Solution sketch to problems: STK 4051 / STK 9051 exam spring 2023

Problem 1

a) Integral is approximated by:

$$\int_{R^n} h(\boldsymbol{x})g(\boldsymbol{x})d\boldsymbol{x} \approx \overline{h_a} = \frac{1}{B}\sum_{i=1}^{B} h(\boldsymbol{x}_i)$$

If the samples are independent, the error in approximation measured in terms of the variance is: $\mathrm{Var}\{\overline{h_a}\} = \frac{\mathrm{Var}_g\{h(X)\}}{B}$. Where the subscript indicate that the variance is over the distribution $g(\boldsymbol{x})$. Thus if $\mathrm{Var}\{h(X)\} < \infty$, and the samples are independent then the limit is zero when $B \to \infty$.

b) Integral can be approximated by:

$$\int_{R^n} h(\boldsymbol{x})f(\boldsymbol{x})d\boldsymbol{x} \approx \overline{h_b} = \frac{1}{B}\sum_{i=1}^{B} \frac{f(\boldsymbol{x}_i)}{g(\boldsymbol{x}_i)} h(\boldsymbol{x}_i)$$

If the samples are independent, then we can utilize the result from above with the function $\frac{f(\boldsymbol{x})}{g(\boldsymbol{x})}h(\boldsymbol{x})$, thus the error in approximation measured in terms of the variance is:

$$\mathrm{Var}\{\overline{h_b}\} = \frac{\mathrm{Var}_g\left\{\frac{f(\boldsymbol{X})}{g(\boldsymbol{X})}h(\boldsymbol{X})\right\}}{B},$$

where the subscript indicate that the variance is over the distribution $g(\boldsymbol{x})$. For the convergence: If $\mathrm{Var}_g\left\{\frac{f(\boldsymbol{X})}{g(\boldsymbol{X})}h(\boldsymbol{X})\right\} < \infty$, and the samples are independent then the limit is zero when $B \to \infty$, which gives convergence. Importantly however for the solution to converge to the right number, we also need that the support of $g(x)$ contains the support of $h(x)f(x)$. Alternatively we could require $E_f\left\{\frac{f(\boldsymbol{X})}{g(\boldsymbol{X})}h(\boldsymbol{X})^2\right\} < \infty$.
Utilizing $q(x)$ rather than $f(x)$ we get:

$$\overline{h_{b_2}} = \frac{\sum_{i=1}^{B} \frac{q(\boldsymbol{x}_i)}{g(\boldsymbol{x}_i)} h(\boldsymbol{x}_i)}{\sum_{i=1}^{B} \frac{q(\boldsymbol{x}_i)}{g(\boldsymbol{x}_i)}}$$

These methods are denoted by the common name importance sampling. Using un-normalized weights and normalized weights respectively.

c) In step 2 A1, we sample form the standard normal distribution using the inversion formula. $P(X < x) = P(\Phi^{-1}(u) < x) = P\big(u < \Phi(x)\big) = \Phi(x)$.
The last equality is a consequence of the normal distribution, the second to last is due to the monotonicity of the transform.
In step 2 A2, we sample form an exponential distribution shifted such that the minimum value is x_0 rather than 0.

In formula (3) we use Monte Carlo simulation, such as (1) to compute the integral.

In formula (4) we use importance sampling. Note that we do not need to add the indicator since all samples are larger than x_0.

d) Since the jitter is much stronger for algorithm 1 than for algorithm 2. This indicate that the sample variance is larger for this computation. Theoretically, it is much better to have all samples centered at the region where the function is non-zero, since we the investigate the part which contribute to the integral much better. Also note that the decay of exponential distribution is slower than the normal distribution. The latter is important for the variance of the ratio. $\mathrm{Var}_g\left\{\frac{f(\boldsymbol{X})}{g(\boldsymbol{X})}h(\boldsymbol{X})\right\}$. The jitter is caused by Monte Carlo variability, i.e. the fact that we do not have an infinite number of samples. In order to reduce the variability, we can increase the number of samples. It is also possible to trade some variance in the estimate for bias, by using common random numbers $(u_i\,, i = 1 \ldots, B)$ for all values of x_0.

Problem 2

a) The likelihood is:

$$L(p,\lambda\,) = \prod_{i=1}^{n}(p \cdot \lambda \cdot \exp -\lambda x + (1-p) \cdot \lambda^2 x \cdot \exp -\lambda x)$$

In In Newton algorithm, we update the estimate with

$$x_{i+1} = x_i - \left(H_f\right)^{-1} \cdot \ \nabla f$$

Where $f(x)$ is the function, we want to optimize, ∇f is the gradient of this function, and H_f is the Hessian. In the quasi-Newton algorithm we use an approximation to replace the inverse Hessian. This expression is faster to compute. Note that even if the quasi-newton algorithm converges the approximation for the Hessian need not converge to the Hessian.

In the table we clearly see that there are two modes. One of the modes proposes a probability which is larger than zero. To avoid this issue, we could either use constrained optimization or transform the probability into an unconstrained parameter, e.g. $\theta = \log\left(\frac{1}{p} - 1\right)$.

b) One expression for the complete log likelihood is:

$$l(p,\lambda|\boldsymbol{x},\boldsymbol{C}) = \sum_{i=1}^{n} I(C_i = 1)[\log p + \log\lambda - \lambda x_i] + \sum_{i=1}^{n} I(C_i = 2)[\log(1-p) + 2\log\lambda + \log\,(x_i) - \lambda x_i]$$

The Q(|) function is the expected value of the complete log likelihood given the observed data and the current estimate of the parameters. In the expression for the

complete log likelihood above we see that the only random ness is in the indicator functions, which are in the expression in a sum. The expectation can be applied to each term in the sum, when we use that $E(I(C_i = 1)|\boldsymbol{x}, p, \lambda) = P(C_i = 1|x_i, p, \lambda)$, we get the desired result.

$$P(C_i = 1|x_i, p, \lambda) = \frac{P(C_i = 1, x_i|p, \lambda)}{P(x_i|p, \lambda)}$$

$$= \frac{P(C_i = 1|p, \lambda)p(x_i|C_i = 1, p, \lambda)}{P(C_i = 1|p, \lambda)p(x_i|C_i = 1, p, \lambda) + P(C_i = 2|p, \lambda)p(x_i|C_i = 2, p, \lambda)}$$

$$= \frac{p \cdot \text{Exp}(x_i; \lambda)}{p \cdot \text{Exp}(x_i; \lambda) + (1 - p) \cdot \text{Erlang}(x_i; 2, \lambda)}$$

Where we have used definition of conditional probability and inserted the expressions.

c) Differentiate Q($p, \lambda|p^{(t)}, \lambda^{(t)}$), wrt p:

$$\frac{\partial \text{Q}}{\partial p} = \sum_{i=1}^{n} P\left(C_i = 1\middle|x_i, p^{(t)}, \lambda^{(t)}\right)\left[\frac{1}{p}\right] - \sum_{i=1}^{n}\left(1 - P\left(C_i = 1\middle|x_i, p^{(t)}, \lambda^{(t)}\right)\right)\left[\frac{1}{1 - \text{p}}\right] = 0$$

$$\Rightarrow \; p = \frac{1}{n}\sum_{i=1}^{n} P\left(C_i = 1\middle|x_i, p^{(t)}, \lambda^{(t)}\right)$$

Differentiate Q($p, \lambda|p^{(t)}, \lambda^{(t)}$), wrt λ:

$$\frac{\partial \text{Q}}{\partial \lambda} = \sum_{i=1}^{n} P\left(C_i = 1\middle|x_i, p^{(t)}, \lambda^{(t)}\right)\left[\frac{1}{\lambda} - x_i\right] + \sum_{i=1}^{n}\left(P\left(C_i = 2\middle|x_i, p^{(t)}, \lambda^{(t)}\right)\right)\left[\frac{2}{\lambda} - x_i\right] = 0$$

$$\Rightarrow \; \lambda = \frac{\sum_{i=1}^{n} P\left(C_i = 1\middle|x_i, p^{(t)}, \lambda^{(t)}\right) + 2 \cdot P\left(C_i = 2\middle|x_i, p^{(t)}, \lambda^{(t)}\right)}{\sum_{i=1}^{n} x_i}$$

In the EM algorithm each iteration will improve the likelihood from 2a.

d) In a non-parametric bootstrap, we resample the dataset with replacement to create a dataset of equal size as the initial. Then we estimate the parameters using the prescribed algorithm. One repeat of this procedure creates one point in the scatterplot in figure 2. We repeat this procedure B times. And uses the set of estimates to represent the sampling uncertainty of the estimator. In the figure we see that the parameter estimates are highly correlated (negative correlation). The lower the value of p the larger the estimate of λ. This is natural since reducing p means that we overlook more samples, which would increase the number

of events in the same time interval. The samples seems to have two modes. One close to $p = 1$ and one centred around $p = 0.75$. There is no obvious reason for this.

Problem 3

a) The conditional probability is computed to proportionality by:

$$p(\boldsymbol{\lambda}, \beta | \alpha, \gamma, \delta, \boldsymbol{x}, \boldsymbol{t}) \propto p(\boldsymbol{\lambda}, \beta, \boldsymbol{x} | \alpha, \gamma, \delta, \boldsymbol{t})$$

$$= p(\beta | \gamma, \delta, \alpha, \boldsymbol{t}) p(\boldsymbol{\lambda} | \beta, \gamma, \delta, \alpha, \boldsymbol{t}) p(\boldsymbol{x} | \boldsymbol{\lambda}, \beta, \gamma, \delta, \alpha, \boldsymbol{t})$$

$$= p(\beta | \gamma, \delta) p(\boldsymbol{\lambda} | \beta, \alpha) p(\boldsymbol{x} | \boldsymbol{\lambda}, \boldsymbol{t})$$

$$= p(\beta | \gamma, \delta) \prod_{i=1}^{10} p(\lambda_i | \beta, \alpha) \prod_{i=1}^{10} p(x_i | \lambda_i, t_i)$$

$$= \frac{\delta^\gamma \beta^{\gamma-1}}{\Gamma(\gamma)} \exp(-\delta\beta) \prod_{i=1}^{10} \frac{\beta^\alpha \lambda_i^{\alpha-1}}{\Gamma(\alpha)} \exp(-\lambda_i \beta) \prod_{i=1}^{10} \frac{(\lambda_i t_i)^{x_i}}{x_i!} \exp(-\lambda_i t_i)$$

Where transition 1 is due to posterior is proportional to the joint distribution (conditioned to given parameters $\alpha, \gamma, \delta, \boldsymbol{t}$). Second transition is due to common factorization. In transition 3, we impose the dependencies of parameters given in the text. In transition 4, we utilize the prior independence of λ_i, and x_i. In the final step we impose the distributions.

Computation of distributions for the Gibbs-sampler:

$$p(\lambda_i | \alpha, \beta, \gamma, \delta\, \boldsymbol{t}, \boldsymbol{x}, \boldsymbol{\lambda_{-i}}) \propto \lambda_i^{\alpha-1} \exp(-\lambda_i \beta)\, \lambda_i^{x_i} \exp(-\lambda_i t_i)$$

$$\propto \lambda_i^{\alpha+x_i-1} \exp\big(-\lambda_i(\beta + t_i)\big)$$

$$\propto \text{Gamma}(\lambda_i; \alpha + x_i, \beta + t_i)$$

$$p(\beta | \alpha, \gamma, \delta, \boldsymbol{\lambda}, \boldsymbol{x}, \boldsymbol{t}) \propto \beta^{\gamma-1} \exp(-\delta\beta) \prod_{i=1}^{10} \beta^\alpha \exp(-\lambda_i \beta)$$

$$\propto \beta^{\gamma+10\alpha-1} \exp\left(-\left(\delta + \sum_{i=1}^{10} \lambda_i\right)\beta\right)$$

$$\propto \text{Gamma}\left(\beta; \gamma + 10\alpha, \delta + \sum_{i=1}^{10} \lambda_i\right)$$

b) In R_H, the $f(x)$, and $f(x^*)$ are the target densities, evaluated in the current sample, x, and the proposed sample, x^*. The function $g(x^*|x)$, is the density for proposing a change to the proposed sample, $x^*$from the current sample. The function $g(x^*|x)$ is the density for the reverse proposal.

M-H pseudo code:

Initialize $x = x_0$

For i in 1: B

Sample $x^* \sim g(x^*|x_{i-1})$, u_i~Uniform[0,1]

Compute R_{MH}

if $u_i < R_{MH}$: set $x_i = x^*$ otherwise set $x_i = x_{i-1}$

The transition kernel in Metropolis Hastings ($x \neq x^*$):

$$P(x, x^*) = g(x^*|x) \max\left\{1, \frac{f(x^*)g(x|x^*)}{f(x)g(x^*|x)}\right\}$$

c) Starting with the integral:

$$\int f(x)P(x, x^*)dx = \int f(x^*)P(x^*, x)dx = f(x^*) \underbrace{\int P(x^*, x)dx}_{1} = f(x^*)$$

First transition is just utilizing the detailed balance, next we set what is constant in the integration outside the integral and recognize that the integral of the transition kernel is 1.

For convergence the Markov chain need to be:

- Irreducible – possible to move to any value in a finite number of steps
- Aperiodic – not go into predictable cycles
- Recurrent – always return to sets that has a positive probability