METROPOLIS-HASTINGS OPTIMIZATION FOR GAUSSIAN PROCESSES

GAL AV-GAY

Date: April 18, 2016.

Contents

1. Abstract

The Metropolis-Hastings algorithm [3] [5] is often used to obtain Markov Chian Monte Carlo samples from highly complex posterior distributions, such as those involved in full bayesian inference of hyper-parameters in Gaussian Process models. Here we compare four implementations of the Metropolis-Hastings algorithm in the context of sampling from the posterior distributions of hyper-parameters in a Gaussian Process. The implementations are compared in terms of their initialization biases and convergence rates, as well as in terms of their performance on higher dimensional data. Our experiments involve sampling from GP posterior distributions using the four different implementations and comparing the quality of these samples. A discrepency measure is devised based on the Kolmogorov-Smirnov test to measure the convergence rate of each algorithm. Issues in generating MCMC samples for high dimensional data are discussed and an optimal approach to sampling is proposed. All of the algorithms in this paper are implemented in an R package called 'gpMCMC' [1].

2. Introduction

Gaussian Processes (GPs) have been shown to provide a robust and flexible fit for high-dimensional data. Using gaussian likelihoods for the hyper-parameters in the correlation function, inference in GP models is analytically tractable. Some applications, however, involve GP models with highly complex posterior distributions for the hyper-parameters. In these cases there exist algorithms for maximizing the posterior, yet sampling from these distributions remains difficult. Markov Chain Monte Carlo (MCMC) methods are often used to sample from these nonstandard multivariate distributions using the Metropolis algorithm.

This algorithm is favoured since it will work for almost any likelihood function. Unfortunately, it often requires a long time to converge to the desired stationary distribution. In addition, it is difficult to quantify the initialization bias of the metropolis algorithm, i.e we don't know how long we need to run our algorithm before it converges to the true distribution that we seek.

In *"Optimizing and Adapting the Metropolis Algorithm"* [6], Jeffrey S. Rosenthal reviews a number of different approaches for optimizing the Metropolis algorithm. This review is primarily concerned with the Goldilocks principle in relation to Metropolis algorithm step lengths. Rosenthal shows that if the step length in the Metropolis algorithm is too small, the Markov chain will not do a good job of exploring the target density, and will require many iterations to properly converge to the target density. Alternatively, if the step-length is too large, most proposals are not accepted, and once again many iterations are required for convergence to the target density. By the Goldilocks-Principle, the algorithm is optimized by choosing a generating density (step-length) that results in a moderate acceptance rate. This is also based on a weak asymptotic result [4] where a random walk MCMC is shown to converge to a Langevin diffusion process, for which optimization is achieved analytically. Remarkably, the paper finds that the optimal acceptance rate for a Metropolis algorithm with a particular multivariate-normal proposal density is ≈ 0.234, in the sense that this acceptance rate will result in the quickest convergence to the stationary distribution, asymptotically speaking.

The efficiency of our Markov chain is dependent on more than just the acceptance rate. The shape of the proposal distribution is also crucial. *Roberts and Rosenthal (2001)* [8] show that the optimal shape

for the proposal distribution is defined by the covariance matrix for the parameters of the model. In cases where it is difficult to approximate this covariance matrix, adaptive MCMC methods are used to iteratively approximate the covariance matrix of the parameters from the samples as they are being generated by the Metropolis algorithm. However, these adaptive algorithms require additional regularity conditions to be satisfied in order to guarantee convergence, due to the non-homogeneity of the markov-chain. As an alternative we consider the laplace approximation to this covariance matrix and discuss its drawbacks, which are more evident at higher dimensions.

Since our focus is on hyper-parameter inference for Gaussian Processes, we investigate four different methods for sampling from the log-posterior distribution of such GPs, where each method is characterized by its proposal density. The first two implementations involve univariate step lengths, such that each dimension of the candidate sample in the Metropolis algorithm is generated separately. These include uniformly distributed step lengths and normally distributed step lengths. The second two implementations involve multivariate steps, the first of which assumes independence between each hyper-parameter and the second using a laplace approximation to the covariance of the posterior distribution of hyper-parameters as the covariance of the proposal density. The optimality of the latter of these methods (involving the Laplace approximation) is based on a result in Roberts et al. [8]. We optimize the different implementations by the goldilocks principle. In the case of the independent multivariate normal proposal density, doing so simply requires choosing a step-length that results in an acceptance rate of approximately 0.234.

In order to assess the convergence rates of the different implementations, we devise a measure of the discrepancy between two multivariate empirical distributions. This measure is based on the Kolmogorov-Smirnov test. Using this discrepancy measure we compare the convergence rates of different MH algorithm implementations. This involves generating samples from the posterior density of the hyper-parameters conditional on some data, and then comparing the discrepancy between these samples and a gold standard.

We perform the aforementioned simulations using both the G-Protein and Borhole data, however we run into some issues when the Laplace approximation is used to approximate the covariance of the posterior-distribution of hyper-parameters for the Borhole data. The G-Protein data results in well-behaved Laplace approximations to the covariance of the hyper-parameter distribution. However, the Borhole data yields some flat profile likelihoods, which result in numerically-singular covariance matrices.

2.1. **The Metropolis Algorithm.** Given an arbitrary starting point, the M-H algorithm draws samples from a Markov chain that converges over time to a desired posterior distribution $\pi = p(\theta|Y) \propto p(Y|\theta)p(\theta)$. Given the current estimate θ^i, a new proposal θ^* will be generated via a transition kernel $q(\cdot|\theta^i)$. The next state θ^{i+1} is defined given the current state and the kernel:

$$\theta^{i+1} = \begin{cases} \theta^* & \text{with probability } \alpha \\ \theta^i & \text{with probability } 1-\alpha \end{cases}$$

where α is equal to:

$$\alpha = min\left\{1, \frac{p(Y|\theta^*)p(\theta^*)q(\theta^i|\theta^*)}{p(Y|\theta^i)p(\theta^i)q(\theta^*|\theta^i)}\right\}$$

It is important to note that the proposal density, as defined by the kernel, $q(\cdot|\theta^i)$, must satisfy certain regularity conditions required for convergence of the Markov chain. These conditions are irreducibility and aperiodicity [7]: meaning that if x and y are points in the domain of the target density, there must be a non-zero probability of moving from x to dy in a finite number of iterations. These conditions are often but not exclusively satisfied when the proposal distribution has a positive density on the same support as the target density.

The efficiency of the algorithm, i.e. the convergence rate of the Markov chain, is dependent on both the shape and spread (step-length) of the proposal density. Both of these factors influence the acceptance rate α. Roberts et al. (1997) [4] assess the asymptotic properties of an independent multivariate normal proposal density:

$$Q = \mathcal{N}(0, \frac{l^2}{d} I_d) \tag{1}$$

Where d is the number of covariates and l is a scaling constant that we refer to as the step-length. Roberts et al. evaluate the speed of the algorithm as a function of this step-length as $d \to \infty$, as well as the asymptotic acceptance rate as a function of the step-length. Furthermore, they show that there is relationship between the speed and the acceptance rate, such that the optimal acceptance rate is 0.234 for an independent multivariate normal proposal density as above (1). Different implementations of the Metropolis algorithm exist that use other proposal distributions. One option, for example, is to make univariate steps. A possible advantage of this method is that the step length can

be optimized independently for each covariate, although doing so may be difficult as it is possible that optimization of the step length for one covariate depends on the chosen step length for other covariates; this, however, requires further investigation. Another drawback is that the posterior density must be computed more often.

The shape of the proposal distribution is also important. Roberts and Rosenthal (2001) [8] prove under strong assumptions that there exists an optimal Gaussian proposal distribution:

$$Q = \mathcal{N}(0, \frac{2.38^2}{d}\Sigma_\pi) \tag{2}$$

Such that Σ_π is the covariance matrix of the target density, π. Furthermore, it was shown that the acceptance rate under this proposal distribution is approximately 0.234. This method therefore has a distinct logistical advantage over others in that there is no need to optimize its step length in order to achieve the desired acceptance rate. A disadvantage is that the covariance matrix of the target density is often unknown. Attempts have been made to approximate the covariance matrix iteratively via something called Adaptive MCMC [9], however this method requires additional regularity conditions for convergence. Instead, we consider a laplace approximation to the covariance matrix of the target distribution, which is computed for the Gaussian Process posterior distribution in Section 5. Computational issues are sometimes observed and are discussed later.

2.2. **Metropolis Hastings MCMC for Gaussian Processes.** Consider n replications of k dimensional data. In this report we consider

the case of Gaussian Processes (GPs) with Gaussian correlation functions between pairs of points:

$$R(x, x^{'}) = exp\left\{-\sum_{i=1}^{k} \theta_i |x_i - x_i^{'}|^2\right\}$$

The goal here is to simulate from the distribution of these θ hyper-parameters in order to do inference and prediction. Using a prior on the θ hyper-parameters $p(\theta)$, the posterior distribution of these GP hyper-parameters is as follows:

$$p(\theta|y) \propto \frac{\pi(\theta)}{(\widehat{\sigma}^2)^{(n-k)/2} det^{1/2}(\mathbf{R}) det^{1/2}(F^T\mathbf{R}^{-1}F)} \tag{3}$$

where the maximum likelihood estimate of the variance is :

$$\widehat{\sigma^2} = \frac{y^T\mathbf{R}^{-1}y - y^T\mathbf{R}^{-1}F(F^T\mathbf{R}^{-1}F)^{-1}F^T\mathbf{R}^{-1}y}{n-k} \tag{4}$$

Where F is the design matrix. For example, in the case of a constant regression model, F would be a column-vector of ones.

Computing the posterior density here requires an evaluation of the covariance function, which is on the order of $O(kn^2)$, and an inversion of this covariance function costs on the order of $O(n^3)$. This posterior density needs to be evaluated at every iteration of our M-H algorithm. This poses a disadvantage to univariate methods in that they may require up to k times as many iterations in order to achieve similar convergence rates as methods that involve multivariate proposal densities.

2.3. **Implementations of the Metropolis Algorithm.** In this report we consider four separate implementations of the Metropolis Algorithm and compare their convergence rates via a series of Kolmogorov-Smirnov [2] like tests. Each of the implementations is differentiated by its specific proposal density:

(1) Univariate Uniform;
(2) Univariate Normal;
(3) Multivariate Gaussian using an Independent Covariance Matrix (1);
(4) Multivariate Gaussian using the Laplace approximation to the Covariance Matrix (2).

Assessment of these implementations was performed using three data-sets of increasing dimension. Here we outline how each of these implementations were optimized and compared.

For the univariate proposal densities, we attempt to optimize the step length so that it corresponds to an acceptance rate between 0.2 and 0.4. Although there is no asymptotic result that suggests this acceptance rate is optimal for the given proposal densities, we are basing this loose optimization on the *Goldilocks*-Principle, as it was discussed in *Rosenthal* (2014) [6]. Optimization was also performed on proposal density (2) by finding the step length that corresponds to an acceptance rate of 0.234. Proposal density (4) did not require previous optimization as its optimality is defined by the asymptotic result in [8]. The laplace approximation requires computation of the second derivative of the posterior distribution with respect to the theta parameters, evaluated at the approximate posterior mode of the posterior distribution. Gold standard simulations are generated for each data-set, and the

convergence rates of each of the implementations are compared using a measure of discrepancy between each sample and the gold-standard.

3. Measuring the discrepancy between two Empirical Distributions

A 'gold standard' MCMC sample is obtained by running the MH algorithm for a very large number of iterations (1 hundred million). The empirical density of this gold standard is designated as being arbitrarily different from that of the target distribution. The discrepancy of a sample from the gold standard is obtained in order to assess the convergence rates of the different MH implementations to the target density. We therefore require a measure of the discrepancy between two multivariate empirical densities.

One option would be to choose the Kolmogorov-Smirnov test as our measure, which considers the maximum difference between two empirical cumulative distribution functions. This would require considering each dimension separately. This, however, does not take the correlation between the hyper-parameters into account. We formulate a measure of the discrepancy between two two-dimensional empirical cumulative distribution functions based on the Kolmogorov-Smirnov test. Evaluating this discrepancy for a particular pair of dimensions involves taking a grid over the shared domain of the samples and evaluating the empirical cumulative distribution function (CDF) at each square of the grid. The difference in the empirical CDF for each square is taken between the gold standard and the sample, and the maximum difference in these proportions over all grid boxes is taken to be the discrepancy between the two 2-dimensional distributions. Given two sets of samples of k dimensions, we measure the maximum discrepancy for each pair

of dimensions separately and report the maximum discrepancy over all the pairs. This discrepancy value will therefore be between 0 and 1.

For any two disjoint empirical distributions the discrepancy will be at its maximum of 1. The discrepancy between any two samples with substantially different means will be reasonably high (higher than 0.1). In addition, the discrepancy between samples from differently shaped distributions is also expected to be quite high. The question remains as to what is a reasonable value for this discrepancy between two samples that follow a similarly shaped distribution. This 'reasonable value' should depend on the nature of the distribution.

In order to assess the performance of our measure, we simulate two samples of size 100000 from bivariate normal distributions. One distribution has an independent covariance matrix with variances of 1, while the other has covariances of 0.5. Given ten pairs of these samples, the average discrepancy between each pair samples was found to be 0.079, whereas the average discrepancy between two samples with identical covariance matrices was found to be 0.002. This gives us a perspective on the range of discrepancy values between two similarly distributed samples. Furthermore, we find the specific grid square at which this maximum difference of 0.08 is observed. We find that for two bivariate distributions with equal means, this maximum is observed at the center. This is expected as the value of the CDF at the center grid point is over the top left quadrant of the grid. Evaluating the differences between the upper left quadrants of the grid between two empirical CDFs seems to most accurately identify differences in the covariance matrix.

FIGURE 1. Samples from two different bivariate normal distributions with identical means. On the left the covariance between the two dimensions is 0.5 and on the right it is 0. The discrepancy between these two samples is 0.078.

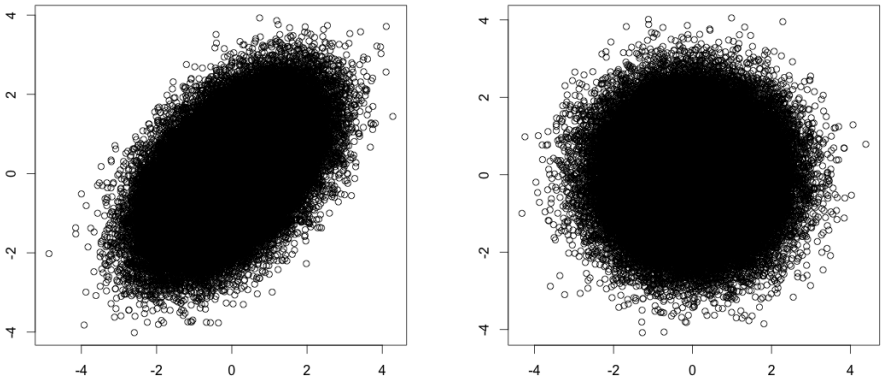
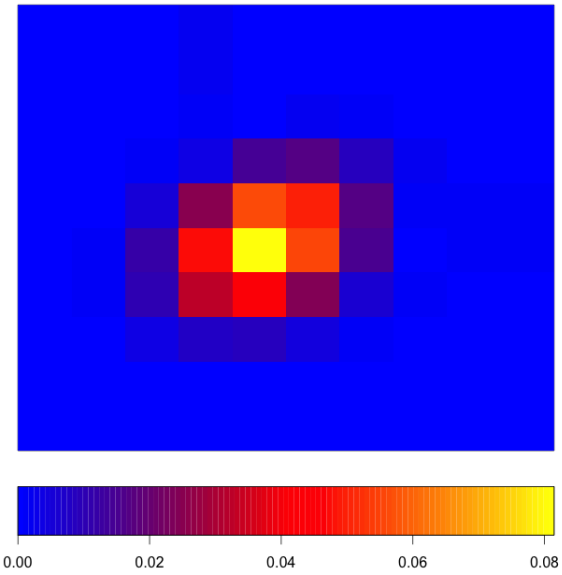


FIGURE 2. Heat-map of the differences in CDF values between grid points for the two samples above illustrated in Figure 1. We find that the maximum difference of 0.08 is at the center.



4. Results

First we investigate the convergence rates of the four MH implementations using the G-Protein data. We highlight a computational issue that arises for higher dimensional data using a simple example involving a fabricated data-set, and propose methods for circumventing this issue. Finally we evaluate the performance of our four implementations on the Borhole data, for which this computational issue arises. Recall that our four implementations are: 1) Univariate Uniform, 2) Univariate Normal, 3) Multivariate Gaussian with and independent covariance matrix, 4) Multivariate Gaussian using the Laplace approximation to the covariance. We will refer to these methods as 1) uniform, 2) normal, 3) multivariate-normal, 4) laplace.

4.1. **G-Protein.** We simulate using MCMC from the posterior of a Gaussian Process used to fit the G-Protein data-set. This is a 4-dimensional data-set where all parameters are important in the model. We use an exponential prior for the posterior distribution of the theta hyper-parameters with lambda equal to 0.1. Below we trace the acceptance rate of three of the four Metropolis algorithms we are testing and find the point at which the acceptance rate is approximately 0.234. The purpose of this is to loosely optimize these algorithms based on results discussed in *"Optimizing and Adapting the Metropolis Algorithm"* [6]. The Laplace method has a fixed proposal distribution, the optimality of which is based on results in *Roberts and Rosenthal (2001)* [8].

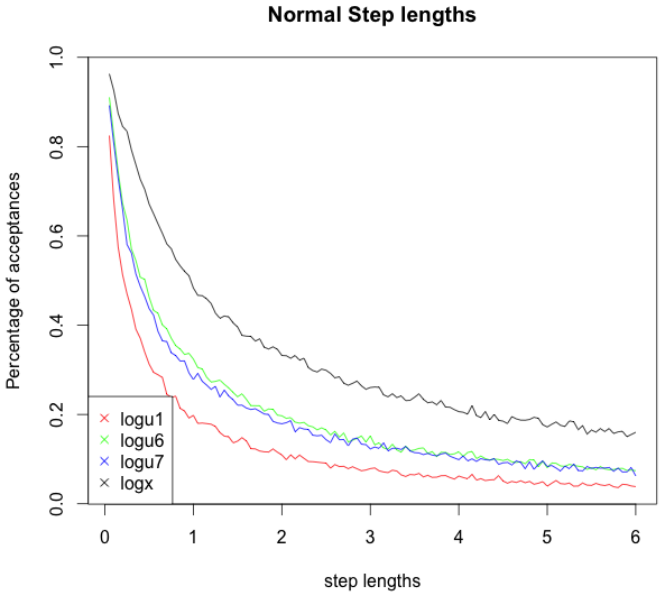
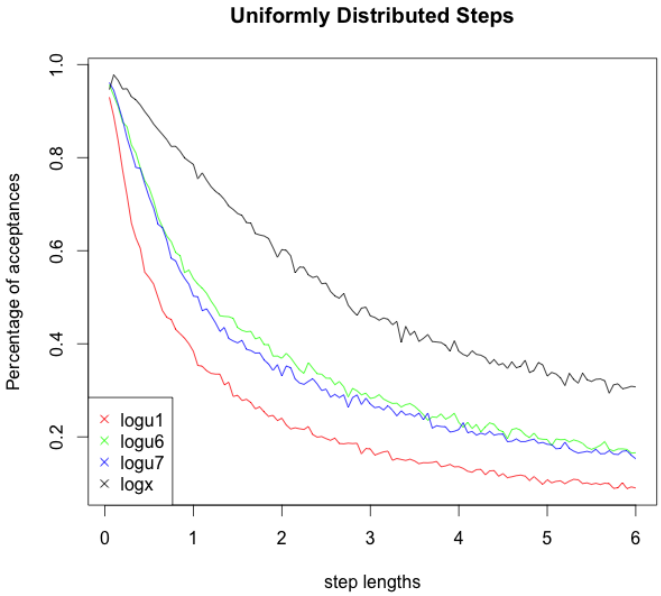
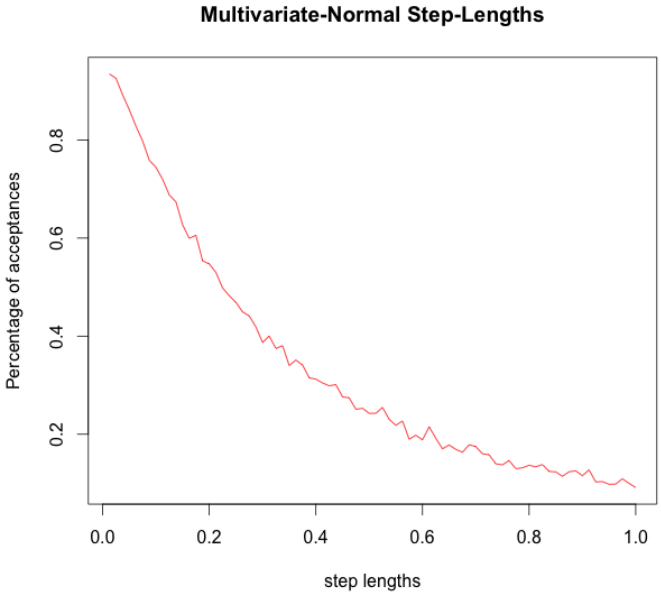


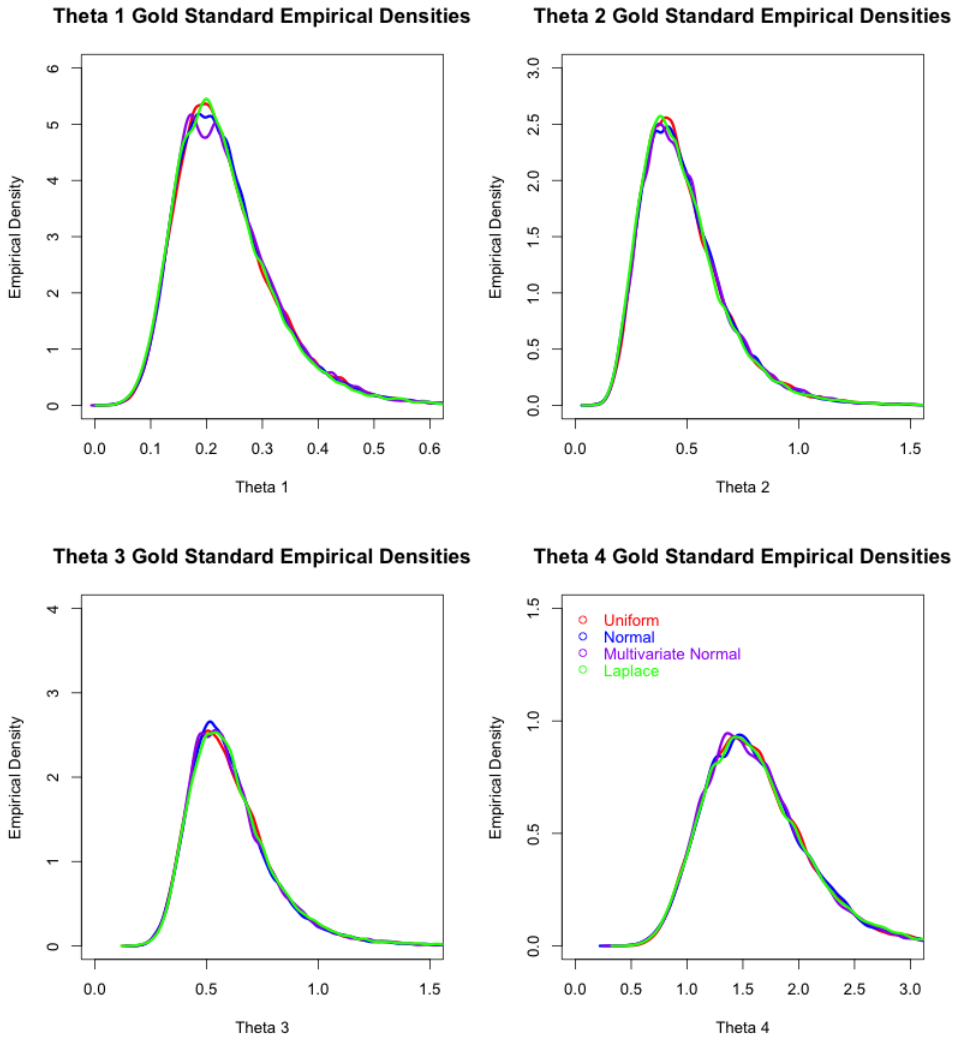
FIGURE 3. Acceptance rate of each MH implementation against the step-length, used to optimize the step-length for each method.





Four different gold standard samples of a million samples each are obtained, one for each implementation, using an Exponential prior with a lambda value of 0.1. These samples, as well as all future G-Protein MCMC samples discussed are obtained using the optimal step-lengths found via Figure 3. We evaluate the discrepancy between all the pairs of gold standard samples and find that the maximum discrepancy observed is 0.017 between the multivariate-normal and laplace methods. We therefore use this level of discrepancy as a benchmark for good convergence. It can be seen from Figure 4 that the profile empirical density for each theta hyper-parameter is relatively identical under each implementation of the MH algorithm.

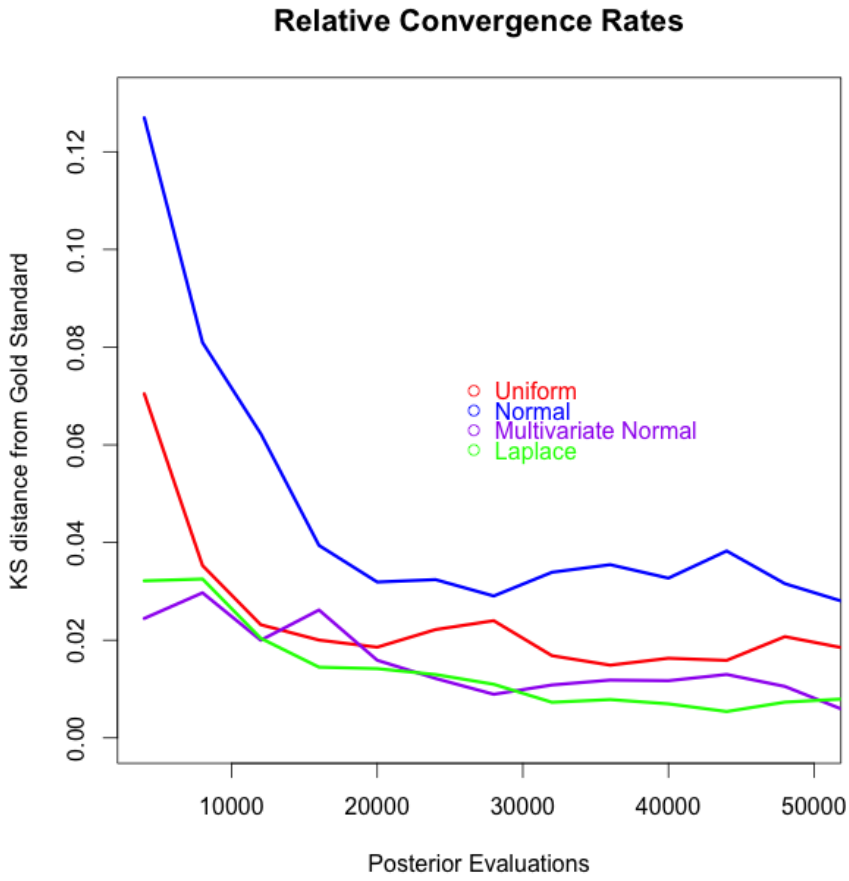
FIGURE 4. Empirical Densities for the gold standards of each Theta hyper-parameter in a GP fit of the G-Protein data.



Finally we evaluate the discrepancy of each implementation against the number of posterior density evaluations. We find, as expected, that the laplace and multivariate normal methods converge slightly faster than the univariate methods. It should be noted that for each new proposal, the univariate methods require k times as many posterior

evaluations as the multivariate methods (where k is the number of dimensions, 4 in the case of the G-Protein data). Therefore the number of samples generated for a fixed number of posterior evaluations is k times higher for the multivariate methods compared to the univariate methods.

FIGURE 5. Convergence rate for each implementation in terms of our devised KS distance against the number of posterior evaluations.

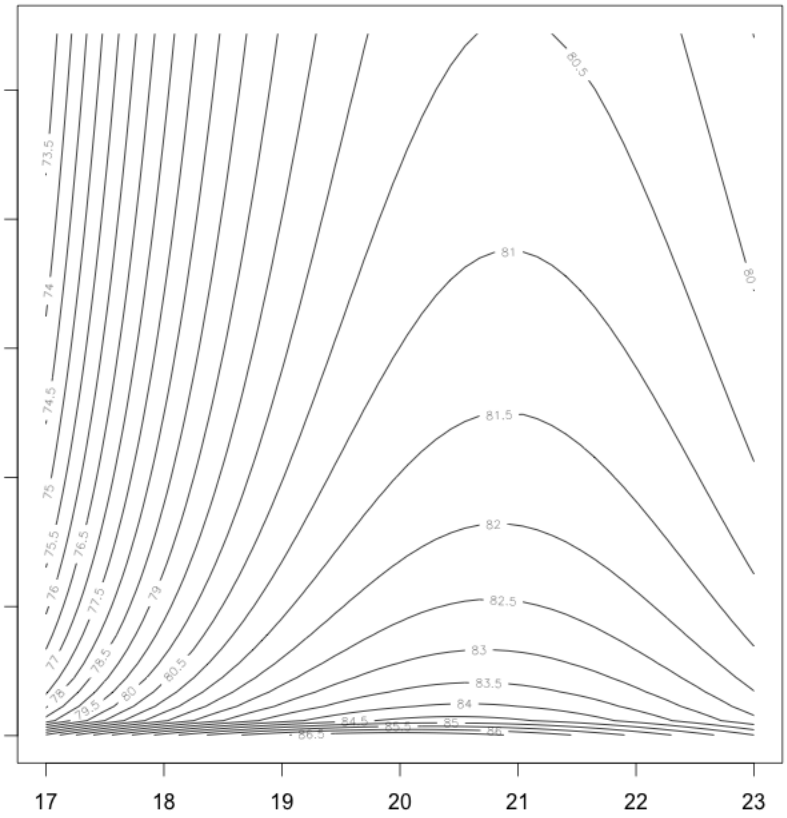


4.2. **Simple Example.** For high dimensional data-sets, we are more likely to observe that some variables have a negligible effect on our model. For these variables, the GP fit for the hyper-parameter theta

will be close to zero, resulting in a non-Gaussian profile likelihood for there parameters. Below we highlight a simple 2-dimensional example with one relevant variable and one that has no effect on the response.

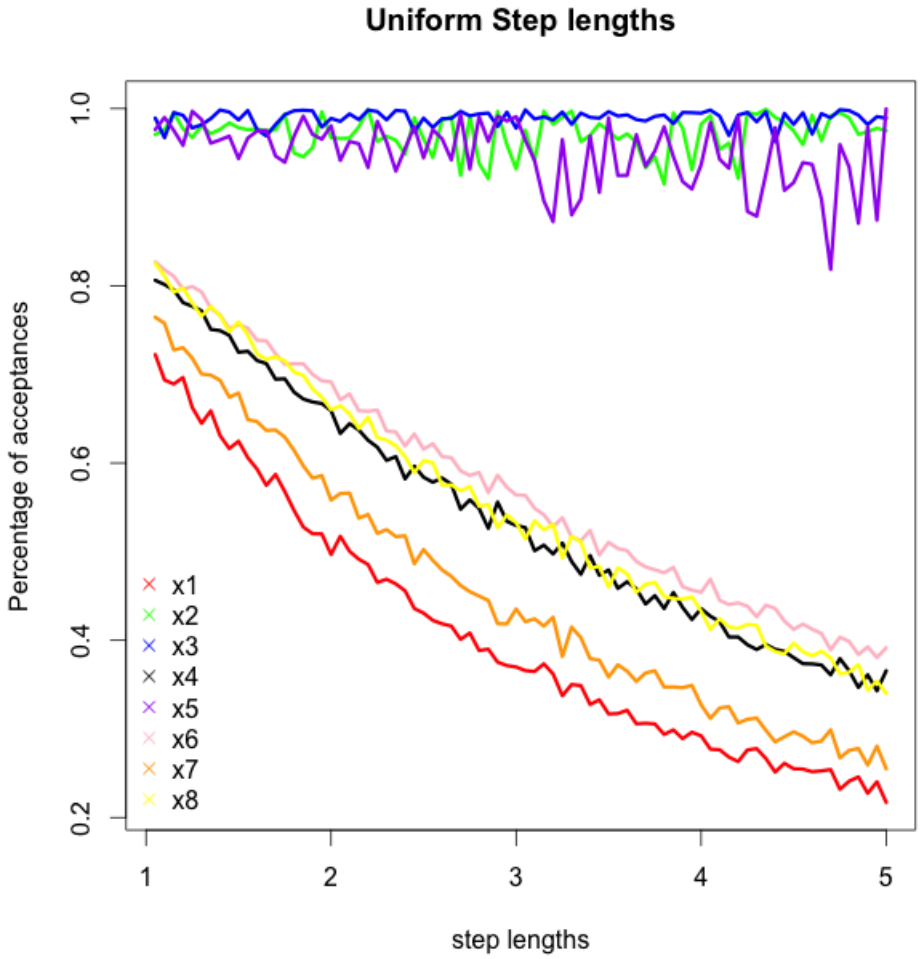
The uniform, normal, and multivariate methods will simulate proposals as before. The laplace approximation will yield a non positive semi-definite matrix due to the non-Gaussian shape of the likelihood. This prevents us from simulating from a multivariate normal distribution with the covariance matrix given by the laplace approximation. This is the main drawback of this method. Fortunately we can use variable selection to avoid this issue by setting a threshold for the theta hyper-parameters, such that all parameters with a value lower than the threshold will be simulated separately using one of the other MH implementations, or otherwise not simulated at all. The approach taken towards simulating these near-zero thetas is not so crucial as they will have little effect on the model.

FIGURE 6. Likelihood for a simple example exhibiting a non-Gaussian likelihood with the true value of Theta 1 being 20 and the true value of Theta 2 being 0.



4.3. **Borhole.** The borehole data is 8-dimensional, however, three of the variables have little effect on the model and have relatively flat likelihoods. This can be seen from the acceptance rate versus step length plot for the uniform MH method, where three variables have a very high acceptance rate regardless of the step length.

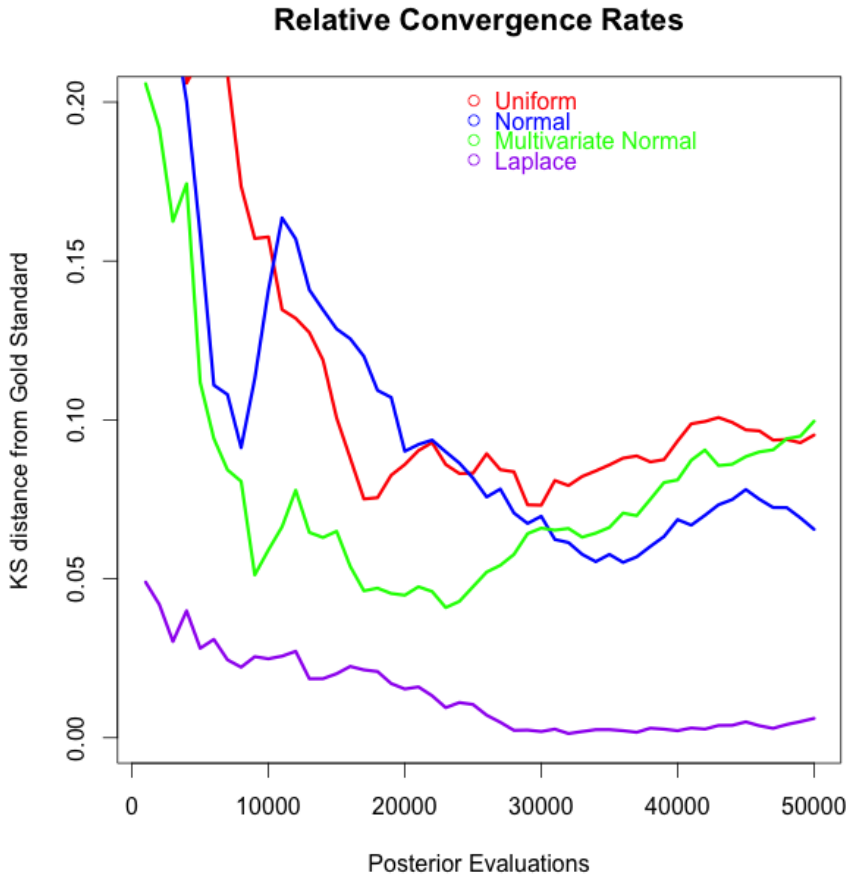
FIGURE 7. Acceptance rates versus step length for variables simulated using the univariate uniform MH method. Three variables have distinctly flat likelihoods, x_2, x_3, and x_5.



In order for the laplace approximation of the covariance to work, we subset the data containing variables 1,4,6,7,8 and determine the laplace approximation to the covariance for these variables only. Samples for the remaining variables (2,3, and 5) are generated using the uniform method. Details of how these variables are sampled are not important

as they have little effect on the model.

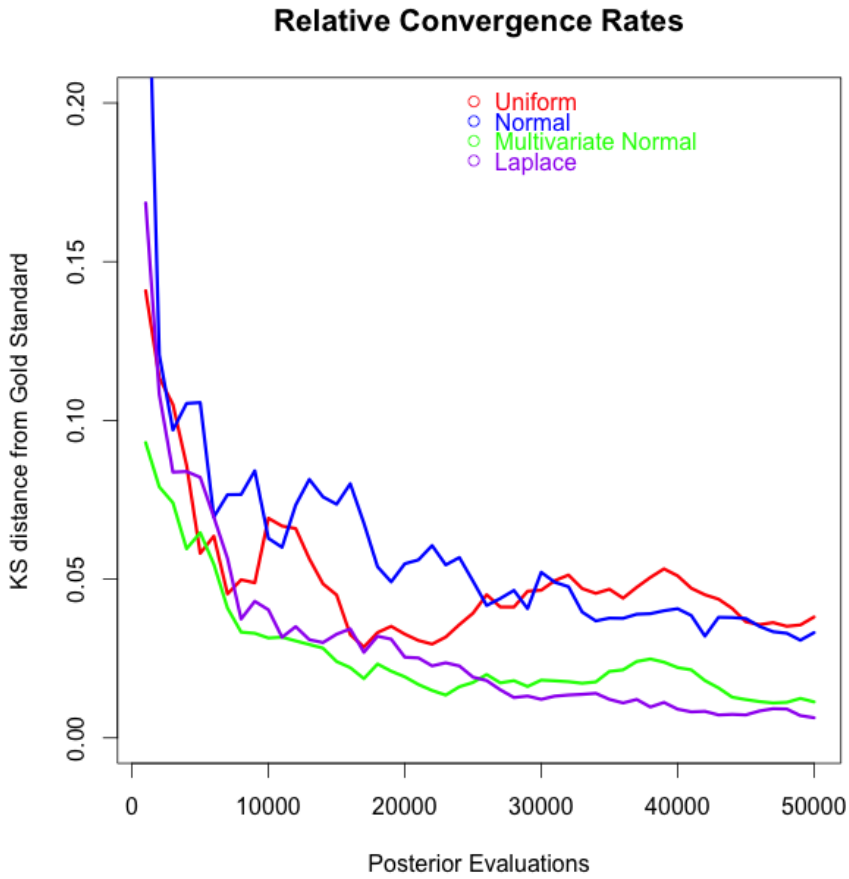
FIGURE 8. Convergence rate of each method using the borhole data, displayed as KS-distance against number of posterior evaluations. Sampling was performed separately for the set of negligible variables, x_2, x_3, and x_5 for the laplace method.



The convergence rate of each method is compared in Figure 8, where we see that the laplace method is far superior. It seems to be the case, however, that the superiority of the laplace method comes as a result of the fact that the negligible variables are being sampled

separately. This is shown in Figure 9, where we sample each set of variables separately (as was done for the laplace method in Figure 8) for all methods and compare their convergence rates. Figure 9 shows an improved convergence rate for all the other methods, with the laplace method still being superior.

FIGURE 9. Convergence rate of each method using the borhole data, displayed as KS-distance against number of posterior evaluations. Sampling was performed separately for the set of negligible variables, x_2, x_3, and x_5 for all methods.



5. Discussion

Of the four methods we have discussed, the laplace method consistently exhibits the fastest convergence. In addition, the laplace method requires significantly less optimization in that the step length is fixed by the Fisher approximation to the covariance. Finding the optimal step-length is more costly than computing the Fisher approximation to the covariance.

The laplace method runs into some issues for data where there are negligible variables, which result in near-zero theta hyper-parameters. The near-zero hyper-parameters result in non-Gaussian likelihoods, which cause the laplace method to fail due to our inability to sample from a multivariate normal distribution with a non semipositive-definite covariance matrix. We circumvent this issue by segregating the data based on which theta values are below some threshold, and conducting two separate simulations on each set of variables. Doing so allows us to implement the laplace method. Furthermore, doing so for the remaining methods also improves their performance. We are unsure why this is the case, as the likelihood for the remaining 'negligible' variables is relatively flat and most proposals are accepted.

References

[1] gpMCMC r (gnu) package. `https://github.com/galotalp/gpMCMC`. Accessed: 2016-04-20.

[2] Wayne W. Daniel. Kolmogorov-smirnov one-sample test. In *Applied Nonparametric Statistics (2nd ed.)*, pages 319–330. PWS-Kent, 1990.

[3] N. Metropolis et. al. Equations of state calculations by fast computing machines. *The Journal of Chemical Physics*, 21:1087–1092, 1953.

[4] Roberts G.O. Gelman A., Gilks W.R. Weak convergence and optimal scaling of random walk metropolis algorithms. *The Annals of Applied Probability*, 7:110–120, 1997.

[5] W. Hastings. Monte carlo sampling methods using markov chains and their applications. *Biometrika*, 57:97–109, 1970.

[6] Rosenthal S. Jeffrey. Optimizing and adapting the metropolis algorithm. In ed. J.F. Lawless, editor, *Statistics in Action: A Canadian Outlook*, chapter 6, pages 93–108. Chapman & Hall CRC, 2014.

[7] Smith A.F.M Roberts G.O. Bayesian computation via the gibbs sampler and related markov chain monte carlo methods. *Journal of the Royal Statistical Society, Ser. B*, 55:3–24, 1993.

[8] Roberts G.O. Rosenthal J.S. Optimal scaling for various metropolis hastings algorithms. *Statistical Science*, 16:351–367, 2001.

[9] Haario H. Waksman E., Tamminen J. An adaptive metropolis algorithm. *Bernoulli*, 7:223–242, 2001.