

Article

A Short Note on Generating a Random Sample from Finite Mixture Distributions

Luai Al-Labadi * and Anna Ly 

Department of Mathematical & Computational Sciences, University of Toronto Mississauga,
Mississauga, ON L5L 1C6, Canada; annahuynh.ly@mail.utoronto.ca

* Correspondence: luai.allabadi@utoronto.ca

Abstract: Computational statistics is a critical skill for professionals in fields such as data science, statistics, and related disciplines. One essential aspect of computational statistics is the ability to simulate random variables from specified probability distributions. Commonly employed techniques for sampling random variables include the inverse transform method, acceptance–rejection method, and Box–Muller transformation, all of which rely on sampling from the uniform $(0, 1)$ distribution. A significant concept in statistics is the finite mixture model, characterized by a convex combination of multiple probability density functions. In this paper, we introduce a modified version of the composition method, a standard approach for sampling finite mixture models. Our modification offers the advantage of relying on sampling from the uniform $(0, 1)$ distribution, aligning with prevalent methods in computational statistics. This alignment simplifies teaching computational statistics courses, as well as having other benefits. We offer several examples to illustrate the approach.

Keywords: composition method; computational statistics; finite mixture distribution; simulation

MSC: 62-08; 62-04



Citation: Al-Labadi, L.; Ly, A. A Short Note on Generating a Random Sample from Finite Mixture Distributions. *Axioms* **2024**, *13*, 307. <https://doi.org/10.3390/axioms13050307>

Academic Editors: Hans J. Haubold and Stelios Zimeras

Received: 6 March 2024

Revised: 18 April 2024

Accepted: 7 May 2024

Published: 8 May 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Computational statistics has gained significant importance in recent years due to the exponential growth of data and the increasing complexity of data-driven problems. Within computational statistics, the ability to simulate or generate random samples from a probability distribution is fundamental. These generated random samples are utilized for estimating probabilities, expectations, and testing hypotheses. The inverse transform method and the acceptance–rejection method are two of the most fundamental techniques for generating random samples, and these can be found in well-known computational statistics textbooks such as *Statistical Computing with R* by [1]. These methods rely on generating numbers from the uniform $(0, 1)$ distribution. The choice of method depends on the specific distribution being generated and the desired properties of the generated sample, such as efficiency or accuracy.

In certain cases, the data may not conform to commonly known distributions such as the normal or exponential distributions. Instead, they can be represented as a finite mixture model, which combines multiple probability density functions in a convex manner. These models find applications in various scientific domains. For instance, normal mixture distributions are used as parametric density estimators [2], whereas finite mixture models are employed in medical studies [3] and financial analyses [4]. Finite mixture models have also been used by [5] in the analysis of wind speeds, and Ref. [6] have demonstrated their usefulness in Bayesian density estimation. Furthermore, Ref. [7] provide a comprehensive overview of the different applications of mixture models.

Sampling from finite mixture models is a standard topic covered in many computational statistics textbooks, including works by [1,8], among others. In these texts, the primary approach for sampling from finite mixture models is typically the *composition*

method. However, although the composition method is effective, it does not directly use the uniform distribution.

The goal of this paper is to modify the standard composition algorithm by incorporating sampling from the uniform $(0, 1)$ distribution to ensure consistency with primary sampling algorithms such as the inverse transform method and the acceptance–rejection method. This aspect could prove beneficial in teaching computational statistics courses, as sampling from the uniform $(0, 1)$ distribution becomes a standard step in various sampling algorithms.

The remainder of this paper is organized as follows. Section 2 provides a relevant background on finite mixture models and discusses the proposed modification. Section 3 presents several examples demonstrating the effectiveness of the proposed method. Finally, Section 4 offers concluding remarks.

2. Finite Mixture Models and Simulation Theorem

In this section, we define a finite mixture model and introduce a theorem for sampling this model via an adaptation of the composition method. The proof of this theorem is also included.

A finite mixture model is a statistical model that represents a probability distribution as a mixture of several component distributions. Mathematically, given k component distributions $f_1(x), \dots, f_k(x)$, each with associated *mixing probabilities* (also known as *mixing weights*) π_1, \dots, π_k , a *finite mixture model* $f(x)$ is defined as:

$$f(x) = \sum_{i=1}^k \pi_i f_i(x), \tag{1}$$

where $0 \leq \pi_i \leq 1$ and $\sum_{i=1}^k \pi_i = 1$. Further insights into Equation (1) can be found in studies by [9,10].

In the literature, simulating a variable from a finite k -mixture distribution is typically carried out by the composition method [1,11]:

1. Generate an integer $I \in \{1, \dots, k\}$ such that

$$P(I = i) = \pi_i, \quad \text{for } i = 1, \dots, k;$$

2. Deliver X with cumulative distribution function F_I .

The following theorem introduces an algorithm for generating a sample from (1). This theorem presents a modified version of the composition method, utilizing the uniform distribution. Aligning with well-established algorithms such as the inverse transform and acceptance–rejection method enhances accessibility for learners.

Theorem 1. Consider $F(x)$ as defined in (1). The following algorithm generates a random variate from X with the cumulative distribution function $F(x)$:

1. Generate a random u from the uniform $(0, 1)$ distribution;
2. If $\sum_{i=1}^{l-1} \pi_i \leq u < \sum_{i=1}^l \pi_i$, generate a random x from $F_l(x)$, where $l = 1, \dots, k$, with the convention that $\sum_{i=1}^0 \pi_i = 0$.

Proof. We show that the generated sample has the same distribution as X . By the law of total probability, we have

$$\begin{aligned}
 P(X \leq x) &= \int_0^1 P(X \leq x|U = u)du \\
 &= \int_0^{\pi_1} P(X \leq x|U = u)du + \int_{\pi_1}^{\pi_1+\pi_2} P(X \leq x|U = u)du + \\
 &\quad \dots + \int_{\sum_{i=1}^{l-1} \pi_i}^{\sum_{i=1}^l \pi_i} P(X \leq x|U = u)du + \\
 &\quad \dots + \int_{\sum_{i=1}^{k-1} \pi_i}^1 P(X \leq x|U = u)du \\
 &= \int_0^{\pi_1} F_1(x)du + \int_{\pi_1}^{\pi_1+\pi_2} F_2(x)du + \\
 &\quad \dots + \int_{\sum_{i=1}^{l-1} \pi_i}^{\sum_{i=1}^l \pi_i} F_l(x)du + \dots + \int_{\sum_{i=1}^{k-1} \pi_i}^1 F_k(x)du \\
 &= \pi_1 F_1(x) + \pi_2 F_2(x) + \dots + \pi_l F_l(x) + \dots + \pi_k F_k(x) \\
 &= \sum_{i=1}^k \pi_i F_i(x) = F(x).
 \end{aligned}$$

□

The proof of Theorem 1 reveals that the approach is overly general, encompassing not only mixtures of continuous distributions but also extending to other scenarios. This includes mixtures involving continuous and discrete distributions, as well as mixtures comprising only discrete distributions. Additionally, the framework can be extended to sample mixtures of multivariate distributions. In the following section, we explore specific examples that illustrate these various cases.

3. Examples

In this section, we demonstrate the proposed algorithm outlined in Theorem 1 with six illustrative examples. The R code is provided in the Supplementary Materials.

Example 1. *Mixture of three normal distributions [10].*

Suppose $X_1 \sim N(\mu = 0, \sigma^2 = 1)$, $X_2 \sim N(\mu = 5, \sigma^2 = 0.25)$, and $X_3 \sim N(\mu = 2, \sigma^2 = 9)$ are independent. Let

$$F(x) = 0.3F_1(x) + 0.5F_2(x) + 0.2F_3(x).$$

Using Theorem 1, we generated a sample of size 10^6 from $F(x)$. Figure 1 shows the histogram of the generated sample with the true density superimposed. It is evident from Figure 1 that the proposed method performs exceptionally well in this example.

Example 2. *Mixture of five gamma distributions: different shapes with same scale parameters [1].*

Consider $F(x) = \sum_{i=1}^5 \pi_i F_i(x)$, where $X_i \sim \text{gamma}(r = 3, \beta_i = i)$ are independent and the mixing probabilities are $\pi_i = i/15$, $i = 1, \dots, 5$. Using Theorem 1, we generated a sample of size 10^6 from $F(x)$. Figure 2 displays the histogram plot of the generated sample with the true density superimposed. The proposed procedure also performs well in this example.

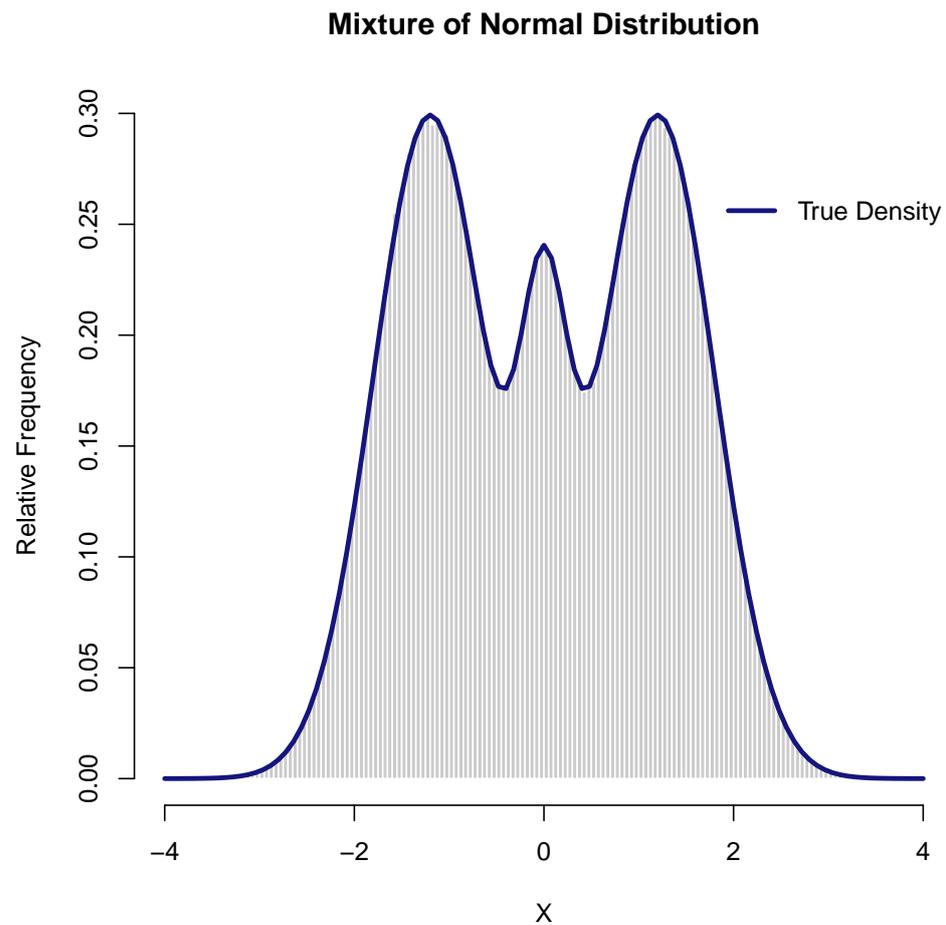


Figure 1. Mixture of three normal distributions in Example 1.

Example 3. Mixture of five gamma distributions: different scale with same shape parameters.

Let $F(x)$ be as described in Example 2, with $X_i \sim \text{gamma}(r_i = i, \beta_i = 3)$. Employing Theorem 1, we generated a sample of size 10^6 from $F(x)$. Figure 3 presents the histogram plot of the generated sample with the true density superimposed. The proposed procedure demonstrates effective performance in this example as well.

Example 4. Comparing empirical and true mixed distributions.

In this example, we compare $F_n(x)$, the empirical cumulative distribution function (ECDF) of the simulated data, with the true mixed distribution

$$F(x) = \sum_{i=1}^3 \pi_i F_i(x),$$

where F_i represents three cases:

- **Case 1:** $X_1 \sim t(5)$, $X_2 \sim t(10)$, and $X_3 \sim t(15)$. Here, $t(v)$ represents the t distribution with v degrees of freedom;
- **Case 2:** $X_1 \sim \text{beta}(2, 5)$, $X_2 \sim \text{beta}(2, 10)$, and $X_3 \sim \text{beta}(2, 15)$;
- **Case 3:** $X_1 \sim \text{Pareto}(1, 1)$, $X_2 \sim \text{Pareto}(2, 2.5)$, and $X_3 \sim \text{Pareto}(3, 3)$. Here, $\text{Pareto}(x_m, \alpha)$ is the Pareto distribution with x_m as the minimum possible value (scale parameter) and α as the shape parameter.

In all three cases, we let $\pi_1 = 9/20$, $\pi_2 = 9/20$, and $\pi_3 = 1/10$. As a measure of proximity, we utilize the Cramér–von Mises distance defined as

$$D = \int (F_n(x) - F(x))^2 dF(x).$$

We examine various sample sizes $n \in \{20, 50, 100, 1000\}$. For each generated sample X_1, \dots, X_n , we estimate D using

$$\hat{D} = \frac{1}{n} \sum_{i=1}^n (F_n(X_i) - F(X_i))^2.$$

For each sample, we compute 10^4 values of \hat{D} and report $\bar{\hat{D}}$ and $\text{sd}(\hat{D})$, representing the mean and standard deviation of the 10^4 values of \hat{D} . Additionally, for comparison, we include results obtained using samples generated from the composition method described in Section 2. The results are reported in Table 1. It is clear that both simulation algorithms work well as both $\bar{\hat{D}}$ and $\text{sd}(\hat{D})$ approach zero, especially as we increase the sample size.

Mixture of Gamma Distribution

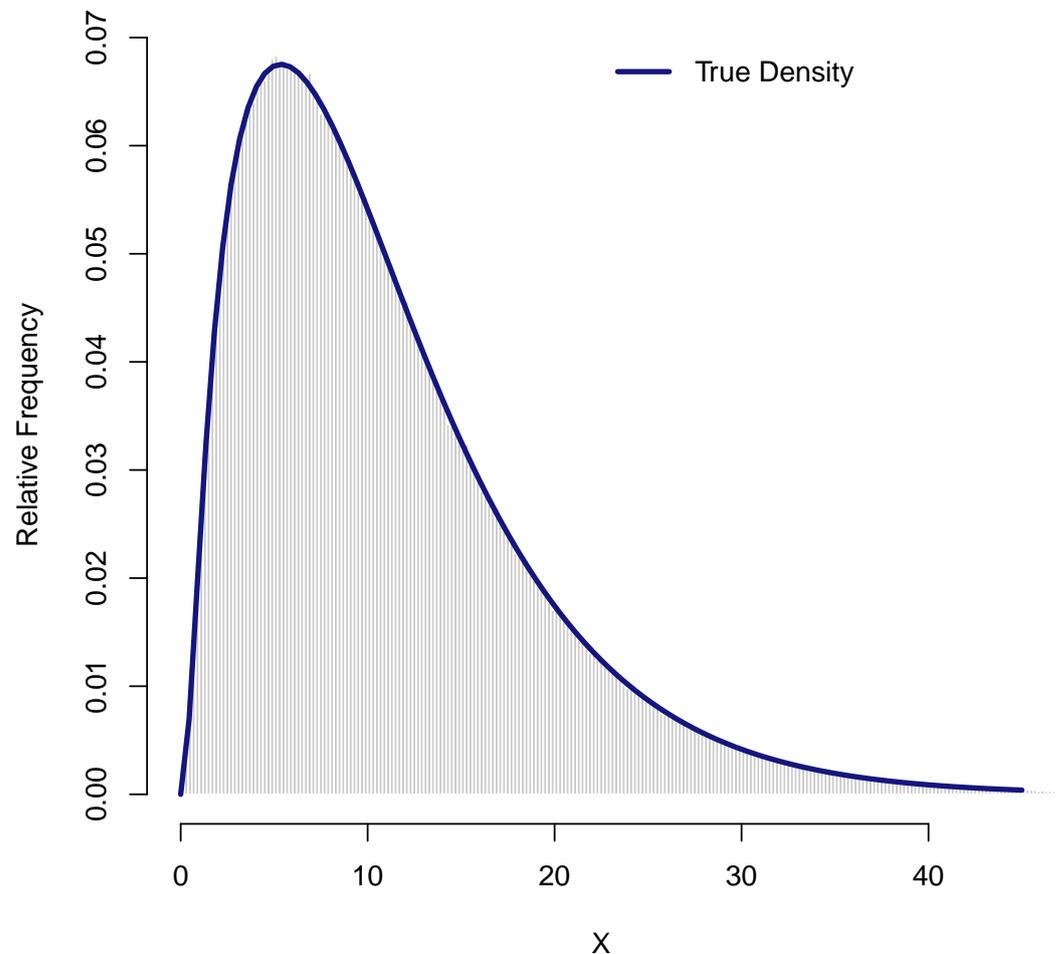


Figure 2. Mixture of five gamma distributions with different shapes and same scale parameters in Example 2.

Mixture of Gamma Distribution

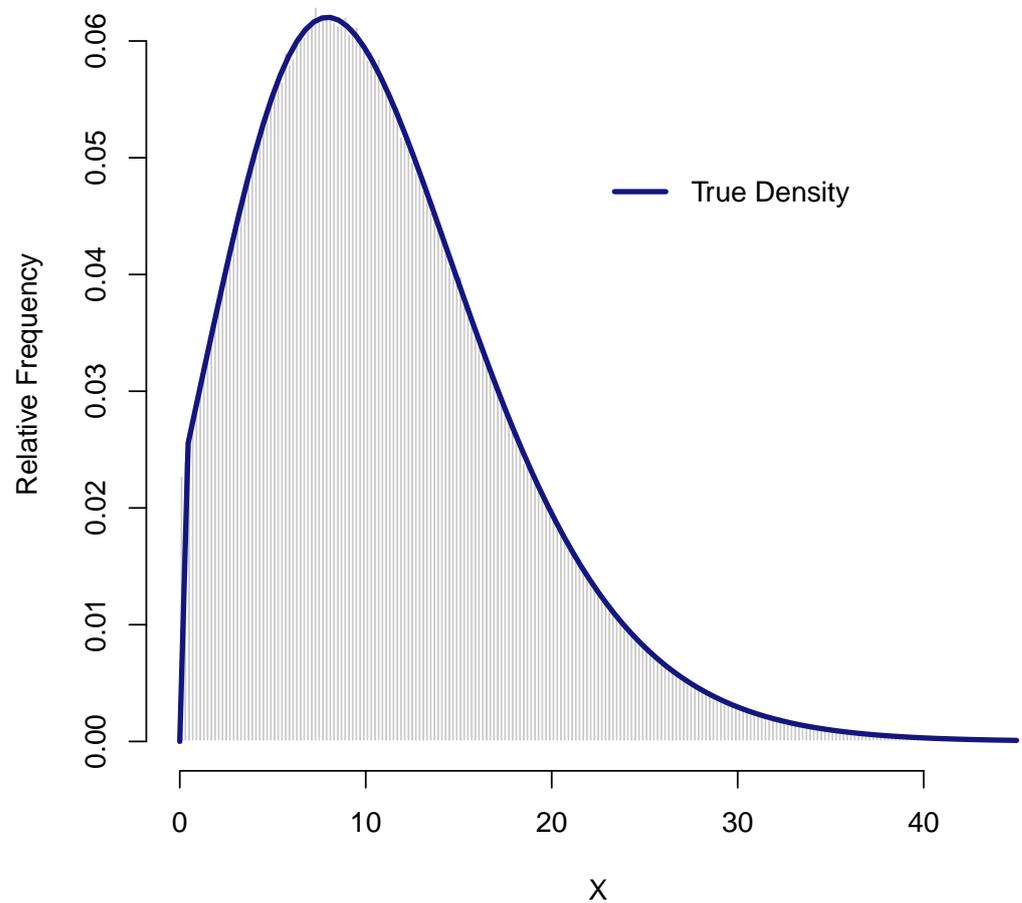


Figure 3. Mixture of five gamma distributions with different scale parameters and same shape parameters in Example 3.

Example 5. Mixture of four binomial distributions [12].

Consider

$$F(x) = \sum_{i=1}^4 \pi_i F_i(x),$$

where $X_i \sim \text{binomial}(m = 10, \theta_i)$ are independent, with $\theta_1 = 0.1$, $\theta_2 = 0.2$, $\theta_3 = 0.6$, and $\theta_4 = 0.9$. The mixing probabilities are $\pi_1 = \pi_2 = \pi_3 = 0.2$ and $\pi_4 = 0.4$. Using Theorem 1, a sample of size 10^6 was generated from $F(x)$. For comparison, we analyzed the theoretical mean and variance alongside the sample mean and variance. As stated by [9], we have $E[X] = \mu = \sum_{i=1}^4 \pi_i \mu_i$ and $V[X] = \sum_{i=1}^4 \pi_i \sigma_i^2 + \sum_{i=1}^4 \pi_i (\mu_i - \mu)^2$. In this example, $\mu_i = m\theta_i$ and $\sigma_i^2 = m\theta_i(1 - \theta_i)$. Thus, $\mu = 16.20$ and $\sigma^2 = 106.98$. Additionally, the sample mean and variance are 16.2050 and 106.9789, respectively. This indicates a close correspondence between the theoretical and sample statistics.

Example 6. Mixture of normal and Poisson distributions.

Consider the mixture distribution given by

$$F(x) = 0.7F_1(x) + 0.3F_2(x),$$

where X_1 follows a normal distribution with mean 10 and variance 4 and X_2 follows a Poisson distribution with mean 4. X_1 and X_2 are independent. Utilizing Theorem 1, a sample of size 10^6 was generated from $F(x)$. As in Example 4, the exact mean and the exact variance of the mixture distribution are $\mu = 0.7 \times 10 + 0.3 \times 5 = 8.50$ and $\sigma^2 = 0.7(4 + (10 - \mu)^2) + 0.3(5 + (5 - \mu)^2) = 9.55$. Additionally, the simulated mean and variance of the mixture distribution are 8.4963 and 9.5485, respectively. This demonstrates a close correspondence between the theoretical and sample statistics.

Table 1. Comparison of proposed and composition methods.

$F(x)$	n	Proposed		Composition	
		\bar{D}	$sd(\hat{D})$	\bar{D}	$sd(\hat{D})$
Case 1	20	0.008771	0.007934	0.008814	0.008036
	40	0.004291	0.003903	0.004295	0.003871
	60	0.002812	0.002515	0.002851	0.002566
	80	0.002126	0.00188	0.002109	0.001861
	100	0.001687	0.001539	0.001674	0.001517
Case 2	20	0.008795	0.008048	0.008814	0.008207
	40	0.004269	0.003854	0.004325	0.003945
	60	0.002822	0.00256	0.002842	0.002528
	80	0.002087	0.00189	0.002108	0.001951
	100	0.001692	0.001546	0.0017	0.001518
Case 2	20	0.008813	0.008194	0.008857	0.008049
	40	0.004299	0.003889	0.004293	0.003844
	60	0.002842	0.002514	0.00285	0.002601
	80	0.002148	0.001903	0.002173	0.002008
	100	0.001678	0.001485	0.001711	0.001569

4. Conclusions

This paper introduces a modified version of the composition method for sampling finite mixture distributions. By incorporating sampling from the uniform $(0, 1)$ distribution, our modification aligns with prevalent methods in computational statistics, such as the inverse transform and acceptance–rejection methods. This modification not only enhances the consistency and accuracy of sampling procedures but also simplifies the teaching of computational statistics courses, where sampling from the uniform $(0, 1)$ distribution is a common step in various algorithms.

The effectiveness of the proposed modification is demonstrated through several illustrative examples, showcasing its robust performance across different scenarios. From mixtures of normal and gamma distributions to binomial and Poisson mixtures, the proposed algorithm consistently generates samples that closely match the theoretical distributions. Moreover, comparison metrics such as the Cramér–von Mises distance provide quantitative evidence of the algorithm’s efficiency and accuracy, especially as sample sizes increase.

Overall, the modified composition method presented in this paper offers a valuable addition to the toolkit of computational statisticians and educators alike. Its simplicity, consistency, and performance make it a practical choice for sampling finite mixture distributions in various applications.

Supplementary Materials: The following supporting information can be downloaded at: www.mdpi.com/xxx/s1. R Code: A Short Note on Generating a Random Sample from Finite Mixture Distributions.

Author Contributions: Methodology, L.A.-L.; Software, A.L.; Writing—original draft, A.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: No new data were created or analyzed in this study. Data sharing is not applicable to this article.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Rizzo, M. *Statistical Computing with R*; CRC Press: Boca Raton, FL, USA; Taylor & Francis: Abingdon, UK, 2019.
2. Hothorn, T.; Everitt, B.S. *A Handbook of Statistical Analyses Using R*, 2nd ed.; Chapman and Hall/CRC: Boca Raton, FL, USA, 2009.
3. Everitt, B.S. An introduction to finite mixture distributions. *Stat. Methods Med. Res.* **1996**, *5*, 107–127. [[CrossRef](#)] [[PubMed](#)]
4. Lin, W.C.; Emura, T.; Sun, L.H. Estimation under copula-based Markov normal mixture models for serially correlated data, *Commun. Stat.-Simul. Comput.* **2021**, *50*, 4483–4515. [[CrossRef](#)]
5. Cai, J.; Xu, Q.; Cao, M.; Yang, Y. Capacity credit evaluation of correlated wind resources vsing vine copula and improved importance sampling. *Appl. Sci.* **2019**, *9*, 199. [[CrossRef](#)]
6. Escobar, M.D.; West, M. Bayesian density estimation and inference using mixtures. *J. Am. Stat. Assoc.* **1995**, *90*, 577–588. [[CrossRef](#)]
7. Titterington, D.M.; Smith, A.F.M.; Makov, U.E. *Statistical Analysis of Finite Mixture Distributions*; John Wiley & Sons: New York, NY, USA, 1985.
8. Tanizaki, H. *Computational Methods in Statistics and Econometrics*, 1st ed.; Marcel Dekker: New York, NY, USA, 2004.
9. Hogg, R.V.; McKean, J.W.; Craig, A.T. *Introduction to Mathematical Statistics*, 8th ed.; Person: Boston, MA, USA, 2019.
10. McLachlan, G.; Peel, D. *Finite Mixture Models*; John Wiley & Sons, Inc.: New York, NY, USA, 2000.
11. Ghorbanzadeh, D.; Dur, P.; Jaupi, L. A method for the generate a random sample from a finite mixture distributions. In Proceedings of the 6th Annual International Conference on Computational Mathematics, Computational Geometry & Statistics (CMCGS 2017), Singapore, 6–7 March 2017. [[CrossRef](#)]
12. Everitt, B.S.; Hand, D.J. *Finite Mixture Distributions*; Chapman & Hall: New York, NY, USA, 1981.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.