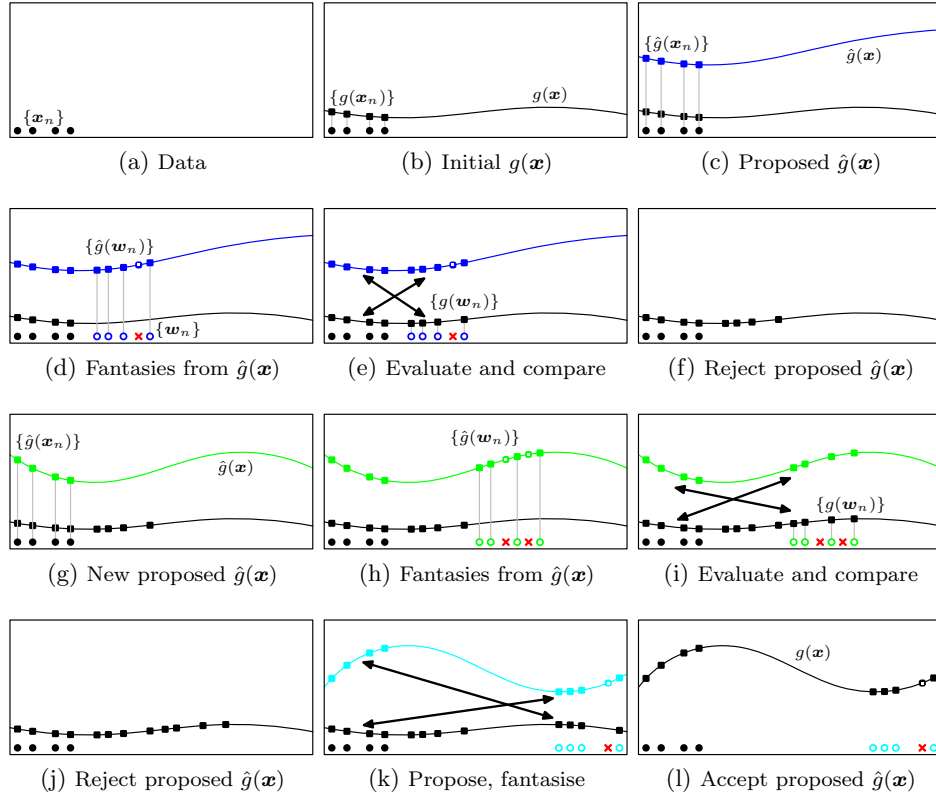


Fig 2: A cartoon of three exchange sampling transitions. In the first two transitions, the proposals are rejected to demonstrate the expanding Markov state due to retrospective sampling of $g(\mathbf{x})$. The third proposal is accepted and the previously-accumulated state is discarded. (a) The observed data, illustrated as \bullet . (b) The initial $g(\mathbf{x})$ evaluated at the data, shown as \blacksquare . (c) The proposed $\hat{g}(\mathbf{x})$ evaluated at the data, shown as \bullet . (d) Fantasies are drawn from $\hat{g}(\mathbf{x})$, illustrated as \circ . There is one rejected proposal, shown as \times , with the corresponding function value illustrated as \square . (e) $g(\mathbf{x})$ is evaluated at the fantasies and the two explanations are compared using Equation 4.4. (f) The proposal $\hat{g}(\mathbf{x})$ is rejected. The Markov state expands to include the fantasies. (g) A new $\hat{g}(\mathbf{x})$, shown in green, is evaluated at the data. (h) Fantasies are drawn from $\hat{g}(\mathbf{x})$ (two rejected fantasy proposals). (i) $g(\mathbf{x})$ is evaluated at the fantasies and the explanations are compared. (j) The proposal was rejected. The Markov state expands to twelve function evaluations. (k) Skipping the intermediate steps, propose, fantasise, evaluate and compare, using the function shown in cyan. (l) The proposal is accepted and made the new $g(\mathbf{x})$. All of the information about the old function is thrown away, but the new $g(\mathbf{x})$ must keep information in its conditioning set about the fantasies it generated.

To achieve a better acceptance rate, it is better to make conservative, perturbative proposals. This can be achieved by introducing a set of B “control points” in \mathcal{X} , denoted $\mathcal{C} = \{\mathbf{x}_b \in \mathcal{X}\}_{b=1}^B$. These control points have associated function values, which we denote as $\mathcal{G} = \{g(\mathbf{x}_b)\}_{b=1}^B$. We assume that \mathcal{C} is a superset of the observed data, i.e., $\mathcal{D} \subseteq \mathcal{C}$. The function values at the control points are explicitly included in the Markov state and all retrospective function draws condition on these points. New discoveries about the function continue to accumulate in the conditioning sets as before. The difference now is that the conditioning sets are initialised with the control points and small, perturbative proposals can be made on the function values at those initial points.

To make this construction explicit, Equation 4.3 is extended to

$$(4.5) \quad p(\mathcal{G}, \mathbf{g}_{\setminus \mathcal{C}}, \mathcal{D}, \hat{\mathcal{G}}, \hat{\mathbf{g}}_{\setminus \mathcal{C}}, \mathcal{W} | \mathcal{C}, \theta, \psi) = \\ \mathcal{GP}(\mathcal{G} | \mathcal{C}, \theta) \mathcal{GP}(\mathbf{g}_{\setminus \mathcal{C}} | \mathcal{G}, \theta) \mathcal{Z}_{\pi}[\mathbf{g}]^{-N} \left[\prod_{n=1}^N \Phi(g(\mathbf{x}_n)) \pi(\mathbf{x}_n | \psi) \right] \\ \times q(\hat{\mathcal{G}} \leftarrow \mathcal{G}) \mathcal{GP}(\hat{\mathbf{g}}_{\setminus \mathcal{C}} | \hat{\mathcal{G}}, \theta) \mathcal{Z}_{\pi}[\hat{\mathbf{g}}]^{-N} \prod_{n=1}^N \Phi(\hat{g}(\mathbf{w}_n)) \pi(\mathbf{w}_n | \psi),$$

where $\hat{\mathcal{G}}$ indicates the proposal of the function values at the control points, i.e. $\hat{\mathcal{G}} = \{\hat{g}(\mathbf{x}_b)\}_{b=1}^B$, and $\mathbf{g}_{\setminus \mathcal{C}}$ denotes the function values, excluding those at \mathcal{C} . The proposal density $q(\hat{\mathcal{G}} \leftarrow \mathcal{G})$ can be chosen to take smaller steps than the prior draws of the previous section. With the joint distribution in Equation 4.5, and using the conditional retrospective exchange sampling as before, the acceptance ratio of exchanging the pair $(\mathcal{G}, \mathbf{g}_{\setminus \mathcal{C}})$ for $(\hat{\mathcal{G}}, \hat{\mathbf{g}}_{\setminus \mathcal{C}})$ is

$$(4.6) \quad a_{\text{exch-cp}} = \frac{q(\mathcal{G} \leftarrow \hat{\mathcal{G}}) \mathcal{GP}(\hat{\mathcal{G}} | \mathcal{C}, \theta)}{q(\hat{\mathcal{G}} \leftarrow \mathcal{G}) \mathcal{GP}(\mathcal{G} | \mathcal{C}, \theta)} \prod_{n=1}^N \frac{\Phi(\hat{g}(\mathbf{x}_n)) \Phi(g(\mathbf{w}_n))}{\Phi(g(\mathbf{x}_n)) \Phi(\hat{g}(\mathbf{w}_n))}.$$

Superficially, this might seem similar to the knot-based imputation method of Tokdar [2007]. However, whereas Tokdar [2007] uses knots as a finite-dimensional approximation, we use the control points simply to constrain the proposal distribution. The control points only initialise the retrospective sampling procedure. As we enforce a Gaussian process prior on the function values of the control points, the inference procedure still yields the correct posterior distribution on the uncompromised fully-nonparametric Gaussian process density sampler model. The number and locations of the control points are free parameters.

A new function is proposed by first choosing values at the control points close to the existing function. The remainder of the function is drawn from

the prior, conditioned on the values at the control points. We always include the locations of the observed data as control points, i.e., $\mathcal{D} \subset \mathcal{C}$. This is not required for the algorithm to be valid, but is convenient as all proposed functions must be evaluated at the data in any case. Taking account of the arbitrary proposal density at the control points, $q(\{\hat{g}(\mathbf{x}_k)\}_{k=1}^K \leftarrow \{g(\mathbf{x}_k)\}_{k=1}^K)$, the exchange sampling acceptance ratio becomes

$$(4.7) \quad a_{\text{exch-control}} = \frac{q(\{g(\mathbf{x}_k)\}_{k=1}^K \leftarrow \{\hat{g}(\mathbf{x}_k)\}_{k=1}^K) p(\{\hat{g}(\mathbf{x}_k)\}_{k=1}^K | \theta)}{q(\{\hat{g}(\mathbf{x}_k)\}_{k=1}^K \leftarrow \{g(\mathbf{x}_k)\}_{k=1}^K) p(\{g(\mathbf{x}_k)\}_{k=1}^K | \theta)} \\ \times \prod_{n=1}^N \frac{\Phi(\hat{g}(\mathbf{x}_n)) \Phi(g(\mathbf{w}_n))}{\Phi(g(\mathbf{x}_n)) \Phi(\hat{g}(\mathbf{w}_n))}.$$

The functions drawn from the Gaussian process must still be evaluated at a larger conditioning set that includes the locations of fantasies. As before, these can be drawn “retrospectively” as needed, but now these Gaussian process samples are conditioned on the values at the control points.

4.1.2. *Hyperparameter inference.* One of the benefits of the Bayesian approach is the ability to perform hierarchical inference. In this case, it allows us to infer the hyperparameters θ of the Gaussian process and the hyperparameters ψ of the base density. We augment the exchange sampling algorithm slightly to sample from the posterior on hyperparameters: before proposing a new function $\hat{g}(\mathbf{x})$, we propose new hyperparameters $\hat{\theta}$ and $\hat{\psi}$ from a proposal density $q(\hat{\theta}, \hat{\psi} \leftarrow \theta, \psi)$. When samples of the new function are drawn, it is done with these proposed hyperparameters. The new joint distribution is

$$(4.8) \quad p(\mathbf{g}, \{\mathbf{x}_n\}_{n=1}^N, \theta, \psi, \hat{\mathbf{g}}, \{\mathbf{w}_n\}_{n=1}^N, \hat{\theta}, \hat{\psi}) = \\ p(\theta, \psi) p(\mathbf{g} | \theta) p(\{\mathbf{x}_n\}_{n=1}^N | \mathbf{g}, \psi) \\ \times q(\hat{\theta}, \hat{\psi} \leftarrow \theta, \psi) p(\hat{\mathbf{g}} | \hat{\theta}) p(\{\mathbf{w}_n\}_{n=1}^N | \hat{\mathbf{g}}, \hat{\psi})$$

where $p(\theta, \psi)$ is an appropriate hyperprior. The proposal is now to exchange the triplets $(\mathbf{g}, \theta, \psi)$ and $(\hat{\mathbf{g}}, \hat{\theta}, \hat{\psi})$. The acceptance of this swap has Metropolis–Hastings ratio

$$(4.9) \quad a_{\text{exch-hyper}} = \frac{q(\theta, \psi \leftarrow \hat{\theta}, \hat{\psi}) p(\hat{\theta}, \hat{\psi})}{q(\hat{\theta}, \hat{\psi} \leftarrow \theta, \psi) p(\theta, \psi)} \\ \times \prod_{n=1}^N \frac{\Phi(\hat{g}(\mathbf{x}_n)) \pi(\mathbf{x}_n | \hat{\psi}) \Phi(g(\mathbf{w}_n)) \pi(\mathbf{w}_n | \psi)}{\Phi(g(\mathbf{x}_n)) \pi(\mathbf{x}_n | \psi) \Phi(\hat{g}(\mathbf{w}_n)) \pi(\mathbf{w}_n | \hat{\psi})}.$$

This acceptance ratio generalises straightforwardly to the case with control points discussed in Section 4.1.1.

4.1.3. *Sampling from the predictive distribution.* An important task for density inference is estimation of the predictive density. The predictive distribution arises on data space when the posterior is integrated out. For the GPDS, this density is

$$(4.10) \quad p(\mathbf{x} | \mathcal{D}) = \int d\theta \int d\psi \int d\mathbf{g} p(\mathbf{x} | \mathbf{g}, \theta, \psi) p(\mathbf{g}, \theta, \psi | \mathcal{D}).$$

The predictive distribution can also be thought of as the distribution on the next datum to arrive, given the N already seen and taking uncertainty into account. In the GPDS, the predictive density in Equation 4.10 is not available analytically. We nevertheless have all the tools in place to generate samples from the predictive distribution. We do this by using the generative procedure of Section 3 to generate additional data after each Metropolis–Hastings step. We use a very similar method to Algorithm 3.1, but initialise the conditioning set using the current state of the Markov chain.

4.2. *Sampling over latent histories.* An alternative to inference via exchange sampling is to model the *latent history* of the generative process. By using the GPDS prior to model the data, we are asserting that the data can be explained as the result of Algorithm 3.1. However, we did not observe any of the intermediate states of the rejection sampling algorithm, such as the number and locations of the rejected proposals, and the value of the function sampled from the Gaussian process prior. Nevertheless, Algorithm 3.1 provides a well-defined probabilistic model over both the observed data and this latent state. By modeling this larger joint distribution we can avoid evaluating the intractable normalisation constant that would otherwise appear in the likelihood function.

We model the data $\mathcal{D} = \{\mathbf{x}_n\}_{n=1}^N$ as having been generated exactly as in Algorithm 3.1, i.e., run until exactly N proposals were accepted. The state space of the Markov chain on latent histories in the GPDS consists of: 1) the values of the latent function $g(\mathbf{x})$ at the data, denoted $\mathcal{G}_N = \{g(\mathbf{x}_n)\}_{n=1}^N$, 2) the number of rejections M , 3) the locations of the M rejected proposals, denoted $\mathcal{M} = \{\mathbf{x}_m\}_{m=1}^M$, and 4) the values of the latent function $g(\mathbf{x})$ at the M rejected proposals, denoted $\mathcal{G}_M = \{g(\mathbf{x}_m)\}_{m=1}^M$. The joint distribution over the data and the ordered history of the GPDS generative procedure, given the hyperparameters, is

$$(4.11) \quad p(\mathcal{D}, \mathcal{G}_N, \mathcal{M}, \mathcal{G}_M | \theta, \psi) = \mathcal{GP}(\mathcal{G}_N, \mathcal{G}_M | \mathcal{D}, \mathcal{M}, \theta) \\ \times \left[\prod_{n=1}^N \Phi(g(\mathbf{x}_n)) \pi(\mathbf{x}_n | \psi) \right] \prod_{n=1}^M (1 - \Phi(g(\mathbf{x}_m))) \pi(\mathbf{x}_m | \psi).$$

We sample from the posterior distribution over all unknowns, which is proportional to the joint distribution with the observations, \mathcal{D} , clamped. The Markov chain algorithm applies three types of update in sequence: 1) modification of the number of rejections M , 2) updating of the rejection locations \mathcal{M} , and 3) modification of the latent function values \mathcal{G}_M and \mathcal{G}_N . We will maintain an explicit ordering of the latent rejections for reasons of clarity, although this is not necessary due to exchangeability. At any time we could propose a reshuffling of the latent history, subject to it ending in an acceptance, and this proposal would always be accepted, as the two permutations have the same probability under the model.

Inference by Markov chain Monte Carlo of the history of a probabilistic computational procedure has been studied previously. Beskos et al. [2006] sampled from the state of a rejection sampler for diffusions. Murray [2007], who coined the phrase ‘‘latent history,’’ modeled data as having been the result of a Markov chain which had provably mixed via coupling from the past [Propp and Wilson, 1996]. Another example is Huber and Wolpert [2009], who model the history of the Matérn Type III process to perform tractable inference. The Church programming language [Goodman et al., 2008] also exploits this idea, by treating probabilistic procedures as first class objects on which inference can be performed.

4.2.1. *Modifying the number of latent rejections.* We propose a new number of latent rejections \hat{M} by drawing it from a proposal density $q(\hat{M} \leftarrow M)$. If \hat{M} is greater than M , we must also propose new rejections to add to the latent state. We take advantage of the exchangeability of the process to generate the new rejections: we imagine these proposals were made *after* the last observed datum was accepted, and our proposal is to call them rejections and move them *before* the last datum. If \hat{M} is less than M , we do the opposite by proposing to move some rejections to after the last acceptance.

When proposing additional rejections, we must also propose times for them among the current latent history. There are $\binom{\hat{M}+N-1}{\hat{M}-M}$ such ways to insert these additional rejections into the existing latent history, such that the sampler terminates after the N th acceptance. When removing rejections, we must choose which ones to place after the data, and there are $\binom{M}{M-\hat{M}}$ possible sets. Upon simplification, the proposal ratios for both addition and removal of rejections are identical:

$$\frac{\overbrace{q(M \leftarrow \hat{M}) \binom{\hat{M}+N-1}{\hat{M}-M}}^{\hat{M} > M}}{\underbrace{q(\hat{M} \leftarrow M) \binom{\hat{M}}{\hat{M}-M}}_{\hat{M} < M}} = \frac{\overbrace{q(M \leftarrow \hat{M}) \binom{M}{M-\hat{M}}}_{\hat{M} < M}}{\underbrace{q(\hat{M} \leftarrow M) \binom{M+N-1}{M-\hat{M}}}_{\hat{M} > M}} = \frac{q(M \leftarrow \hat{M}) M! (\hat{M} + N - 1)!}{q(\hat{M} \leftarrow M) \hat{M}! (M + N - 1)!}.$$

When inserting rejections, we propose the locations of the additional proposals, denoted \mathcal{M}^+ , and the corresponding values of the latent function, denoted \mathcal{G}_M^+ . We generate \mathcal{M}^+ by making $\hat{M} - M$ independent draws from the base density. We draw \mathcal{G}_M^+ jointly from the Gaussian process prior, conditioned on all of the current latent state, i.e., $(\mathcal{M}, \mathcal{G}_M, \mathcal{D}, \mathcal{G}_N)$. The joint probability of this state is

$$(4.12) \quad p(\mathcal{D}, \mathcal{M}, \mathcal{M}^+, \mathcal{G}_N, \mathcal{G}_M, \mathcal{G}_M^+ | \theta, \psi) = \left[\prod_{n=1}^N \pi(\mathbf{x}_n | \psi) \Phi(g(\mathbf{x}_n)) \right] \\ \times \left[\prod_{m=1}^M \pi(\mathbf{x}_m | \psi) (1 - \Phi(g(\mathbf{x}_m))) \right] \left[\prod_{m=M+1}^{\hat{M}} \pi(\mathbf{x}_m | \psi) \right] \\ \times \mathcal{GP}(\mathcal{G}_M, \mathcal{G}_N, \mathcal{G}_M^+ | \mathcal{D}, \mathcal{M}, \mathcal{M}^+, \theta).$$

The joint distribution in Equation 4.12 expresses the probability of all the base density draws, the values of the function draws from the Gaussian process, and the acceptance or rejection probabilities of the proposals *excluding* the newly generated points. When we make an insertion proposal, exchangeability allows us to shuffle the ordering without changing the probability; the only change is that now we must account for labeling the new points as rejections. In the acceptance ratio, all terms except for the “labeling probability” cancel. The reverse proposal is similar, however we denote the removed proposal locations as \mathcal{M}^- and the corresponding function values as \mathcal{G}_M^- . The overall acceptance ratios for insertions or removals are

$$(4.13) \quad a_{\text{hist-num}} = \begin{cases} \frac{q(M \leftarrow \hat{M}) M! (\hat{M} + N - 1)!}{q(\hat{M} \leftarrow M) \hat{M}! (M + N - 1)!} \prod_{g \in \mathcal{G}_M^+} (1 - \Phi(g)) & \text{if } \hat{M} > M \\ \frac{q(M \leftarrow \hat{M}) M! (\hat{M} + N - 1)!}{q(\hat{M} \leftarrow M) \hat{M}! (M + N - 1)!} \prod_{g \in \mathcal{G}_M^-} (1 - \Phi(g))^{-1} & \text{if } \hat{M} < M. \end{cases}$$

A simple and convenient way of implementing this procedure is to make limited proposals that either insert or delete only one latent rejection at a time. We define a function $\zeta(M, N) : \mathbb{N} \times \mathbb{N}^+ \rightarrow (0, 1]$ and propose inserting a new latent rejection with probability ζ . Otherwise, with probability $1 - \zeta$, we propose removing a rejection. We must, of course, enforce $\zeta(0, N) = 1$. With these limited proposals, the first case of Equation 4.13 (proposing one new latent rejection, i.e., $\hat{M} = M + 1$) can be written as

$$(4.14) \quad a_{\text{hist-ins}} = \frac{(1 - \zeta(M + 1, N)) (M + N) (1 - \Phi(g(\mathbf{x}^+)))}{\zeta(M, N) (M + 1)},$$

where \mathbf{x}^+ is the proposed rejection location. The location \mathbf{x}^+ is drawn from the base density $\pi(\mathbf{x} | \psi)$. In the second case, if there is at least one latent rejection in the current history ($M > 0$), then the deletion of a single rejection is proposed, i.e., $\hat{M} = M - 1$. This deletion proposal has Metropolis–Hastings acceptance ratio

$$(4.15) \quad a_{\text{hist-del}} = \frac{\zeta(M-1, N) M}{(1 - \zeta(M, N)) (M + N - 1) (1 - \Phi(g(\mathbf{x}^-)))},$$

where \mathbf{x}^- is the location of the proposed removal. The rejection to remove is chosen uniformly from among the M currently in the history.

4.2.2. Modifying latent rejection locations. Given the number of latent rejections M and the current latent function, we would like to sample from the locations of the rejections. Given the latent function, the locations of the rejections are independent. We make perturbative proposals of new locations, conditionally sample the function from the Gaussian process and then accept or reject with Metropolis–Hastings.

The current locations of the rejections are denoted \mathcal{M} and we draw a proposal $\hat{\mathcal{M}}$ from a proposal distribution $q(\hat{\mathcal{M}} \leftarrow \mathcal{M})$. The values of the latent function at \mathcal{M} are denoted \mathcal{G}_M and we sample the function at $\hat{\mathcal{M}}$ jointly from the Gaussian process prior given \mathcal{D} , \mathcal{G}_N , \mathcal{M} , and \mathcal{G}_M . The Metropolis–Hastings acceptance ratio of this proposal is

$$(4.16) \quad a_{\text{hist-locs}} = \frac{q(\mathcal{M} \leftarrow \hat{\mathcal{M}})}{q(\hat{\mathcal{M}} \leftarrow \mathcal{M})} \prod_{m=1}^M \frac{\pi(\hat{\mathbf{x}}_m | \psi) (1 - \Phi(\hat{g}(\mathbf{x}_m)))}{\pi(\mathbf{x}_m | \psi) (1 - \Phi(g(\mathbf{x}_m)))}.$$

4.2.3. Modifying the latent function values. Conditioned on the number and location of the latent rejections, we must also sample from the latent function at both the data and rejection locations. The conditional joint posterior distribution is

$$(4.17) \quad p(\mathcal{G}_N, \mathcal{G}_M | \mathcal{M}, \mathcal{D}, \theta) = \mathcal{GP}(\mathcal{G}_N, \mathcal{G}_M | \mathcal{D}, \mathcal{M}, \theta) \\ \times \left[\prod_{n=1}^N \Phi(g(\mathbf{x}_n)) \right] \left[\prod_{m=1}^M (1 - \Phi(g(\mathbf{x}_m))) \right].$$

This joint distribution is easily sampled using Hybrid (Hamiltonian) Monte Carlo [Duane et al., 1987]. For numerical reasons we suggest performing gradient calculations in the “whitened” space resulting from applying the inverse Cholesky decomposition of the covariance matrix to the function values.

Algorithm 4.2 implements the latent history algorithm in pseudocode, with the simple $q(\hat{M} \leftarrow M)$ that proposes increasing or decreasing the number of latent rejections M by one.

4.2.4. *Hyperparameter inference.* Given a sample from the posterior on the latent history, we can also perform a Metropolis–Hastings step in the space of hyperparameters. As in Section 4.1.2, we have hyperparameters θ for the Gaussian process and ψ for the base density, with joint prior density $p(\theta, \psi)$. We introduce the proposal density $q(\hat{\theta}, \hat{\psi} \leftarrow \theta, \psi)$ to make proposals $\hat{\theta}$ and $\hat{\psi}$. The acceptance ratio for a Metropolis–Hastings step in the posterior of the hyperparameters, given the latent history, is

$$(4.18) \quad a_{\text{hist-hp}} = \frac{q(\theta, \psi \leftarrow \hat{\theta}, \hat{\psi}) p(\hat{\theta}, \hat{\psi}) \mathcal{N}(\{\mathcal{G}_M, \mathcal{G}_N\} | \mathcal{M}, \mathcal{D}, \hat{\theta})}{q(\hat{\theta}, \hat{\psi} \leftarrow \theta, \psi) p(\theta, \psi) \mathcal{N}(\{\mathcal{G}_M, \mathcal{G}_N\} | \mathcal{M}, \mathcal{D}, \theta)} \\ \times \left[\prod_{m=1}^M \frac{\pi(\mathbf{x}_m | \hat{\psi})}{\pi(\mathbf{x}_m | \psi)} \right] \left[\prod_{n=1}^N \frac{\pi(\mathbf{x}_n | \hat{\psi})}{\pi(\mathbf{x}_n | \psi)} \right].$$

4.2.5. *Generating predictive samples.* As with the exchange sampling approach in Section 4.1.3, it is possible to generate samples from the predictive density. As each state in the Markov chain of the latent history inference is a rejection sampler state, it is simply a matter of continuing the rejection procedure forward to produce a new sample.

4.3. *Calculating the predictive density.* We have shown that each inference method can yield predictive samples, but it is also natural to require that a density model provide an estimate of the normalized predictive density itself. We use the method of Chib and Jeliazkov [2001], which considers Metropolis–Hastings moves between a pair \mathbf{x} and \mathbf{x}' . Using the base density $\pi(\mathbf{x} | \psi)$ as the proposal density, the detailed balance condition for Metropolis–Hastings gives the identity

$$(4.19) \quad p(\mathbf{x}, \mathbf{g}, \theta, \psi) \pi(\mathbf{x}' | \psi) \min \left(1, \frac{\Phi(g(\mathbf{x}'))}{\Phi(g(\mathbf{x}))} \right) = \\ p(\mathbf{x}', \mathbf{g}, \theta, \psi) \pi(\mathbf{x} | \psi) \min \left(1, \frac{\Phi(g(\mathbf{x}))}{\Phi(g(\mathbf{x}'))} \right).$$

We integrate both sides of this identity over \mathbf{x}' and take the expectation of each side under the posterior over the function \mathbf{g} and the hyperparameters

Algorithm 4.2 Simulate R steps of a Markov chain on the latent history

Inputs:

- Number of MCMC iterations R
- Observed data $\mathcal{D} = \{\mathbf{x}_n\}_{n=1}^N$
- Gaussian process covariance function $C(\mathbf{x}, \mathbf{x}'; \theta)$
- Base density $\pi(\mathbf{x} | \psi)$
- Location proposal density $q(\hat{\mathbf{x}}_m \leftarrow \mathbf{x}_m)$
- Insert proposal probability function $\zeta(M, N)$

Outputs:

- R samples of the latent history $\{\mathcal{M}^{(r)}, \mathcal{G}_N^{(r)}, \mathcal{G}_M^{(r)}\}_{r=1}^R$

```

1:  $\mathcal{M} \leftarrow \emptyset, \mathcal{G}_M \leftarrow \emptyset$  ▷ Start out with no latent rejections.
2:  $\mathcal{G}_N \sim \mathcal{GP}(g | \mathcal{D}, \theta)$  ▷ Initialise the function at the data.
3: for  $r \leftarrow 1 \dots R$  do ▷ Take  $R$  MCMC steps on the latent history.
4:    $u_\zeta \sim \text{Un}(0, 1)$  ▷ Draw a uniform random variate on  $(0, 1)$ .
5:   if  $u_\zeta < \zeta(|\mathcal{M}|, N)$  then ▷ Decide whether to insert or delete.
6:      $\mathbf{x}^+ \sim \pi(\mathbf{x} | \psi_\pi)$  ▷ Draw a proposed rejection location.
7:      $g(\mathbf{x}^+) \sim \mathcal{GP}(g | \mathbf{x}^+, \mathcal{D}, \mathcal{M}, \mathcal{G}_M, \mathcal{G}_N, \theta)$  ▷ Draw the proposed function value.
8:      $a_{\text{hist-ins}} \leftarrow \frac{(1 - \zeta(|\mathcal{M}| + 1, N)) (|\mathcal{M}| + N) (1 - \Phi(g(\mathbf{x}^+)))}{\zeta(|\mathcal{M}|, N) (|\mathcal{M}| + 1)}$  ▷ Acceptance ratio.
9:      $u_{\text{ins}} \sim \text{Un}(0, 1)$  ▷ Draw a uniform random variate on  $(0, 1)$ .
10:    if  $u_{\text{ins}} < a_{\text{hist-ins}}$  then ▷ Metropolis–Hastings acceptance rule.
11:       $\mathcal{M} \leftarrow \mathcal{M} \cup \mathbf{x}^+, \mathcal{G}_M \leftarrow \mathcal{G}_M \cup g(\mathbf{x}^+)$  ▷ Add this new rejection.
12:    end if
13:  else if  $|\mathcal{M}| > 0$  then
14:     $m \sim \lceil \text{Un}(0, |\mathcal{M}|) \rceil$  ▷ Select one of the  $M$  rejections at random.
15:     $a_{\text{hist-del}} = \frac{\zeta(|\mathcal{M}| - 1, N) |\mathcal{M}|}{(1 - \zeta(|\mathcal{M}|, N)) (|\mathcal{M}| + N - 1) (1 - \Phi(g(\mathbf{x}_m)))}$  ▷ Acceptance ratio.
16:     $u_{\text{del}} \sim \text{Un}(0, 1)$  ▷ Draw a uniform random variate on  $(0, 1)$ .
17:    if  $u_{\text{del}} < a_{\text{hist-del}}$  then ▷ Metropolis–Hastings acceptance rule.
18:       $\mathcal{M} \leftarrow \mathcal{M} \setminus \mathbf{x}_m, \mathcal{G}_M \leftarrow \mathcal{G}_M \setminus g(\mathbf{x}_m)$  ▷ Remove the  $m$ th rejection.
19:    end if
20:  end if
21:  for  $m \leftarrow 1 \dots M$  do ▷ Loop over the latent rejections.
22:     $\hat{\mathbf{x}}_m \sim q(\hat{\mathbf{x}}_m \leftarrow \mathbf{x}_m)$  ▷ Propose a new location.
23:     $g(\hat{\mathbf{x}}_m) \sim \mathcal{GP}(g | \hat{\mathbf{x}}_m, \mathcal{D}, \mathcal{M}, \mathcal{G}_N, \mathcal{G}_M, \theta)$  ▷ Draw a function value from the GP.
24:     $a_{\text{hist-loc}} = \frac{q(\mathbf{x}_m \leftarrow \hat{\mathbf{x}}_m) \pi(\hat{\mathbf{x}}_m) (1 - \Phi(g(\hat{\mathbf{x}}_m)))}{q(\hat{\mathbf{x}}_m \leftarrow \mathbf{x}_m) \pi(\mathbf{x}_m) (1 - \Phi(g(\mathbf{x}_m)))}$  ▷ Acceptance ratio.
25:     $u_{\text{loc}} \sim \text{Un}(0, 1)$  ▷ Draw a uniform random variate from  $(0, 1)$ .
26:    if  $u_{\text{loc}} < a_{\text{hist-loc}}$  then ▷ Metropolis–Hastings acceptance rule.
27:       $\mathbf{x}_m \leftarrow \hat{\mathbf{x}}_m, g(\mathbf{x}_m) \leftarrow g(\hat{\mathbf{x}}_m)$  ▷ Update the rejection.
28:    end if
29:  end for
30:   $\mathcal{G}_N, \mathcal{G}_M \sim \text{HMC}(\mathcal{G}_N, \mathcal{G}_M | \mathcal{D}, \mathcal{M}, \theta)$  ▷ Update function via Hybrid Monte Carlo.
31:   $\mathcal{M}^{(r)} \leftarrow \mathcal{M}, \mathcal{G}_N^{(r)} \leftarrow \mathcal{G}_N, \mathcal{G}_M^{(r)} \leftarrow \mathcal{G}_M$  ▷ Store the current version of the history.
32: end for
33: return  $\{\mathcal{M}^{(r)}, \mathcal{G}_N^{(r)}, \mathcal{G}_M^{(r)}\}_{r=1}^R$ 

```

θ and ψ :

$$\int d\theta \int d\psi \int d\mathbf{g} p(\mathbf{g}, \theta, \psi | \mathcal{D}) \int dx' p(\mathbf{x} | \mathbf{g}, \theta, \psi) \pi(\mathbf{x}' | \psi) \min\left(1, \frac{\Phi(g(\mathbf{x}'))}{\Phi(g(\mathbf{x}))}\right) = \int d\theta \int d\psi \int d\mathbf{g} p(\mathbf{g}, \theta, \psi | \mathcal{D}) \int dx' p(\mathbf{x}' | \mathbf{g}, \theta, \psi) \pi(\mathbf{x}' | \psi) \min\left(1, \frac{\Phi(g(\mathbf{x}))}{\Phi(g(\mathbf{x}'))}\right).$$

We observe that

$$p(\mathbf{g}, \theta, \psi | \mathcal{D}) p(\mathbf{x} | \mathbf{g}, \theta, \psi) = p(\mathbf{x}, \mathbf{g}, \theta, \psi | \mathcal{D}) = p(\mathbf{x} | \mathcal{D}) p(\mathbf{g}, \theta, \psi | x, \mathcal{D})$$

and so we may find the predictive density via

(4.20)

$$p(\mathbf{x} | \mathcal{D}) = \frac{\int d\theta \int d\psi \int d\mathbf{g} \int dx' p(\theta, \psi, \mathbf{g}, x' | \mathcal{D}) \pi(\mathbf{x} | \psi) \min\left(1, \frac{\Phi(g(\mathbf{x}))}{\Phi(g(\mathbf{x}'))}\right)}{\int d\theta \int d\psi \int d\mathbf{g} \int dx' p(\theta, \psi, \mathbf{g} | x, \mathcal{D}) \pi(\mathbf{x}' | \psi) \min\left(1, \frac{\Phi(g(\mathbf{x}'))}{\Phi(g(\mathbf{x}))}\right)}$$

Both the numerator and the denominator in Equation 4.20 are expectations. The top is an expectation under the posterior and the bottom is an expectation under the posterior where the data has been augmented with x :

$$(4.21) \quad p(\mathbf{x} | \mathcal{D}) = \frac{\mathbb{E}_{p(\mathbf{g}, \theta, \psi, x' | \mathcal{D})} \left[\pi(\mathbf{x} | \psi) \min\left(1, \frac{\Phi(g(\mathbf{x}))}{\Phi(g(\mathbf{x}'))}\right) \right]}{\mathbb{E}_{p(\mathbf{g}, \theta, \psi | \mathcal{D}, x)} \left[\mathbb{E}_{\pi(\mathbf{x}' | \psi)} \left[\min\left(1, \frac{\Phi(g(\mathbf{x}'))}{\Phi(g(\mathbf{x}))}\right) \right] \right]}.$$

The numerator can be estimated directly as part of the MCMC inference. After each Markov step, generate a predictive sample x' and record the transition probabilities. The denominator requires a Markov chain to be run with a data set augmented by the predictive location x . At each step in the Markov chain, a sample x' is generated from the base density and the transition probabilities are evaluated.

5. Examples.

5.1. *One-Dimensional Bounded Density.* We examined the GPDS on the one-dimensional problem studied by Lenk [1991] and Tokdar [2007]. It is a mixture of an exponential and normal density on $[0, 1]$:

$$(5.1) \quad f_1(x) = \frac{3}{4} \cdot 3 \exp\{-3x\} + \frac{1}{4} \cdot \left(\frac{\pi}{32}\right)^{-\frac{1}{2}} \exp\left\{-32\left(x - \frac{3}{4}\right)^2\right\}.$$

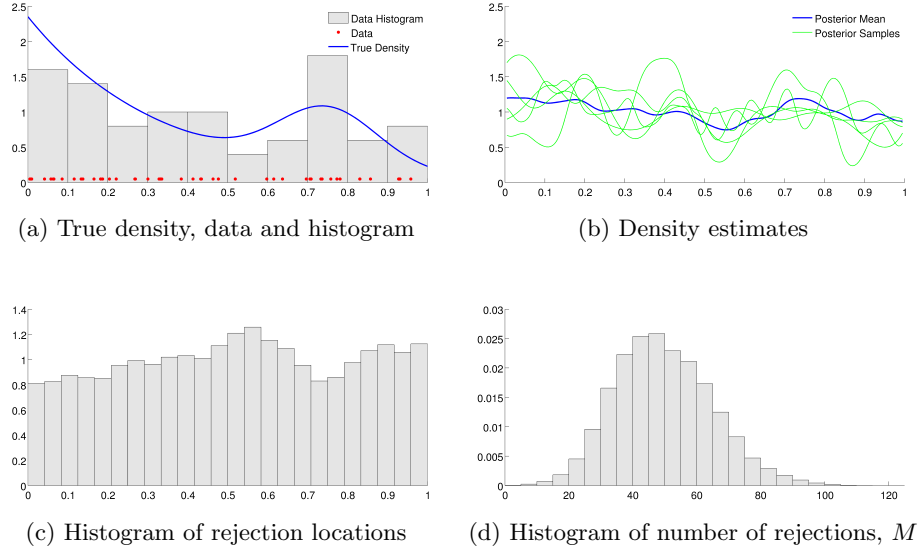


Fig 3: Bounded one-dimensional example

50 independent observations were drawn from this density and latent history inference with the GPDS was applied to it. Figure 3a shows a histogram of the observations and the true density. The base density for the GPDS was chosen to be the uniform distribution on $(0, 1)$. The covariance function used was the stationary squared exponential:

$$(5.2) \quad C(x, x') = \alpha^2 \exp \left\{ -\frac{1}{2} \left(\frac{x - x'}{\ell} \right)^2 \right\}$$

and the parameters α and ℓ were included in MCMC sampling for both the GPDS and the logistic Gaussian process. The priors used for the Gaussian process hyperparameters were

$$(5.3) \quad \ln \alpha \sim \mathcal{N}(\mu = 1, \sigma = 0.5)$$

$$(5.4) \quad \ln \ell \sim \mathcal{N}(\mu = 0.05, \sigma = 0.5).$$

The Markov chain was simulated for 50,000 iterations, with the first 10,000 discarded as burn-in. Figure 3b shows the predictive density from the MCMC run, along with several posterior samples. Figure 3c shows a histogram of the locations of the rejections in the latent history inference, and Figure 3d is a histogram of the number of rejections in samples from the latent history.

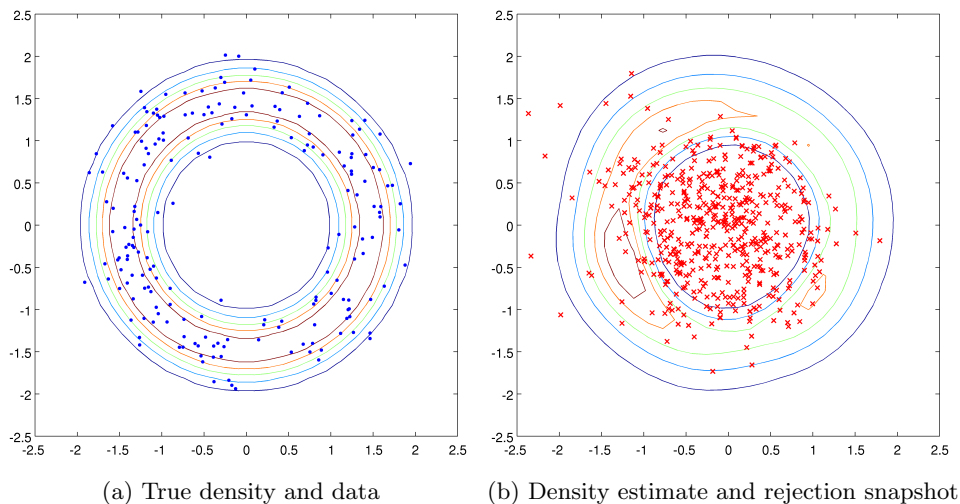


Fig 4: Synthetic ring mixture example

5.2. *Two-Dimensional Unbounded Density.* We generated 200 independent observations from a two-dimensional location mixture of Gaussians, where the means are drawn uniformly from a ring of radius $3/2$, centred at the origin. The Gaussians have a variance of $1/16$ so that the data density is

$$f_2(x_1, x_2) = \frac{4}{9\pi} \int_{-\pi}^{\pi} d\vartheta \mathcal{N}(x_1; 3/2 \cos \vartheta, \sigma = 1/4) \mathcal{N}(x_2; 3/2 \sin \vartheta, \sigma = 1/4).$$

We used the two-dimensional isotropic variant of the covariance function given by Equation 5.2 and used a Gaussian distribution for the base density, inferring the mean and covariance as in Section 4.2.4. We simulated the Markov chain for 50,000 iterations, discarding 10,000 as burn-in. The true density and the observed data are shown in Figure 4a, while the posterior predictive density and a posterior sample of rejection location are shown in Figure 4b. As expected, the rejections tend to accumulate in the center of the ring, where the base density places mass but the predictive density should be low.

6. Discussion.

6.1. *Computational issues.* Computation with a Gaussian process is expensive. If the GP is realised on R points, the space complexity of storing the

covariance (Gram) matrix is $O(R^2)$ and the time complexity of decomposing (or inverting) the matrix is $O(R^3)$. The time cost of this decomposition will be the asymptotically-dominating factor when performing GPDS inference using either exchange sampling or the latent history method.

6.2. *Comparing exchange sampling and latent history inference.* Modeling of probability densities is fundamentally different from regression. In regression and classification, one conditions on having seen data in the input space when performing inference and prediction. In these cases, it is necessary only to model the function at places where data have been observed, or at predictive query locations. In density modeling, however, the places with low density are just as important to the model as those with high density. Unfortunately, it is unlikely to have observed data in regions with low density, so a representation of the function only at locations where there are data is not adequate for the inference we wish to perform. One might think of defining a density as analogous to putting up a tent: pinning the canvas down with pegs (or stakes) is just as important as putting up poles. In exchange sampling, the “pegs” are inferred implicitly as rejections along the way to generating fantasy data. At each exchange sampling step, a new tent is constructed — complete with its own pegs — and asked to explain the data. In the latent history model, however, the tent is modified one piece at a time: pegs and poles are inserted, removed, and adjusted gradually to explain the data.

It is possible also to see that the latent history model is likely to require fewer samples from the Gaussian process as it proceeds. Consider the Gaussian process density sampler: when the latent history method is at equilibrium, its state will have some typical number of latent rejections M . This is about the same number of rejections as would be expected to occur during an exchange sampling fantasy. However, to find the acceptance ratio in exchange sampling it is *also* necessary to evaluate against the observed data after fantasising. This means that the Gaussian process in exchange sampling requires at least $2N + M$ evaluations to make a Metropolis–Hastings move, while the latent history method requires only $N + M$. This does not even consider the expansion of state that occurs when exchange sampling rejects a proposal, and additional fantasy data are incorporated into the Markov state. As the time complexity of computation in the Gaussian process grows cubically in the number of data, exchange sampling can become rapidly more expensive.

Another reason that the latent history method is preferable to exchange sampling is that it requires less bookkeeping about the function $g(\mathbf{x})$. The

state of the exchange sampling Markov chain is the uncountably-infinite object $g(\mathbf{x})$. The innovation of the method is that through retrospective sampling we are able to make Metropolis–Hastings moves with only a finite number of computations. This retrospective sampling, however, means that information discovered about a particular $g(\mathbf{x})$ must be retained for as long as that function is relevant to the current Markov state. In contrast, the state of the Markov chain when performing latent history inference only includes $g(\mathbf{x})$ at the latent rejections or thinned events. That is, rather than an uncountably-infinite object $g(\mathbf{x})$, the Gaussian process in the latent history model conditions on a finite set of points in the input space. This means that the values of the function do not need to be kept in memory, except for at the data and at the locations of the rejections or thinned events. This contrast can also be seen in the difference between the joint distributions that describe the two inference methods for the Gaussian process density sampler. In exchange sampling, when writing Equation 4.3, we use \mathbf{g} to denote $g(\mathbf{x})$ as an infinite vector. When writing the posterior distribution on the latent history, however, we do not need to denote an infinite function. Equation 4.11 only defines a distribution on the function values at the data and the latent rejections.

Finally, while the latent history method enables efficient Hamiltonian Monte Carlo sampling of the latent function values, it is not clear how to combine HMC with exchange sampling.

6.3. Restricting the function space. With both the exchange sampling and latent history methods, incorporating fewer latent rejections (“tent pegs”) into the Gaussian process results in improved efficiency. For a given $g(\mathbf{x})$, the expected number of rejections is $N(\mathcal{Z}_\pi[g]^{-1} - 1)$. This expression is derived from the observation that $\pi(\mathbf{x} | \psi)$ provides an upper bound on the function $\Phi(g(\mathbf{x})) \pi(\mathbf{x} | \psi)$ and the ratio of acceptances to rejections is determined by the proportion of the mass of $\pi(\mathbf{x} | \psi)$ contained by $\Phi(g(\mathbf{x})) \pi(\mathbf{x} | \psi)$. One problem with inference is that there are many functions $g(\mathbf{x})$ that can explain the data equivalently, as $\Phi(g(\mathbf{x})) \pi(\mathbf{x} | \psi)$ is unnormalised. Many of these $g(\mathbf{x})$ will cause $\Phi(g(\mathbf{x}))$ to be close to zero, resulting in many rejections. The Gaussian process prior might only provide weak regularisation to prevent this.

One way to improve this situation is to require that the function $g(\mathbf{x})$ be pinned to zero for some \mathbf{x}_0 . This prevents $\Phi(g(\mathbf{x})) \pi(\mathbf{x} | \psi)$ from being small everywhere and reduces the redundancy in the prior that occurs due to normalisation. We use the base density $\pi(\mathbf{x} | \psi)$ as a prior on \mathbf{x}_0 and treat it as a hyperparameter for the Gaussian process. We can then use the

inference methods of Sections 4.1.2 and 4.2.4 to infer an appropriate \mathbf{x}_0 .

6.4. *The logistic Gaussian process.* The Gaussian process is an appealing prior on functions due to the ability to specify the smoothness and differentiability properties of sample realizations via a covariance function, without choosing an explicit set of basis functions. This flexibility and intuition has led to interest in applying Gaussian processes to density modeling via the logistic Gaussian process introduced by Leonard [1978] and further developed by Lenk [1988, 1991]. If $g(\mathbf{x})$ is a random function drawn from a Gaussian process, then the logistic GP arrives at a density $f(\mathbf{x})$ on a closed interval \mathcal{I} via

$$(6.1) \quad f(\mathbf{x}) = \frac{e^{g(\mathbf{x})}}{\int_{\mathcal{I}} e^{g(\mathbf{x})}}$$

for $\mathbf{x} \in \mathcal{I}$. On a bounded interval and with minor constraints on the covariance function, the integral in the denominator exists and the density is well-defined Tokdar [2007]. The distribution on densities is closed under Bayesian updating. As in Equation 2.1, however, it is generally impossible to integrate an infinite-dimensional random function and so likelihood-based calculations are intractable. The GP-based prior we have presented in this paper allows exact inference computation despite this intractability by constructing a generative model, but no such method is known for the logistic Gaussian process.

In order to perform inference with the logistic Gaussian process, several finite-dimensional approximations have been introduced. Lenk [1991, 2003] proposes an approximation of the logistic GP by a truncated Karhunen–Loève expansion evaluated on a grid. Tokdar [2007] uses a finite-dimensional approximation to the logistic Gaussian process by parameterizing the function values on a grid and then imputing other values from the conditional mean. The normalisation constant is estimated via a numeric method, e.g., the trapezoidal rule or Simpson’s rule.

The approach of expanding the density as a finite Fourier series, as described by Lenk [2003], is appealing in a single dimension as one can parameterize the function in terms of coefficients with independent Gaussian priors. The variances of these priors arise directly from the Gaussian process covariance function. As noted by Lenk [2003], however, the number of Fourier coefficients required grows exponentially with dimension. Finding higher-dimensional bases that are rich enough to express interesting structure while also allowing efficient computation is considered an open problem.

The imputation method of Tokdar [2007] extends to the multivariate case more straightforwardly. While a lattice does not scale well to many dimen-

sions, the imputation approximation does not necessarily require a grid. Tokdar [2007] proposes a method of inferring appropriate knot locations and explores this on a two-dimensional test problem using reversible jump Markov chain Monte Carlo [Green, 1995]. This has a similar motivation to the model presented in this paper: adapt the parameterization of the Gaussian process as the data demands. The GPDS achieves this via a fully-nonparametric generative model, Tokdar [2007] specifies a finite-dimensional surrogate model with the dimensionality selected as a part of inference. Additionally, it is unclear in Tokdar [2007] how the normalization constant is to be effectively estimated when the knots are irregularly arranged. It is suggested to perform imputation to a grid from the known knots, but this reintroduces some aspects of the problems of lattices in high dimensions. In contrast, the GPDS inference discussed in the present paper explicitly avoids these problems by performing computation without evaluating $\mathcal{Z}_\pi[\mathbf{g}]$.

Acknowledgements. The authors wish to thank Radford Neal and Zoubin Ghahramani for valuable comments.

References.

- A. Beskos, O. Papaspiliopoulos, G. O. Roberts, and P. Fearnhead. Exact and computationally efficient likelihood-based estimation for discretely observed diffusion processes (with discussion). *Journal of the Royal Statistical Society: Series B*, 68:333–382, 2006.
- S. Chib and I. Jeliazkov. Marginal likelihood from the Metropolis–Hastings output. *Journal of the American Statistical Association*, 96(453):270–281, 2001.
- L. Csató. *Gaussian processes - iterative sparse approximations*. PhD thesis, Aston University, Birmingham, UK, March 2002.
- I. DiMatteo, C. R. Genovese, and R. E. Kass. Bayesian curve-fitting with free-knot splines. *Biometrika*, 88(4):1055–1071, 2001.
- S. Duane, A. D. Kennedy, B. J. Pendleton, and D. Roweth. Hybrid Monte Carlo. *Physics Letters B*, 195(2):216–222, 1987.
- M. D. Escobar and M. West. Bayesian density estimation and inference using mixtures. *Journal of the American Statistical Association*, 90(430):577–588, June 1995.
- T. S. Ferguson. A Bayesian analysis of some nonparametric problems. *The Annals of Statistics*, 1(2):209–230, 1973.
- N. D. Goodman, V. K. Mansinghka, D. M. Roy, K. Bonawitz, and J. B. Tenenbaum. Church: a language for generative models. In *Proceedings of the 24th Annual Conference on Uncertainty in Artificial Intelligence*, 2008.
- P. J. Green. Reversible jump Markov chain Monte Carlo computation and Bayesian model selection. *Biometrika*, 82:711–732, 1995.
- M. L. Huber and R. L. Wolpert. Likelihood-based inference for Matérn type III repulsive point processes. *Advances in Applied Probability*, 41(4), 2009. In press.
- H. Ishwaran and L. F. James. Gibbs sampling methods for stick-breaking priors. *Journal of the American Statistical Association*, 96(453):161–173, March 2001.
- M. Lavine. Some aspects of Pólya tree distributions for statistical modelling. *Annals of Statistics*, 20(3):1222–1235, 1992.

- M. Lavine. More aspects of Pólya tree distributions for statistical modelling. *Annals of Statistics*, 22(3):1161–1175, 1994.
- P. J. Lenk. The logistic normal distribution for Bayesian, nonparametric, predictive densities. *Journal of the American Statistical Association*, 83(402):509–516, 1988.
- P. J. Lenk. Towards a practicable Bayesian nonparametric density estimator. *Biometrika*, 78(3):531–543, 1991.
- P. J. Lenk. Bayesian semiparametric density estimation and model verification using a logistic-Gaussian process. *Journal of Computational and Graphical Statistics*, 12(3):548–565, 2003.
- T. Leonard. Density estimation, stochastic processes and prior information. *Journal of the Royal Statistical Society, Series B*, 40(2):113–146, 1978.
- A. Y. Lo. On a class of Bayesian nonparametric estimates: I. density estimates. *The Annals of Statistics*, 12(1):351–357, March 1984.
- D. J. C. MacKay. Bayesian interpolation. *Neural Computation*, 4(3):415–447, 1992.
- J. Møller, A. N. Pettit, R. Reeves, and K. K. Bethelsen. An efficient Markov chain Monte Carlo method for distributions with intractable normalising constants. *Biometrika*, 93(2):451–458, 2006.
- I. Murray. *Advances in Markov chain Monte Carlo methods*. PhD thesis, Gatsby Computational Neuroscience Unit, University College London, London, 2007.
- I. Murray, Z. Ghahramani, and D. J. C. MacKay. MCMC for doubly-intractable distributions. In *Proceedings of the 22nd Annual Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 359–366, 2006.
- R. M. Neal. Defining priors for distributions using Dirichlet diffusion trees. Technical Report 0104, Department of Statistics, University of Toronto, 2001.
- R. M. Neal. Density modeling and clustering using Dirichlet diffusion trees. In *Bayesian Statistics 7*, pages 619–629, 2003.
- A. O’Hagan. Curve fitting and optimal design for prediction. *Journal of the Royal Statistical Society, Series B*, 40:1–42, 1978.
- O. Papaspiliopoulos and G. O. Roberts. Retrospective Markov chain Monte Carlo methods for Dirichlet process hierarchical models. *Biometrika*, 95(1):169–186, 2008.
- J. Pitman and M. Yor. The two-parameter Poisson-Dirichlet distribution derived from a stable subordinator. *Annals of Probability*, 25:855–900, 1997.
- J. G. Propp and D. B. Wilson. Exact sampling with coupled Markov chains and applications to statistical mechanics. *Random Structures and Algorithms*, 9(1&2):223–252, 1996.
- C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning*. MIT Press, Cambridge, MA, 2006.
- D. Thorburn. A Bayesian approach to density estimation. *Biometrika*, 73(1):65–75, 1986.
- S. T. Tokdar. Towards a faster implementation of density estimation with logistic Gaussian process priors. *Journal of Computational and Graphical Statistics*, 16(2):1–23, 2007.
- S. T. Tokdar and J. K. Ghosh. Posterior consistency of logistic Gaussian process priors in density estimation. *Journal of Statistical Planning and Inference*, 137:34–42, 2007.

RYAN P. ADAMS, IAIN MURRAY
 DEPARTMENT OF COMPUTER SCIENCE
 UNIVERSITY OF TORONTO
 10 KING’S COLLEGE ROAD
 TORONTO, ONTARIO M5S 3G4, CA
 E-MAIL: rpa@cs.toronto.edu
murray@cs.toronto.edu

DAVID J.C. MACKAY
 CAVENDISH LABORATORY
 UNIVERSITY OF CAMBRIDGE
 MADINGLEY ROAD
 CAMBRIDGE CB3 0HE, UK
mackay@mrao.cam.ac.uk