TP 5 : Sampling from multimodal distributions

We consider a target distribution π with support $X \subset \mathbb{R}^d$ $(d \in \mathbb{N}^*)$. When the target distribution is multimodal, especially with well-separated modes, classical MCMC algorithms (such as the Metropolis-Hastings algorithm or the Gibbs sampler) can perform very poorly and exhibit poor mixing. Indeed, a Metropolis-Hastings algorithm with local proposal can get stuck for a long time in a local mode of the target distribution.

To overcome these difficulties, we will introduce two MCMC algorithms which are particularly efficient with multimodal target distributions. The first algorithm is the *Parallel tempering* algorithm and the second one is the *Equi-Energy sampler*.

Exercise 1: A toy example

In the following, we consider a target distribution π – taken from [LW01] and plotted at figure 1 – defined on $\mathbb{R}^2$ as a mixture of 20 Gaussian distributions. The target distribution writes :

$$\pi(\mathbf{x}) = \sum_{i=1}^{20} \frac{w_i}{2\pi\sigma_i^2} \exp\left(- \frac{1}{2\sigma_i^2} {}^t(\mathbf{x} - \boldsymbol{\mu}_i)(\mathbf{x} - \boldsymbol{\mu}_i) \right)$$

where, $\forall i \in \{1, \dots, 20\}$, $w_i = 0.05$ and $\sigma_i = 0.1$. The 20 means $\boldsymbol{\mu}_i$ are defined as follows :

$$\begin{aligned}(\boldsymbol{\mu}_1, \dots, \boldsymbol{\mu}_{20}) \quad = \quad \Bigg(& \begin{pmatrix}2.18\\5.76\end{pmatrix}, \begin{pmatrix}8.67\\9.59\end{pmatrix}, \begin{pmatrix}4.24\\8.48\end{pmatrix}, \begin{pmatrix}8.41\\1.68\end{pmatrix}, \begin{pmatrix}3.93\\8.82\end{pmatrix}, \begin{pmatrix}3.25\\3.47\end{pmatrix}, \begin{pmatrix}1.70\\0.50\end{pmatrix}, \\ & \begin{pmatrix}4.59\\5.60\end{pmatrix}, \begin{pmatrix}6.91\\5.81\end{pmatrix}, \begin{pmatrix}6.87\\5.40\end{pmatrix}, \begin{pmatrix}5.41\\2.65\end{pmatrix}, \begin{pmatrix}2.70\\7.88\end{pmatrix}, \begin{pmatrix}4.98\\3.70\end{pmatrix}, \begin{pmatrix}1.14\\2.39\end{pmatrix}, \\ & \begin{pmatrix}8.33\\9.50\end{pmatrix}, \begin{pmatrix}4.93\\1.50\end{pmatrix}, \begin{pmatrix}1.83\\0.09\end{pmatrix}, \begin{pmatrix}2.26\\0.31\end{pmatrix}, \begin{pmatrix}5.54\\6.86\end{pmatrix}, \begin{pmatrix}1.69\\8.11\end{pmatrix} \Bigg).\end{aligned}$$

1. Write a Metropolis-Hastings Symmetric Random Walk algorithm (you may use your code from previous tutorial classes) to sample from π.
2. Show that the Metropolis-Hastings algorithm (even the adaptive Metropolis-Hastings algorithm) fails to sample from π.

Exercise 2: Parallel Tempering

The general idea of the Parallel Tempering (PT) [Gey91, ED05] algorithm is to use *tempered* versions of the distribution π and run parallel Metropolis-Hastings algorithm to sample from these tempered distributions. The tempered distributions are obtained by "warming up" the target distribution π at different *temperatures*. At each iteration of the algorithm, a *swap* between two chains (chains running at different temperature levels) is proposed. The Parallel Tempering uses the fast mixing of the chains at high temperature to improve the mixing of the chains at low temperatures.

Let K denote a positive integer. We consider a sequence of temperatures $(T_i)_{1\leq i\leq K}$ such that :

$$T_1 > T_2 > \ldots > T_K = 1 .$$

In the Parallel Tempering algorithm, K chains run in parallel : for $i \in [\![1, K]\!]$, the i-th chain targets the tempered distribution $\pi_i := \pi^{1/T_i}$; the distribution of interest corresponds to the lowest temperature, $T_K = 1$. Let $(X_n^{(i)})_{n\in\mathbb{N}}$ denote the i-th chain, sampling from the tempered distribution π_i.
At the n-th iteration of the Parallel Tempering algorithm, a candidate $Y_{n+1}^{(i)}$ for the i-th chain is proposed using the transition kernel $P^{(i)}(X_n^{(i)}, \cdot)$ of a Metropolis-Hastings algorithm. The next step consists in proposing a swap between two different chains (running at different temperatures) : given $(i, j) \in [\![1, K]\!]^2$, with $i \neq j$, a swap is proposed with probability $\alpha(i, j)$. More precisely, the PT algorithm writes :

Algorithm 1: Parallel Tempering

```
For all $i \in [\![1, K]\!]$, initialize $X_0^{(i)}$ ;
for $n = 1$ to $\mathrm{N}_{\mathrm{iter}}$ do
    For all $i \in [\![1, K]\!]$, draw $Y_{n+1}^{(i)}$ using the transition kernel $P^{(i)}(X_n^{(i)}, \cdot)$ ;
    Choose uniformly $(i, j) \in [\![1, K]\!]^2$, with $i \neq j$ ;
    Compute the swap acceptance probability $\alpha(i, j)$, given by :
        $$\alpha(i,j) = \min\left(1\,,\, \frac{\pi_i(Y_{n+1}^{(j)})\,\pi_j(Y_{n+1}^{(i)})}{\pi_i(Y_{n+1}^{(i)})\,\pi_j(Y_{n+1}^{(j)})}\right) ;$$
    Draw $U \sim \mathcal{U}([0, 1])$ ;
    if $U \leqslant \alpha(i, j)$ then
        $$\begin{cases} X_{n+1}^{(i)} = Y_{n+1}^{(j)} \\ X_{n+1}^{(j)} = Y_{n+1}^{(i)} \end{cases} ;$$
    else
        $$\begin{cases} X_{n+1}^{(i)} = Y_{n+1}^{(i)} \\ X_{n+1}^{(j)} = Y_{n+1}^{(j)} \end{cases} ;$$
    end
    For all $k \in [\![1, K]\!]$, $k \neq i, j$, set $X_{n+1}^{(k)} = Y_{n+1}^{(k)}$ .
end
```

1. Implement the Parallel Tempering algorithm.

2. In order to illustrate the performance of the algorithm, use your code to sample from the distribution π of the previous exercise. Use the algorithm with $K = 5$ and with the following temperatures ladder :

$$(T_1,\ldots,T_5) = (\,60,\ 21.6,\ 7.7,\ 2.8,\ 1\,)\,.$$

For the Metropolis-Hastings step (line 3), take as proposal distribution the bivariate Gaussian distribution centered at $X_n^{(i)}$, with variance-covariance matrix $\tau_i^2 \mathrm{I}_2$:

$$\forall i \in [\![1, K]\!], \qquad Y_{n+1}^{(i)} \sim \mathcal{N}_{\mathbb{R}^2}(X_n^{(i)}, \tau_i^2 \mathrm{I}_2) \quad \text{where} \quad \tau_i = 0.25\sqrt{T_i}\,.$$

The scale parameters τ_i are tuned to ensure a reasonable acceptance rate in the algorithm.

In practice, the performance of the Parallel Tempering algorithm strongly depends on the choice of the temperatures ladder, the number of chains and the choice of proposal kernels. For most distributions, tuning these parameters may be infeasible. In [MMV13], the authors have proposed an adaptive Parallel Tempering algorithm to address these difficulties.

Exercise 3: Equi-Energy Sampler

Unlike the Parallel Tempering algorithm, the Equi-Energy Sampler (EES) [KZW06] does not use only the current state of the auxiliary chains. In the Equi-Energy Sampler, the whole past of the auxiliary chains is used to improve the mixing of the algorithm (the chains move more efficently in the support of the target distribution).

Let S be a positive integer. We consider *energy levels* $(\xi_0,\ldots,\xi_S)$ such that :

$$\xi_0 = 0 < \xi_1 < \ldots \xi_{S-1} < \xi_S = +\infty\,.$$

The energy levels allow to define *energy rings* $(\mathcal{A}_i)_{1\leqslant i\leqslant S}$ as follows :

$$\forall i \in [\![1, S]\!], \qquad \mathcal{A}_i = \{\,\mathbf{x} \in X \mid \xi_{i-1} \leqslant \pi(\mathbf{x}) < \xi_i\,\}\,.$$

As in the Parallel Tempering algorithm, in the Equi-Energy Sampler, K chains run in parallel and, for $1 \le i \le K$, the i-th chain targets the tempered distribution $\pi_i = \pi^{1/T_i}$, where $(T_1,\ldots,T_K)$ is a temperatures ladder such that

$$T_1 > \ldots > T_K = 1\,.$$

In the algorithm, the first chain $(X_n^{(1)})_{n\in\mathbb{N}}$ is defined using a Metropolis-Hastings transition kernel and the $K-1$ other chains $(X_n^{(j)})_{n\in\mathbb{N}}$ is constructed using the previous chain : For all $j \in [\![2, K]\!]$,

- With probability $\varepsilon > 0$, the new state $X_{n+1}^{(j)}$ is chosen uniformly among the past of the chain $(X_n^{(j-1)})_{n\in\mathbb{N}}$ and in the same energy ring as the current state $X_n^{(j)}$.
- With probability $1-\varepsilon$, a new state $X_{n+1}^{(j)}$ is proposed using a Metropolis-Hastings kernel $P^{(j)}(X_n^{(j)},\cdot)$ having π_j as stationnary distribution.

More precisely, Equi-Energy Sampler is described at algorithm 2.

Algorithm 2: Equi-Energy Sampler

```
For all $i \in [\![1, K]\!]$, initialize $X_0^{(i)}$ ;
for $n = 1$ to $\mathrm{N_{iter}}$ do
    Draw $Y_{n+1}^{(1)}$ using the transition kernel $P^{(1)}(X_n^{(1)}, \cdot)$ ;
    for $j = 2$ to $K$ do
        Draw $U \sim \mathcal{U}([0,1])$ ;
        if $U \leqslant \varepsilon$ then
            Compute the energy ring $\mathcal{A}_\ell$ of the current state $X_n^{(j)}$ ;
            Choose a new candidate $Y_{n+1}^{(j)}$ uniformly among $\{X_k^{(j-1)}, 1 \le k \le n\} \cap \mathcal{A}_\ell$ ;
            Compute the acceptance probability $\alpha(X_n^{(j)}, Y_{n+1}^{(j)})$ given by :
                $$\alpha(X_n^{(j)}, Y_{n+1}^{(j)}) = \min\left(1\,,\, \frac{\pi_j(Y_{n+1}^{(j)})\,\pi_{j-1}(X_n^{(j)})}{\pi_{j-1}(Y_{n+1}^{(j)})\,\pi_j(X_n^{(j)})}\right) ;$$
            With probability $\alpha(X_n^{(j)}, Y_{n+1}^{(j)})$, set $X_{n+1}^{(j)} = Y_{n+1}^{(j)}$ and with probability
            $1 - \alpha(X_n^{(j)}, Y_{n+1}^{(j)})$, set $X_{n+1}^{(j)} = X_n^{(j)}$ ;
        else
            Draw $X_{n+1}^{(j)}$ using the transition kernel $P^{(j)}(X_n^{(j)}, \cdot)$.
        end
    end
end
```

1. Implement the Equi-Energy Sampler

2. Use the Equi-Energy Sampler to sample from the target distribution π. To do so, use the algorithm with $K = 5$ and with the following temperatures ladder

$$(T_1, \dots, T_5) = (\,60,\ 21.6,\ 7.7,\ 2.8,\ 1\,)$$

and the energy levels are chosen as follows :

$$(\xi_1, \dots, \xi_4) = (\,e^{-63.2}\,,\ e^{-20}\,,\ e^{-6.3}\,,\ e^{-2}\,)\,.$$

For the Metropolis-Hastings steps, take as proposal distribution the bivariate Gaussian distribution centered at $X_n^{(i)}$, with variance-covariance matrix $\tau_i^2 \mathrm{I}_2$:

$$\forall i \in [\![1, K]\!], \qquad Y_{n+1}^{(i)} \sim \mathcal{N}_{\mathbb{R}^2}(X_n^{(i)}, \tau_i^2 \mathrm{I}_2)\,.$$

The scale parameters τ_i are chosen as before. The parameters ε is chosen equal to 0.1.

3. Compare the performance of the Equi-Energy Sampler to the Parallel Tempering algorithm. In order to compare the mixing of the two algorithms, in [KZW06], the authors propose to run both algorithms several times and monitor the number of times the samples visit the modes of π during the last 2000 iterations. Discuss the performance of the Equi-Energy Sampler with different ε.

As for the Parallel Tempering algorithm, in practice, the performance of the Equi-Energy Sampler depends on the choice of the energy levels and proposal distributions. Tuning these parameters may be very difficult when the target distribution π is known only up to a normalizing constant. In [SFM13], A. Schreck et al. proposed an adaptive Equi-Energy Sampler which automatically tunes these parameters.

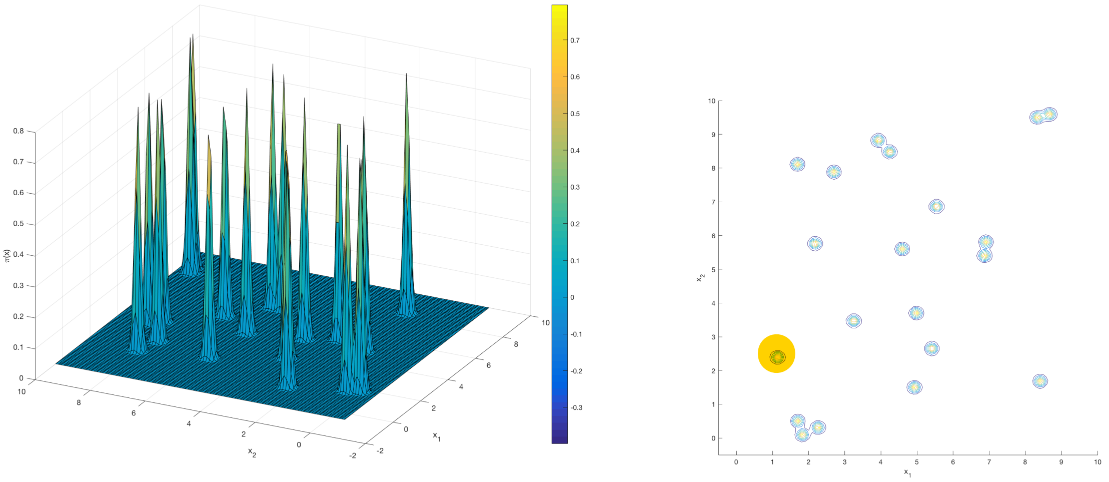

Figure 1: Mixture of 20 Gaussian distributions

References

[ED05] David J Earl and Michael W Deem. Parallel tempering: Theory, applications, and new perspectives. *Physical Chemistry Chemical Physics*, 7(23):3910–3916, 2005.

[Gey91] Charles J Geyer. Markov chain monte carlo maximum likelihood. 1991.

[KZW06] SC Kou, Qing Zhou, and Wing Hung Wong. Discussion paper equi-energy sampler with applications in statistical inference and statistical mechanics. *The annals of Statistics*, pages 1581–1619, 2006.

[LW01] Faming Liang and Wing Hung Wong. Real-parameter evolutionary monte carlo with applications to bayesian mixture models. *Journal of the American Statistical Association*, 96(454):653–666, 2001.

[MMV13] Błażej Miasojedow, Eric Moulines, and Matti Vihola. An adaptive parallel tempering algorithm. *Journal of Computational and Graphical Statistics*, 22(3):649–664, 2013.

[SFM13] Amandine Schreck, Gersende Fort, and Eric Moulines. Adaptive equi-energy sampler: Convergence and illustration. *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, 23(1):5, 2013.