Monte Carlo Project 2

Introduction

In this independent project, I will be examining the elliptical slice sampling method and its generalisation called generalised elliptical slice sampling, and also parallelisation methods to estimate inputs of generalised elliptical slice sampling.

Posterior distributions of real world data tend to have dependencies between the latent variables. One way we can capture these dependencies in the posterior is to factor the posterior into a Gaussian prior and an arbitrary likelihood. Say for example, the posterior distribution is a long and thin density, and we want to explore the state space quickly and efficiently. We can then apply the elliptical slice sampling method to take advantage of the elliptical shaped density due to the dependence structure, since it is able to propose large moves in the direction of dependence due to the way slice sampling adaptively pick step sizes to match the local density. This is also a computationally efficient method since we are able to exploit the shape of the density that arised from the density without computing gradients and Hessians, as opposed to Hamiltonian Monte Carlo. Also in contrast, the Metropolis Hastings algorithm with only spherical proposals are relatively inefficient when there are dependencies between the variables, as it is not able to exploit the elliptical shape of the density with its spherical proposal. The elliptical slice sampling method is also easy to implement as it contains no tuning parameters, and thus requires no expert tuning for the chain to mix well.

Method

Elliptical slice sampling is a multivariate extension of slice sampling, where each proposal is a trigonometric rotation of the current state $\mathbf{f}$ and auxiliary variable $\boldsymbol{\nu}$. The angle of rotation θ is sampled uniformly from a bracket starting from $[0, 2\pi]$ and this bracket is shrunk until an acceptance occurs, which gives a probability of acceptance of the proposal state of 1. Thus, the algorithm is efficient in the sense that no samples are wasted.

Elliptical slice sampling as introduced in (Murray et al., 2010) can be used to sample from target distributions of the form

$$p^*(\mathbf{f}) = \frac{1}{Z} N(\mathbf{f}; 0, \Sigma) L(\mathbf{f})$$

,

where Z is the normalisation constant, $\mathbf{f}$ is the variable of interest, L is an arbitrary likelihood, and $N(\mathbf{f}; 0, \Sigma)$ is the prior.

This method can be generalised to priors with arbitrary mean μ

$$p^*(\mathbf{f}) = \frac{1}{Z} N(\mathbf{f}; \mu, \Sigma) L(\mathbf{f})$$

,

since Gaussian means can be shifted.

Elliptical slice sampling is a modification of the pCN method which is described as follows. Given initial state $\mathbf{f}$, the new proposed state is

$$\mathbf{f}' = \sqrt{1 - \epsilon^2}(\mathbf{f} - \mu) + \epsilon(\boldsymbol{\nu} - \mu) + \mu$$

,

where $\epsilon \in [-1, 1]$ is a step-size parameter, and $\boldsymbol{\nu} \sim N(\mu, \Sigma)$.

For a fixed auxiliary variable ν, if we vary $\epsilon \in [-1, 1]$, the locus of proposed states is half an ellipse. But by defining the proposed state in a new way:

$$\mathbf{f}' = (\mathbf{f} - \mu)cos\,\theta + (\boldsymbol{\nu} - \mu)sin\,\theta + \mu$$

,

where $\theta \in [0, 2\pi]$. The locus of proposed states become a full ellipse that gives a richer set of possible proposed states, since $\forall\theta \in [0, 2\pi]$, we can find a $\epsilon \in [-1, 1]$, such that the the original pCN expression agrees with our modified version. Since the proposed state for elliptical slice sampling is a modified version of pCN, it should inherit the dimension robustness of pCN, where the convergence of the algorithm does not deteriorate with the dimension of the parameter $\mathbf{f}$. This makes it suitable for high dimensional sampling problems such as Gaussian processes.

This expression for the proposed state uses the fact that the sum of two Gaussian randomm variable is a Gaussian random variable. i.e: Given two independent Gaussian draws $\mathbf{f} \sim N(\mu, \Sigma)$ and $\nu \sim N(\mu, \Sigma)$, the proposed state $\mathbf{f}'$ defined as

$$\mathbf{f}' = (\mathbf{f} - \mu)cos\,\theta + (\boldsymbol{\nu} - \mu)sin\,\theta + \mu$$

,

is sum of two Gaussian random variables. Hence, marginally, $\mathbf{f}' \sim N(\mu, \Sigma)$ for any $\theta \in [0, 2\pi]$, since

$$\begin{aligned} Cov(\mathbf{f}') &= \Sigma\, cos^2\theta\ + \Sigma\, sin^2\theta \\ &= \Sigma \end{aligned} \tag{1}$$

for any $\theta \in [0, 2\pi]$. Thus the proposed states will have the dependence structure of the prior.

We would like to use slice sampling to adaptively pick θ to get our proposed state. But naively applying slice sampling to obtain θ does not give a Markov kernel that satisfies reversibility. Since the locus of proposed states is defined using the current state $\mathbf{f}$, the markov kernel becomes not reversible. Therefore, we need to augment the proposed state expression with auxiliary variables to get rid of the $\mathbf{f}$ in the expression so that we can have a Markov kernel that satisfies reversibility.

The paper introduced two new auxiliary variables:

$$\nu_0 = (\mathbf{f} - \mu)\, sin\,\theta + (\boldsymbol{\nu} - \mu)\, cos\,\theta + \mu$$

and

$$\nu_1 = (\mathbf{f} - \mu)\, cos\,\theta - (\boldsymbol{\nu} - \mu)\, sin\,\theta\ + \mu$$

.

Similar to derivations earlier, we can see that $\nu_0 \sim N(\mu, \Sigma)$ and $\nu_1 \sim N(\mu, \Sigma)$. These new auxiliary variables will replace $\mathbf{f}$. With these new auxiliary variables we can now redefine,

$$\mathbf{f} = (\boldsymbol{\nu}_0 - \mu)\, sin\,\theta + (\boldsymbol{\nu}_1 - \mu)\, cos\,\theta + \mu$$

,

And again similarly to the above derivations, we can see that $\mathbf{f} \sim N(\mu, \Sigma)$. Thus, if we can jointly sample from the augmented distribution:

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

,

then samples from the target distribution $p^*(\mathbf{f})$ can be obtained from the expression

$$\mathbf{f} = (\boldsymbol{\nu}_0 - \mu)\, sin\, \theta + (\boldsymbol{\nu}_1 - \mu)\, cos\, \theta + \mu$$

Algorithm 1

Input: Current state $\mathbf{f}$, prior parameters $\mu\,, \Sigma$.

1) Sample from $p^*(\boldsymbol{\nu}_0, \boldsymbol{\nu}_1|\theta)$ by sampling $\boldsymbol{\nu} \sim N(\mu, \Sigma)$ and setting $\boldsymbol{\nu}_0 = (\mathbf{f} - \mu)\, sin\theta + (\boldsymbol{\nu} - \mu)\, cos\theta$ and $\boldsymbol{\nu}_1 = (\mathbf{f} - \mu)\, cos\theta - (\boldsymbol{\nu} - \mu)\, sin\theta$
2) Sample from $p^*(\theta|\boldsymbol{\nu}_0, \boldsymbol{\nu}_1) \propto L(\boldsymbol{\nu}_0\, sin\theta + \boldsymbol{\nu}_1 cos\theta)$ via 1D slice sampling:

(a) Initialise $\theta_{min} = 0$ and $\theta_{max} = 2\pi$ and draw $\theta \in [\theta_{min}, \theta_{max}]$
(b) Draw $\ell \in [0, L(\boldsymbol{\nu}_0\, sin\theta + \boldsymbol{\nu}_1 cos\theta)$ uniformly
(c) Sample $\theta_{new} \in [\theta_{min}, \theta_{max}]$ uniformly
(d) If $L(\boldsymbol{\nu}_0\, sin\theta_{new} + \boldsymbol{\nu}_1 cos\theta_{new}) > \ell$, set $\theta \leftarrow \theta_{new}$. Otherwise, if $\theta_{new} < \theta$, set $\theta_{min} \leftarrow \theta_{new}$ or if $\theta_{new} > \theta_{current}$, set $\theta_{max} \leftarrow \theta_{new}$. And go back to (c)

3) Once we have $\boldsymbol{\nu}_0, \boldsymbol{\nu}_0, \theta$, set $\mathbf{f}' = (\boldsymbol{\nu}_0 - \mu) sin\theta + (\boldsymbol{\nu}_1 - \mu) cos\theta + \mu$. Repeat from step (1).

This algorithm makes the link to slice sampling explicit, but the paper also presented a more neat and equivalent algorithm that gets rid of the $\boldsymbol{\nu}_0$, $\boldsymbol{\nu}_1$, which is as follows:

Algorithm 2

Input: Current state $\mathbf{f}$, prior parameters $\mu\,, \Sigma$

1) Sample $\boldsymbol{\nu} \sim N(\mu, \Sigma)$
2) Sample $u \sim Unif[0, 1]$. Set $log\ y \leftarrow logL(\mathbf{f}) + log\ u$
3) Draw proposal angle $\theta \sim Unif[0, 2\pi]$. Set $[\theta_{min}, \theta_{max}] \leftarrow [\theta - 2\pi, \theta]$
4) Set proposal state $\mathbf{f}' \leftarrow \mathbf{f} cos\theta + \boldsymbol{\nu} sin\theta$
5) If $logL(\mathbf{f}') > log\ y$. Accept $\mathbf{f}$.
6) Else:

(a) if $\theta < 0$, then $\theta_{min} \leftarrow \theta$. else: $\theta_{max} \leftarrow \theta$
(b) Draw proposal angle $\theta \sim Unif[\theta_{min}, \theta_{max}]$

7) Return to step (4)

Minor discussion of Algorithm

1) Auxiliary varriable $\boldsymbol{\nu}$, along with the current state $\mathbf{f}$ defines an ellipse and this ellipse reflects the dependence structure of the priors.

2) Intuitively, the reason we only need to compare likelihood ratios in the slice sampling step is because updates $\{\mathbf{f}_k\}$ and $\{\boldsymbol{\nu}_k\}$ have the same prior probability as the initial states.

3) The proposed state $\mathbf{f}'$ is never the same as the current state $\mathbf{f}$ unless it is the only state with nonzero likelihood. This effectively gives us an algorithm with 100% acceptance rate.

Validity of Algorithm

In order for the proposed state $\mathbf{f}'$ to be valid and from the target distribution, we will require the algorithm to satisfy reversibility and ergodicity.

Proof of reversibility:

Consider the joint distribution of the target distribution and the random variables generated by the algorithm:

$$\begin{aligned} p(\mathbf{f}, y, \boldsymbol{\nu}, \{\theta_k\}) &= p^*(\mathbf{f})p(y|\mathbf{f})p(\boldsymbol{\nu})p(\{\theta_k\}|\mathbf{f}, \boldsymbol{\nu}, y) \\ &= \frac{1}{L(\mathbf{f})}N(\mathbf{f}; \mu, \Sigma)N(\boldsymbol{\nu}; \mu, \Sigma)p(\{\theta_k\}|\mathbf{f}, \boldsymbol{\nu}, y), \end{aligned} \tag{2}$$

where $p(y|\mathbf{f}) = \frac{1}{L(\mathbf{f})}$, since $y \sim Unif[0, L(\mathbf{f})]$ and

$$p(\{\theta_k\}|\mathbf{f}, \boldsymbol{\nu}, y)$$

is the distribution of the set of proposed angles in the algorithm.

Given random variables $\boldsymbol{\nu}, \theta_k$ and current state $\mathbf{f}$, we can define transformations:

$$\boldsymbol{\nu}_k = \boldsymbol{\nu} cos\theta_k - \mathbf{f} sin\theta_k$$

,

$$\mathbf{f}_k = \boldsymbol{\nu} sin\theta_k + \mathbf{f} cos\theta_k, \quad k = 1...K.$$

,

We can easily show that these transformations have unit Jacobian by simple diffentiation and taking determinants of a 2x2 matrix. Having unit Jacobian for any θ_k means the prior distributions are rotation invariant, giving:

$$N(\boldsymbol{\nu}_k; \mu, \Sigma)N(\mathbf{f}_k; \mu, \Sigma) = N(\boldsymbol{\nu}; \mu, \Sigma)N(\mathbf{f}; \mu, \Sigma)$$

for all k. Now, in order to show reversibility, we only need to show that

$$p(\{\theta_k\}|\mathbf{f}, \boldsymbol{\nu}, y) = p(\{\theta'_k\}|\mathbf{f}', \boldsymbol{\nu}', y)$$

The first proposed angle is always $\frac{1}{2\pi}$ since $\theta_1 \sim Unif[0, 2\pi]$ and $\theta'_1 \sim Unif[0, 2\pi]$. And since the reverse transition that starts from initial state $\mathbf{f}'_1 = \mathbf{f_K}$, uses the same state proposals as the forward transition,

they make the same rejection decisions and thus the same set of shrinking decisions. Thus the intermediate probabilities of $\frac{1}{\theta_{max}-\theta_{min}}$ are the same for both forward and backward transitions.

Thus we have shown that the probability for the set of proposed states for backwards and forwards are the same. And since the probability of drawing the same y, and same $\boldsymbol{\nu}' = \boldsymbol{\nu}_K$, is the same as the forward transition, we have proven the reversibility of the Markov kernel

This tells us that the proposed state defined as $\mathbf{f}' = (\mathbf{f} - \mu)cos\theta + \boldsymbol{\nu} sin\theta$ has the correct target distribution if $\mathbf{f}$ was drawn from the stationary distribution and the algorithm was run. i.e: the target distribution is a stationary distribution with respect to the Markov kernel defined by the algorithm.

Proof of ergodicity:

According to (Murray et al., 2010), since the distribution over the first proposed move is $N(\mathbf{f}cos\theta, \Sigma\, sin^2\theta)$, there is a nonzero probability of transitioning to any state under the posterior that has nonzero probability. And according to them, this is enough to guarantee that the chain is irreducible and aperiodic, which proves ergodicity.

Advantages of elliptical slice sampling

1) There are no tuning parameters since the algorithm uses slice sampling to adaptively pick step sizes to match the local shape of the density function. In contrast to competing algorithms that we have covered in class like Metropolis-Hastings, we have to come up with appropriate proposal distributions. Or for Hamiltonian Monte Carlo, we have to chose an appropriate step size and number of steps in order for the chain to mix well. For non-experts of Monte Carlo, this can be difficult.

2) It is efficient in the sense that it has a 100% acceptance probability and no samples are wasted.

3) It is also efficient in the sense that its proposals are in multivariate form, so it is more efficient and does not suffer from slow convergence problems of traditional componentwise univariate slice sampling, where we have to update each variable one by one. And this can be very slow when the variables we are interested in are of high dimension.

4) Since it is a generalisation of the pCN method, elliptical slice sampling should inherit the dimension robustness of the pCN method, making it well-suited for high-dimensional sampling problems.

5) Computation is relatively cheap compared to other methods. We don't have to compute gradients unlike Hamiltonian Monte Carlo and as mentioned in the original paper, drawing $\boldsymbol{\nu}$ will be the dominant cost in high dimensional problems as typical Gaussian samplers which use Cholesky decomposition for Σ will cost $O(N^2)$ for each iteration and since we have N samples, the cost becomes $O(N^3)$. In contrast, the cost of $L(\mathbf{f})$, which is $O(N)$, since individual likelihoods are independent when conditioned on $\mathbf{f}$.

Disadvantages and limitations of elliptical slice sampling

1) Users must specify prior covariance Σ and the algorithm can be sensitive to this specification.

2) As any distribution can be factored as $p^*(\mathbf{f}) = \frac{1}{Z} N(\mathbf{f}; 0, \Sigma) L(\mathbf{f})$, there exists likelihoods $L(\mathbf{f})$, where elliptical slice sampling is not effective. One such case is the chain can get stuck if there are regions of very strong likelihood. Once the chain is in such a region, the likelihood threshold becomes higher, and accepted proposals can only make small moves across the state space.

Ways to improve elliptical slice sampler

1) Even though we are already shrinking the bracket for angles at a fast rate, the original paper mentions a method by (Skilling and Mckay, 2003) that allows us to shrink the bracket even more aggressively and thus increasing the speed of us drawing the appropriate angle.

2) Biasing proposals away from the current state could potentially improve the speed of convergence. As the proposal states become more and more independent from the current state, there will be less redundant information. In our context, the proposal state becomes independent of the current state at $\theta = \frac{\pi}{2}$, since

$$\begin{aligned} \mathbf{f}' &= \mathbf{f} cos\frac{\pi}{2} + \boldsymbol{\nu} sin\frac{\pi}{2} \\ &= \boldsymbol{\nu} \end{aligned} \tag{3}$$

But as mentioned in the original paper, this is only useful when likelihoods are weak and do not provide much information.

3) The original paper mentions that for computational reasons, one might consider updating the variables of interest in blocks instead. If we can write the prior as:

$$\begin{bmatrix} \mathbf{f}_A \\ \mathbf{f}_B \end{bmatrix} \sim N\left(0, \begin{bmatrix} \Sigma_{A,A} & \Sigma_{A,B} \\ \Sigma_{A,B} & \Sigma_{B,B} \end{bmatrix}\right) \tag{4}$$

,

then we can update each block separately using elliptical slice sampling and well known results from multivariate normal conditional distributions.

4) We can also generalise elliptical slice sampling to variables of interests that have a continuous distribution, which I will explore in the next section.

Generalised elliptical slice sampling (Based on Nishihara et al. 2014)

The main idea is to reframe the target distribution $\boldsymbol{\pi}$ such that it can be sampled with elliptical slice sampling. One approach is to approximate the target distribution $\boldsymbol{\pi}$ with some normal approximation $N(\boldsymbol{\mu}, \Sigma)$. Thus, we can write the target distribution as:

$$\boldsymbol{\pi}(\mathbf{f}) = R(\mathbf{f}) N(\mathbf{f}; \boldsymbol{\mu}, \Sigma)$$

,

where

$$R(\mathbf{f}) = \frac{\boldsymbol{\pi}(\mathbf{f})}{N(\mathbf{f}; \boldsymbol{\mu}, \Sigma)}$$

is the residual likelihood of the normal approximation to the likelihood. Even though R is not a likelihood and $N(\mathbf{f}; \boldsymbol{\mu}, \Sigma)$ is not a prior, the target distribution is still in the form where we can apply elliptical slice sampling to. But as mentioned in (Nishihara et al. 2014), the chain might mix slowly in practice. This is

due to our normal approximation not being able to account for heavy tails in the target distribution. As a result, $R(\mathbf{f})$ might explode as $\mathbf{f}$ moves further away from the mean and into the tails, and this causes the chain get stuck and mix slowly.

In order to deal with this, Nishihara et al. expands the family of approximations to:

$$\boldsymbol{\pi}(\mathbf{f}) \propto R(\mathbf{f}) \int N(\mathbf{f}; \boldsymbol{\mu}(s), \Sigma(s))\phi(ds)$$

,

where $\int N(\mathbf{f}; \boldsymbol{\mu}(s), \Sigma(s))\phi(ds)$ represents a mixture of Gaussians and ϕ is a measure we can choose such that the integral $\int N(\mathbf{f}; \boldsymbol{\mu}(s), \Sigma(s))\phi(ds)$ becomes a distribution more close to the target in shape.

Nishihara et al. chose ϕ such that the integral is a multivariate t distribution, but he notes that the approximation equation is actually very general and there are many other choices of ϕ.

Since

$$\boldsymbol{\pi}(\mathbf{f}) = \int \boldsymbol{\pi}(\mathbf{f}, s)ds$$

,

we can view the above approximation of $\boldsymbol{\pi}(\mathbf{f})$ as a marginal density of the joint distribution over $\mathbf{f}$ and dummy variable s. Giving the equation:

$$\boldsymbol{\pi}(\mathbf{f}, s) = R(\mathbf{f})N(\mathbf{f}; \boldsymbol{\mu}(s), \Sigma(s))\lambda(s)$$

,

where λ is the density of ϕ with respect to the base measure over s. Therefore, similar to Gibbs sampling, we can sample this joint distribution by sampling the conditionals of $\mathbf{f}$ and s. And we know from class that this is valid because joint distributions are able to be fully characterised by their conditionals.

$$p(\mathbf{f}|s) \propto R(\mathbf{f})N(\mathbf{f}; \boldsymbol{\mu}(s), \Sigma(s))$$

and

$$p(s|\mathbf{f}) \propto N(\mathbf{f}; \boldsymbol{\mu}(s), \Sigma(s))\lambda(s)$$

Notice that the conditional of $\mathbf{f}$ can be sampled using elliptical slice sampling as previously described earlier in this section. Thus we can obtain samples from the marginal $\boldsymbol{\pi}$ by simply dropping the s dummy variable from our samples of the joint distribution.

As mentioned before, we can choose ϕ such that the approximation yields a multivariate t-distribution with ν degrees of freedom:

$$T_\nu(\mathbf{f}; \boldsymbol{\mu}, \Sigma) = \int IG(s; \frac{\nu}{2}, \frac{\nu}{2})N(\mathbf{f}; \boldsymbol{\mu}, s\Sigma)\, ds, \quad s > 0$$

,

and where λ has the density of an inverse gamma distribution:

$$IG(s; \alpha, \beta) = \frac{\beta^\alpha}{\Gamma(\alpha)} s^{-\alpha-1} e^{-\frac{\beta}{s}}$$

We then notice that the conditional $p(s|\mathbf{f})$ has an inverse gamma posterior distribution due to conjugacy. ie:

$$p(s|\mathbf{f}) = IG(s; \alpha', \beta')$$

,

where

$$\alpha' = \frac{D+\nu}{2}$$

,

where D is the dimension and,

$$\beta' = \frac{1}{2}(\nu + (\mathbf{f}-\boldsymbol{\mu})^T\Sigma^{-1}(\mathbf{f}-\boldsymbol{\mu}))$$

,

We can then draw from these distributions as previously mentioned and obtain samples from the target $\boldsymbol{\pi}$. By writing $\boldsymbol{\mu}(s) = \boldsymbol{\mu}$ and $\Sigma(s) = s\Sigma$, we can write the algorithm for generalised elliptical slice sampling as follows.

Algorithm 3 (Generalised Elliptical Slice Sampling)

1) Input: current state $\mathbf{f}$, parameters ν,$\boldsymbol{\mu}$,Σ, D.
2) Set $\alpha' \leftarrow \frac{D+\nu}{2}$.
3) Set $\beta' \leftarrow \frac{1}{2}(\nu + (\mathbf{f}-\boldsymbol{\mu})^T\Sigma^{-1}(\mathbf{f}-\boldsymbol{\mu}))$.
4) Draw $s \sim IG(\alpha', \beta')$.
5) Set $log\, L \leftarrow log\, \pi - log\, T_\nu(\mathbf{f}; \boldsymbol{\mu}, \Sigma)$.
6) Draw $\mathbf{f}'$ using elliptical slice sampling with input parameters $\mathbf{f}, \boldsymbol{\mu}, s\Sigma, log\, L$.

Minor discussion about algorithm

1) We can see that that this algorithm uses the same ideas from the elliptical slice sampling method in the discrete case by comparing the algorithms of both cases.
2) The multivariate t distribution was chosen because its flexibility allows it to capture most of the mass of a distribution. But Nishihara et al. notes that finer and more accurate approximations can be made for situations where the multivariate t distribution struggles to approximate the target distribution. Namely, distributions with multiple modes.
3) The (Nishihara et al. 2014) paper goes on to improve this method further by parallelising the generalised elliptical slice sampling method, which is actually the main focus of the paper.

Estimating prior inputs to Generalised Elliptical Slice Sampling (GESS) using parallelisation (based on Nishihara et al. 2014)

Instead of arbitrarily choosing input prior parameters for the generalised elliptical slice sampling, Nishihara et al. proposes a method that exploits data available from samples of the posterior to get the maximum likelihood estimate of the input prior parameters. This method uses two groups of Markov Chains, each obtained through parallelisation, and uses each group's chain to update the input parameters of the other group's Markov Chain. Let $F = \{\boldsymbol{f}_1, ..., \boldsymbol{f}_{K_1}\}$ and $G = \{\boldsymbol{g}_1, ..., \boldsymbol{g}_{K_2}\}$ denote the states of Markov Chains in these two groups, where $K_1 = K_2$ is usually set to the number of cores the machine has in practice. Since the individual chains are independent, and each chain is ergodic and the stationary distribution π is invariant. The collection of chains' joint distribution is just the product of their stationary distributions:

$$\begin{aligned}\Pi(F,G) &= \Pi_1(F)\Pi_2(G) \\ &= \prod_{k=1}^{K_1} \pi(\boldsymbol{f}_k) \prod_{k=1}^{K_2} \pi(\boldsymbol{g}_k)\end{aligned} \tag{5}$$

We then use the data from each chain to get the maximum likelihood of the input prior parameters fot the generalised elliptical slice sampling algorithm of the other chain. We define transition operators

$$Q_1 = S(\boldsymbol{f} \to \boldsymbol{f}'; \nu_G, \boldsymbol{\mu}_G, \boldsymbol{\Sigma}_G)$$

,

and

$$Q_2 = S(\boldsymbol{g} \to \boldsymbol{g}'; \nu_F, \boldsymbol{\mu}_F, \boldsymbol{\Sigma}_F)$$

,

and apply the transition operators to each individual chain in their respective groups in parallel. To maximise the likelihood of input prior parameters, Nishihara et al. applies the Expectation Maximisation algorithm to the samples from each chain. That is:

$$\nu_G, \boldsymbol{\mu}_G, \boldsymbol{\Sigma}_G = argmax_{\nu,\boldsymbol{\mu},\boldsymbol{\Sigma}} \prod_{k=1}^{K_2} T_v(\boldsymbol{g}_k; \boldsymbol{\mu}, \boldsymbol{\Sigma})$$

.

Algorithm 4 (Approximating input prior parameters of generalised elliptical slice sampling)

1) Input: States $F = \{\boldsymbol{f}_1, ..., \boldsymbol{f}_{K_1}\}$ and $G = \{\boldsymbol{g}_1, ..., \boldsymbol{g}_{K_2}\}$
2) $\nu, \boldsymbol{\mu}, \boldsymbol{\Sigma} \leftarrow EM(G)$
3) for all $\boldsymbol{f}_k \in F, \boldsymbol{f}'_k \leftarrow GESS(\boldsymbol{f}_k, \nu, \boldsymbol{\mu}, \boldsymbol{\Sigma})$
4) $F' \leftarrow \{\boldsymbol{x}'_1, ..., \boldsymbol{x}'_{K_1}\}$
5) $\nu, \boldsymbol{\mu}, \boldsymbol{\Sigma} \leftarrow EM(F')$
6) for all $\boldsymbol{g}_k \in G, \boldsymbol{g}'_k \leftarrow GESS(\boldsymbol{g}_k, \nu, \boldsymbol{\mu}, \boldsymbol{\Sigma})$
7) $G' \leftarrow \{\boldsymbol{g}'_1, ..., \boldsymbol{g}'_{K_2}\}$

Validity of algorithm

We show that the Markov transition operator defined by the algorithm satisfies general balance and that the chain is ergodic.

Proof of general balance

As the cases for Q_1 and Q_2 are identical, we only show one of them here.

$$\begin{aligned}\int \Pi_1(F)Q_1(F \to F')dF &= \prod_{k=1}^{K_1}(\int \pi(\boldsymbol{f}_k)S(\boldsymbol{f}_k \to \boldsymbol{f}'_k; \nu_Y, \boldsymbol{\mu}_Y, \boldsymbol{\Sigma}_Y)df_k) \\ &= \Pi_1(F')\end{aligned} \tag{6}$$

The last equality is true because generalised elliptical slice sampling satisfies general balance, and thus π is invariant.

Proof of ergodicity

Since the transition operator Q_1 updates chains independently, and that generalised elliptical slice sampling is ergodic. It follows that the collection of independent chains F is also ergodic.

Implementation of Vanilla Elliptical Slice Sampling

```
library(mvtnorm)
```

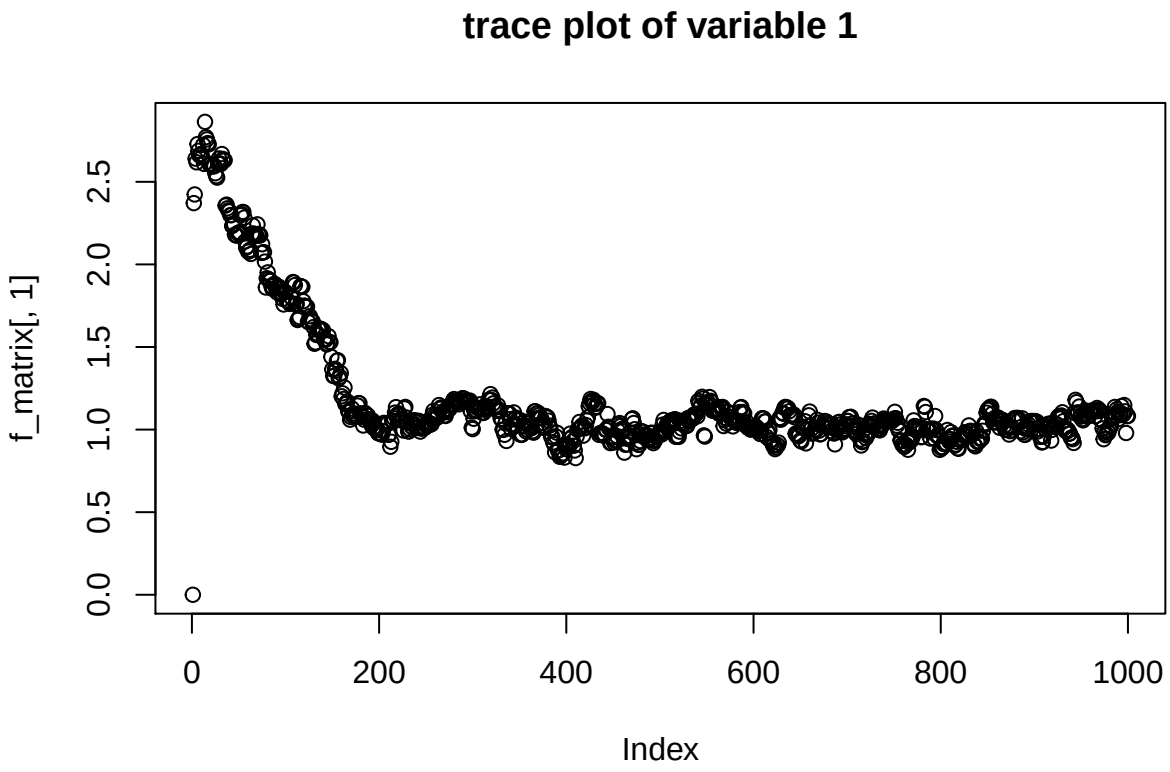
```
## Warning: package 'mvtnorm' was built under R version 3.6.2
```

```
#simulating the multivariate normal data
X = rmvnorm(200, mean=c(1,2,3,4,5)) #mean 1,2,3,4,5
```

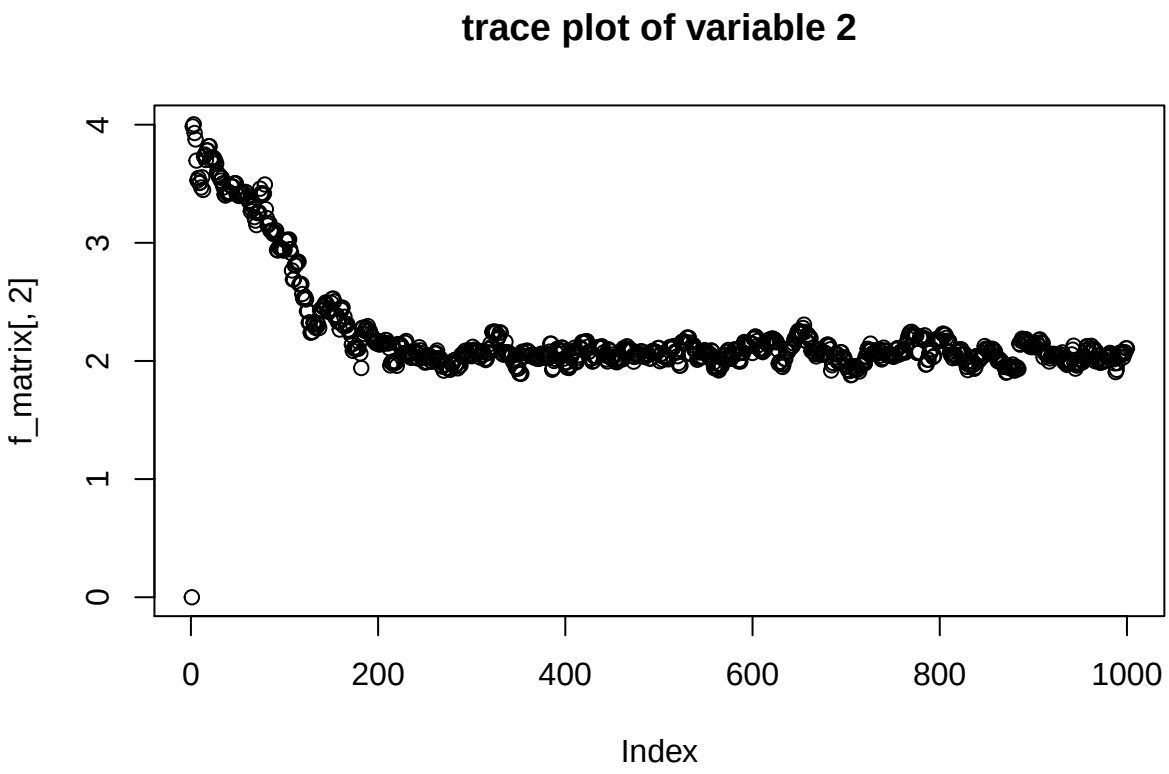
```
#Elliptical slice sampler
niter = 1000
f_matrix = matrix(0,nrow=niter, ncol=5)
mu = rep(10,5) #prior mean

for (i in 2:niter){

  f_curr = f_matrix[i-1,] #initialisation
  nu = rmvnorm(1, mean=mu, sigma = cov(X)) #drawing the ellipse
  u = runif(1)
  log_y =  sum(dmvnorm(X, mean= f_curr, sigma = cov(X), log = TRUE)) + log(u) #likelihood threshold

  theta = runif(1,min=0,max=2*pi) #initialising theta
  theta_min = theta-2*pi
  theta_max = theta

  f_next = (f_curr - mu) * cos(theta) + (nu - mu) * sin(theta) + mu

  #comparing likelihoods
  while (sum(dmvnorm(X, mean= f_next, sigma=cov(X), log = TRUE)) < log_y){

    f_next = (f_curr - mu) * cos(theta) + (nu - mu) * sin(theta) + mu
```

```
    if (theta < 0){theta_min = theta} #shrinking the bracket
    else{theta_max = theta}

    theta = runif(1,min=theta_min, max=theta_max)
  }

  f_matrix[i,] = f_next
}
```

```
#trace plots
plot(f_matrix[,1], main="trace plot of variable 1")
```




```
plot(f_matrix[,2], main="trace plot of variable 2")
```



It seems that the chain converged very quickly to the target distribution at iteration number 200, even when it started very far away.

```
mean(f_matrix[,1])
```

```
## [1] 1.189311
```

```
mean(f_matrix[,2])
```

```
## [1] 2.230061
```

```
mean(f_matrix[,3])
```

```
## [1] 3.126817
```

```
mean(f_matrix[,4])
```

```
## [1] 3.965668
```

```
mean(f_matrix[,5])
```

```
## [1] 4.894884
```

The chain managed to converge to the true values of mu.

Discussion and Bibliography

1) I. Murray, R. Prescott Adams, and D. J. C. MacKay. Elliptical slice sampling. 2010.
2) P. Richard Hahn, Jingyu He & Hedibert F. Lopes (2019) Efficient Sampling for Gaussian Linear Regression With Arbitrary Priors, Journal of Computational and Graphical Statistics, 28:1, 142-154, DOI: 10.1080/10618600.2018.1482762
3) Nishihara, Robert, Iain Murray, and Ryan P. Adams. "Parallel MCMC with generalized elliptical slice sampling." The Journal of Machine Learning Research 15.1 (2014): 2087-2112.
4) Fagan, Francois, Jalaj Bhandari, and John Cunningham. "Elliptical Slice Sampling with Expectation Propagation." UAI. 2016.