An efficient algorithm for sampling from $\sin^k(x)$ for generating random correlation matrices

Enes Makalic\textsuperscript{a} and Daniel F. Schmidt\textsuperscript{b}

\textsuperscript{a}Melbourne School of Population and Global Health, University of Melbourne, Carlton, VIC, Australia; \textsuperscript{b}Faculty of Information Technology, Monash University, Clayton, VIC, Australia

ABSTRACT
In this note, we develop a novel algorithm for generating random numbers from a distribution with a probability density function proportional to $\sin^k(x)$, $x \in (0, \pi)$ and $k \geq 1$. Our algorithm is highly efficient and is based on rejection sampling where the envelope distribution is an appropriately chosen beta distribution. An example application illustrating how the new algorithm can be used to generate random correlation matrices is discussed.

ARTICLE HISTORY
Received 31 January 2019
Accepted 28 November 2019

KEYWORDS
Correlation matrix; Inverse transform sampling; Rejection sampling

MATHEMATICS SUBJECT CLASSIFICATION
62D05; 62H20; 62-04

1. Introduction
The generation of random high dimensional correlation matrices is an important problem with particular application to simulation studies of supervised and unsupervised learning procedures. The recent algorithm proposed by Pourahmadi and Wang (2015) provides a method for the random generation of correlation matrices based on the hyperspherical parametrization. A key step of this algorithm involves the sampling of a potentially large number of random variates from a specific, nonstandard probability density; the suggestion in the original paper to use the inverse cumulative distribution function technique to sample from this density can be very slow when the dimensionality of the correlation matrix is moderate or large. In this note we develop a simple rejection sampler for the density in question which is substantially faster than the inverse cumulative distribution function approach.

Consider a positive random variable $X$ with probability density function

$$f_X(x) = c_k \sin^k(x), \quad x \in (0, \pi), \quad k \geq 1,$$

where the normalization constant $c_k$ is given by

$$c_k = \frac{\Gamma\left(\frac{k}{2} + 1\right)}{\sqrt{\pi} \ \Gamma\left(\frac{k}{2} + 1\right)}.$$ 

This is a symmetric continuous probability density function for which the mean (mode and median) are
\[ E_X[X] = \pi/2, \quad f_X(\pi/2) = c_k, \]
for all \( k \geq 1 \). Generating random numbers from this distribution is not straightforward as the corresponding cumulative density function, although available in closed form, is defined recursively and requires \( O(k) \) operations to evaluate. The nature of the cumulative density function makes any procedure based on inverse transform sampling computationally inefficient, especially for large \( k \). Furthermore, an alternative approach based on numerical integration is not recommended due to numerical issues with the probability density function when \( k \) is large.

In this note, we develop a novel algorithm (see Section 2) for generating random numbers from the distribution defined in (1). Our algorithm is based on rejection sampling (Robert and Casella 2004) (pp. 50–53) where the envelope distribution is a particular beta distribution. We compute the exact unconditional acceptance probability of the proposed algorithm and show that the maximum expected average number of iterations per sample is \( \pi/(2\sqrt{2}) \) (\( \approx 1.11 \)) for all \( x \in (0, \pi) \) and \( k \geq 1 \), which highlights the efficiency of the proposed algorithm. Lastly, we use our novel rejection sampler to implement the algorithm for generating high dimensional correlation matrices based on the hyperspherical parametrization (see Section 3) proposed by Pourahmadi and Wang (Pourahmadi and Wang 2015). Our implementation of the correlation matrix sampling algorithm for the Matlab and R numerical computing platforms is available for download from Matlab Central (File ID: 68810) and CRAN (the randcorr package).

2. Rejection sampling algorithm

The proposed algorithm for sampling from density (1) utilizes rejection sampling where the envelope distribution is an appropriately scaled symmetric Beta\((k + 1, k + 1)\) distribution with probability density function

\[
g_Y(x) = \frac{x^k(\pi - x)^k}{B(k + 1, k + 1)\pi^{2k+1}}, \quad x \in (0, \pi), \quad k \geq 1, \tag{2} \]

where \( B(\cdot, \cdot) \) is the Euler beta function. Theorem 2.1 proves that \( g_Y(y) \) is an envelope density for this problem and gives the expected average sampling efficiency for all \( k \geq 1 \). The proof of the theorem is given in Appendix A.

**Theorem 2.1.** Let \( f_X(x) \) denote the probability density function given by (1) and let \( Y \) denote a Beta\((k + 1, k + 1)\) random variable with probability density function \( g_Y(x) \) (2). Then,

\[
f_X(x) \leq M_k \, g_Y(x), \quad x \in (0, \pi), \quad k \geq 1, \tag{3} \]

where \( M_k \) is the upper-bound of the likelihood ratio \( f_X(x)/g_Y(x) \) and is given by

\[
M_k = \frac{f_X(\pi/2)}{g_Y(\pi/2)} = \frac{\sqrt{\pi}2^{k-1}\Gamma\left(\frac{k}{2} + 1\right)^2}{\Gamma\left(k + \frac{3}{2}\right)}. \tag{4} \]

Based on Theorem 2.1, our algorithm to generate a random variable \( X \) with probability density function (1) is given below:
Generate $X \sim \text{Beta}(k + 1, k + 1)$

Generate $U$ from the uniform distribution $U \sim U_{[0,1]}$

Accept $X$ if

$$(1/k) \log U \leq \log \left( \frac{\pi^2 \sin X}{4X(\pi - X)} \right)$$

Otherwise, return to Step 1.

The average number of iterations required to obtain a sample from the distribution (1) is given by $M_k$ (4). This is a strictly increasing function of $k \geq 1$ where the first few values are:

$$M_1 = \frac{\pi}{3}, \quad M_2 = \frac{16}{15}, \quad M_3 = \frac{12\pi}{35}, \quad M_4 = \frac{1024}{945}.$$  

The maximum of $M_k$ is given by

$$\lim_{k \to \infty} M_k = \frac{\pi}{2\sqrt{2}} \approx 1.11$$

which emphasizes the efficiency of the proposed algorithm. In particular, for any value of $k \geq 1$, the maximum expected average number of iterations per sample is $\pi/(2\sqrt{2}) \approx 1.11$.

### 3. Application

The proposed rejection sampling algorithm is suitable for applications requiring repeated sampling from density (1), especially for large $k$. An example application is the problem of random sampling of a $p \times p$ correlation matrix $R$, which may be high dimensional. For this problem, we implement the procedure outlined in (Pourahmadi and Wang 2015) which consists of two steps: (i) generate $p(p-1)/2$ angles $0 < \theta_{ij} < \pi$ from the distribution (1) where $k = \{1, ..., p-1\}$; and (ii) compute a special lower triangular matrix $B$ (Equation (1) in Pourahmadi and Wang (2015)) from the angles $\theta_{ij}$; the corresponding $p \times p$ correlation matrix is then given by $R = BB^T$ (see Pourahmadi and Wang (2015) for details).

The procedure of Pourahmadi and Wang requires a computationally efficient method for sampling of the angles $\theta_{ij}$. This is especially true for high dimensional correlation matrices where $p$ (and therefore $k$) is large and the number of random variates required is of the order $O(p^2)$. Using our rejection sampling algorithm, we have implemented the Pourahmadi and Wang sampler in the Matlab and R numerical computing environments. Our program can generate a random $100 \times 100$ correlation matrix (i.e., 4,950 samples of $\theta_{ij}$) in $\approx 0.02$ s, and a random $1,000 \times 1,000$ (i.e., 499,500 samples of $\theta_{ij}$) correlation matrix in $\approx 0.5$ s on a commodity laptop. In contrast, the inverse transform sampling approach implemented in (Pourahmadi and Wang 2015) and running on comparable computing hardware, requires approximately 0.9 s and 42 s to generate a $100 \times 100$ and a $600 \times 600$ correlation matrix, respectively.

Additionally, we compared the computational performance of our algorithm to the method proposed in Numpacharoen and Atsawarungruangkit (2012) (NA), which is also based on the hyperspherical parametrization of the Cholesky factor, and the
Matlab built-in function randcorr (RC) that is based on the algorithm in (Bendel and Mickey 1978) (see Figure 1). When the dimensionality of the random correlation matrix is low ($p < 20$), all three algorithms are highly efficient with the method proposed by Numpacharoen and Atsawarungruangkit (2012) being faster than the other two algorithms. As the dimensionality of the correlation matrix is increased, our proposed algorithm is clearly the most efficient and is an order of magnitude faster than the other methods. As noted by Pourahmadi and Wang (2015), the NA algorithm can become numerically unstable for large $p$ and may require modification of the default value for the threshold parameter that is used by the method. In contrast, our proposed method is numerically stable even for large $p$ and is free of any user-settable hyperparameters.

Appendix A. Proof of Theorem 2.1

Proof. We wish to show that $f_X(x) \leq M_k g_Y(x)$, for all $x \in (0, \pi)$ and $k \geq 1$, with $f_X(x)$ given by (1), $g_Y(x)$ given by (2) and $M_k$ given by (4). It suffices to prove that

$$f_X(x) \leq M_k g_Y(x),$$

$$\frac{\Gamma\left(\frac{k}{2} + 1\right) \sin^k(x)}{\sqrt{\pi} \Gamma\left(\frac{k}{2} + \frac{1}{2}\right)} \leq \left(\frac{\sqrt{\pi} 2^{k-1} \Gamma\left(\frac{k}{2} + 1\right)^2}{\Gamma\left(k + \frac{3}{2}\right)}\right) \frac{x^k (\pi - x)^k}{B(k + 1, k + 1) \pi^{2k+1}},$$

$$\sin^k(x) \leq 4^k \pi^{-2k} ((\pi - x)x)^k,$$

$$\sin(x) \leq 4\pi^{-2} (\pi - x)x.$$

To show the latter, we note that the function $p_3(x) = 4\pi^{-2} (\pi - x)x$ is an interpolation polynomial at the nodes $x_0 = 0, x_1 = x_2 = \pi/2$ and $x_3 = \pi$ for the function $\sin(x)$ for all $x \in (0, \pi)$ ((user:42969) MR 2016). The interpolation error for the polynomial $p_3(x)$ and some $\xi \in (0, \pi)$ is
\[
\sin(x) - p_3(x) = \frac{\sin(\tilde{\zeta})}{24} \prod_{j=0}^{3} (x - x_j) = \frac{x}{24} \left( x - \frac{\pi}{2} \right)^2 (x - \pi) \sin(\tilde{\zeta}) \leq 0,
\]

as the only term that is negative is \((x - \pi) \leq 0\). The inequality is strict everywhere except the boundary points \((x = 0, \pi)\) and the maximum of \(f_X(x)\) and \(g_Y(x)\) \((x = \pi/2)\). □

**ORCID**

Enes Makalic [http://orcid.org/0000-0003-3017-0871](http://orcid.org/0000-0003-3017-0871)

**References**

Bendel, R. B., and M. R. Mickey. 1978. Population correlation matrices for sampling experiments. *Communications in Statistics - Simulation and Computation* 7 (2):163–82. doi:10.1080/03610917808812068.

Numpacharoen, K., and A. Atsawarungruangkit. 2012. Generating correlation matrices based on the boundaries of their coefficients. *PLoS One* 7 (11):e48902. doi:10.1371/journal.pone.0048902.

Pourahmadi, M., and X. Wang. 2015. Distribution of random correlation matrices: Hyperspherical parameterization of the Cholesky factor. *Statistics & Probability Letters* 106: 5–12. doi:10.1016/j.spl.2015.06.015.

Robert, C. P., and G. Casella. 2004. *Monte Carlo statistical methods*. Berlin, Heidelberg: Springer-Verlag.

(user:42969) MR. 2016. How to prove \(\sin(x) \leq 4x(\pi - x)/\pi^2\) for all \(x \in (0, \pi)\)? [Mathematics]. [https://math.stackexchange.com/q/2030069](https://math.stackexchange.com/q/2030069).