Appendices to "A Repelling-Attracting Metropolis Algorithm for Multimodality"

Hyungsuk Tak
Statistical and Applied Mathematical Sciences Institute

Xiao-Li Meng
Department of Statistics, Harvard University

David A. van Dyk
Statistics Section, Department of Mathematics, Imperial College London

A The average number of density evaluations in Section 3.1

Kou et al. (2006) implement the EE (equi-energy) sampler by running five parallel chains under five different temperature levels. The chain under the highest temperature adopts only MH transitions, and the other four chains use an EE jump with probability 0.1 and an MH transition otherwise at each iteration. The EE sampler begins by running a chain under the highest temperature for 75,000 iterations; the first 25,000 are burn-in iterations and the next 50,000 iterations form an energy ring at the highest temperature. The first chain uses only MH transitions. After running the first chain for 75,000 iterations, the sampler initiates the next chain under the second highest temperature and runs it for 75,000 iterations; the first 25,000 are burn-in iterations and the next 50,000 iterations form an energy ring at the second highest temperature. From the second chain, the sampler adopts an EE jump with probability 0.1 and an MH transition otherwise at each iteration. This process continues until the EE sampler finishes running the fifth chain under the unit temperature for 75,000 iterations; the first 25,000 iterations are discarded. All chains keep running until the end of the fifth chain, which means that the first chain runs for $5\times$75,000 iterations in total and the second one runs for $4\times$75,000 iterations, etc. Each EE jump needs to evaluate the target density twice (with caching). Thus, the (expected) total number of the density evaluations is $16\times$75,000 and that per iteration is 16.0.

Similarly, Kou et al. (2006) implement parallel tempering with five temperature levels and propose four swaps with probability 0.1 at the end of each iteration. Five chains under five different temperature levels are run simultaneously for 75,000 iterations, using MH transitions. At the end of each iteration, four swaps occur with probability 0.1 and

no swaps otherwise. Each swap requires two additional evaluations of the target (with caching). Thus, the (expected) total number of the target density evaluations is 435,000 and the average number of the density evaluations per iteration is 5.8 (=435,000/75,000).

B Implementation details in Section 3.3

Tempered transitions require several tuning parameters, e.g., the number of rungs of the temperature ladder and the temperature and jumping scale of each rung, and setting these parameters is known to be challenging in practice (Behrens et al., 2012). At each iteration, the tempered transitions ascend the temperature ladder to explore a flatter surface where the modes are melted down, and then descend the ladder, accepting the last candidate with a modified acceptance probability to maintain the stationary distribution (Neal, 1996). To sample π_1 in (12) at iteration i, for example, suppose $\pi_{1j}(x_1) \propto \{\pi_1(x_1 \mid x_2^{(i-1)}, x_3^{(i-1)}, x_4^{(i-1)}, y, w)\}^{1/T_j}$, where T_j is the temperature at rung j for $j = 1, \ldots, J$. The target density is $\pi_{10}(x_1)$ and the ladder has J rungs with $T_0 = 1 < T_1 < \cdots < T_J$. Within each iteration i, starting from $j = 1$ to J, we generate $\hat{x}_{1j}$ from $\mathrm{N}_2(\hat{x}_{1,j-1}, \Sigma_j)$, where $\hat{x}_{10} = x_1^{(i-1)}$, and accept it with probability $\min\{1, \pi_{1j}(\hat{x}_{1j})/\pi_{1j}(\hat{x}_{1,j-1})\}$ and set $\hat{x}_{1j} = \hat{x}_{1,j-1}$ otherwise. Once we reach $j = J$, we reverse the process from $j = J$ to 1 and generate $\check{x}_{1,j-1}$ from $\mathrm{N}_2(\check{x}_{1j}, \Sigma_j)$ where $\check{x}_{1J} = \hat{x}_{1J}$, and accept it with probability $\min\{1, \pi_{1,j-1}(\check{x}_{1,j-1})/\pi_{1,j-1}(\check{x}_{1j})\}$ and set $\check{x}_{1,j-1} = \check{x}_{1j}$ otherwise until we reach the bottom of the temperature ladder, collecting $\check{x}_{1,J-1}, \ldots, \check{x}_{10}$. After generating the last proposal $\check{x}_{10}$, we set $x_1^{(i)} = \check{x}_{10}$ with an MH acceptance probability of

$$\min\left\{1, \ \frac{\pi_{11}(x_1^{(i-1)})}{\pi_{10}(x_1^{(i-1)})} \times \cdots \times \frac{\pi_{1J}(\hat{x}_{1,J-1})}{\pi_{1,J-1}(\hat{x}_{1,J-1})} \frac{\pi_{1,J-1}(\check{x}_{1,J-1})}{\pi_{1J}(\check{x}_{1,J-1})} \times \cdots \times \frac{\pi_{10}(\check{x}_{10})}{\pi_{11}(\check{x}_{10})}\right\},$$

and set $x_1^{(i)} = x_1^{(i-1)}$ otherwise.

To fit the simulated data, we set three rungs with temperature equal to 2^j for the jth rung. Because the longest observed distance between two sensors is about 0.9, we set the jumping covariance matrix $\Sigma_j = (0.9 \times 1.2^{j-1})^2 \times I_2$ for each Metropolis update of the tempered transitions for the jth rung. For Metropolis and RAM, we set $\Sigma = 1.08^2 \times I_2$. This is the same as the jumping covariance matrix of tempered transitions at the middle rung, i.e., Σ_2. An initial value for each unknown location for each Markov chain is randomly selected from the unit square, $[0, 1] \times [0, 1]$.

C The average number of π_{11} evaluations in Section 3.4

Since the kernels are used only to sample π_{11} in *Step* 1 of Table 5, the average number of π_{11} evaluations at each iteration ($\mathrm{N}_{\pi_{11}}$) is not proportional to the entire CPU time needed for sampling the full posterior π in (18). For reference, $\mathrm{N}_{\pi_{11}}^{\mathrm{M}} = 2$ (with either the Gaussian or mixture proposal), $\mathrm{N}_{\pi_{11}}^{\mathrm{RAM}} = 8.76$ ($\mathrm{N}_d = 1$, $\mathrm{N}_u = 4.48$, $\mathrm{N}_z = 1.28$), and $\mathrm{N}_{\pi_{11}}^{\mathrm{TT}} = 11$. Each Metropolis transition evaluates π_{11} twice, once for the current state and once for the

proposal at each iteration. Caching the density value of the current state does not help reduce $N_{\pi_{11}}$ because the density of the current state changes according to the updates of the other parameters in *Steps* 2 and 3 of Table 5.

References

Behrens, G., Friel, N., and Hurn, M. (2012). Tuning Tempered Transitions. *Statistics and Computing*, 22(1):65–78.

Kou, S. C., Zhou, Q., and Wong, W. H. (2006). Discussion Paper: Equi-Energy Sampler with Applications in Statistical Inference and Statistical Mechanics. *The Annals of Statistics*, 34(4):1581–1619.

Neal, R. M. (1996). Sampling From Multimodal Distributions Using Tempered Transitions. *Statistics and Computing*, 6(4):353–366.