NTNU – Trondheim
Norwegian University of
Science and Technology

Department of Mathematical Sciences

Examination paper for
TMA4300 Computer Intensive Statistical Methods

Academic contact during examination: Andrea Riebler

Phone: 4568 9592

Examination date: May 24th 2014

Examination time (from–to): 09:00–13:00

Permitted examination support material: C:

- Calculator HP30S, CITIZEN SR-270X or CITIZEN SR-270X College, Casio fx-82ES PLUS with empty memory.
- Statistiske tabeller og formler, Tapir.
- K. Rottmann: Matematisk formelsamling.
- One yellow, stamped A5 sheet with own handwritten formulas.

Other information:

- All nine sub-problems in this exam count approximately the same.
- All answers must be justified.
- In your solution you can use English and/or Norwegian.

Language: English

Number of pages: 6

Number pages enclosed: 0

Checked by:

Date Signature

Problem 1

Assume we only have a method to generate random numbers that are uniformly distributed between 0 and 1.

a) Describe how you can generate samples from a normal distribution with mean $\mu \in \mathbb{R}$ and variance $\sigma^2 > 0$. Choose an approach that does not include a rejection step.

b) Based on the approach in part a) describe how to obtain random samples from a mixture density created as the mixture of two normal distributions, i.e.

$$w \cdot \mathcal{N}(\mu_1, \sigma_1^2) + (1 - w) \cdot \mathcal{N}(\mu_2, \sigma_2^2),$$

with $0 < w < 1$ and where w denotes the mixture weight.

Problem 2

Assume we have data on overall mortality aggregated to certain age groups and calendar-time intervals. Let E_{ij} denote the expected number of cases (known) in age group $i = 1, \ldots, I$ and time interval $j = 1, \ldots, J$. We assume that the number of cases y_{ij} in age group i during calendar period j are conditionally independent and follow a Poisson distribution:

$$y_{ij} \mid \eta_{ij} \sim \text{Poisson}(E_{ij} \exp(\eta_{ij}))$$

where

$$\eta_{ij} = \theta_j + z_{ij}.$$

Component $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_J)^\top$ is temporally structured. A common way to introduce a temporally structured effect is to assume that effects adjacent in time are similar. Here, we do this using a prior based on first-order differences:

$$\begin{aligned} p(\boldsymbol{\theta}|\kappa_\theta) &\propto \kappa_\theta^{(J-1)/2} \exp\left(-\frac{\kappa_\theta}{2} \sum_{j=2}^{J} (\theta_j - \theta_{j-1})^2\right) \\ &= \kappa_\theta^{(J-1)/2} \exp\left(-\frac{\kappa_\theta}{2} \boldsymbol{\theta}^\top \mathbf{R} \boldsymbol{\theta}\right). \end{aligned}$$

Here, $\mathbf{R}$ is defined as

$$\mathbf{R} = \begin{pmatrix} 1 & -1 & & & & & \\ -1 & 2 & -1 & & & & \\ & -1 & 2 & -1 & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & -1 & 2 & -1 & \\ & & & & -1 & 2 & -1 \\ & & & & & -1 & 1 \end{pmatrix}$$

and κ_θ is the precision (inverse variance) parameter that determines the degree of smoothing.

Component $\boldsymbol{z} = (z_{11}, \ldots, z_{IJ})^\top$ is unstructured white noise with precision parameter κ_z, i.e. $\mathcal{N}(\mathbf{0}, \kappa_z^{-1} \mathbf{I}_{I \times J})$, where $\mathbf{I}_{I \times J}$ denotes the identity matrix of dimension $I \times J$.

The distribution of η_{ij}, conditional on the component θ_j and κ_z, is now

$$\eta_{ij} \mid \theta_j, \kappa_z \sim \mathcal{N}(\theta_j, \kappa_z^{-1}).$$

The precision terms are assigned gamma prior distributions:

$$\kappa_\theta \sim \text{Gamma}(\alpha_\theta, \beta_\theta),$$
$$\kappa_z \sim \text{Gamma}(\alpha_z, \beta_z).$$

The gamma distribution $\text{Gamma}(a, b)$ has density function:

$$p(x) = \frac{b^a}{\Gamma(a)} x^{a-1} \exp(-bx), \quad \text{with } x > 0 \text{ and } a, b > 0.$$

a) Explain shortly the principles of Gibbs-sampling.

b) Derive the full-conditional distributions for

- $p(\kappa_z \mid \boldsymbol{y}, \kappa_\theta, \boldsymbol{\eta}, \boldsymbol{\theta})$ and $p(\kappa_\theta \mid \boldsymbol{y}, \kappa_z, \boldsymbol{\eta}, \boldsymbol{\theta})$
- $p(\eta_{ij} \mid \boldsymbol{y}, \kappa_z, \kappa_\theta, \boldsymbol{\eta}_{-ij}, \boldsymbol{\theta})$ and $p(\boldsymbol{\theta} \mid \boldsymbol{y}, \kappa_z, \kappa_\theta, \boldsymbol{\eta})$.

Motivate your derivations.

If possible define the parametric distribution and its parameters.

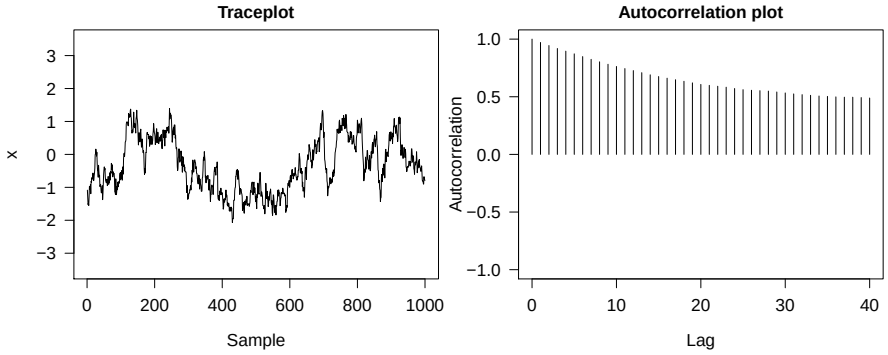
Problem 3

The lines on the next page show a MCMC program to generate samples from a specific univariate continuous target distribution $X|a, b \sim \text{Target}(a, b)$, with $x \in \mathbb{R}$ and parameters $a = 0$ and $b = 1$. The target density on log-scale is defined in the R-function `dtarget(x, a, b, log=TRUE)` (not explicitly given).

a)

- In lines 33–35 we see that only samples after the burn-in period are saved. Define the term "burn-in" period. Why do we not use the samples produced during the burn-in period?
- What type of proposal distribution is used and why is the `logproposal.ratio` on line 17 equal to 0?

b) Below you see the traceplot and autocorrelation plot for samples obtained with `my.mcmc(nburnin=100, numit=1000, sd=0.2)`.



- Do you expect a high or low overall acceptance rate? Please explain.
- What is the role of the parameter `sd`? How would you change the value of `sd` to explore the target distribution more efficiently? Explain your choice.

c) Suppose we have generated 1000 samples for our random variable X:

- Assume you obtain an effective sample size of 23. What does this mean?
- Suppose you are interested in the probability $q = P(X > 0)$. Explain how you could estimate q using the generated samples?

```
my.mcmc <- function(nburnin, numit, sd){

   xsamples <- rep(NA, numit)
   yes <- 0
   no <- 0
   # specify a starting value
   x <- 0.0

   for(k in -(nburnin-1):numit){
     # propose a new value
     proposal <- rnorm(1, mean=x, sd=sd)

     # compute log posterior ratio
     logposterior.ratio <- dtarget(proposal, a=0, b=1, log=TRUE) -
                           dtarget(x, a=0, b=1, log=TRUE)
     # compute log proposal ratio
     logproposal.ratio <- 0

     # derive the acceptance probability (on log scale)
     alpha <- logposterior.ratio + logproposal.ratio
     # accept-reject step
     if(log(runif(1)) <= alpha){
       # accept the proposed value
       x <- proposal
       # increase counter of accepted values
       yes <- yes + 1
     }
     else{
       # stay with the old value
       no <- no + 1
     }

     if(k > 0){
       xsamples[k] <- x
     }
     if(k %% 100 == 0){
       # print every 100 iterations the acceptance rate
       cat("The acceptance rate is:", round(yes/(yes+no)*100,2), "%\n")
     }
   }
   return(xsamples)
 }
```

Problem 4

Assume we have a data set $(y_1, x_1), \ldots, (y_n, x_n)$, where x_i denotes an observation and y_i the class label. We would like to estimate the misclassification rate of a classification method using k-fold cross-validation.

a) Explain how k-fold cross-validation is implemented.

b) What are the advantages and disadvantages of k-fold cross-validation relative to leave-one-out cross-validation when $k < n$. Consider in your argumentation computational aspects and accuracy of the obtained misclassification rate.