Mandatory assignment 3

Deadline: November 13th 2020 @ 12:00 (noon, Norwegian time).

Read carefully through the information about the page on Mandatory assignments on Canvas. Notice in particular that the assignments should be solved individually.

Hand in on Canvas. Submissions should be of **either** of the following types

- Submit two files: one pdf-file with a report containing the answers to the theory questions, and one file including the R-code.
- Submit two files: One R markdown (Rmd) file containing both theory answers and R-code, and a pdf-file with the output you obtain when running you R markdown file.

The first line of the R-code should be: `rm(list=ls())` . Check that the R-code file runs before you submit it. Use comments in the R-code to clearly identify which question each part of the R-code belong to. Also try to add some comments to explain important parts of the code. The file ending of the R-code file should be .R or .r. The report can be handwritten and scanned to pdf-file, or written in your choice of text editor and converted to pdf. Cite the sources you use.

Problems marked with an $^{\text{R}}$ should be solved in R, the others are theory questions.

Some of the questions are marked with "**Open-ended, no help**". These are intended to test whether you are able to answer more over-arching questions using the theory you have learned. No help will be provided for these questions in the exercise sessions.

The following files are associated with this assignment:

- `logistic_regression_data.txt` (see Problem 3)
- `linear_regression_data.txt` (see Problems 4 and 5)
- `seir_mcmc_funs.R` (see Problem 6)
- `SEIR_hospital_data.txt` (see Problem 6)
- `seir_mcmc_output.txt` (optional, see Problem 6)

and should be stored in your working directory (but should not be submitted).

Problem 1: "Random number generation and Monte Carlo integration"

a)$^{\mathrm{R}}$ **Open-ended, no help**: Write a function that generates n independent random variables for the distribution with density

$$p(x) = \frac{2^{\frac{1}{4}}\Gamma\left(\frac{3}{4}\right)}{\pi} \exp\left(-\frac{x^4}{2}\right), -\infty < x < \infty$$

(Here Γ is the gamma function, whose numerical values you obtain in R using the function `gamma()`) Verify that your function produces correct results by comparing a histogram of the produced random variables with the probability density.

b)$^{\mathrm{R}}$ **Open-ended, no help**: Write a function that generates n independent random variables for the distribution with density

$$p(x) = 2x \exp(-x^2),\ 0 < x < \infty$$

Additional points will be given if you can do this without using for/while-loops. Verify that your function produces correct results by comparing a histogram of the produced random variables with the probability density.

c)$^{\mathrm{R}}$ **Open-ended, no help**: Consider the integral

$$\int_0^\infty \exp(\sqrt{x}) \exp(-20(x-4)^2))dx$$

Evaluate the integral using importance sampling.

Problem 2: "Smile-shaped target"

Consider the distribution which is characterized by the log-density kernel

$$\log g(\boldsymbol{\theta}) = -\frac{\theta_1^2}{2} - \frac{(\theta_2 - \theta_1^2)^2}{2},\ -\infty < \theta_1, \theta_2 < \infty$$

where $\boldsymbol{\theta} = (\theta_1, \theta_2)$.

The θ_1-marginal of this distribution is $N(0, 1)$, which may be used for checking the correctness of your algorithm.

a)$^{\mathrm{R}}$ **Open-ended, no help**: Use the MCMC algorithm you think would be most suitable for obtaining MCMC samples distributed according to $\pi(\boldsymbol{\theta}) \propto g(\boldsymbol{\theta})$. (Please write the complete algorithm from scratch on your own, do not use the 2D random walk MH code from the lectures/exercise sets).
Argue for why you are using your chosen MCMC algorithm for this particular problem.
Please spend time tuning your algorithm so that it is as efficient as possible.
Run your MCMC algorithm long enough that you get at least an ESS of 1000 for both θ_1 and θ_2 after any removal of burn-in.

Problem 3: "IMH for simple logistic regression problem"

Logistic regressions are widely used in statistics and machine learning for situations where we wish to model binary (0 or 1) outcomes rather that outcomes on the real line (as for in linear regression). Specifically, we model the probability $P(y_i = 1)$ of outcome $y_i,\ i = 1, \ldots, n$ as a function of a covariate[1] x_i. The logistic regression model under consideration here may be written as

$$y_i \sim Bernoulli(p_i), \text{ where } p_i = \frac{\exp(\alpha + \beta x_i)}{1 + \exp(\alpha + \beta x_i)},\ i = 1, \ldots, n,$$

where $\boldsymbol{\theta} = (\alpha, \beta)$ are parameters, $\mathbf{x} = (x_1, \ldots, x_n)$ are fixed covariates. The observations $\mathbf{y} = (y_1, \ldots, y_n)$ are assumed to be independent.

Further we consider a Bayesian analysis, where the priors are given by

$$\alpha \sim N(0, 10^2),\ \beta \sim N(0, 10^2).$$

and the overarching task of this problem is to obtain MCMC samples from the posterior distribution $p(\boldsymbol{\theta}|\mathbf{y})$.

The below code, which may be copied into the R-code you hand in, first loads a data set, and then defines a function `logistic.lp` which returns $\log g(\boldsymbol{\theta})$ where $g(\boldsymbol{\theta}) \propto p(\boldsymbol{\theta}|\mathbf{y})$.

```
# load the data set
df <- data.frame(read.table("logistic_regression_data.txt"))
x <- df$x
y <- df$y

# function returning a log-posterior kernel for theta=(alpha,beta)
logistic.lp <- function(theta){
  alpha <- theta[1]
  beta <- theta[2]
  # log-likelihood
  Eeta <- exp(alpha+beta*x)
  p <- Eeta/(1.0+Eeta)
  log.like <- sum(dbinom(y,size=1,prob = p,log=TRUE))

  # priors
  log.prior <- dnorm(alpha,sd=10,log=TRUE) + dnorm(beta,sd=10,log=TRUE)

  # log-posterior kernel
  return(log.like+log.prior)
}
```

The data set is given in the file `logistic_regression_data.txt`, which is assumed to be in your working directory.

[1] E.g. $y_1 = 1$ indicates patient i cured, whereas $y_i = 0$ indicates patient i still sick. x_i is the dose of some medication patient i was given.

Performing an initial maximum likelihood-based classical analysis[2] we obtain the point estimate

$$\hat{\boldsymbol{\theta}} = \begin{pmatrix} -0.102 \\ 1.993 \end{pmatrix}$$

and an estimate of the covariance matrix of the parameter given by

$$\hat{\Sigma} = \begin{pmatrix} 0.00653 & -0.00058 \\ -0.00058 & 0.01689 \end{pmatrix}$$

a)$^{\mathrm{R}}$ Write an Independent MH sampler with target distribution $p(\boldsymbol{\theta}|\mathbf{y})$, using a $N(\hat{\boldsymbol{\theta}}, \delta\hat{\Sigma})$ proposal distribution. Here $\delta > 0$ is a tuning parameter that should be selected for highest possible performance. Ensure that you have at least 1000 effective samples after removing burn-in.

Now you are tasked with making a predictions of (the random variable) $m(x^*) = \frac{\exp(\alpha+\beta x^*)}{1+\exp(\alpha+\beta x^*)}$ for some new observation y^* with associated covariate x^*. Here (α, β) are distributed according to posterior distribution.

b)$^{\mathrm{R}}$ Make a plot showing the median and the 0.05 and 0.95 quantiles of $m(x^*)$ for values of x^* ranging between -5 and 5.

c)$^{\mathrm{R}}$ **Open-ended, no help** Conditional having observed the data, for which values of the covariate x^* are you at least 99% certain that $m(x^*) > 0.8$

[2]You may do this on your own using the lines given in the Appendix at the end of this document.

Problem 4: "Gibbs sampler for simple linear regression model"
Consider the simple linear regression model

$$y_i \sim N(\alpha + \beta x_i, \tau^{-1}), i = 1, \ldots, n,\ \tau > 0,$$

where $\boldsymbol{\theta} = (\alpha, \beta, \tau)$ are parameters, x_i are fixed covariates, and the observations y_i are independent. We assume the following priors:

$$\alpha \sim N(0, 10^2),\ \beta \sim N(0, 10^2),\ \tau \sim Gamma(1, 1).$$

The joint posterior of $\boldsymbol{\theta} = (\alpha, \beta, \tau)$ of has a log-density kernel

$$\log g(\boldsymbol{\theta}) = \frac{n}{2}\log(\tau) - \frac{\tau}{2}\sum_{i=1}^{n}(y_i - \alpha - \beta x_i)^2 - \frac{\alpha^2 + \beta^2}{200} - \tau$$

i.e. $p(\boldsymbol{\theta}|\mathbf{y}) = \pi(\boldsymbol{\theta}) \propto g(\boldsymbol{\theta})$. The conditional posterior τ, i.e. $\tau|\alpha, \beta, \mathbf{y}$ is a Gamma distribution with shape parameter $n/2 + 1$ and scale parameter

$$\frac{1}{\frac{1}{2}\sum_{i=1}^{n}(y_i - \alpha - \beta x_i)^2 + 1}$$

a) Find the distribution of the conditional posteriors $\alpha|\beta, \tau, \mathbf{y}$ and $\beta|\alpha, \tau, \mathbf{y}$. Show the steps you take for deriving them.

Load the data set we are considering using the commands

```
df <- data.frame(read.table(file="linear_regression_data.txt"))
x <- df$x
y <- df$y
```

Be sure to have the file `linear_regression_data.txt` in your working directory.

b)$^{\mathrm{R}}$ Write a Gibbs sampler (with 3 blocks) targeting the joint posterior $p(\boldsymbol{\theta}|\mathbf{y})$ associated with this data set, using updates according to $\alpha|\beta, \tau, \mathbf{y}$, $\beta|\alpha, \tau, \mathbf{y}$ and $\tau|\alpha, \beta, \mathbf{y}$.

c)$^{\mathrm{R}}$ Run the sampler for at least 10000 iterations. Remove burn-in as needed, and make sure your Gibbs output appears to have converged (trace plots, Geweke test). You should have at least an Effective sample size of 1000 for all of α, β, τ.
Also make sure your posterior means are approximately the same as the estimates you obtain when running a (classical) linear regression.

Hint, in this case you obtain a (classical) linear regression via e.g.

```
lm.out <- lm(y~x, data=df)
```

You now want to improve the performance of the Gibbs sampling procedure by reducing the number of blocks to two. For this purpose you obtain the joint conditional posterior of α and β, i.e. the distribution of $(\alpha, \beta)|\tau, \mathbf{y}$ to be a bivariate normal distribution with covariance matrix $-\mathbf{c}^{-1}$ and mean vector $-\mathbf{c}^{-1}\mathbf{b}$, i.e. $(\alpha, \beta)|\tau, \mathbf{y} \sim N(-\mathbf{c}^{-1}\mathbf{b}, -\mathbf{c}^{-1})$, where

$$\mathbf{c} = \begin{pmatrix} -n\tau - 0.01 & -\tau\sum_{i=1}^{n} x_i \\ -\tau\sum_{i=1}^{n} x_i & -\tau\sum_{i=1}^{n} x_i^2 - 0.01 \end{pmatrix},\ \mathbf{b} = \begin{pmatrix} \tau\sum_{i=1}^{n} y_i \\ \tau\sum_{i=1}^{n} y_i x_i \end{pmatrix}.$$

d)[R] Write a new Gibbs sampler with two blocks, namely updating according to $(\alpha, \beta)|\tau, \mathbf{y}$ and $\tau|\alpha, \beta, \mathbf{y}$.
Check the output of the algorithm applied to the above data set as described in point c).
Does the output of the two-block sampler improve relative to the three-block sampler in written in point b)?

It is generally believed that $\alpha + \beta = 0$ for the situation giving rise to the data we consider here.

e)[R] **Open-ended, no help**: Does your analysis of the data suggest that $\alpha + \beta \neq 0$?

Problem 5: "Bootstrapping"

In this problem, we consider the same linear regression situation as in the previous problem, including the same data set (but now consider a classical approach rather than Bayesian).

a)[R] **Open-ended, no help** Using the bootstrap, check if there following claims hold:
Claim 1: $\sigma = 1/\sqrt{\tau}$ is strictly smaller than 1.0.
Claim 2: α is equal to 0.
Claim 3: $\alpha + \beta$ is equal to 0.
In all cases, back up your conclusion with a graphical representation.

Problem 6: "Bayesian inference for the SEIR model"

Before doing this problem, please refresh your memory of the SEIR model also considered in mandatory assignments 1 and 2.

In this problem, we consider estimating the parameters R (`R0max` in the code) and S (`soc_dist_Rfac` in the code), $\boldsymbol{\theta} = (R, S)$, based on a data set $\{y_t\}_{t=1}^T$ of counts of the number of hospitalized persons (out of a population of 1 million) recorded every day between March 21st 2020 and July 18th 2020 (and hence $T = 120$)[3].

We model the count data using independent Poisson distributions

$$y_t \sim Poisson(\lambda(t, \boldsymbol{\theta})), t = 1, \ldots, T, \tag{1}$$

where $\lambda(t, \boldsymbol{\theta})$ is taken to be the combined (SEIR-model implied) number of persons in the hospital states HH, HC and CC (i.e. $1000000(HH + HC + CC)$) for the days t we have data for. Notice that $\lambda(t, \boldsymbol{\theta})$ depends on $\boldsymbol{\theta}$ as all of HH, HC and CC are functions of $\boldsymbol{\theta}$.

The prior is taken to be the joint distribution for (R, S) which was derived in mandatory assignment 2, and has the joint probability distribution

$$p(\boldsymbol{\theta}) = \begin{cases} \frac{\exp\left(-\frac{(R-\mu)^2}{2\sigma^2}\right)}{\sqrt{2\pi\sigma^2}(1-F(l))} \frac{10R}{3} & \text{if } R \geq l \text{ and } 0.5/R \leq S \leq 0.8/R \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

where $l = 1.8, \mu = 2.2, \sigma = 1.5$.

In the R-script `seir_mcmc_funs.R` includes functions needed for

- evaluating (2)
- loading the data set from file `SEIR_hospital_data.txt`, be sure to have this file in your working directory.
- the function `seir.lp()` which evaluates $\log g(\boldsymbol{\theta})$ where $g(\boldsymbol{\theta}) = L(\boldsymbol{\theta})p(\boldsymbol{\theta}) \propto p(\boldsymbol{\theta}|\mathbf{y})$.

Have a look at the contents of `seir_mcmc_funs.R` and make sure you understand what is going on there.

a)[R] Obtain MCMC samples targeting $p(\boldsymbol{\theta}|\mathbf{y})$. You should have at least an effective sample size of 200 for both R and S. Show the steps you do in order to tune the sampler.

Hint; the bivariate RWMH code `twoDRWMH()` in the solution of exercise set 7 may be useful in point a). A reasonable starting point for the tuning a RWMH would be a diagonal proposal covariance matrix with diagonal elements `3e-4` and `5e-7`.

If you, for some reason, failed to obtain meaningful MCMC samples in point a), the file `seir_mcmc_output.txt` (loaded e.g. via `data.frame(read.table(file="seir_mcmc_output.txt"))`) contains samples of reasonable quality obtained from a long run of RWMH. You may use (a subset) of these MCMC samples in the next questions.

[3] This corresponds to day number $81, 82, \ldots, 200$ in the output of the SEIR model code.

b)$^{\mathrm{R}}$ **Open-ended, no help**: Compare the posterior distribution $p(\boldsymbol{\theta}|\mathbf{y})$ and the prior distribution $p(\boldsymbol{\theta})$. Would you say that observing the hospital data provides us with much information with respect to $\boldsymbol{\theta}$?

c)$^{\mathrm{R}}$ **Open-ended, no help**: Re-do problem 4 f) of mandatory assignment 2, but this time using samples from $p(\boldsymbol{\theta}|\mathbf{y})$ (rather than $p(\boldsymbol{\theta})$ which was done mandatory assignment 2). Compare the plots you obtain with and without conditioning on data - are they different?

d)$^{\mathrm{R}}$ **Open-ended, no help**: The hospital managers wish for budgeting purposes to know the distribution of the total number of "hospital-bed-days" (per 1 million persons) required by Covid-19 patients in each of the months December 2020, January 2021 and February 2021. Please use the additional randomness characterized by (1) in our calculations. Give a graphical representation and provide a summary that could be presented to a non-technical person. Do the calculations both under the prior- (before seeing data) and posterior (after seeing data) distributions on $\boldsymbol{\theta}$.

Here "hospital-bed-days" refers to the sum of all one day stays in a hospital bed due to Covid-19 over the given months.

Appendix

Code for maximum likelihood analysis for the logistic regression model

```
glm.out <- glm(y~x,data=df,family = binomial(link="logit"))
summary(glm.out) # point estimate
vcov(glm.out) # covariance matrix
```