Penn State STAT 540
Homework #1, due Thursday, October 4, 2018

What you have to submit in a Canvas submission folder: (i) Your `R` code in a file titled PSUemailidHW1.R (e.g. muh10HW1.R), (ii) pdf file that contains a clear writeup for the questions below named PSUemailidHW1.pdf (e.g. muh10HW1.pdf). Note that your code should be readily usable, without any modifications.

1. Matrix inversion: Consider matrices of the form $\Sigma = \sigma I + KK'$ where K is an $N \times M$ matrix of iid Gamma($\alpha = 2, \beta = 2$) random variates and $\sigma = 0.2$. Fix $M = 50$. Compute the inverse of the matrix using two different algorithms: (i) directly (by using the `solve` function in R) and (ii) using the Sherman-Morrison-Woodbury identity discussed in class. I have provided example code for simulating the matrix for the case where K is based on iid N(0,1) random variables here `http://personal.psu.edu/muh10/540/hwdir/hw01.R`

   (a) Plot the CPU time versus N for algorithm 1 and algorithm 2 (you will have to determine the grid and range of N values that are feasible).

   (b) A plot of floating point operations (flops) versus N for algorithm 1 and algorithm 2.

   (c) Briefly summarize what you observe based on a comparison of the computational costs for the two algorithms using flops and using CPU time. Do you see any differences? If so, what is your explanation for the differences?

   (d) How does the computational cost scale with M? Run the code for matrix inversions and plot the increase in cost for $N = 1,000$ for $M = 50, 100, \ldots, 1,000$ and overlay the cost for $N = 5,000$ for $M = 50, 100, \ldots, 1,000$. (You should have a total of 4 curves.)

2. Let $O(p)$ be the set of $p \times p$ orthogonal matrices. Consider H, a random matrix that is uniformly distributed on $O(p)$, h_{ij}, the (i,j)th entry of H. Let X be any $p \times p$ matrix. Define

$$\begin{aligned} M_2(X) :=& \mathrm{tr}(X^2), \\ M_{11}(X) :=& (1/2)[(\mathrm{tr}(X))^2 - \mathrm{tr}(X^2)] \\ C(X) :=& M_2(X) - (2/3)M_{11}(X), \end{aligned} \tag{1}$$

   where $\mathrm{tr}(X)$ is the trace of the matrix X. $A = \mathrm{diag}(a_1, , a_p)$, a diagonal matrix with $a_1 > \ > a_p > 0$, $B = \mathrm{diag}(b_1, , b_p)$, a diagonal matrix with $b_1 > \ > b_p > 0$. Suppose that $1 \le s < t \le p$, and $1 \le j \le p$. Define $f_{s,t,j}(H) = \sum_{i=1}^{j}(h_{is}^2 - h_{it}^2)$. Approximate $E[C(HAHB)f_{s,t,j}(H)]$ using Monte Carlo. Do this for $A = \mathrm{diag}(a_1, \ldots, a_p)$ where $a_i = 1/i, i = 1, \ldots, p$, and $B = \mathrm{diag}(b_1, \ldots, b_p)$ where $b_i = 1/i, i = 1, \ldots, p$. Note: To simulate a matrix uniformly distributed on $O(p)$: (i) Generate a $p \times p$ matrix Z

whose entries are i.i.d. N(0,1) random variables, and (ii) form the matrix $H = (ZZ')^{-1/2}Z$.

(a) Report the Monte Carlo approximation, along with its Monte Carlo standard error, for each of $p = 10, 100, 1000, 10000$. What is the largest p for which you are able to do this approximation within 2 hours of wall time on your computer? If this p is different from the four values listed above, report your results for it as well. (Note that wall time is the actual time elapsed as measured by a clock.) Report the Monte Carlo approximation for this p, along with its Monte Carlo standard error.

(b) What is the computational complexity of this Monte Carlo algorithm? You may assume that the cost of generating a N(0,1) random variate is 1 flop.

(c) Based on the Monte Carlo approximations above, would you say that the true expectation is positive? (This question is based on a conjecture from Sheena (2005), via Don Richards, that this expectation is always positive.)

3. Define the univariate Poisson kernel density function (Yang, 2004; or see "wrapped Cauchy" in Levy (1939) and Wintner (1947)) as follows:

$$f(\theta; \mu, \rho) = \frac{1}{2\pi} \frac{1 - \rho^2}{1 - 2\rho\cos(\theta - \mu) + \rho^2}, \quad \mu - \pi \le \theta \le \mu + \pi \qquad (2)$$

Approximate the expectation $E_f(\theta^2)$ for $\mu = 3, \rho = 0.7$ using rejection sampling. State the proposal distribution you used and report associated Monte Carlo standard errors.

4. Consider approximating $\mu = E_\pi g(X)$ for some real-valued function g and distribution π. Construct a ratio important sampling approximation, $\tilde{\mu}_n$, with different importance functions for the numerator and denominator, q_1, q_2 respectively. Derive a formula for the Monte Carlo standard error approximation of $\tilde{\mu}$ using the samples $X^{(1)}, \ldots, X^{(n)} \overset{iid}{\sim} q_1(\cdot)$ and $Y^{(1)}, \ldots, Y^{(m)} \overset{iid}{\sim} q_2(\cdot)$. You may adapt the derivation sketch provided in the lecture notes to answer this question, but you must show your work.

5. Consider the conditionally independent random variates $Y_1, \ldots, Y_n|\alpha \sim$ Poisson$(\exp(\alpha))$, and $\alpha \sim t_{10}(0, 100)$, a t-density with $\nu = 10$ degrees of freedom and mean μ=0, $\sigma^2 = 100$ (so variance=$\nu\sigma^2/(\nu - 2)$) The observed data are 5,4,7,4,4,3,2,5,10,6,6,8,6,6,4,5,8,11,4,3. For both questions below, provide Monte Carlo standard errors and other details about your importance sampling algorithm.

(a) Approximate $E(\alpha|Y)$ using importance sampling.

(b) Approximate $E(\alpha|Y)$ using importance sampling, but now assume $\alpha \sim t_3(0, 100)$. Do not change the importance function you used before. Is this a good approximation? If it is not, change your importance function, and explain the argument for the change.

(c) Approximate $P(\alpha < 0.5|Y)$ using importance sampling.