Elliptical Slice Sampling

Francesca Panero Xuewen Yu

October 18, 2017

Abstract

In this project, we presented the R package 'ESS', which provides an implementation of the Elliptical Slice Sampling method presented by Murray, Adams and MacKay in 2010 (3). We validated this algorithm on the simplest senario of Gaussian regression model and compared it with Metropolis Hastings algorithms on two Gaussian Process based likelihood models.

1 Introduction

Slice sampling was introduced by Neal in 2003 (5) in the homonym paper published in the Annals of Statistics. Neal's idea was to present a new approach to sample from a probability density function, that tried to overcome some drawbacks of the two major MCMC methods of sampling: the Gibbs sampler and the Metropolis Hastings method.

The aim of these three methods is to sample from a probability density function from which we are not able to sample directly. Their basic idea is to construct a Markov chain that have as stationary distribution the target one we care about, and samples from this chain will eventually come from the desired distribution.

The basic idea of slice sampling lies in the fact that if we want to sample x from a distribution $p(x)$ that is proportional to a certain function $f(x)$, it would be sufficient just to sample uniformly from the area below $f(x)$. By defining an auxiliary random variable y, we exploit Gibbs sampler to sample from the two full conditional distributions. In particular, $y|x$ is distributed as an $Uniform(0, f(x))$ and $x|y$ is distributed as an uniform on the so called slice $S = \{x : y < f(x)\}$. The joint density of x and y is, then

$$p(x, y) = \begin{cases} 1/Z & \text{for } 0 < y < f(x) \\ 0 & \text{otherwise} \end{cases}$$

where $Z = \int f(x)dx$. From this it can easily be seen that $p(x) = \int_0^{f(x)} p(x, y)dy = (1/Z)f(x)$ as desired.

Based on this idea, Murray, Adams and Mackay proposed Elliptical Slice Sampling for multivariate Gaussian priors and indicated the advantage of this new method over MCMC methods in terms of computational complexity and performance of the sampled Markov chains (3). This project aims to implement Elliptical Slice Sampling in R and assess the performance of Elliptical Slice Sampling on Gaussian regression model and Log Gaussian Cox process model.

2 Elliptical Slice Sampling

Elliptical Slice Sampling is a particular case of MCMC methods that avoids the tuning of parameters, simpler and often faster than other methods to sample from the posterior distribution of models with multivariate Gaussian prior.

Let $\mathbf{f}$ be the vector of latent variables distributed as a Gaussian distribution with zero vector mean and covariance matrix Σ:

$$\mathbf{f} \sim \mathcal{N}(\mathbf{f}; \mathbf{0}, \Sigma) = |2\pi\Sigma|^{-1/2} \exp\left(-\frac{1}{2}\mathbf{f}^T\Sigma^{-1}\mathbf{f}\right)$$

Let

$$L(\mathbf{f}) = p(\text{data}|\mathbf{f})$$

be the likelihood function. Our target distribution is the posterior of this model:

$$p^*(\mathbf{f}) \propto \mathcal{N}(\mathbf{f}; \mathbf{0}, \Sigma) L(\mathbf{f}).$$

Neal, in 1999 (4), introduced the idea of a Metropolis-Hastings algorithm for this type of problems, by proposing as new state

$$\mathbf{f}' = \sqrt{1-\epsilon^2}\mathbf{f} + \epsilon\boldsymbol{\nu}, \qquad \boldsymbol{\nu} \sim \mathcal{N}(\mathbf{0}, \Sigma)$$

where $\epsilon \in [-1, 1]$ is a step-size parameter, and its varying defines half of an ellipse, having $\boldsymbol{\nu}$ fixed. When $\epsilon = 0$ we are resampling the same value. The probability of accepting the move is

$$p(\text{accept}) = \min(1, L(\mathbf{f}')/L(\mathbf{f}))$$

otherwise the next state is still $\mathbf{f}$. A drawback of this algorithm is that ϵ needs to be tuned so that the Markov chain can mix efficiently. A representation of the whole ellipse gives a richer choice of updates, so we will exploit:

$$\mathbf{f}' = \boldsymbol{\nu}\sin\theta + \mathbf{f}\cos\theta$$

that represents the ellipse centered in the origin of the axes and passing through $\mathbf{f}$ and $\boldsymbol{\nu}$, and θ is a new step-size parameter.

To avoid the tuning of the parameter, the idea is to augment the prior to make θ a random variable and sample it exploiting the slice sampling. The augmented model becomes

$$\begin{aligned}
\boldsymbol{\nu}_0 &\sim \mathcal{N}(\mathbf{0}, \Sigma) \\
\boldsymbol{\nu}_1 &\sim \mathcal{N}(\mathbf{0}, \Sigma) \\
\theta &\sim \text{Uniform}[0, 2\pi] \\
\mathbf{f}' &= \boldsymbol{\nu}_0\sin\theta + \boldsymbol{\nu}_1\cos\theta
\end{aligned}$$

which assures that the distribution of $\mathbf{f}$ is always $\mathcal{N}(\mathbf{0}, \Sigma)$. The new target distribution is

$$p^*(\boldsymbol{\nu}_0, \boldsymbol{\nu}_1, \mathbf{f}) \propto \mathcal{N}(\boldsymbol{\nu}_0; \mathbf{0}, \Sigma)\mathcal{N}(\boldsymbol{\nu}_1; \mathbf{0}, \Sigma) L(\mathbf{f}(\boldsymbol{\nu}_0, \boldsymbol{\nu}_1, \theta))$$

The slice sampling exploits the following algorithm:

Algorithm 1 ESS Algorithm

Input: $\mathbf{f}$, $\log L$
Output: $\mathbf{f}' \sim \mathbf{p}^*(\mathbf{f})$

1: Sample from $p(\boldsymbol{\nu}_0, \boldsymbol{\nu}_1, \theta | \boldsymbol{\nu}_0\sin\theta + \boldsymbol{\nu}_1\cos\theta = \mathbf{f})$:

$$\begin{aligned}
\theta &\sim \text{Uniform}[0, 2\pi] \\
\boldsymbol{\nu} &\sim \mathcal{N}(\mathbf{0}, \Sigma) \\
\boldsymbol{\nu}_0 &\leftarrow \mathbf{f}\sin\theta + \boldsymbol{\nu}\cos\theta \\
\boldsymbol{\nu}_1 &\leftarrow \mathbf{f}\cos\theta - \boldsymbol{\nu}\sin\theta
\end{aligned}$$

2: Sample θ using slice sampling on $p^*(\theta | \boldsymbol{\nu}_0, \boldsymbol{\nu}_1) \propto L(\boldsymbol{\nu}_0\sin\theta + \boldsymbol{\nu}_1\cos\theta)$
3: $\mathbf{f}' = \boldsymbol{\nu}_0\sin\theta + \boldsymbol{\nu}_1\cos\theta$

To avoid proposing $\boldsymbol{\nu}_0$ and $\boldsymbol{\nu}_1$ in every iteration of the algorithm and to shrink the range of θ after every iteration, Murray, Adams and MacKay (3) suggested a mature algorithm as follows:

Algorithm 2 Neater ESS Algorithm

Input: $\mathbf{f}$, log$L$

Output: $\mathbf{f}^{'} \sim \mathbf{p}^{*}(\mathbf{f})$

1: Sample $\boldsymbol{\nu} \sim \mathcal{N}(\boldsymbol{\nu}; \mathbf{0}, \Sigma)$ (this defines the ellipse centered in the origine passing through $\mathbf{f}$ and $\boldsymbol{\nu}$).

2: Define the slice by sampling the height y:

$$u \sim \text{Uniform}[0, 1]$$
$$\log y \leftarrow \log L(\mathbf{f}) + \log u$$

3: Define the bracket for the angles:

$$\theta \sim \text{Uniform}[0, 2\pi]$$
$$[\theta_{\min}, \theta_{\max}] \leftarrow [\theta - 2\pi, \theta]$$

4: Propose a new status $\mathbf{f}^{'} \leftarrow \mathbf{f}\cos\theta + \boldsymbol{\nu}\sin\theta$

5: **if** $\log L(\mathbf{f}^{'}) > \log y$ **then**

6: Accept: **return** $\mathbf{f}^{'}$

7: **else** Shrink the bracket:

8: **if** $\theta < 0$ **then** $\theta_{min} \leftarrow \theta$

9: **else** $\theta_{max} \leftarrow \theta$

10: $\theta \sim \text{Uniform}[\theta_{min}, \theta_{max}]$

11: Go to 4.

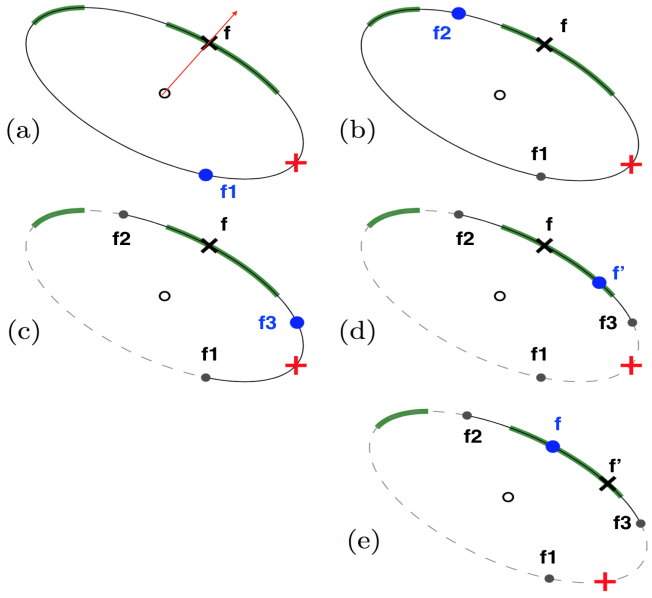

Figure 1: (a)-(d): Implementing the algorithm by inputting $\mathbf{f}$, which is denoted by black cross, and update to $\mathbf{f}'$. The red cross represents the auxiliary variable ν, green slice represents the likelihood threshold, blue points are the new proposal in each step, and solid lines are the range that θs are sampled from. (e) is the reverse procedure proposing $\mathbf{f}$ back from f', where ν' is the red cross.

An example of updating $\mathbf{f}$ to $\mathbf{f}'$ by implementing this algorithm is shown in Figure 1. $\mathbf{f}$ is the location where $\theta = 0$. The four proposals are $\mathbf{f_1}, \mathbf{f_2}, \mathbf{f_3}$ and $\mathbf{f_4}$, where $\mathbf{f_4} = \mathbf{f}'$. These states correspond to angles θ_k $k = 1, ..., 4$. The range that θs are sampled from shrinks during the implementation of the algorithm such that $\mathbf{f}$ is always included. The update stops when the proposed new state lies on the green slice, i.e. the range of likelihood satisfying step 5 in the algorithm. This process is reversible and original state $\mathbf{f}$ can be reached, as shown in (e). When we update the state from $\mathbf{f}'$, $\theta = 0$ at this point. A rotation of $-\theta_4$ will return back to the original state.

3 Experiments

In this section, we will validate the elliptical slice sampling algorithm on 2 models: Gaussian regression and Log Gaussian Cox process, and compare this algorithm with the Metropolis-Hastings algorithm adapted by Neal (4).

3.1 Model Description

3.1.1 Gaussian Regression

Observations y_n are drawn from Normal distribution with mean f_n and variance σ_n^2, for $n = 1, ..., N$. Let N denote the sample size, D denote the number of dimensions. $\mathbf{f} = (f_1, ..., f_N) \sim N(0, \Sigma)$. To simulate $\mathbf{f}$, we define the covariance matrix as

$$\Sigma_{i,j} = \sigma_f^2 \exp\left(-\frac{1}{2}\sum_{d=1}^{D}(x_{d,i} - x_{d,j})^2/l^2\right) \tag{1}$$

The covariance matrix Σ is computed by inputing $\mathbf{X} = [\mathbf{x}_1, ..., \mathbf{x}_N]^{'}$, which is a $D \times N$ matrix where the column vector $\mathbf{x_n}$, $n = 1, ..., N$, are the input D-dimensional vectors that will be drawn from a D-dimensional unit hypercube for all n. Fixing l (called "lengthscale" parameter), σ_f^2 and σ_n^2, we can generate f_n and therefore simulate observations y_n, since $\mathbf{y}|\mathbf{f} \sim \mathcal{N}\left(\mathbf{f}, \sigma_n^2\mathbf{I}\right)$and therefore their likelihood function as a function of $\mathbf{f}$ looks like:

$$L_r(\mathbf{f}) = \prod_{n=1}^{N} \mathcal{N}(y_n; f_n, \sigma_n^2) \tag{2}$$

3.1.2 Log Gaussian Cox process

Cox process is a "doubly stochastic" Poisson process with a stochastic intensity measure (2). Log Gaussian Cox process is introduced by Moller (2) as the Cox process where the logarithm of the internsity function is a Gaussian process. Mathematically, let y_n denote the observations. Then $y_n \sim Poisson(\lambda_n)$ with mean λ_n. The intensity function can be estimated given the log Gaussian Cox process observation within a bounded subset. This means we can partition the space finitely into N bins and y_n is the number of events in bin n for all $n = 1, ..., N$. We assume that every bin has a constant intensity function λ_n. Let m be the offset to the log mean λ_n, and define it as the sum of the mean log-intensity of the Poisson process and the log of the bin size (3).

$$y_n|f_n \sim Poisson(exp(f_n + m)) \tag{3}$$

$$\mathbf{f} \sim \mathbf{N}(\mathbf{0}, \mathbf{\Sigma}) \tag{4}$$

Σ is defined in Equation (7). Then the likelihood of $\mathbf{y}$ is:

$$L_p(\mathbf{f}) = \prod_{\mathbf{n=1}}^{\mathbf{N}} \frac{\lambda_{\mathbf{n}}^{\mathbf{y_n}}\mathbf{exp}(-\lambda_{\mathbf{n}})}{\mathbf{y_n!}}, \lambda_{\mathbf{n}} = \mathbf{e}^{\mathbf{f_n}+\mathbf{m}} \tag{5}$$

3.2 Implementation and Results

3.2.1 Gaussian Regression

Let $l = 1$, $\sigma_f^2 = 1$, $\sigma_n^2 = 0.3^2$. Firstly, we validate the Elliptical Slice Sampling algorithm on Gaussian regression model when $\mathbf{f}$ is bivariate Normal variable. In this case, let $N = 2$ and $D = 1$ such that $\{\mathbf{x_n}\}_{\mathbf{n=1}}^{\mathbf{2}}$ is one dimensional. Since both prior distribution and likelihood function are Gaussian, the posterior distribution of $\mathbf{f}$ should be Gaussian. We perform Henze-Zirkler's test to assess whether the outputs follow Bivariate Normal distribution. This is based on the measure of distance, which is nonnegative, between the characteristic function of the multivariate normality and the empirical characteristic function. The distribution of the test statistic is approximately log normal. We achieved a p-value much larger than 0.05, which indicates that there is no strong evidence to reject the null hypothesis that the outputs of the algorithm follows multivariate normal distribtuion. We can visualise the distribution of outputs in Figure 1. As a necessary condition for mutivariate normality, each variable should have normal distribution. We confirm this condition from QQ-plot.

Then we assess the performance of the algorithm on Gaussian regression when $N = 200$, i.e. $\mathbf{f}$ is 200-dimensional. As stated in the model description section, $\mathbf{f}$ can be generated by inputing X to covariance matrix. So we first simulated datasets X. In order to compare the performance of algorithm for different dimensions, i.e. D, of feature vectors, 2 synthetic datasets X_1,$X_1 0$ will be simulated with $D = 1, 10$ respectively. This will give different distribution of prior distribution of $\mathbf{f}$. Denote the observations simulated from these two cases as $R1$ and $R10$ respectively.

3.2.2 Log Gaussian Cox process

Data of mining disasters are provided by Jarrett et al. (1). There were 191 events happening during 40550 days which were partitioned into 811 bins each of which contains 50 days according to Murray, Adams and MacKay (3). This means f is 811 dimensional, we notice that both Elliptical Slice Samping and metropolis hastins method take a long time. This makes it harder to validate algorithms on this model. Hence, we partition the period into 102 intervals, such that each interval includes 400 days (except the last one, which is consist of 150 days). Given the date of every event, the number of event happened in each bin can be computed, i.e. y_n. Let $l = 13516$, $\sigma_f^2 = 1$, $N = 102$, $D = 1$ and $m = log(191/102)$. We observe frequent jumps in trace plot and the autocorrelation function goes to 0 within ?? lag(??).

3.2.3 Comparison

For Gaussian regression model, we will compare the performance of Elliptical Slice Sampling algorithm with the theorethical Bayesian model, the Metropolis-Hasting algorithm introduced by Neal (4) and an adapted Metropolis-Hastings algorithm proposed by Roberts et al. in (6).

In order to plot and compare the densities, we restricted the full 4-models comparison on the 2-dimensional case. Hence, $N = 1$, $D = 2$ and, as suggested in (3), we put $\sigma_n = 0.3$, $\sigma_f = 1$ and $l = 1$. The full Bayesian model will look like this:

$$\begin{aligned}
\mathbf{f} &\sim \mathcal{N}(\mathbf{0}, \Sigma) \\
\mathbf{y}|\mathbf{f} &\sim \mathcal{N}\left(\mathbf{f}, \Sigma^{'}\right) \\
\mathbf{f}^{'} &\sim \mathcal{N}\left(\left(\Sigma^{-1} + \Sigma^{'-1}\right)^{-1} \Sigma^{'-1}\mathbf{y}, \left(\Sigma^{-1} + \Sigma^{'-1}\right)^{-1}\right)
\end{aligned}$$

where Σ, as proposed before, is such that $\Sigma_{i,j} = \exp\left(-\frac{1}{2}\sum_{d=1}^{2}(x_{d,i} - x_{d,j})^2\right)$ $i, j = 1, 2$ and $\Sigma^{'} = \begin{bmatrix} 0.09 & 0 \\ 0 & 0.09 \end{bmatrix}$. We tested the normality of these samples with the 'mvnorm.skew.test' from the 'ICS' pckage and the null hypothesis was not rejected.

For Neal's Metropolis Hastings algorithm, we discover that the performance of the MCMC chain is sensitive to the value of step-size ϵ because the mixture and convergence of the chain depend on ϵ. We take $\epsilon = 0.2$ after trails of different values.

With the same target distribution, adaptive Metropolis-Hastings use proposals for n^{th} iteration as follows:

$$q_n(\mathbf{f}, \mathbf{f}') = \begin{cases} N(\mathbf{f}, (\mathbf{0.1^2})\mathbf{I_d}/\mathbf{d}) & n \leq 2d \\ (1-\beta)N(\mathbf{f}, (\mathbf{2.38^2})\mathbf{\Sigma_n}/\mathbf{d}) + \beta\mathbf{N}(\mathbf{f}, (\mathbf{0.1^2})\mathbf{I_d}/\mathbf{d}) & n > 2d \end{cases} \tag{6}$$

where Σ_n is the empirical estimate of the covariance of the target distribution based on the run so far and β is a small positive constant (7).

Compared to other algorithms, Adaptive Metropolis Hastings algorithm takes longer time to reach a stationary variance and good mixture. To avoid complexity of estimating Σ_n, in this example, we simply take the covariance of the posterior distribution, which is known. In this case, it is not nessary to take the weighted sum as proposal according to previous research (7). Then the proposal turns out to be $N(\mathbf{f}, (\mathbf{2.38^2}) \left(\mathbf{\Sigma^{-1}} + \mathbf{\Sigma'^{-1}}\right)^{\mathbf{-1}} \mathbf{d})$.

In the figure 3.2.3 we can compare the contour plots of the four samples coming from the theoretical model, the ESS algorithm, the Neal MH and the Adaptive MH. The number of samples for each model was 100000. It is evident that the stationary distributions of the three algorithms are the correct one.

Then we compare Elliptical Slice Sampling with Neal's MH in terms of effective sample size and CPU time. We run 10^6 iterations and remove the first 10^5 burn-in. We encountered problems when implementing Adaptive MH with proposal as shown in Equation (6), which is resulted from the empirical estimation of the covariance matrix. In addition, the update of state is infrequent. In the first 400 iterations, $\mathbf{f}$ still stay in the same state. Therefore, we only compare Elliptical Slice sampling with Neal's MH. The results are shown in Table 1, which show that Elliptical slice sampling is more efficient compared to Neal's MH because the former one has more effective samples in all 3 examples. The effective samples occupy approximately 47% for Elliptical slice sampling method while consists of only 21% of total samples for Neal's MH. The effective sample size is much smaller for data where $\mathbf{f}$ is 200 dimensional where input vector is 10 dimensional. On the other hand, the elliptical slice sampling takes longer time. But the effective samples per unit time for Elliptical Slice Sampling is larger in general. For example, for $R1$, the effective samples per unit time is 2.1 for Elliptical Slice Sampling, which is approximately twice as large as Neal's MH.

Table 1: Compare Elliptical Slice Sampling with Neal's Metropolis Hastings on data $R1$, $R10$ and $Mining$ in terms of effective sample size and CPU time.

	R1		R10		Mining	
	ESS	NMH	ESS	NMH	ESS	NMH
effective samples	33316.79	13330.98	2139.793	1159.535	429181.7	186571
CPU time	15898.001	9472.684	16374.79	10497.30	3448.755	3271.421

4 Discussion and Conclusion

Neal's MH requires extra time to tune step-size. Combined the results discovered in Section 3, Elliptical Slice Sampling outperforms Neal's MH in terms of time complexity, efficiency and mixture of Markov chains (??).

References

[1] Jarrett, R. G. (1979) *A note on the intervals between coal-mining disasters*, Biometrika.

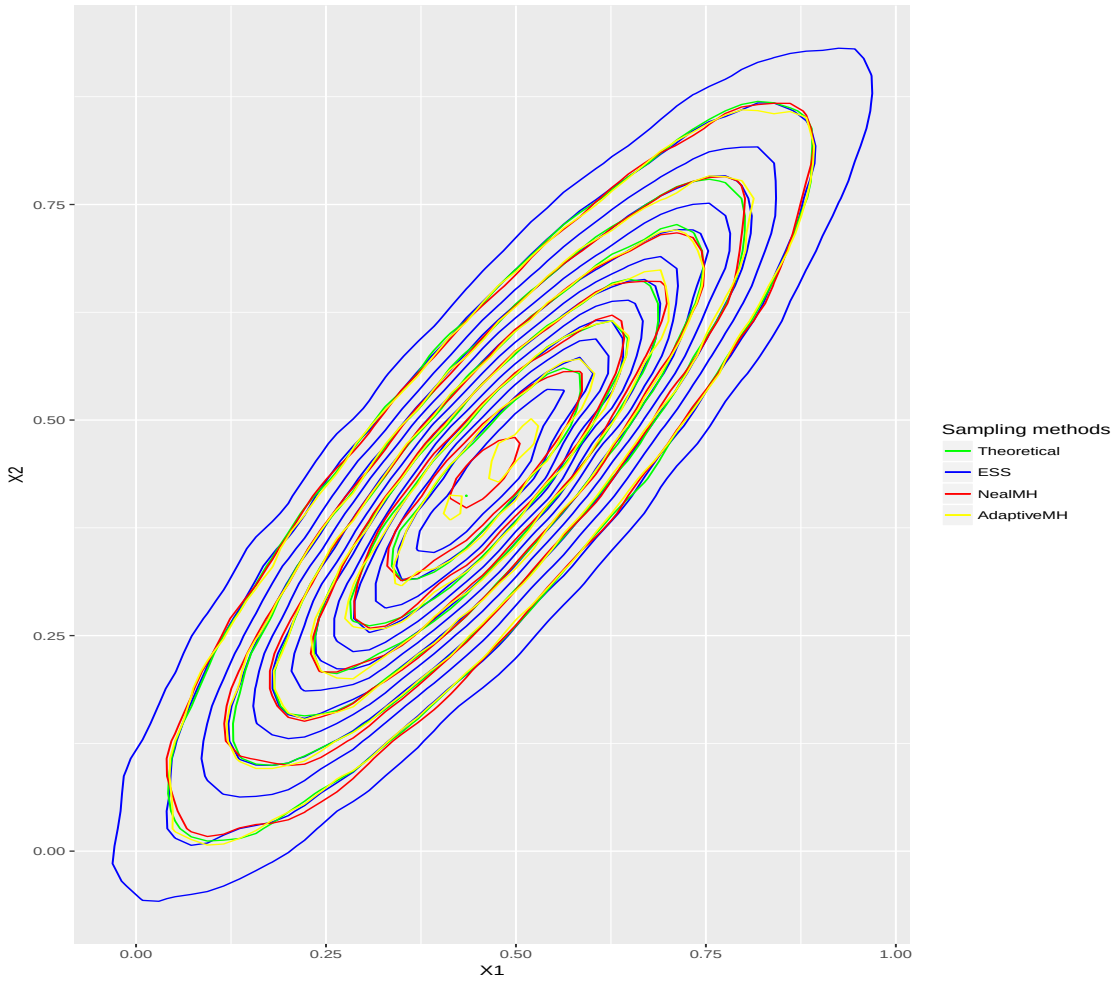

Figure 2: Comparison of contour plots for Gaussian regression

[2] Møller, J., Syversveen, A. R., Waagepetersen (1998) *Log Gaussian Cox processes*, Scandinavian Journal of Statistics.

[3] Murray, I., Adams, R. P., and MacKay, D. J. C. (2010) *Elliptical slice sampling*, Journal of Machine Learning Research.

[4] Neal, R. M. (1999) *Regression and Classification Using Gaussian Process Priors*, Bayesian Statistics 6 (475-501), OU Press.

[5] Neal, R. M. (2003) *Slice sampling*, Annals of Statistics.

[6] Roberts, G. O., Rosenthal, J. S. (2009) *Examples of Adaptive MCMC*, Journal of Computational and Graphical Statistics.

[7] Gareth O. Roberts Jeffrey S. Rosenthal (2009) *Examples of Adaptive MCMC*, Journal of COmputational and Graphical Statistics, 18:2, 349-367, DOI:10.1198/icgs.2009.06134.