Stochastic Simulation
Professor: Prof. Fabio Nobile
Assistant: Eliott Van Dieren

Week of 2025-11-24 to 2025-11-28

Lab 11 of Thursday 27th November 2025

Exercice 1.

In many applications of interest, it is not uncommon to encounter the need for sampling from a multi-modal distribution f. The theory developed so far is directly applicable to these types of distributions. However, in practice, sampling from these distributions using MCMC can be computationally challenging, as we will investigate in this problem. Throughout this exercise, we will consider the bi-modal distribution on $\mathbb{R}$

$$f(x;\gamma,x_0)=\frac{e^{-\gamma(x^2-x_0)^2}}{Z},\quad \gamma>0, \tag{1.1}$$

where Z is some normalizing constant. Depending on the values of γ and x_0, designing a sampling strategy to properly sample from (1.1) can become challenging.

To solve this exercise, we will use the random walk Metropolis (RWM) algorithm which is a Metropolis-Hastings algorithm that uses a proposal distribution of the form $q(x,y)=\mathcal{N}(y;x,\sigma^2)$, where

$$\mathcal{N}(y;x,\sigma^2):=\frac{1}{\sqrt{2\pi\sigma^2}}\exp\left(-\frac{(y-x)^2}{2\sigma^2}\right)$$

is the Gaussian density with mean x and variance σ^2. The parameter σ is a tuning parameter that controls the size of the steps taken by the Markov chain.

Intuitively, if the two modes of (1.1) are too far apart, using a RWM might not work, if the step-size is too small since the chain might get stuck in one of the modes and not be able to explore the other mode with acceptable probability. Conversely, a RWM with very large *steps* might tend to reject quite often, thus rendering the whole sampling procedure inefficient. We begin by verifying this. Implement the RWM algorithm using as proposal distribution $q(x,y)=\mathcal{N}(y;x,\sigma^2)$ and target distribution $f(x;\gamma,x_0)$ for $\gamma=1$, $x_0=1,4,9,25$. Try different choices of σ. Discuss the quality of your samples by analyzing the trace-plots (one realization of the chain), autocorrelation functions and histograms of the chains obtained.

Solution

An exemplary implementation of this problem is given at the end of the exercise. We implement the sampler for different values of x_0 and σ and show the results in Figure 1. As we can see from Figure 1, when x_0 and σ are of similar magnitude (top row), the sampler is able to correctly explore the distribution. This is contrary to what happens in the bottom row, where x_0 is much larger than σ and as such, the sampler tends to get stuck at one of the peaks of the distribution. This in turn can be fixed by increasing the size of σ, as shown in figure 2. Notice here that for σ around 10, the chain has better mixing and a more rapidly decaying auto-correlation plot, albeit the acceptance rate is 0.017, which is quite small. In general, it is difficult to choose an appropriate σ to sample from this type of multi-modal distributions when using random walk

Metropolis. Some recent advances to overcome this difficulty are the so-called Hamiltonian Monte Carlo, the delayed-rejection Metropolis-Hastings, and the parallel-tempering algorithm.[1]

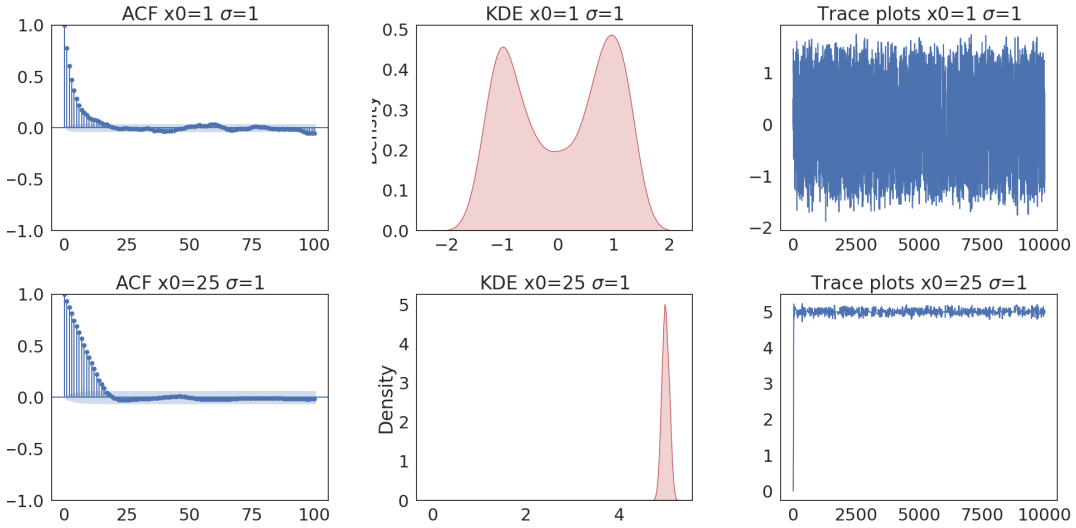

Figure 1: Results for $\sigma = 1$ and $x_0 = 1$ (top) and $x_0 = 25$ (bottom). The blue-shaded part on the auto-correlation plot joins the boundaries of an approximate 95% interval for the individual correlations.

[1]We refer the interested reader to S. Brooks, A. Gelman. *Handbook of Markov Chain Monte Carlo*, 2011, CRC press.

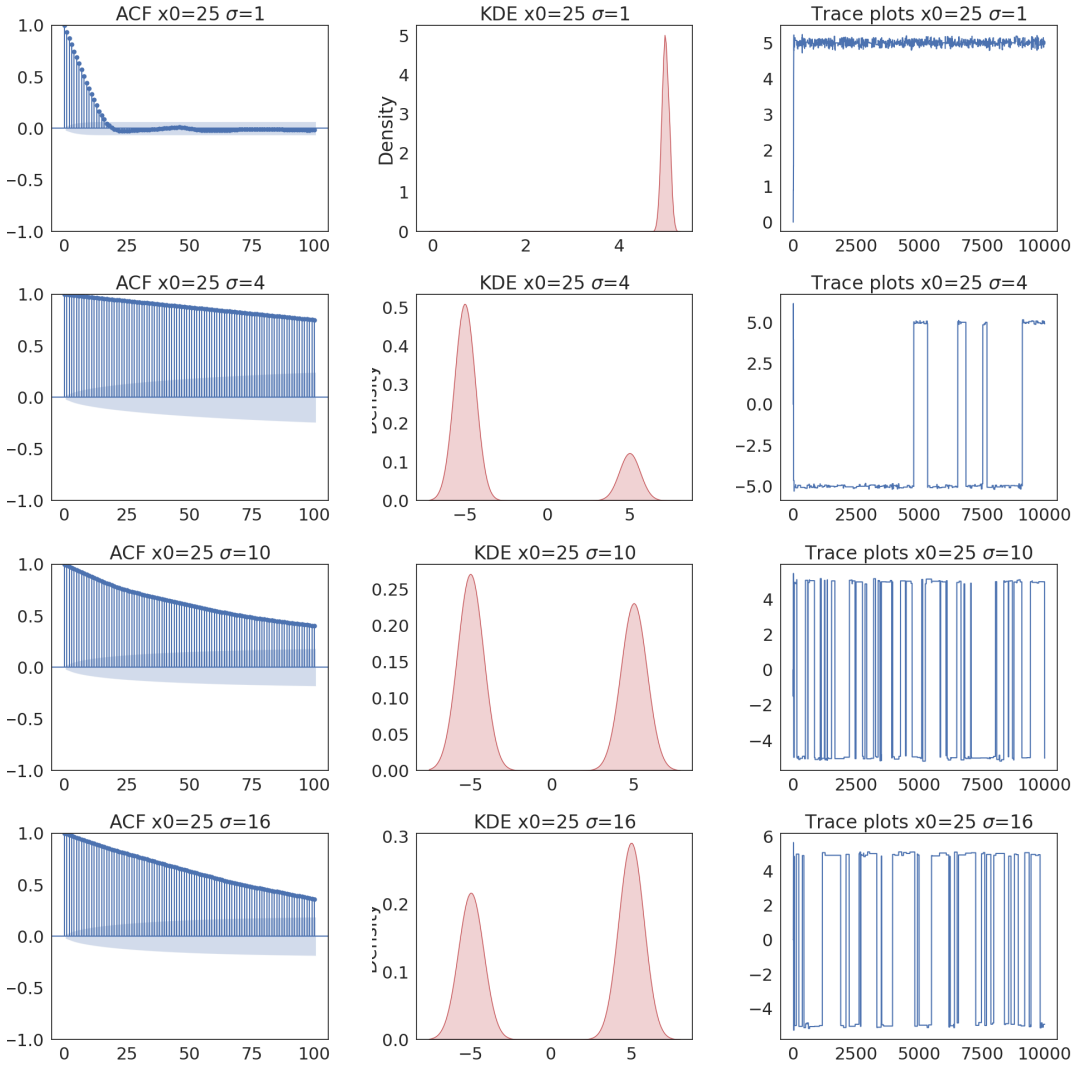

Figure 2: Results for $x_0 = 25$ and different values of σ, from top to bottom $\sigma = 1, 4, 10, 16$. The blue-shaded part on the autocorrelation plot joins the boundaries of an approximate 95% interval for the individual correlations.

Python code:

```
import numpy as np
import matplotlib.pyplot as plt
import statsmodels.graphics.tsaplots as sm
import seaborn as sns; sns.set(color_codes=True)# Defines the pdf
sns.set(font_scale=2) #fontsize in plots
sns.set_style("white")

np.random.seed(42)

def f(x,x0,gamma=1):
    return np.exp(-gamma*(x**2-x0)**2 )
```

```
#defines the metropolis-hastings routine
def MH(sigma,x0):
    N=10000
    X=np.zeros(N)
    X[0]=0;
    px=f(X[0],x0)

    for i in range(N-1):
        y=X[i]+sigma*np.random.standard_normal(1)
        py=f(y,x0)
        px=f(X[i],x0)
        if py/px > np.random.random(1):
            X[i+1]=y
        else:
            X[i+1]=X[i]

    plt.plot(X)
    plt.gca().set_rasterized(True)
    plt.title('Trace plots x0='+str(x0)+' $\sigma$='+str(sigma))
    plt.show()

    sm.plot_acf(X,lags=100)
    plt.gca().set_rasterized(True)
    plt.title('ACF x0='+str(x0)+' $\sigma$='+str(sigma))
    plt.show()

    sns.kdeplot(X, shade=True, color="r")
    plt.gca().set_rasterized(True)
    plt.title('KDE x0='+str(x0)+' $\sigma$='+str(sigma))
    plt.show()

    return X

## Runs experiments
xx=np.array([1,25])
ss=np.array([1,4,10,16] )

for i in range(len(xx)):
    for j in range(len(ss)):
        MH(ss[j],xx[i])
```

Exercice 2.

Ideally, we would like to obtain (approximately) i.i.d samples from a target distribution f using Markov Chain Monte Carlo (MCMC) algorithms. One practical way of doing so is via *sub-sampling* (also called *batch sampling*), which is implemented to reduce or eliminate correlation between the successive values in the Markov chain. That is, instead of considering the entire chain $\{X_n : n \geqslant 0\}$, say, this technique sub-samples the chain with a batch size $k > 1$, so that only the values $\{X_{kn} : n \geqslant 0\}$ are considered. If the covariance $\mathrm{Cov}_f(X_0, X_n)$ vanishes as $n \to \infty$, then the idea of sub-sampling is quite natural since X_{kn} and $X_{k(n+1)}$ can be considered to be approximately independent for k sufficiently big; estimating such a k may be difficult in practice though. While sub-sampling provides a way of generating (approx.) i.i.d. samples from f and may thus be useful to assess the convergence of a MCMC method, it necessarily leads to an efficiency loss. Let $\{X_n \in \mathbb{R}^d : n \geqslant 0\}$ be a Markov chain with a unique stationary distribution f, and $X_0 \sim f$ (i.e., the chain is at equilibrium). Take $\phi : \mathbb{R}^d \to \mathbb{R}$ such that $\mathbb{E}_f(|\phi|^2) < \infty$ and consider two estimators for $\mu = \mathbb{E}_f(\phi)$, namely one that uses the entire Markov chain ($\hat{\mu}$) and one

based on sub-sampling $(\hat{\mu}_k)$ using only every k-th value:

$$\hat{\mu} = \frac{1}{Nk}\sum_{n=1}^{Nk}\phi(X_n)\ , \quad \text{and} \quad \hat{\mu}_k = \frac{1}{N}\sum_{n=1}^{N}\phi(X_{nk})\ .$$

Show that the variance of $\hat{\mu}$ satisfies $\mathrm{Var}_f(\hat{\mu}) \leqslant \mathrm{Var}_f(\hat{\mu}_k)$ for every $k > 1$.

Solution

Let $k > 1$. Then define $\hat{\mu}_k^{(0)}, \hat{\mu}_k^{(1)}, \ldots, \hat{\mu}_k^{(k-1)}$ as the shifted versions of $\hat{\mu}_k$, in the sense that:

$$\hat{\mu}_k^{(j)} = \frac{1}{N}\sum_{n=1}^{N}\phi(X_{nk-j})\ , \quad j = 0, 1, \ldots, k-1\ .$$

Being the chain stationary, all $\hat{\mu}_k^{(j)}$ are identically distributed (but not independent), and in particular $\hat{\mu}_k^{(0)} = \hat{\mu}_k$ in distribution. Notice that the estimator $\hat{\mu}$ can be written as:

$$\hat{\mu} = \frac{1}{k}\sum_{j=0}^{k-1}\hat{\mu}_k^{(j)}\ ,$$

so that the variance of $\hat{\mu}$ satisfies:

$$\begin{aligned}\mathrm{Var}_f(\hat{\mu}) = \mathrm{Var}_f\left(\frac{1}{k}\sum_{j=0}^{k-1}\hat{\mu}_k^{(j)}\right) &= \frac{\mathrm{Var}_f(\hat{\mu}_k^{(0)})}{k} + \sum_{i\neq j}\frac{\mathrm{Cov}_f(\hat{\mu}_k^{(i)}, \hat{\mu}_k^{(j)})}{k^2} \\ &\leqslant \frac{\mathrm{Var}_f(\hat{\mu}_k^{(0)})}{k} + \sum_{i\neq j}\frac{\sqrt{\mathrm{Var}_f(\hat{\mu}_k^{(i)})\,\mathrm{Var}_f(\hat{\mu}_k^{(j)})}}{k^2}\end{aligned}$$

in view of the Cauchy–Schwarz inequality and the stationarity. The claim then follows form the stationarity of the Markov chain again, indeed

$$\begin{aligned}\mathrm{Var}_f(\hat{\mu}) &\leqslant \frac{\mathrm{Var}_f(\hat{\mu}_k^{(0)})}{k} + \sum_{i\neq j}\frac{\sqrt{\mathrm{Var}_f(\hat{\mu}_k^{(i)})\,\mathrm{Var}_f(\hat{\mu}_k^{(j)})}}{k^2} \\ &= \frac{\mathrm{Var}_f(\hat{\mu}_k^{(0)})}{k} + \frac{k-1}{k}\,\mathrm{Var}_f(\hat{\mu}_k^{(0)}) = \mathrm{Var}_f(\hat{\mu}_k)\ .\end{aligned}$$

Exercice 3.

Let $X \subset \mathbb{R}^d$ and $P_i : X \times \mathcal{B}(X) \to [0,1]$, $i = 1 \ldots, m$ be Markov transition kernels on X with $\mathcal{B}(X)$ the associated $\sigma-$algebra.

(a) Given $a_1, \ldots, a_m \in \mathbb{R}^+$, such that $\sum_{i=1}^m a_i = 1$, show that $P(x, A) = \sum_{i=1}^m a_i P_i(x, A)$ is a Markov transition kernel.

(b) Suppose that a measure $\pi : \mathcal{B}(X) \to [0,1]$ is invariant for each kernel P_i. Show that it is also invariant for $P = \sum_{i=1}^m a_i P_i$, where $a_1, \ldots, a_m \in \mathbb{R}^+$, such that $\sum_{i=1}^m a_i = 1$. If each P_i is reversible, is P reversible?

(c) Under the same assumptions of point (b), define the Markov operator $\mathcal{P}_i$ associated to P_i (i.e., $\pi\mathcal{P}_i(A) = \int P(x, A)d\pi(x), \forall A \in \mathcal{B}(X)$). Then, show that π is also invariant for $\mathcal{P} = \mathcal{P}_{i_1} \circ ... \circ \mathcal{P}_{i_k}$, for any choice of $i_1, ..., i_k$. If each P_i is reversible, for which choice of $i_1, ..., i_k$ is $\mathcal{P}$ reversible?

Solution

(a) We need to verify that

1) $\forall A \in \mathcal{B}(X)$, $P(\cdot, A)$ is measurable.
2) $\forall x \in X$, $P(x, \cdot)$ is a probability measure on $(X, \mathcal{B}(X))$.

For the first we have that $\forall A \in \mathcal{B}(X)$, $P(\cdot, A) = \sum a_i P_i(\cdot, A)$ which is measurable as a linear combination of measurable functions. Moreover, $\forall x \in X$, $P(x, \cdot) = \sum_i a_i P_i(x, \cdot)$ is a probability measure on $(X, \mathcal{B}(X))$ since it is a convex combination of probability measures. Notice, in particular, that $P(x, X) = \sum_i a_i P_i(x, X) = \sum_i a_i = 1$.

(b) Each kernel P_i preserves π, that is

$$\int_X P_i(x, A)\pi(dx) = \pi(A), \quad \forall A \in \mathcal{B}(X), \quad \forall i = 1, ..., m. \tag{3.1}$$

We have then $\int_X P(x, A)\pi(dx) = \sum_i a_i \int_X P_i(x, A)\pi(dx) = \sum_i a_i \pi(A) = \pi(A), \forall A \in \mathcal{B}(X)$, hence P also preserves π.

(c) Notice that, since $\pi\mathcal{P}_i = \pi$ by assumption, we have $\pi\mathcal{P} = \pi\mathcal{P}_{i_1} \circ ... \circ \mathcal{P}_{i_k} = \pi$. Furthermore, if each $\mathcal{P}_i$ is reversible, and $(i_1, ..., i_k) = (i_k, ..., i_1)$, then $\mathcal{P}$ is reversible (sufficient condition).