Avd. Matematisk statistik

KTH Matematik

HOME ASSIGNMENT 2, SF2955 COMPUTER INTENSIVE METHODS IN MATHEMATICAL STATISTICS

Examiner: Jimmy Olsson
All MATLAB-files needed are available through the course home page.
The following is to be submitted in Canvas by **Wednesday 15 May, 12:00:00**:

- A report, named `group number-HA2-report.pdf`, of **maximum 7 pages** in `pdf` format. The report should provide detailed solutions to all problems. The presentation should be self-contained and understandable without access to the code.
- *All* your `m`-files (or similar depending on your language of choice) along with a file named `group number-HA2-matlab.m` that runs your analysis.

Discussion between groups is permitted, as long as your report reflects your own work.

Statistical inference from coal mine disaster and complex posterior sampling using Markov chain Monte Carlo

29 April 2024

Bayesian analysis of coal mine disasters—constructing a complex MCMC algorithm

In this problem we will generalise the coal mining example in the book (See Chapter 11.2.1) from one breakpoint to $d-1$ breakpoints. First we need some notation. Let $t_1 = 1851$ and $t_{d+1} = 1963$ be the fixed end points of the dataset and denote by t_i, $i = 2, \ldots, t_d$, the breakpoints. We collect end points and break points in a vector $\boldsymbol{t} = (t_1, \ldots, t_{d+1})$. The disaster intensity in each interval $[t_i, t_{i+1})$ is λ_i and we let $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_d)$.
Another difference from the example in the book is that instead of calculating the number of disasters each year we will use time continuous data where $\boldsymbol{\tau} = (\tau_1, \ldots, \tau_n)$ denotes the time points of the $n = 191$ disasters (available in the file `coal-mine.csv`). We model the data on the interval $t_1 \leq t \leq t_{d+1}$ using an inhomogeneous Poisson process with intensity

$$\lambda(t) = \sum_{i=1}^{d} \lambda_i \mathbb{1}_{[t_i,t_{i+1})}(t).$$

From the time points of the disasters we compute

$$n_i(\boldsymbol{\tau}) = \text{number of disasters in the sub-interval } [t_i, t_{i+1}) = \sum_{j=1}^{n} \mathbb{1}_{[t_i,t_{i+1})}(\tau_j).$$

We put a $\Gamma(2, \theta)$ prior on the intensities with a $\Gamma(2, \vartheta)$ hyperprior on θ, where ϑ is a fixed hyperparameter that needs to be specified. In addition, we put a prior

$$f(\boldsymbol{t}) \propto \begin{cases} \prod_{i=1}^{d}(t_{i+1} - t_i), & \text{for } t_1 < t_2 < \ldots < t_d < t_{d+1}, \\ 0, \quad \text{otherwise}, \end{cases}$$

on the breakpoints, preventing the same from being located too closely. Using theory of Poisson processes, it can be shown that

$$f(\boldsymbol{\tau}|\boldsymbol{\lambda}, \boldsymbol{t}) \propto \exp\left(-\sum_{i=1}^{d} \lambda_i(t_{i+1} - t_i)\right) \prod_{i=1}^{d} \lambda_i^{n_i(\boldsymbol{\tau})}.$$

To sample from the posterior $f(\theta, \boldsymbol{\lambda}, \boldsymbol{t} \mid \boldsymbol{\tau})$ we will construct a hybrid MCMC algorithm as follows. All components except the breakpoints $\boldsymbol{t}$ can be updated using Gibbs sampling. To update the breakpoints we use a Metropolis-Hastings step. There are several possible proposal distributions for the MH step:

- *Random walk proposal:* update one breakpoint at a time. For each breakpoint t_i we generate a candidate t_i^* according to

  $$t_i^* = t_i + \epsilon, \quad \text{with} \quad \epsilon \sim \text{Unif}(-R, R)$$

  and $R = \rho(t_{i+1} - t_{i-1})$.

- *Independent proposal:* update one breakpoint at a time. For each breakpoint t_i we generate a candidate t_i^* according to

  $$t_i^* = t_{i-1} + \varepsilon(t_{i+1} - t_{i-1}), \quad \text{with} \quad \varepsilon \sim \text{Beta}(\rho, \rho).$$

This corresponds to a scaled and shifted beta-distribution for t_i^* with density function

$$f(t_i|t_{i+1}, t_{i-1}) = \frac{\Gamma(2\rho)}{\Gamma(\rho)^2} \frac{(t_i - t_{i-1})^{\rho-1} (t_{i+1} - t_i)^{\rho-1}}{(t_{i+1} - t_{i-1})^{2\rho-1}}.$$

In both cases ρ is a tuning parameter of the proposal distributions.

Problem 1

(a) Compute, up to normalizing constants, the marginal posteriors $f(\theta \mid \boldsymbol{\lambda}, \boldsymbol{t}, \boldsymbol{\tau})$, $f(\boldsymbol{\lambda} \mid \theta, \boldsymbol{t}, \boldsymbol{\tau})$, and $f(\boldsymbol{t} \mid \theta, \boldsymbol{\lambda}, \boldsymbol{\tau})$. In addition, try to identify the distributions.

(b) Construct a hybrid MCMC algorithm that samples from the posterior $f(\theta, \boldsymbol{\lambda}, \boldsymbol{t} \mid \boldsymbol{\tau})$. Pick *one* of the possible updating options for $\boldsymbol{t}$.

(c) Investigate the behavior of the MCMC chain for 1, 2, 3, and 4 breakpoints.

(d) How sensitive are the posteriors to the choice of the hyperparameter ϑ?

(e) How sensitive is the mixing and the posteriors to the choice of ρ in the proposal distribution?

Sampling from a circle-shaped posterior using Hamiltonian Monte Carlo

Consider the following Bayesian model with conditionally independent observations

$$y_i \mid (\theta_1, \theta_2) \sim \mathrm{N}(\theta_1^2 + \theta_2^2, \sigma^2), \quad i = 1, \ldots, n,$$

where σ is known but $\boldsymbol{\theta} := (\theta_1, \theta_2) \in \mathbb{R}^2$ unknown. The parameter $\boldsymbol{\theta}$ has prior distribution $\mathrm{N}_2(\mathbf{0}, \Sigma)$ where the covariance matrix Σ is a known hyperparameter. Given the data $\mathbf{y} = (y_1, \ldots, y_n)$, we want to sample from the posterior $f(\boldsymbol{\theta} \mid \mathbf{y})$ using HMC. The density $f(\boldsymbol{\theta} \mid \mathbf{y})$, up to a normalizing constant, is displayed in Figure 1.

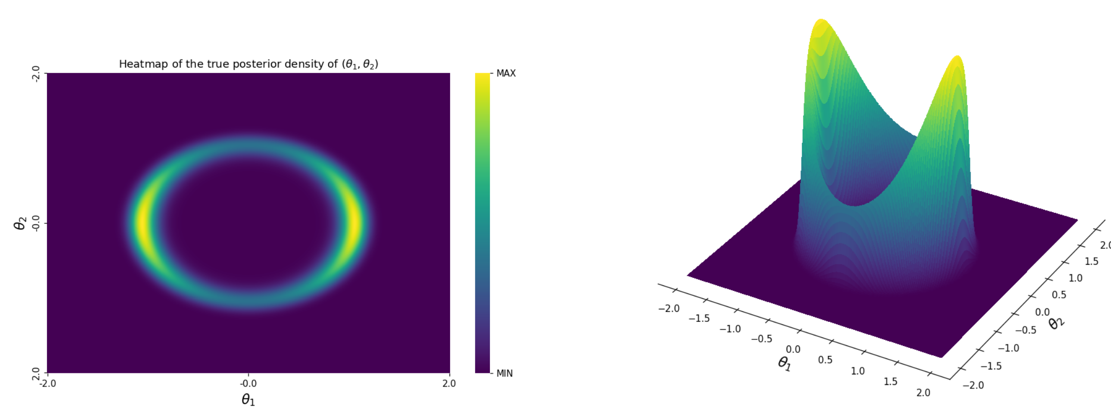

Figure 1: Heatmap and 3D plot of the posterior density $f(\boldsymbol{\theta} \mid \mathbf{y})$, up to a normalizing constant.

Problem 2

(a) Compute, up to an additive constant not depending on $\boldsymbol{\theta}$, the logarithm of the posterior $\ln f(\boldsymbol{\theta} \mid \mathbf{y})$ and its gradient $\nabla_{\boldsymbol{\theta}} \ln f(\boldsymbol{\theta} \mid \mathbf{y})$ with respect to $\boldsymbol{\theta}$.

(b) Design an HMC algorithm to sample from $f(\boldsymbol{\theta} \mid \mathbf{y})$ (provide a pseudocode).

(c) Given the known parameters

$$\sigma = 2, \quad \Sigma = \begin{bmatrix} 5 & 0 \\ 0 & 0.5 \end{bmatrix},$$

and the data $\mathbf{y}$ (with $n = 100$, stored in the file `hmc-observations.csv`), implement your HMC algorithm and compare it to a standard Metropolis-Hastings approach with bivariate random walk proposal, *i.e.*, given $\boldsymbol{\theta}$, letting the candidate have distribution

$$\boldsymbol{\theta}^* \sim \mathrm{N}_2(\boldsymbol{\theta}, \zeta^2 I)$$

for some $\zeta > 0$ to be calibrated.

(i) How efficiently is the support of the target distribution explored in each case?[1] Compare the acceptance rates and the autocorrelation functions of the samples.

(ii) Investigate the performances of both algorithms varying the step size ε and the number of steps L of the leapfrog integrator, as well as the standard deviation ζ of the Gaussian random walk.

Good luck!

[1] In order to compare your samples with the heatmap of Figure 1, it may be useful to display a 2D histogram plot, using for instance the Matlab function `HIST2D(...,'tile')` or, in Python, `matplotlib.pyplot.hist2d`.