TMA4300 Final Exam 2023 Solutions

Problem 1

a)

The following algorithm works, although it's about half as efficient as it should be. Mathematical operations are assumed to be vectorized when possible. We apply the Box-Muller algorithm:

Algorithm 1 `stdGauss(p)`

Set $\boldsymbol{U}_1 \leftarrow$ `runif(p)`

Set $\boldsymbol{U}_2 \leftarrow$ `runif(p)`

Set $\boldsymbol{Z}_0 \leftarrow \sqrt{-2\ln(\boldsymbol{U}_1)}\cos(2\pi U_2)$

Return $\boldsymbol{Z}_0$

b)

Algorithm 2 `mvn(N, mu, L)`

Set $p \leftarrow$ length of μ times N

$\boldsymbol{Z} \leftarrow$ `matrix(stdGauss(p), ncol=N)`

$\boldsymbol{X}_0 \leftarrow \boldsymbol{L}\boldsymbol{Z}$

Add $\boldsymbol{\mu}$ to the columns of $\boldsymbol{X}_0$, and set `xmat` to the result

Return `xmat`

c)

Algorithm 3 `mvt(N, mu, L, nu)`

```
xmat ← mvn(N, mu, L)
uvec ← colSums(matrix(runif(N*nu), ncol=N)^2)
tmat ← xmat
for i in 1:N do
    tmat[,i] = (xmat[,i] - mu)√(ν/uvec[i]) + mu
end for
Return tmat
```

Problem 2

a)

First, we will show that

$$P(\text{accept } \boldsymbol{z}) = \frac{N - p + 1}{\prod_{i=1}^{p} z_i}$$

using the definitions of C and κ as described in the problem statement. Then,

$$\begin{aligned} C &\equiv \sup_{\boldsymbol{z} \in A} \frac{P(\boldsymbol{X} = \boldsymbol{z} | X_1 \geq 1, \ldots, X_p \geq 1)}{P(\boldsymbol{Y} = \boldsymbol{z} - \mathbf{1})} \\ &= \sup_{\boldsymbol{z} \in A} \frac{P(\boldsymbol{X} = \boldsymbol{z}) / P(X_1 \geq 1, \ldots, X_p \geq 1)}{P(\boldsymbol{Y} = \boldsymbol{z} - \mathbf{1})} \qquad \text{(since } z_i \geq 1 \ \forall \ i \text{ if } \boldsymbol{z} \in A) \\ &\equiv \frac{1}{\kappa} \sup_{\boldsymbol{z} \in A} \frac{P(\boldsymbol{X} = \boldsymbol{z})}{P(\boldsymbol{Y} = \boldsymbol{z} - \mathbf{1})}, \end{aligned}$$

where $\kappa = P(X_1 \geq 1, \ldots, X_p \geq 1)$. Now,

$$
\begin{aligned}
C^* &= \sup_{\boldsymbol{z} \in A} \frac{P(\boldsymbol{X} = \boldsymbol{z})}{P(\boldsymbol{Y} = \boldsymbol{z} - \mathbf{1})} \\
&= \sup_{\boldsymbol{z} \in A} \frac{\frac{N!}{z_1! \ldots z_p!} \pi_1^{z_1} \ldots \pi_p^{z_p}}{\frac{(N-p)!}{(z_1-1)! \ldots (z_p-1)!} \pi_1^{z_1-1} \ldots \pi_p^{z_p-1}} \\
&= \left(\frac{N!}{(N-p)!} \prod_{i=1}^{p} \pi_i \right) \sup_{\boldsymbol{z} \in A} \prod_{i=1}^{p} \frac{1}{z_i} \\
&= \left(\frac{N!}{(N-p)!} \prod_{i=1}^{p} \pi_i \right) \frac{1}{N-p+1} \qquad \text{(maximized if all but 1 } z_i \text{ is 1).}
\end{aligned}
$$

The acceptance probability is then:

$$
\begin{aligned}
P(\text{accept } \boldsymbol{z}) &= \frac{1}{C} \frac{P(\boldsymbol{X} = \boldsymbol{z} | X_1 \geq 1, \ldots, X_p \geq 1)}{P(\boldsymbol{Y} = \boldsymbol{z} - \mathbf{1})} \\
&= \frac{1}{C^*} \frac{P(\boldsymbol{X} = \boldsymbol{z})}{P(\boldsymbol{Y} = \boldsymbol{z} - \mathbf{1})} \\
&= \left(\frac{(N-p)!}{n!} \frac{N-p}{\prod_{i=1}^{p} \pi_i} \right) \frac{\frac{N!}{z_1! \ldots z_p!} \pi_1^{z_1} \ldots \pi_p^{z_p}}{\frac{(N-p)!}{(z_1-1)! \ldots (z_p-1)!} \pi_1^{z_1-1} \ldots \pi_p^{z_p-1}} \\
&= \frac{N-p+1}{\prod_{i=1}^{p} z_i},
\end{aligned}
$$

giving the desired result.

b)

The expected proportion of proposals that are accepted is given by

$$
\begin{aligned}
\frac{1}{C} &= \frac{\kappa}{C^*} \\
&= \kappa \left(\frac{(N-p)!}{N!} \prod_{i=1}^{p} \frac{1}{\pi_i} \right) (N-p+1) \\
&= P(\boldsymbol{X} \in A) \frac{(N-p)!}{N!} p^p (N-p+1).
\end{aligned}
$$

Since there is no closed form for $P(\boldsymbol{X} \in A)$ this is as far as we can simplify the above expression. However, it is worth noting that if $N = p$ exactly, then the acceptance probability must be 1. On the other hand, as $N \to \infty$ for any fixed p the acceptance probability converges to 0. Hence, this sampler likely performs better (i.e. has higher acceptance probabilities) when N is close to p.

Problem 3

a)

Let $\bar{\boldsymbol{Y}}_i = \frac{1}{n}\sum_{j=1}^{n} Y_{ij}$. The posterior is then:

$$\begin{aligned}
p(\boldsymbol{\lambda}, \alpha, \beta \mid \boldsymbol{Y}_1, \ldots, \boldsymbol{Y}_m) &\propto p(\boldsymbol{Y}_1, \ldots, \boldsymbol{Y}_m \mid \boldsymbol{\lambda}, \alpha, \beta) p(\boldsymbol{\lambda} \mid \alpha, \beta) p(\alpha) p(\beta) \\
&= \left(\prod_{i=1}^{m}\prod_{j=1}^{n} \frac{\lambda_i^{Y_{ij}} e^{-\lambda_i}}{Y_{ij}!}\right)\left(\prod_{i=1}^{m} \frac{\beta^\alpha}{\Gamma(\alpha)} \lambda_i^{\alpha-1} e^{-\beta\lambda_i}\right)\left(\frac{B^A}{\Gamma(A)}\alpha^{A-1}e^{-B\alpha}\right)\left(\frac{D^C}{\Gamma(C)}\beta^{C-1}e^{-D\beta}\right) \\
&\propto \left(\prod_{i=1}^{m} \lambda_i^{n\bar{\boldsymbol{Y}}_i} e^{-n\lambda_i}\right)\left(\prod_{i=1}^{m} \frac{\beta^\alpha}{\Gamma(\alpha)} \lambda_i^{\alpha-1} e^{-\beta\lambda_i}\right)\left(\alpha^{A-1}e^{-B\alpha}\right)\left(\beta^{C-1}e^{-D\beta}\right).
\end{aligned}$$

The full conditional for λ_i is then,

$$\begin{aligned}
p(\lambda_i \mid \alpha, \beta, \boldsymbol{Y}_1, \ldots, \boldsymbol{Y}_m) &\propto \left(\lambda_i^{n\bar{\boldsymbol{Y}}_i} e^{-n\lambda_i}\right)\left(\lambda_i^{\alpha-1} e^{-\beta\lambda_i}\right) \\
&= \lambda_i^{\alpha + n\bar{\boldsymbol{Y}}_i - 1} e^{-(\beta+n)\lambda_i},
\end{aligned}$$

which is the kernel/core of a Gamma$(\alpha + n\bar{\boldsymbol{Y}}_i, \beta + n)$. The full conditional of λ_i is therefore a Gamma$(\alpha + n\bar{\boldsymbol{Y}}_i, \beta + n)$ distribution.

b)

The acceptance probability is:

$$P\left(\text{accept}\begin{pmatrix}\alpha' \\ \beta'\end{pmatrix}\right) = \min\left\{1, \frac{p(\boldsymbol{\lambda}, \alpha', \beta' \mid \boldsymbol{Y}_1, \ldots, \boldsymbol{Y}_m)}{p(\boldsymbol{\lambda}, \alpha, \beta \mid \boldsymbol{Y}_1, \ldots, \boldsymbol{Y}_m)}\right\}.$$

Now,

$$\frac{p(\boldsymbol{\lambda}, \alpha', \beta' \mid \boldsymbol{Y}_1, \ldots, \boldsymbol{Y}_m)}{p(\boldsymbol{\lambda}, \alpha, \beta \mid \boldsymbol{Y}_1, \ldots, \boldsymbol{Y}_m)} = \frac{\left(\prod_{i=1}^m \frac{(\beta')^{\alpha'}}{\Gamma(\alpha')} \lambda_i^{\alpha'-1} e^{-\beta'\lambda_i}\right) \left((\alpha')^{A-1} e^{-B\alpha'}\right) \left((\beta')^{C-1} e^{-D\beta'}\right)}{\left(\prod_{i=1}^m \frac{\beta^{\alpha}}{\Gamma(\alpha)} \lambda_i^{\alpha-1} e^{-\beta\lambda_i}\right) \left(\alpha^{A-1} e^{-B\alpha}\right) \left(\beta^{C-1} e^{-D\beta}\right)}$$

$$= \left(\prod_{i=1}^m \frac{\Gamma(\alpha)(\beta')^{\alpha'}}{\Gamma(\alpha')\beta^{\alpha}} \lambda_i^{\alpha'-\alpha-1} e^{-(\beta'-\beta)\lambda_i}\right)$$

$$\times \left(\left(\frac{\alpha'}{\alpha}\right)^{A-1} e^{-B(\alpha'-\alpha)}\right) \left(\left(\frac{\beta'}{\beta}\right)^{C-1} e^{-D(\beta'-\beta)}\right)$$

$$= \left(\frac{\Gamma(\alpha)^m (\beta')^{m\alpha'}}{\Gamma(\alpha')^m \beta^{m\alpha}} \prod_{i=1}^m \lambda_i^{\alpha'-\alpha-1} e^{-(\beta'-\beta)\lambda_i}\right)$$

$$\times \left(\left(\frac{\alpha'}{\alpha}\right)^{A-1} e^{-B(\alpha'-\alpha)}\right) \left(\left(\frac{\beta'}{\beta}\right)^{C-1} e^{-D(\beta'-\beta)}\right)$$

$$= \left(\frac{\Gamma(\alpha)^m (\beta')^{m\alpha'+C-1}}{\Gamma(\alpha')^m \beta^{m\alpha+C-1}} \left(\prod_{i=1}^m \lambda_i\right)^{\alpha'-\alpha-1} e^{-(\beta'-\beta)\sum_{i=1}^m \lambda_i}\right)$$

$$\times \left(\left(\frac{\alpha'}{\alpha}\right)^{A-1} e^{-B(\alpha'-\alpha)-D(\beta'-\beta)}\right)$$

$$= \frac{\Gamma(\alpha)^m (\beta')^{m\alpha'+C-1}}{\Gamma(\alpha')^m \beta^{m\alpha+C-1}} \left(\prod_{i=1}^m \lambda_i\right)^{\alpha'-\alpha-1} \left(\frac{\alpha'}{\alpha}\right)^{A-1}$$

$$\times \exp\left\{-(\beta'-\beta)\left(D + \sum_{i=1}^m \lambda_i\right) - B(\alpha'-\alpha)\right\}$$

so the acceptance probability is the maximum of the above value and 1.

To avoid numerical roundoff issues and to reduce computation time it is important to simplify acceptance probabilities as much as possible and to calculate them on a log scale. In addition, the gamma function and factorials (including binomial coefficients) as well as some other functions should be calculated directly on a log scale rather than taking the log of the function itself to avoid numerical problems.

c)

No, it is not possible in general to know if the sampler converged. Although diagnostics exist to show that the sampler did not converge (or didn't draw enough samples), such as trace plots, Geweke tests, and effective sample size calculations, they can only be used to show that a sampler has not yet converged. However, in the case that, for example, a distribution is bimodal with modes sufficiently separated, a sampler can appear as if it converged according to those tests when it has only stayed near one mode.

You could select σ^2 by, for example, using traceplots, effective sample size, or average acceptance probability. Traceplots for each parameter should look like a homogeneous band, and poor tuning parameter choices will affect this. The tuning parameter could be chosen to maximize the effective sample size of α and/or β. The tuning parameter could also be chosen so that the average acceptance probability (for the Metropolis steps specifically) is within the optimal range for Metropolis steps, typically between 20% and 50%.

Problem 4

a)

1. Yes, this can be fit in INLA. All latent effects are Gaussian conditional on the hyperparameters, and the responses are iid conditional on the latent effects and hyper parameters.

2. No, this cannot be fit in INLA, since ϵ_i is a latent effect that is non-Gaussian.

3. Yes, this can be fit in INLA. The latent effects are all Gaussian conditional on the hyperparameters, and the responses are iid conditional on the latent effects and hyperparameters. Further, $\boldsymbol{\gamma}$ has sparse precision matrix $\boldsymbol{Q}$, ensuring that computation is feasible.

4. Yes, this can be fit in INLA. The latent effects are joint Gaussian and

random walks are GMRFs, ensuring the precision matrix of $\boldsymbol{\gamma}$ conditional on the hyperparamater τ_γ is sparse and computation is feasible. Further, the responses are iid conditional on the latent effects and hyperparameters.

Problem 5

a)

The full likelihood is:

$$f(\boldsymbol{z}, \boldsymbol{u} \mid p) = \prod_{i:u_i=1} e^{-z_i} \prod_{i:u_i=0} 3e^{-3z_i} \prod_{i=1}^{n} p^{u_i}(1-p)^{1-u_i},$$

so the full log likelihood is:

$$\begin{aligned}\ell(\boldsymbol{z}, \boldsymbol{u} \mid p) &= \sum_{i:u_i=1} (-z_i) + \sum_{i:u_i=0} (\log(3) - 3z_i) + \sum_{i=1}^{n} u_i \log(p) + (1-u_i)\log(1-p) \\ &= \sum_{i=1}^{n} u_i(-z_i) + \sum_{i=1}^{n}(1-u_i)(\log(3) - 3z_i) + \sum_{i=1}^{n} u_i \log(p) + (1-u_i)\log(1-p).\end{aligned}$$

Given the definition of $\delta_{p^{(t)}}(z_i)$,

$$\begin{aligned}E[\ell(p) \mid \boldsymbol{z}, p^{(t)}] &= \sum_{i=1}^{n} E[u_i \mid \boldsymbol{z}, p^{(t)}](-z_i) + \sum_{i=1}^{n}(1 - E[u_i \mid \boldsymbol{z}, p^{(t)}])(\log(3) - 3z_i) \\ &\quad + \sum_{i=1}^{n} E[u_i \mid \boldsymbol{z}, p^{(t)}] \log(p) + (1 - E[u_i \mid \boldsymbol{z}, p^{(t)}])\log(1-p) \\ &= \sum_{i=1}^{n} \big(\delta_{p^{(t)}}(z_i) \cdot (-z_i) + (1 - \delta_{p^{(t)}}(z_i)) \cdot (\log(3) - 3z_i)\big) \\ &\quad + \log(p) \sum_{i=1}^{n} \delta_{p^{(t)}}(z_i) + \log(1-p) \sum_{i=1}^{n}(1 - \delta_{p^{(t)}}(z_i)).\end{aligned}$$

b)

We begin by differentiating the above expression with respect to p and setting the result to 0:

$$
\begin{aligned}
\frac{\partial}{\partial p} E[\ell(p) \mid \boldsymbol{z}, p^{(t)}] = 0 = \frac{1}{p} \sum_{i=1}^{n} \delta_{p^{(t)}}(z_i) - \frac{1}{1-p} \sum_{i=1}^{n} (1 - \delta_{p^{(t)}}(z_i)) \\
p \sum_{i=1}^{n} (1 - \delta_{p^{(t)}}(z_i)) = (1-p) \sum_{i=1}^{n} \delta_{p^{(t)}}(z_i) \\
\sum_{i=1}^{n} (1 - \delta_{p^{(t)}}(z_i)) = np \\
p = \frac{1}{n} \sum_{i=1}^{n} \delta_{p^{(t)}}(z_i).
\end{aligned}
$$

Hence, we set $p^{(t+1)} = \frac{1}{n} \sum_{i=1}^{n} \delta_{p^{(t)}}(z_i)$.

To estimate the standard error of $\hat{p}$ you could use bootstrapping. This would involve sampling, with replacement, bootstrapped pseudodatasets $(z_1^*, \ldots, z_n^*)$ of size n (since the u_i's are not observed), and each time estimating p. Say you do this 1000 times, for example. Then this would yields 1000 different samples from the bootstrapped distribution of $\hat{p}$, and you could estimate the SE from these samples in the standard way.