Appendices

A Adaptive Proposal Distribution

I use an adaptive proposal distribution $q_\theta(\theta)$ to sample the parameter space θ. The adaptive metropolis hasting algorithm provides systematic method for modifying the shape of the proposal distribution based on the accepted steps of the current markov chain, allowing for more efficient mixing of chains. That is the $q_\theta(\theta^{(i)}) = N(\theta^{(i)}, \Sigma^{(i)}(\theta^{(i)}))$ follows a Gaussian distribution. To provide a reasonable estimate for the covariance matrix $\Sigma^{(i)}$, the Markov chain runs for an initial number of steps (T_{init}) from a truncated multivariate normal proposal distribution with a covariance matrix, I_s, whose entries are calculated using the upper and lower bounds of the support of the priors $[s_0^k, s_1^k] \in \mathcal{S}$, through $i_{k,k} = (s_1^k - s_0^k)/\zeta$ and $i_{i,j} = 0$ otherwise, where ζ is a scaling factor.

Problematically, the proposal distribution using the updated covariance matrix, $\Sigma^{(i)}$, is no longer memoryless, and therefore chain may no longer converge to the correct stationary distribution. To overcome this problem, the proposal distribution must also sample from a non-adaptive multivariate Gaussian distribution modified to ensure that changes to the covariance matrix diminish over time. Further, to improve chain mixing and to optimise convergence rates, I include adaptive scaling factors, $\lambda^{(i)}$ and $M^{(i)}$ for the initial non-adaptive and adaptive proposals, respectively, whose magnitude diminishes with the number of steps in the chain. The adaptive scaling factor for the non-adaptive proposal distributions stops once the model starts sampling from the adaptive proposal distributions. Overall, the combined non-adaptive and adaptive proposal distributions for the adaptive Metropolis-Hastings is given by

$$
\begin{array}{c|c|ll}
i & i \le T_{init} & i > T_{init} & \\
\hline
q(\cdot|\theta^{(i)}) & \mathcal{N}(\theta^{(i)}, \exp(\lambda^{(i)})I_s; \mathcal{S}) & \mathcal{N}(\theta^{(i)}, \Sigma^{(i)}; \mathcal{S}) & \text{with probability } \beta, \\
& & \mathcal{N}(\theta^{(i)}, \exp(\lambda^{(T_{init})})I_s; \mathcal{S}) & \text{with probability } 1 - \beta
\end{array}
\tag{1}
$$

where $\Sigma^{(i)} = \exp(M^{(i)})\Gamma^{(i)}$ and $M^{(i)}$, $\lambda^{(i)}$ and $\Gamma^{(i)}$ are updated iteratively through the stochastic approximation algorithm:

$$
\begin{aligned}
\lambda^{(i+1)} &= \lambda^{(i)} + \gamma_1(i)(a(\theta^{(i)}, \theta') - 0.234) \\
M^{(i+1)} &= M^{(i)} + \gamma_2(i)(a(\theta^{(i)}, \theta') - 0.234) \\
\mu^{(i+1)} &= \mu^{(i)} + \gamma_3(i)(\mu^{(i)} - \theta^{(i)}) \\
\Gamma^{(i+1)} &= \Gamma^{(i)} + \gamma_4(i)[(\theta^{(i)} - \mu^{(i+1)})(\theta^{(i)} - \mu^{(i+1)})^T - \Gamma^{(i)}]
\end{aligned}
$$

where $\gamma_x(i)$ are gain factors. Note when $i > T_{init}$ up stop updaing $\lambda^{(i)}$.

In our implementation, we define $\psi_{adapt}^{(i)} = \{M^{(i)}, \mu^{(i)}, \Gamma^{(i)}, \lambda^{(i)}\}$, and choose values, $\beta = 0.05$, $\zeta = 100$, $\lambda^{(0)} = \log(0.1^2/|\theta_i|)$, $M^{(0)} = \log(2.382^2/|\theta^{(i)}|)$, $\mu^{(0)} = \pi_0$, $\Gamma^{(0)} = I_s$, and $\gamma_x(i) = (1+i)^{-0.5}$ for all x.