TMA4300 Computer Intensive Statistical Methods
Exercise 2, Spring 2017

Note: Solutions of Problems B, C and D must be handed in no later than **20th March 2017, 16:00**. All answers including derivations, computer code and graphics (preferably in one pdf document) should be submitted to

Xin Luo (xin.luo@ntnu.no).

Getting started: The aim of this exercise is to get experience with MCMC algorithms by implementing and running such algorithms for various target distributions, and to use the MCMC output to estimate properties of the target distributions. Whenever you need to sample from a standard univariate distributions you may use the build-in random number functions in R or you can use the corresponding functions you coded in Exercise 1. However, remember that there are two common parameterisations for the gamma distribution, so if you use the build-in function *rgamma* you must first check what parameterisation is used by this function.

Important: For each function of code chunk you write in this exercise you should try to check that it is working properly. You typically do not have available analytical properties for the target distribution as in Exercise 1, but you should still carefully consider whether the simulated values are reasonable.

Whenever you are using MCMC output to estimate properties of the target distribution you need to discuss how you decided the length of the burn-in period and include the relevant plots in your solution.

Problem A: MCMC for a toy example

In this problem we consider the following Bayesian model. We have observations $x_1, \ldots, x_n$, which we assume to be independent and identically distributed $x_i|\mu, \varphi \sim \mathrm{N}(\mu, 1/\varphi)$, given some parameters μ and φ. Except in the last item of this problem, we adopt a prior where μ and φ are independent, μ is normal with zero mean and unit variance, and φ is gamma distributed with mean one and standard deviation two.

1. As this is a toy example, start this exercise by generating your own data $x_1, \ldots, x_n$, i.e choose your favourite values for μ and φ and simulate $n = 20$ observations $x_1, \ldots, x_n$ independently from $\mathrm{N}(\mu, 1/\varphi)$. In the rest of this exercise we will pretend that the true values for μ and φ are unknown and learn about these values by simulating values from the posterior distribution $\mu, \varphi|x_1, \ldots, x_n$.

2. Show that the full conditional distribution for μ is a normal distribution and find formulas for the parameters of this normal distribution. Correspondingly, show that the full conditional distribution for φ is a gamma distribution and find formulas for the parameters of this gamma distribution.

3. Implement a Gibbs sampler algorithm for the posterior distribution $\mu, \varphi|x_1, \ldots, x_n$. Estimate the marginal posterior distributions $f(\mu|x_1, \ldots, x_n)$ and $f(\varphi|x_1, \ldots, x_n)$ by making histograms of the simulated values for μ and φ, respectively. Remember to discard the burn-in period! Use also the simulated values to estimate $\mathrm{E}[\mu|x_1, \ldots, x_n]$, $\mathrm{E}[\sqrt{1/\varphi}|x_1, \ldots, x_n]$ and $\mathrm{Corr}[\mu, \sqrt{1/\varphi}|x_1, \ldots, x_n]$. You may repeat the simulation experiment for different simulated data sets (i.e. you may vary the values of μ, φ and n used when simulating your "observations"). Can you intuitively understand what you observe?

4. Implement a single site random walk proposal Metropolis–Hastings algorithm for the posterior distribution $\mu, \varphi|x_1, \ldots, x_n$. Thus, an update for μ is performed by first proposing a potential new value for μ, $\widetilde{\mu}$, according to $\widetilde{\mu}|\mu \sim \mathrm{N}(\mu, \sigma_\mu^2)$ and then accepting or rejecting the proposed value

according to the Metropolis–Hastings acceptance probability. Correspondingly, an update for φ is performed by first proposing a potential new value for φ, $\widetilde{\varphi}$, according to $\widetilde{\varphi}|\varphi \sim \mathrm{N}(\varphi, \sigma_\varphi^2)$ and then accepting or rejecting it according to the Metropolis–Hastings acceptance probability. Run the algorithm for different values of the tuning parameters σ_μ^2 and σ_φ^2 and observe how this influence the length of the burn-in period and mixing properties of the Markov chain, but do not influence the limiting distribution. Also check that the simulation results are consistent with your results in A.3.

5. Implement a block random walk proposal Metropolis–Hastings algorithm for the posterior distribution $\mu, \varphi|x_1, \ldots, x_n$. Thus, one iteration consists of first proposing new values for (μ, φ) according to $\widetilde{\mu}|\mu \sim \mathrm{N}(\mu, \sigma_\mu^2)$ and $\widetilde{\varphi}|\varphi \sim \mathrm{N}(\varphi, \sigma_\varphi^2)$ independently, and thereafter accepting or rejecting $(\widetilde{\mu}, \widetilde{\varphi})$ jointly according to the Metropolis–Hastings acceptance probability. Run the algorithm for different values of the tuning parameters σ_μ^2 and σ_φ^2 and observe how this influence the length of the burn-in period and mixing properties of the Markov chain, but again do not influence the limiting distribution. Again check that the simulated values are consistent with your results in A.3.

6. As the last MCMC algorithm in this exercise, implement a single site random walk Metropolis–Hastings algorithm for the posterior distribution $\mu, \varphi|x_1, \ldots, x_n$ where the update for μ is as in A.4, but the potential new value for φ, $\widetilde{\varphi}$, is generated by first sampling $u \sim \mathrm{Unif}(1/a, a)$ for some value $a > 1$ and then setting $\widetilde{\varphi} = \varphi \cdot u$. Run the algorithm for different values of the tuning parameters σ_μ^2 and a and evaluate the results as before.

Problem B: Ising model

Consider a 2D rectangular lattice consisting of $m \times n$ nodes. To each node (i, j) in the lattice associate a stochastic variable $x_{ij} \in \{0, 1\}$, and let $x = \{x_{ij}; i = 1, \ldots, m, j = 1, \ldots, n\}$. The Ising model is then defined by

$$f(x) \propto \exp\left\{-\beta \sum_{(i,j)\sim(k,l)} I(x_{ij} \neq x_{kl})\right\},$$

where β is a parameter, the sum is over all pairs of nodes that are (first order) neighbours, and $I(\cdot)$ is the indicator function taking the value one if the argument is true and zero otherwise.

1. Define and implement a Metropolis–Hastings algorithm for $f(x)$. Run the Metropolis–Hastings algorithm for $\beta = 0.5$, $\beta = 0.87$ and $\beta = 1.0$. Try four different initial values for x, *a*) all x_{ij} equal to zero, *b*) all x_{ij} equal to one, *c*) independent random values in each node, and *d*) a checkerboard pattern. Compare the results and evaluate the convergence properties of your simulation algorithm. Present results for a 50×50 or 100×100 lattice.

   Hint: When testing your algorithm it is best to use a small lattice, for example a 10×10 *or* 20×20 *lattice. You solution only needs to contain results for a large lattice.*

2. For the values of β (tried above) where you obtained convergence, use the simulation output to estimate to following.

   a) The distribution of

   $$\frac{1}{mn} \sum_{i=1}^{m} \sum_{j=1}^{n} x_{ij}$$

   and in particular the mean value

   $$\mu_a = \mathrm{E}\left[\frac{1}{mn} \sum_{i=1}^{m} \sum_{j=1}^{n} x_{ij}\right].$$

b) The mean value

$$\mu_b = \mathrm{E}\left[\frac{1}{(m-1)n}\sum_{i=1}^{m-1}\sum_{j=1}^{n} I(x_{ij} = x_{i+1,j})\right].$$

3. For μ_a and μ_b defined in B.2, estimate also a variance for each of the estimates you found in B.2, and use this to find 95% confidence intervals for μ_a and μ_b.

Problem C: Microarray data

In the two last exercises we will consider a simple hierarchical Bayesian model to analyse what is called *microarray data*. You can read more about microarray data, how they are produced and what they are used for by searching for microarray data on the web. Here only the stochastic model that we will apply is discussed. Our data consists of a matrix of (real) values, one for each of a set of *genes* for a number of samples (or patients). The patients belong to two groups, this can for example be patients with two different types of cancer or patients that have had two types of treatments. Thus, for a gene g, our data consists of $x_{gi}, i = 1, \ldots, S_1$ for one of the two types of patients, and $y_{gi}, i = 1, \ldots, S_2$ for the other group. Our focus here will be to decide whether gene g shows *differential expression*, i.e. whether the mean values for the two groups of patients differ. For this we will use the following simple model. Assume the data to be independent (given parameters ν_g, Δ_g and τ_g), and

$$x_{gi}|\nu_g, \Delta_g, \tau_g \sim \mathrm{N}\left(\nu_g + \Delta_g, \frac{1}{\tau_g}\right)$$

and

$$y_{gi}|\nu_g, \Delta_g, \tau_g \sim \mathrm{N}\left(\nu_g - \Delta_g, \frac{1}{\tau_g}\right).$$

Note that instead of parameterising by the variance σ_g^2, we use the *precision* $\tau_g = \frac{1}{\sigma_g^2}$. Thus, the question is whether Δ_g differ from zero or not. In this problem we will consider only one gene g at a time, whereas in the next problem we will consider the situation with many genes jointly.

Two data sets to be used both in this and the next problem can be downloaded from the course home page. For the first data the x_{gi} data is found in 'x1.txt' and the y_{gi} data in 'y1.txt'. Correspondingly, 'x2.txt' and 'y2.txt' contain the second data set. In the files each row is a gene and each column is a sample or patient. The second data set is a subset of the first data set.

1. Show that the normal distribution is the (conditional) conjugate distribution for ν_g, i.e. show that if ν_g to have a normal prior distribution then the full conditional $\nu_g|x_{g1}, \ldots, x_{gS_1}, y_{g1}, \ldots, y_{gS_2}, \Delta_g, \tau_g$ is also normal.

2. Similarly, show that the normal distribution is also the (conditional) conjugate distribution for Δ_g, and that the gamma distribution is the (conditional) conjugate distribution for τ_g.

3. Now assume the priors for ν_g and Δ_g to be normal, both with means zero and variances 100, and the prior for τ_g to a gamma distribution with mean 1 and variance 100. Visualise the resulting Bayesian model as a graphical model. Implement and run (for each of the two data sets) a Gibbs algorithm for the posterior distribution for ν_g, Δ_g and τ_g. Use the data in the first three rows (genes) of the data files, i.e. do separate Metropolis–Hastings runs for each of the first three genes in the data sets. Evaluate the convergence properties and visualise your simulation results. Will you conclude that any of these genes are differential expressed?

Problem D: More on microarray data

Now consider the situation with many genes $g = 1, \ldots, G$. For each gene g we adopt the same likelihood as above, but redefine the prior distribution. Note that we now have separate parameters ν_g, Δ_g and τ_g for each gene g. Apriori we now assume the ν_g's to be normally distributed with mean μ and precision ρ, the Δ_g's to be normally distributed with mean m and precision r, and the τ_g's to be gamma distributed with mean α and variance β. The hyper-parameters μ, ρ, m, r, α and β we assume to be apriori independent. For μ and m we assume (improper) uniform prior distributions on $(-\infty, \infty)$. For ρ we assume the (improper) prior $p(\rho) \propto 1/\rho$, and correspondingly for r, $p(r) \propto 1/r$. For α and β we use (improper) uniform prior distributions on $(0, \infty)$.

1. Visualise the resulting Bayesian model as a graphical model. Define a Metropolis–Hastings algorithm to simulate from the resulting posterior distribution, i.e. specify what kind of proposal distributions you will use, what the corresponding acceptance probabilities are, and how you plan to combine your updates.

2. Implement and run (for each of the two data sets) the Metropolis–Hastings algorithm. If your code runs very slowly you may reduce the number of genes you are using. Evaluate the convergence properties of your algorithm and visualise your simulation results. Compare the results for the first three genes with what you obtained in problem C. Discuss.

Oral presentations

Date	Problem	Team
17.03.2017	2: Problems A1, A2, A3 and A4	Solveig Fosal and Frida Marie Bruun
	2: Problems A5 and A6	Sabuj Bhowmick and Venuga Sivarajah
	2: Problem B1	Knut Nordanger
	2: Problems B2 and B3	Anders Sætherø and Scott Macody Lund
	2: Problem C1, C2 and C3	Hans Olav Vogt Myklebust and Markus Brabrand Urfjell