Elliptical Slice Sampling

Francesca Panero Xuewen Yu

October 17, 2017

Abstract

We present the R package 'ESS', which provides an implementation of the Markov chain Monte Carlo method presented by Murray, Adams and MacKay in 2010 in the paper "Elliptical slice sampling" published in Journal of Machine Learning Research (3).

1 Slice sampling

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.

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:

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

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:

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

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

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

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: $\mathbf{f}^{'} \sim \mathbf{p}^*(\mathbf{f})$
Sample from $p(\boldsymbol{\nu}_0, \boldsymbol{\nu}_1, \theta | \boldsymbol{\nu}_0\text{sin}\theta + \boldsymbol{\nu}_1\text{cos}\theta = \mathbf{f})$:
        $\theta \sim \text{Uniform}[0, 2\pi]$
        $\boldsymbol{\nu} \sim \mathcal{N}(\mathbf{0}, \Sigma)$
        $\boldsymbol{\nu}_0 \leftarrow \mathbf{f}\text{sin}\theta + \boldsymbol{\nu}\text{cos}\theta$
        $\boldsymbol{\nu}_1 \leftarrow \mathbf{f}\text{cos}\theta - \boldsymbol{\nu}\text{sin}\theta$
Sample $\theta$ using slice sampling on $p^*(\theta | \boldsymbol{\nu}_0, \boldsymbol{\nu}_1) \propto L(\boldsymbol{\nu}_0\text{sin}\theta + \boldsymbol{\nu}_1\text{cos}\theta)$
$\mathbf{f}^{'} = \boldsymbol{\nu}_0\text{sin}\theta + \boldsymbol{\nu}_1\text{cos}\theta$
```

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:

2.1 Reversibility of Elliptical Slice Sampling

The Elliptical Slice Sampling algorithm updates $\mathbf{f}$ by proposing $\mathbf{f_1}$,...,$\mathbf{f_k}$,... based on θ_1,...,θ_k,..., until $\mathbf{f_K}$ satisfies step 5 in the algorithm. Then $\mathbf{f_K} = \mathbf{f}'$. This procedure is reversible. We can verify this by showing

$$p(\mathbf{f}, y, \boldsymbol{\nu}, \{\theta_k\}) = p(\mathbf{f}', y, \boldsymbol{\nu}', \{\theta_k^{'}\}) \tag{1}$$

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.
```

This joint distribution is the multiplication of target distribution and proposals of θ_k and $\boldsymbol{\nu}$:

$$p(\mathbf{f}, y, \boldsymbol{\nu}, \{\theta_k\}) = p^{*}(\mathbf{f})p(y|\mathbf{f})p(\boldsymbol{\nu})p(\{\theta_k\}|\mathbf{f}, \boldsymbol{\nu}, y) \propto \tag{2}$$

$$\propto \mathcal{N}(\mathbf{f}; \mathbf{0}, \Sigma)L(\mathbf{f})\mathcal{N}(\boldsymbol{\nu}; \mathbf{0}, \Sigma)\frac{1}{L(\mathbf{f})}p(\{\theta_k\}|\mathbf{f}, \boldsymbol{\nu}, y) \tag{3}$$

Another variable $\boldsymbol{\nu}_k$ is added which is updated in every iteration, such that:

$$\boldsymbol{\nu}_k = \boldsymbol{\nu}\cos(\theta_k) - \mathbf{f}\sin(\theta_k) \qquad k = 1, ..., K \tag{4}$$

Then we have:

$$\mathcal{N}(\boldsymbol{\nu}_k; \mathbf{0}, \Sigma)\mathcal{N}(\mathbf{f}_k; \mathbf{0}, \Sigma) = \mathcal{N}(\boldsymbol{\nu}; \mathbf{0}, \Sigma)\mathcal{N}(\mathbf{f}; \mathbf{0}, \Sigma) \qquad \forall k \tag{5}$$

This can be easily proved by computing Jacobian matrix, which has determinat equalling 1 in this case.

$\theta = 0$ corresponds to the location of $\mathbf{f}$. When we reverse the procedure, the location of $\mathbf{f}'$ corresponds to $\theta' = 0$. So the relative angle of the location of $\mathbf{f}$ is $-\theta_K$. By generating the same y, ν and angle:

$$\theta'_k = \theta_k - \theta_K \tag{6}$$

we can return to $\mathbf{f}$. So the following equation holds:

$$p(\{\theta_k\}|\mathbf{f}, \boldsymbol{\nu}, y) = p(\{\theta_k^{'}\}|\mathbf{f}', \boldsymbol{\nu'}, y) \tag{7}$$

This will lead to the reversibility of the algorithm by substituting and in

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

$$\Sigma_{i,j} = \sigma_f^2 \text{exp}\left(-\frac{1}{2}\sum_{d=1}^{D}(x_{d,i} - x_{d,j})^2/l^2\right) \tag{8}$$

The ovariance 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:

$$L_r(\mathbf{f}) = \prod_{n=1}^{N} \mathcal{N}(y_n; f_n, \sigma_n^2) \tag{9}$$

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).

$$y_n|f_n \sim Poisson(exp(f_n + m)) \tag{10}$$

$$\mathbf{f} \sim \mathbf{N}(\mathbf{0}, \mathbf{\Sigma}) \tag{11}$$

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

$$L_p(\mathbf{f}) = \prod_{\mathbf{n=1}}^{\mathbf{N}} \frac{\lambda_\mathbf{n}^{\mathbf{y_n}}\mathbf{exp}(-\lambda_\mathbf{n})}{\mathbf{y_n}!}, \lambda_\mathbf{n} = \mathbf{e}^{\mathbf{f_n+m}} \tag{12}$$

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. This can be verified by QQ-plot as shown in Figure 2.

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, 3 synthetic datasets X_1,X_2,X_3 will be simulated with $D = 1, 5, 10$ respectively. The traces of log-likelihoods and the first dimension of outputs for $D = 1$ are shown in Figure 3 and 4 respectively, which reveal frequent jumps in both chains. The effective sample size for it is 3177.

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. Given the date of every event, the number of event happened in each bin can be computed, i.e. y_n. Let $l = 13516$, $\sigma_f^2 = 1$, $N = 811$, $D = 1$ and $m = log(191/811)$. The trace of the first dimension of the output is shown below.

3.3 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:

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

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. In the figure 3.3 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. We need to mention the fact that the we tried different step-size parameters for Neal's algorithm to obtain a Gaussian sample.

We tested log Gaussian Cox process model.The effective sample size, likelihood and CPU time for running 100000 iterations after 10000 burn in will be evaluated. We will perform 100 runs to discover the mean and standard deviation of the evaluations.

References

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

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

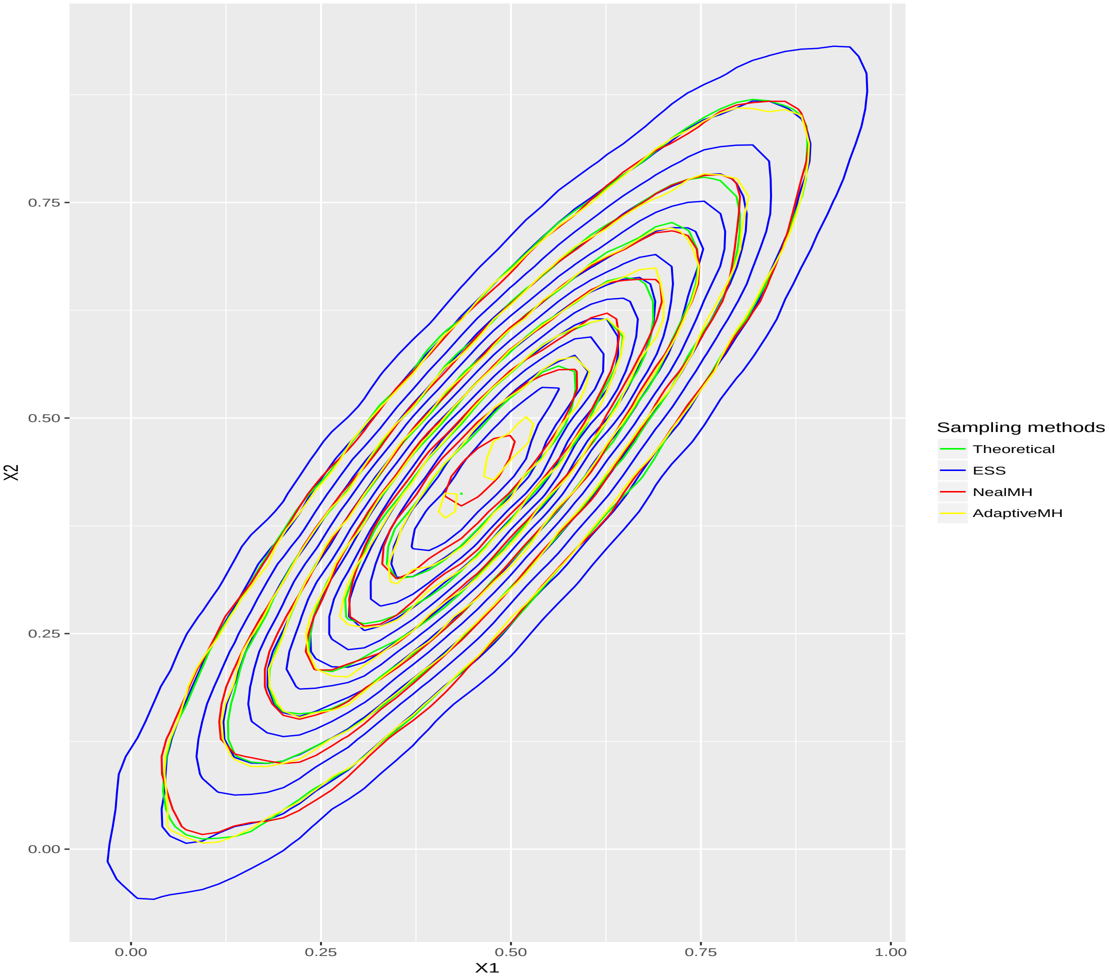

Figure 1: Comparison of contour plots for Gaussian regression

[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.