Proceedings of the 63rd Annual Conference of the South African Statistical Association for 2022

30 November – 2 December 2022
George

Proceedings of the 63rd Annual Conference of the South African Statistical Association for 2022 (SASA 2022)

ISBN 978-0-86886-877-6

Editor

Sheetal Silal University of Cape Town

Assistant Editors

Allan Clark	University of Cape Town
Justin Harvey	Stellenbosch University
Andréhette Verster	University of the Free State

Managing Editor

Charl Pretorius North-West University

Review Process

Five (5) manuscripts were submitted for possible inclusion in the Proceedings of the 63rd Annual Conference of the South African Statistical Association. All submitted papers were assessed by the Editorial team for suitability, after which all papers were sent to be reviewed by at least two independent reviewers each. Papers were reviewed according to the following criteria: relevance to conference themes, relevance to audience, standard of writing, originality and critical analysis. After consideration and incorporation of reviewer comments, all five manuscripts were judged to be suitable for inclusion in the proceedings of the conference.

Reviewers
The editorial team would like to thank the following reviewers:

Tertius de Wet	University of Stellenbosch
Şebnem Er	University of Cape Town
Linda Haines	University of Cape Town
David Hofmeyr	University of Stellenbosch
Dominique Katshunga	University of Cape Town
Retha Luus	University of the Western Cape
Chioneso Marange	University of Fort Hare
Kanshukan Rajaratnam	University of Stellenbosch
Edmore Ranganai	University of South Africa
Hassan Sadiq	University of Stellenbosch
Sean van der Merwe	University of the Free State

Contact Information
Queries can be sent by email to the Managing Editor (managing.editor@sastat.org).

Table of Contents

Proceedings of the 63rd Annual Conference of the South African Statistical Association

Wasserstein distance as discriminator within the Dirichlet family

Tanita Botha, Johan Ferreira and Andriëtte Bekker

Department of Statistics, University of Pretoria, Pretoria, South Africa
Centre of Excellence in Mathematical and Statistical Sciences, University of the Witwatersrand, Johannesburg, South Africa

The Dirichlet distribution is a cornerstone consideration when working with data on the unitary simplex. Several generalisations of the Dirichlet distribution have been developed with more flexible structures which can be applied to data that exhibit departures from the usual Dirichlet, such as multimodality and positive correlation. To gain a deeper insight into the impact of fitting generalisations of the Dirichlet to data that exhibit these structural departures, the Wasserstein distance is considered and investigated between different members of the Dirichlet family. Since this distance gives an intuitive measure of the distance and difference between two (multivariate) distributions, this paper explores the differences between several (competitive) Dirichlet constructions to highlight and examine the effect that these structural changes may result in.

Keywords: Flexible, Generator, Mixture, Noncentral, Poisson.

1. Introduction

The *Wasserstein distance* (d_W), originally introduced in Vaserstein (1969), is known as a fundamental metric in the space of probability measures. The Wasserstein distance has different names (Sommerfeld and Munk, 2018) such as the Mallows distance, the Monte-Kantorovich-Rubinstein distance in the physical sciences, the earth mover's distance in computer science or the optimal transport distance in optimisation, and bears a clear and intuitive interpretation namely "the amount of work required to turn one probability distribution into another." In statistics, it has been used to prove convergence in the context of limit laws (Sommerfeld and Munk, 2018) and has more recently been implemented as a measure of prior impact in Bayesian analysis of computational data (Ley et al., 2017; Ghaderinezhad et al., 2020, 2021). Two recent studies have employed the Wasserstein distance to predict the remaining useful life of rotation machinery using conditional Wasserstein distance-based generative adversarial networks (Man et al., 2022) and research aiming to improve the communication security of Internet of Vehicles nodes in intelligent transportation using models such as the Wasserstein Distance Based Generative Adversarial Network (WaGAN) model (Liu et al., 2022).

The Kolmogorov-Smirnov (KS) distance is closely related to d_W; however, the main drawback from using the KS is that this distance measure finds the maximal difference point between the two

Corresponding author: Tanita Botha (tanita.botha@up.ac.za)
MSC2020 subject classifications: 62E15, 62P12, 60E15

distributions rather than considering the whole space between the two distributions. d_W gives an intuitive measure which can be used to discriminate between two distributions under consideration; but it is necessary to exercise caution as its computation may become complex in high dimensional settings. A high d_W indicates that the two distributions are far apart and hence differ greatly while a small d_W indicates a sense of similarity between the two considered distributions; this measure is defined in the following definition.

Definition 1. The Wasserstein distance $d_W(P_1, P_2)$ (Vallender, 1974) is calculated using the cumulative distribution functions $F_i(\upsilon; x)$ of two distributions P_1 and P_2, where $\upsilon \in \mathbb{R}$ is the parameter of interest and x represents the data (Ghaderinezhad et al., 2021):

$$d_W(P_1, P_2) = \int_a^b |F_1(\upsilon; x) - F_2(\upsilon; x)| d\upsilon,$$

where a and b denote the bounds of the support of υ. d_W can also be extended to $m > 1$ parameters $\upsilon_1, ..., \upsilon_m$ as follows:

$$d_W(P_1, P_2) = \int_{a_1}^{b_1} ... \int_{a_m}^{b_m} |F_1(\upsilon_1, ..., \upsilon_m; x) - F_2(\upsilon_1, ..., \upsilon_m; x)| d\upsilon_1 ... d\upsilon_m \tag{1}$$

where a_i and b_i are the bounds of the support of υ_j, $j = 1, \ldots, m$.

The Dirichlet distribution, a multivariate generalisation of the beta distribution, is used often when working with data on the unitary simplex. This paper aims to explore the differences between members of the Dirichlet family by using d_W to increase understanding regarding the effect that structural changes have between these distributions. Some of the considered members include the Dirichlet generator, of which numerous flexible candidates can be "generated", as explored in Botha et al. (2021) as well as the noncentral Dirichlet construction in Botha et al. (2022), which depends on the noncentrality parameters through the confluent hypergeometric function of several variables. Mixtures of Dirichlet are also considered, such as the flexible Dirichlet distribution as proposed by Ongaro and Migliorati (2013), which is expressed as a finite mixture of Dirichlet components, as well as the double flexible Dirichlet distribution as proposed by Ascari et al. (2021), which is a further generalisation of the Dirichlet structure, which takes advantage of the finite mixture structure of the flexible Dirichlet distribution, as well as the capabilities of modelling positive correlations. These mixtures' properties and structures were further explored in Ferreira et al. (2022). Computational focus is in the two-dimensional case for illustrative purposes in this paper.

The paper is outlined as follows. In Section 2 the considered distributions are briefly revisited. In Section 3, a practical investigation will explore to what extent the d_W discriminates between the different Dirichlet family members under consideration. Section 4 contains conclusions and future work.

2. Preliminary definitions and properties

In this section, the multivariate models of interest are briefly reviewed. The Dirichlet distribution is known for elegant mathematical properties and ease of parameter estimation (Ongaro and Migliorati, 2013), but is poorly parametrised and cannot model many different types of dependence patterns

(Ascari et al., 2021). The following sections review the different (generalised) siblings within the considered Dirichlet family which will be considered in this paper. Let $\boldsymbol{p} = (p_1, \ldots, p_K)$.

2.1 Dirichlet Distribution

The Dirichlet distribution will be our base model of interest.

Definition 2. Suppose $\boldsymbol{p}$ is distributed as a Dirichlet distribution (of type 1, see Sánchez et al., 2006) of order $K \geq 2$ and parameters $\boldsymbol{\Pi} = (\pi_1, \pi_2, \ldots, \pi_{K+1})$ for $\pi_i > 0$, $i = 1, \ldots, K+1$, with respect to the Lebesgue measure on the Euclidean space $\mathbb{R}^K$, then its probability density function (pdf) is given by

$$f(\boldsymbol{p}; \boldsymbol{\Pi}) = \frac{\Gamma(\pi_+)}{\prod_{i=1}^{K+1} \Gamma(\pi_i)} \left(\prod_{i=1}^{K+1} p_i^{\pi_i - 1} \right) \tag{2}$$

on the K dimensional simplex, defined by

$$\begin{aligned} p_1, p_2, \ldots, p_K &> 0, \\ p_1 + p_2 + \cdots + p_K &< 1, \\ p_{K+1} &= 1 - p_1 - \cdots - p_K, \end{aligned}$$

which is denoted by $\mathcal{A}$, where $\Gamma(\cdot)$ denotes the usual gamma function, $\pi_+ = \sum_{i=1}^{K+1} \pi_i$ and p_{K+1} is the vertex boundary to be triangular on the unit simplex.

The contours of this considered model are denoted by "D".

2.2 Noncentral Dirichlet Distribution

The noncentral Dirichlet distribution, constructed via the use of Poisson weights (inspired by Ferreira et al. (2016) and the model of interest in Botha et al. (2022)) is defined as:

Definition 3. Suppose $\boldsymbol{p}$ is noncentral Dirichlet distributed, then the pdf is given by:

$$\begin{aligned} & h_1(\boldsymbol{p}; \boldsymbol{\Pi}, \boldsymbol{\Lambda}) \\ = & \sum_{j_1=0}^{\infty} \cdots \sum_{j_{K+1}=0}^{\infty} \frac{\exp(\frac{\lambda_1}{2})(\frac{\lambda_1}{2})^{j_1}}{j_1!} \cdots \frac{\exp(\frac{\lambda_K}{2})(\frac{\lambda_K}{2})^{j_K}}{j_K!} \frac{\exp(\frac{\lambda_{K+1}}{2})(\frac{\lambda_{K+1}}{2})^{j_{K+1}}}{j_{K+1}!} \\ \times & \frac{\Gamma(\pi_1 + j_1 + \cdots + \pi_K + j_K + \pi_{K+1} + j_{K+1})}{\Gamma(\pi_1 + j_1)\Gamma(\pi_K + j_K)\Gamma(\pi_{K+1} + j_{K+1})} p_1^{\pi_1 + j_1 - 1} \cdots p_K^{\pi_K + j_K - 1} (1 - \sum_{i=1}^{K} p_i)^{\pi_{K+1} + j_{K+1} - 1} \\ = & f(\boldsymbol{p}; \boldsymbol{\Pi}) \exp\left(- \sum_{i=1}^{K+1} \frac{\lambda_i}{2} \right) \\ \times & \sum_{\phi} \frac{(\pi_1 + \cdots + \pi_{K+1})_{j_1 + \cdots + j_{K+1}}}{(\pi_1)_{j_1} \ldots (\pi_{K+1})_{j_{K+1}} j_1! \ldots j_{K+1}!} \left(\frac{\lambda_1}{2} p_1 \right)^{j_1} \cdots \left(\frac{\lambda_K}{2} p_K \right)^{j_K} \left(\frac{\lambda_{K+1}}{2} (1 - \sum_{i=1}^{K} p_i) \right)^{j_{K+1}} \end{aligned} \tag{3}$$

where the vector $\mathbf{p} \in \mathcal{A}$ is thus a noncentral Dirichlet variate with parameters $\boldsymbol{\Pi} = (\pi_1, \pi_2, \ldots, \pi_{K+1})$ for $\pi_i > 0, i = 1, \ldots, K+1$, $\boldsymbol{\Lambda} = (\lambda_1, \ldots, \lambda_K, \lambda_{K+1})$ denoting corresponding noncentral parameters

with $\lambda_i > 0 \; \forall \; i = 1, ..., K+1$ and where $\sum_{\phi} = \sum_{j_1=0}^{\infty} \cdots \sum_{j_{K+1}=0}^{\infty}$. Note: this is also a shape mixture as each p_i in this noncentral model is indexed by a component-specific shape parameter j, and also, the Dirichlet distribution (see (2)) is recovered in the case when $\lambda_i \to 0 \; \forall \; i = 1, ..., K+1$.

The contours of this considered model are denoted by "NC".

2.3 Dirichlet Generator Distribution

The Dirichlet generator distribution, proposed as an alternative candidate to the Dirichlet distribution in Botha et al. (2021), consists of numerous flexible candidates and has the following form:

Definition 4. Suppose $\boldsymbol{p}$ is Dirichlet generator distributed. Then its pdf is given by

$$h_2(\boldsymbol{p}; \boldsymbol{\Pi}) = C p_1^{\pi_1 - 1} p_2^{\pi_2 - 1} \dots p_K^{\pi_K - 1} (1 - \sum_{i=1}^{K} p_i)^{\pi_{K+1} - 1} g(\theta \sum_{i=1}^{K} p_i), \tag{4}$$

with C a normalising constant such that

$$C^{-1} = \int_{\mathcal{A}} \cdots \int p_1^{\pi_1 - 1} p_2^{\pi_2 - 1} \dots p_K^{\pi_K - 1} (1 - \sum_{i=1}^{K} p_i)^{\pi_{K+1} - 1} g(\theta \sum_{i=1}^{K} p_i) d\boldsymbol{p}.$$

The vector $\boldsymbol{p} \in \mathcal{A}$ is thus a Dirichlet generator variate with parameters $\boldsymbol{\Pi} = (\pi_1, \dots, \pi_{K+1})$, scalar $\theta \in \mathbb{R}$, and whichever additional parameters $g(\cdot)$ imposed to ensure that the pdf $h(\cdot)$ is non-negative. The following conditions also apply:

1. $g(\cdot)$ is a Borel-measurable function;
2. $g(\cdot)$ admits a Taylor series expansion;
3. $g(0) = 1$.

The usual Dirichlet distribution with pdf (2) is thus a special case of (4) when $\theta = 0$. Other flexible candidates, which resulted in instances allowing for non-negative correlation, include the three hypergeometric functions (${}_0F_0$; ${}_0F_1$ and ${}_1F_1$), as investigated in Botha et al. (2021) which are commonly considered functions representing exponential, binomial and the confluent hypergeometric functions.

The contours of this considered model are denoted by "GD".

2.4 Flexible Dirichlet Distribution

The flexible Dirichlet distribution as proposed by Ongaro and Migliorati (2013) is defined as follows:

Definition 5. Suppose $\boldsymbol{p}$ is distributed as a flexible Dirichlet distribution. Then its pdf is given by

$$\begin{aligned} h_3(\boldsymbol{p}; \boldsymbol{\Pi}, \tau, \boldsymbol{\beta}) &= \sum_{r=1}^{K+1} \beta_r f(\boldsymbol{p}, \boldsymbol{\Pi} + \tau \boldsymbol{e}_r) \\ &= \frac{\Gamma(\sum_{i=1}^{K+1} \pi_i + \tau)}{\prod_{i=1}^{K+1} \Gamma(\pi_i)} \left(\prod_{i=1}^{K+1} p_i^{\pi_i - 1} \right) \left[\sum_{r=1}^{K+1} \beta_r p_r^{\tau} \frac{\Gamma(\pi_r)}{\Gamma(\pi_r + \tau)} \right], \end{aligned} \tag{5}$$

where the vector $\mathbf{p} \in \mathcal{A}$ is thus a flexible Dirichlet variate with parameters $\mathbf{\Pi} = (\pi_1, \pi_2, \ldots, \pi_{K+1})$ for $\pi_i > 0, i = 1, \ldots, K+1$, $\boldsymbol{\beta} = (\beta_1, \cdots, \beta_{K+1})$ with $0 \leq \beta_i < 1$ and $\sum_{i=1}^{K+1} \beta_i = 1$, the scale parameter $\tau > 0$, $\boldsymbol{e}_r$ is the vector with elements equal to zero except for the r-th entry that is equal to 1. The flexible Dirichlet also includes the Dirichlet as a special case if $\tau = 1$ and $\beta_i = \frac{\pi_i}{\pi_+}, i = 1, 2, ..., K+1$.

The contours of this considered model are denoted by "FD".

2.5 Double Flexible Dirichlet Distribution

The double flexible Dirichlet distribution, an extension of the flexible Dirichlet distributions, and proposed by Ascari et al. (2021) has the following form:

Definition 6. Suppose $\boldsymbol{p}$ is distributed as a double flexible Dirichlet distribution. Then its pdf is given by

$$\begin{aligned} h_4(\boldsymbol{p}; \mathbf{\Pi}, \tau, \boldsymbol{\beta}) &= \sum_{r=1}^{K+1} \sum_{s=1}^{K+1} \beta_{rs} f(\boldsymbol{p}, \mathbf{\Pi} + \tau(\boldsymbol{e}_r + \boldsymbol{e}_s)) \\ &= \frac{\Gamma(\sum_{i=1}^{K+1} \pi_i + 2\tau)}{\prod_{i=1}^{K+1} \Gamma(\pi_i)} \left(\prod_{i=1}^{K+1} p_i^{\pi_i - 1} \right) \\ &\quad \times \left[\sum_{\substack{r=1 \\ r \neq s}}^{K+1} \sum_{s=1}^{K+1} \beta_{rs} p_r^{\tau} p_s^{\tau} \frac{\Gamma(\pi_r)\Gamma(\pi_s)}{\Gamma(\pi_r + \tau)\Gamma(\pi_s + \tau)} + \sum_{r=1}^{K+1} \beta_{rr} p_r^{2\tau} \frac{\Gamma(\pi_r)}{\Gamma(\pi_r + 2\tau)} \right], \end{aligned} \quad (6)$$

where the vector $\mathbf{p} \in \mathcal{A}$ is thus a double flexible Dirichlet variate with parameters $\mathbf{\Pi} = (\pi_1, \pi_2, \ldots, \pi_{K+1})$ for $\pi_i > 0, i = 1, \ldots, K+1$, $0 \leq \beta_{rs} < 1$ and $\sum_{r=1}^{K+1} \sum_{s=1}^{K+1} \beta_{rs} = 1$, the scale parameter $\tau > 0$, and $\boldsymbol{e}_{r,s}$ is the vector with elements equal to zero except for the r, s-th entry that is equal to 1.

The contours of this considered model are denoted by "DFD". A particular valuable aspect of this model is its ability to model positive correlation of $\boldsymbol{p}$ (Ferreira et al., 2022) which is theoretically not possible with the usual Dirichlet distribution.

3. Practical Investigation

Using d_W (1), this section investigates the effect that the different structural changes had on the members of the considered Dirichlet family. The Dirichlet distribution is included as a base to compare against but the differences between all members were included to ensure all possible combinations of changes are investigated (see Figure 3). The structural changes which will be explored include shifts in the distributions, distributional concentration changes, and the effect of modes. This knowledge can advise and guide model selection, discrimination between similar models, and measure model similarity which can advise future investigation and implementation such as utilising alternative distributional structures as priors in (multivariate) Bayesian analysis. The *"transport"* (Schuhmacher et al., 2020) package in R (R Core Team, 2021) has functions which can be used to estimate the Wasserstein distance between two distributions, and can be extended to any order between two sets of samples from different distributions (Ghaderinezhad et al., 2020).

3.1 Considered Distributions

Figures 1 and 2 introduce the contour plots of the distributions that will be considered in this practical investigation. Figure 1 displays the Dirichlet, noncentral Dirichlet and Dirichlet generator family members and Figure 2 the flexible Dirichlet as well as two different β combinations for the double flexible Dirichlet distribution. It is essential to note that each column holds the same parameter values for π across all considered plots:

- The contours in the first column always consist of $\pi_1 = 4$, $\pi_2 = 4$ and $\pi_3 = 4$.
- The contours in the second column always consist of $\pi_1 = 2$, $\pi_2 = 4$ and $\pi_3 = 4$.
- The contours in the third column always consist of $\pi_1 = 10$, $\pi_2 = 4$ and $\pi_3 = 4$,

and the rows containing the different considered candidates. Within these contours the different parameter constructions are adjusted in order to apply and investigate the structural changes with the use of d_W. These changes are obtained through the different distributional parameters $(\boldsymbol{\lambda}, \theta, \tau, \boldsymbol{\beta})$ for each of the distributions introduced in Section 2.

Figure 3 consists of a heat map which displays d_W between all considered distributions. It can be seen that those comparisons with the green and yellow colours resulted in larger d_W results with d_W decreasing as the colours become darker blue.

3.2 Structural Change 1: Shift changes

When investigating the shift changes in all distributions, which occurs when the $\boldsymbol{\pi}$s are changed, it can be seen that the smaller shifts (seen when comparing column 1 with column 2 – see Figures 4 and 5) results in smaller d_W results while the larger shift changes (seen when comparing column 1 and column 3) leads to larger d_W. This is expected as we know that d_W measures the amount of work required to turn one distribution to another which will be greater for the larger shifts.

3.3 Structural Change 2: Noncentral effect

For this investigation, the structural shift as well as the noncentral effect were explored comparing the Dirichlet (2) and noncentral Dirichlet (3) distribution. Even though the shifts seem similar to the small shifts investigated in Section 3.2, the noncentral effect that also comes into effect increased the size of the d_W. This is expected as it is no longer just a structural shift that needs to be considered but also a slight change in the distributional form. The third comparison shows a much smaller d_W which, when compared to the other plots, show that the smaller centroid shift had a much smaller effect than the larger centroid shift in the other two plots.

3.4 Structural Change 3: Influence of modes

The first investigation considers the addition of three modes and is done by comparing the Dirichlet distribution (2) and the double flexible Dirichlet distribution (6) – see Figure 7. We see a slightly higher d_W in the third plot with the distribution more closely condensed around the centroid. This is the case for both investigations in this section.

The second investigation compares the base model from the Dirichlet distribution (2) with the 6 modes of the DFD distribution (6) – see Figure 8. We can see that the d_W are larger than in the instance when we compared against 3 modes in the above step.

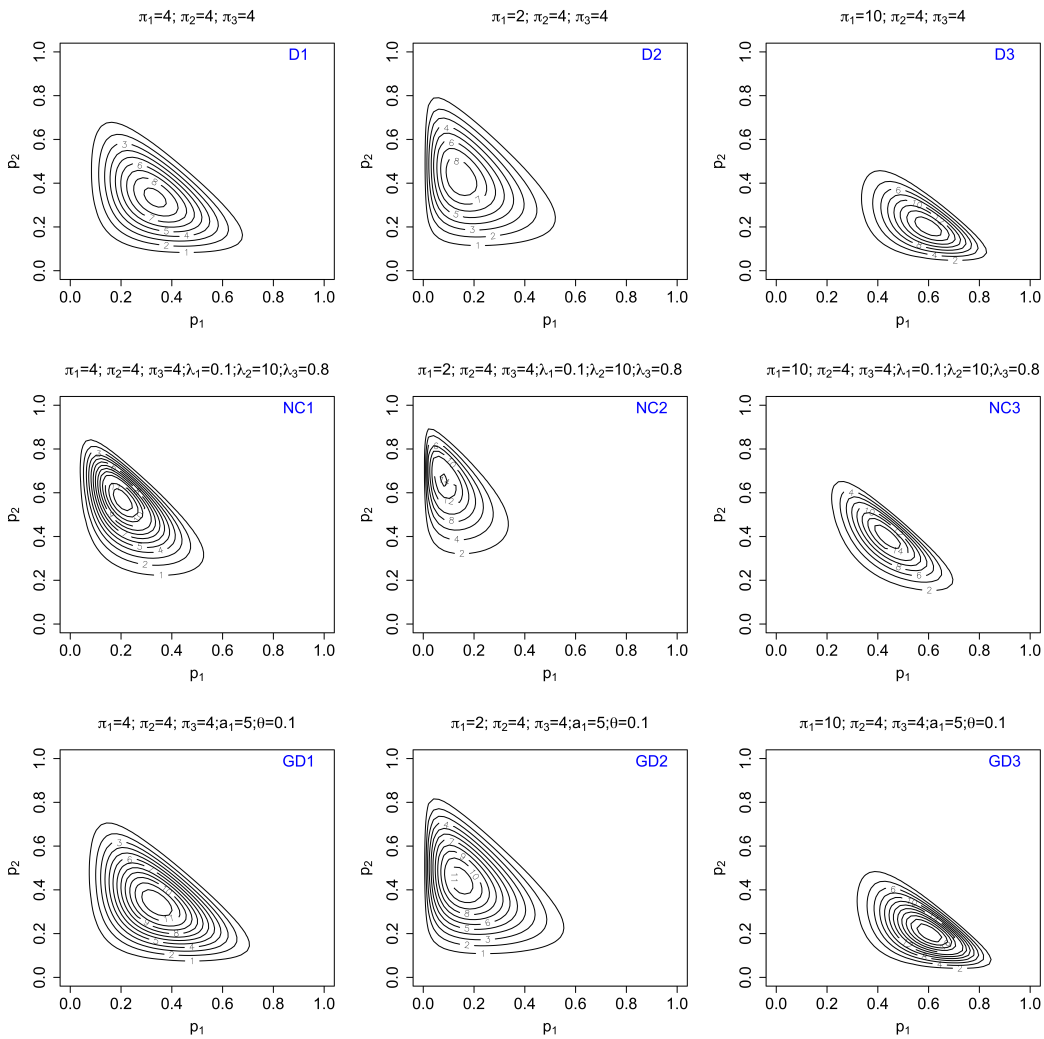


Figure 1. Contours of Dirichlet (2), noncentral Dirichlet (3), and Dirichlet generator (4) distributions.

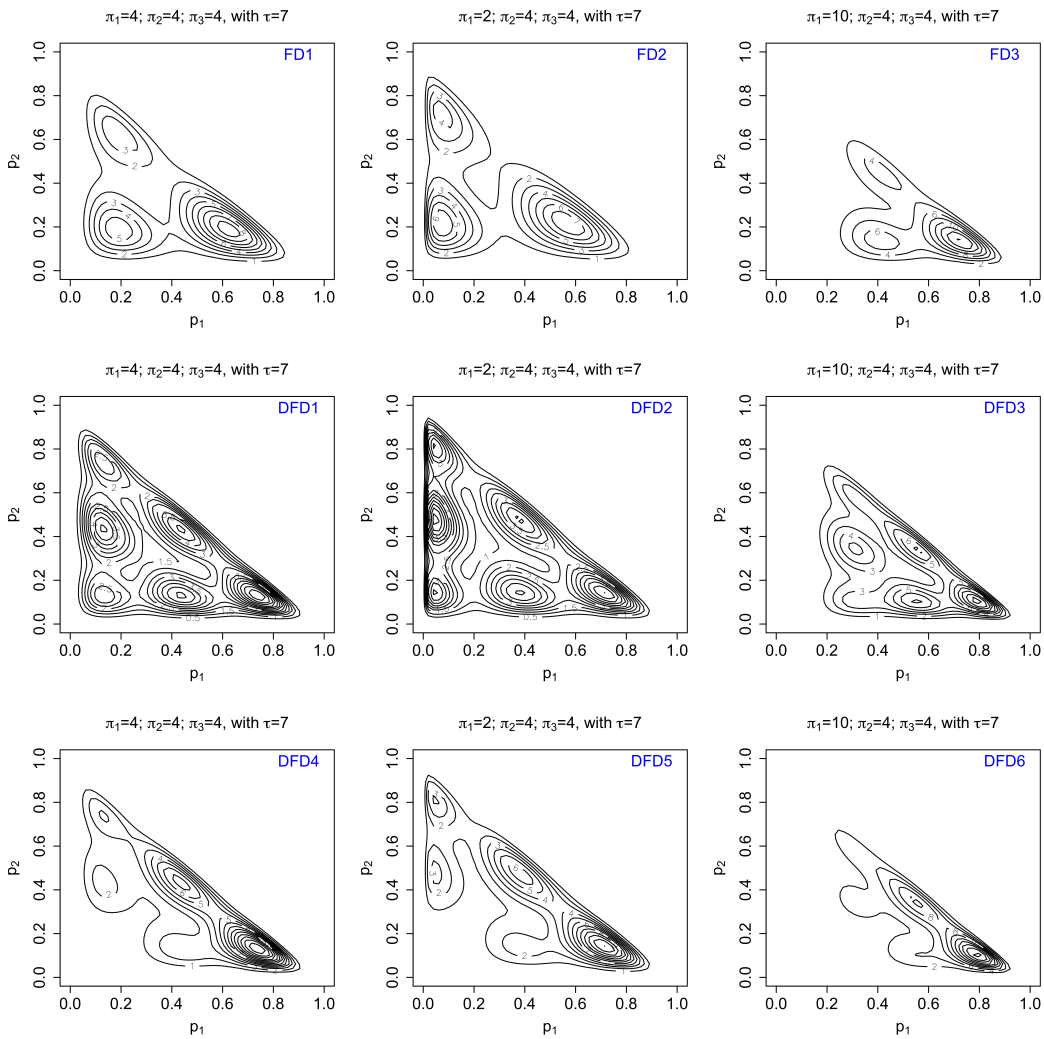


Figure 2. Contours of the flexible (5) and double flexible Dirichlet (6) distributions.

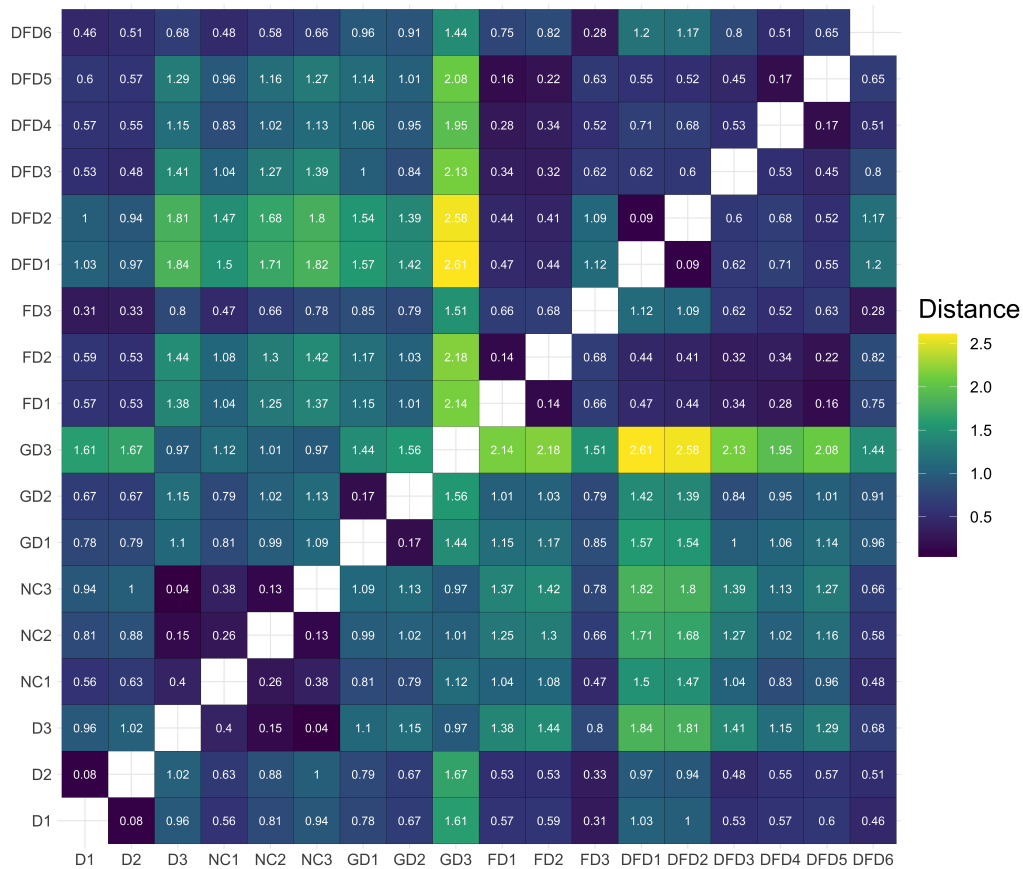


Figure 3. d_W results between all distributions considered with parameters as indicated in Figure 1 and 2.

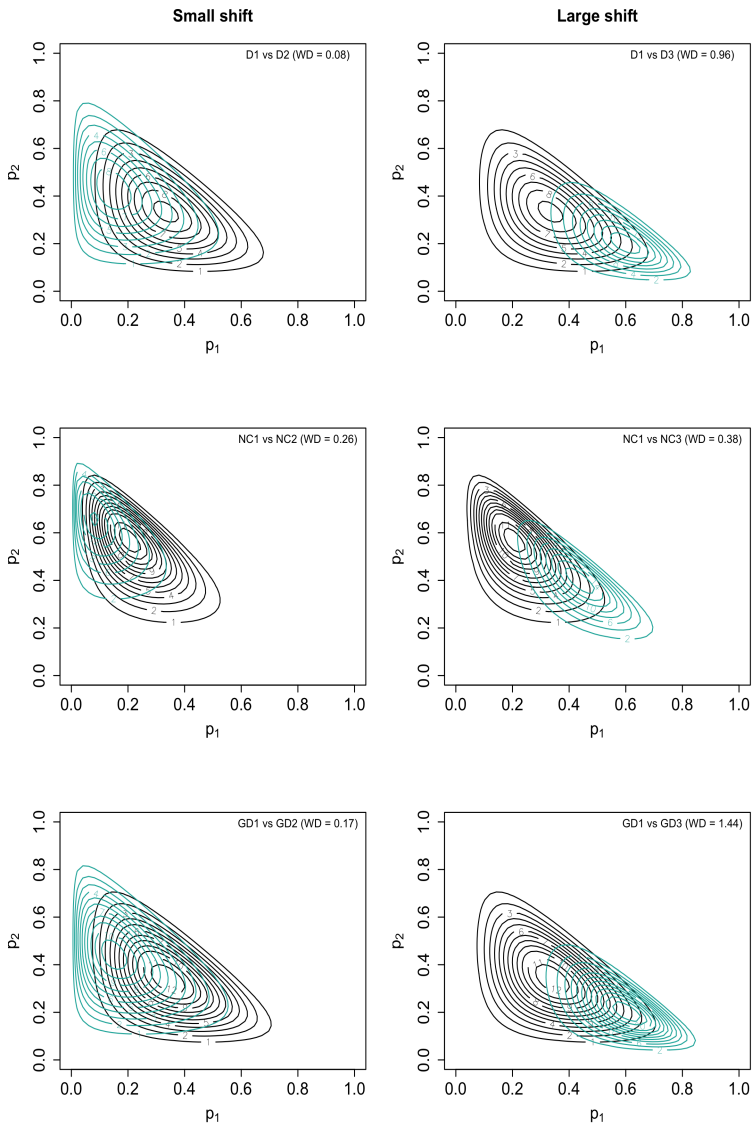


Figure 4. Shifted contours of the distributions being considered.

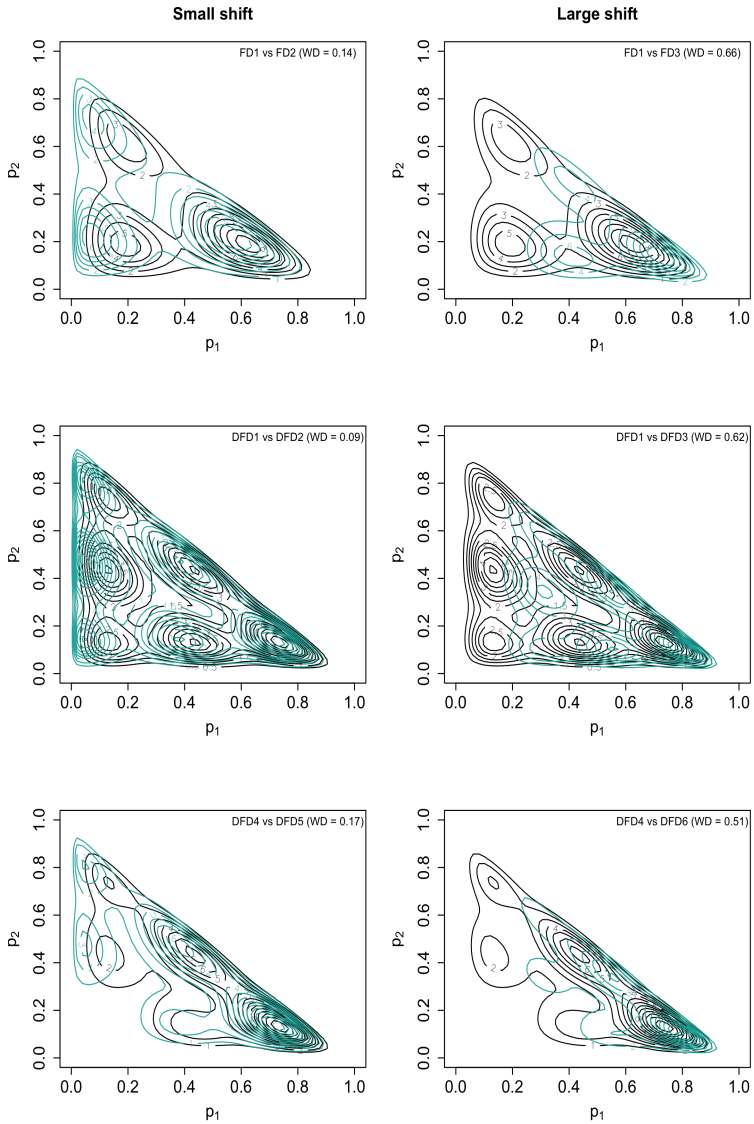


Figure 5. Shifted contours of the distributions being considered.

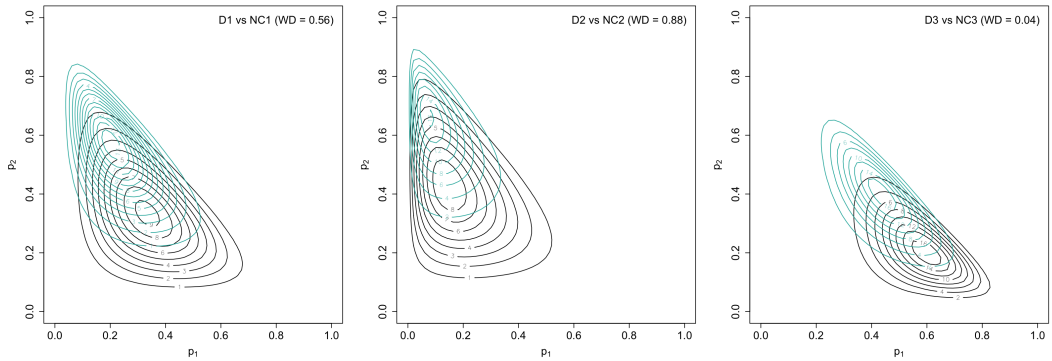


Figure 6. Contours when investigating the noncentral effect.

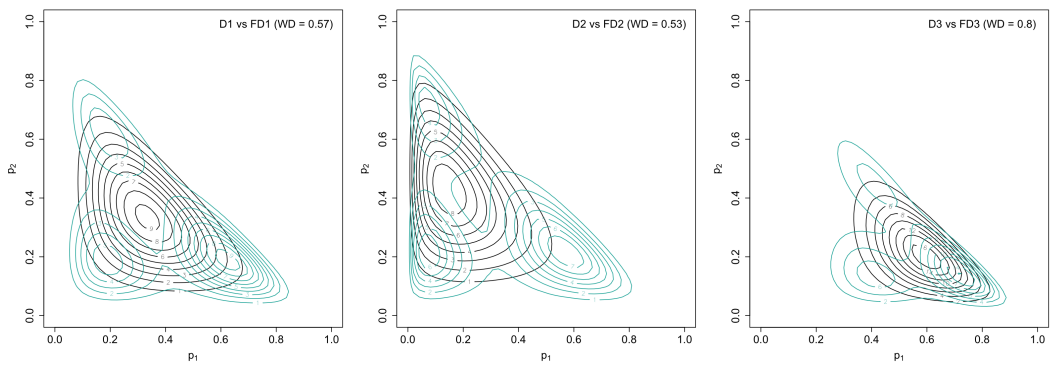


Figure 7. Contours when investigating the first mode's effect.

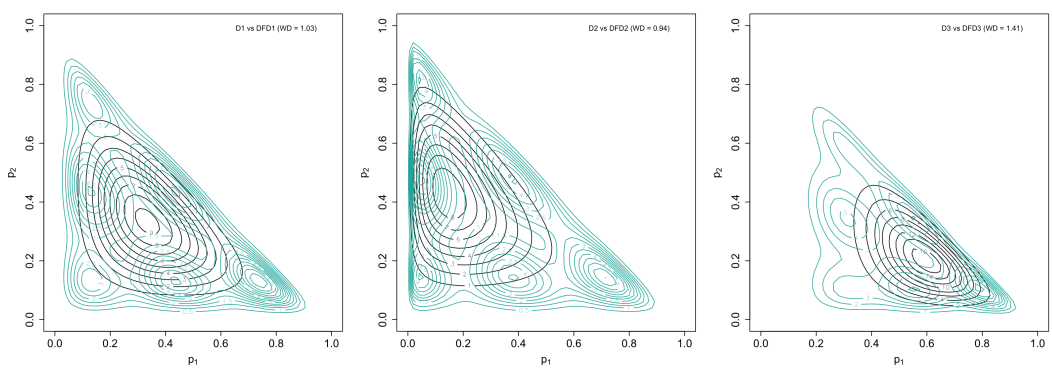


Figure 8. Contours when investigating the second mode's effect.

4. Conclusions and future work

The valuable insights found in this paper highlight the benefit of using this distance measure to discriminate between the members from the family of Dirichlet distributions considered here. Understanding how these distributions differ and how they are related guides the use of these distributions in both practical and theoretical domains. When different structures of distributions are available it is important to determine if the use and implementation of a more complex (multivariate) distributional form will add value to an investigation and how these generalisations differ from the base Dirichlet distribution.

Future work include investigations into the use Wasserstein Impact Measure (WIM), as introduced by Ghaderinezhad et al. (2021), when the generalised Dirichlet distributions are considered as priors in a Bayesian analyses. In this way, the practitioner may determine if a more complex prior added valuable information to the posterior distribution or if the base, less complicated Dirichlet distribution, should rather be used as a prior.

Acknowledgements

This work was based upon research supported in part by the National Research Foundation (NRF) of South Africa (SA), grant RA201125576565, nr 145681; NRF ref. SRUG190308422768 nr. 120839; the RDP296/2022 grant from the University of Pretoria (SA), the Department of Research and Innovation at the University of Pretoria (SA), as well as the Centre of Excellence in Mathematical and Statistical Sciences grant nr 2022-047-STA, based at the University of the Witwatersrand (SA). The opinions expressed and conclusions arrived at are those of the authors and are not necessarily to be attributed to the NRF.

References

Ascari, R., Migliorati, S., and Ongaro, A. (2021). The double flexible Dirichlet: A structured mixture model for compositional data. *Applied Modeling Techniques and Data Analysis 2: Financial, Demographic, Stochastic and Statistical Models and Methods*, **8**, 135–152.

Botha, T., Ferreira, J. T., and Bekker, A. (2021). Alternative Dirichlet priors for estimating entropy via a power sum functional. *Mathematics*, **9**, 1493.

Botha, T., Ferreira, J. T., and Bekker, A. (2022). Some computational aspects of a noncentral Dirichlet family. *In Innovations in Multivariate Statistical Modelling: Navigating Theoretical and Multidisciplinary Domains*. Springer.

Ferreira, J., Bekker, A., and Arashi, M. (2016). Bivariate noncentral distributions: an approach via the compounding method. *South African Statistical Journal*, **50**, 103–122.

Ferreira, J. T., Botha, T., and Bekker, A. (2022). Tsallis and other generalised entropy forms subject to Dirichlet mixture priors. *Symmetry*, **14**, 1110.

Ghaderinezhad, F., Ley, C., and Serrien, B. (2020). The Wasserstein Impact Measure (WIM): a generally applicable, practical tool for quantifying prior impact in Bayesian statistics. *arXiv preprint arXiv:2010.12522*.

Ghaderinezhad, F., Ley, C., and Serrien, B. (2021). The Wasserstein Impact Measure (WIM): A

practical tool for quantifying prior impact in Bayesian statistics. *Computational Statistics & Data Analysis*, 107352.

Ley, C., Reinert, G., and Swan, Y. (2017). Distances between nested densities and a measure of the impact of the prior in Bayesian statistics. *The Annals of Applied Probability*, **27**, 216–241.

Liu, J., Zhang, L., Li, C., Bai, J., Lv, H., and Lv, Z. (2022). Blockchain-based secure communication of intelligent transportation digital twins system. *IEEE Transactions on Intelligent Transportation Systems*, **23**, 22630–22640.

Man, J., Zheng, M., Liu, Y., Shen, Y., and Li, Q. (2022). Bearing remaining useful life prediction based on AdCNN and CWGAN under few samples. *Shock and Vibration*, Article ID 1709071.

Ongaro, A. and Migliorati, S. (2013). A generalization of the Dirichlet distribution. *Journal of Multivariate Analysis*, **114**, 412–426.

R Core Team (2021). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.
URL: *https://www.R-project.org/*

Sánchez, L. E., Nagar, D., and Gupta, A. (2006). Properties of noncentral Dirichlet distributions. *Computers & Mathematics with Applications*, **52**, 1671–1682.

Schuhmacher, D., Bähre, B., Gottschlich, C., Hartmann, V., Heinemann, F., and Schmitzer, B. (2020). *transport: Computation of Optimal Transport Plans and Wasserstein Distances*. R package version 0.12-2.
URL: *https://cran.r-project.org/package=transport*

Sommerfeld, M. and Munk, A. (2018). Inference for empirical Wasserstein distances on finite spaces. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, **80**, 219–238.

Vallender, S. (1974). Calculation of the Wasserstein distance between probability distributions on the line. *Theory of Probability & Its Applications*, **18**, 784–786.

Vaserstein, L. N. (1969). Markov processes over denumerable products of spaces, describing large systems of automata. *Problemy Peredachi Informatsii*, **5**, 64–72.

Proceedings of the 63rd Annual Conference of the South African Statistical Association

Optimal window size detection in Value-at-Risk forecasting: A case study on conditional generalised hyperbolic models

Chun-Kai Huang[1], *Chun-Sung Huang*[2], *Jahvaid Hammujuddy*[3] *and Knowledge Chinhamu*[3]

[1]Curtin University, Bentley, Western Australia
[2]University of Cape Town, Cape Town, South Africa
[3]University of KwaZulu-Natal, Durban, South Africa

The conventional parametric approach for financial risk measure estimation involves determining an appropriate quantitative model, as well as a suitable historical sample period in which the model can be trained. While a lion's share of the existing literature entertains the identification of the most appropriate model for different types of financial assets, or across conflicting market conditions, little is known about the optimal choice of a historical sample period size (or window size) to train the model and estimate model parameters. In this paper, we propose a method to identify an optimal window size for model training when estimating risk measures, such as the widely-utilised Value-at-Risk (VaR) or Expected Shortfall (ES), under the generalised hyperbolic subclasses. We show that the accuracy of VaR estimates may increase significantly through our proposed method of optimal window size detection. In particular, our results demonstrate that, by relaxing the usual restriction of a fixed window size over time, superior VaR forecasts may be produced as a result of improved model parameter estimates.

Keywords: Hyperbolic, MSCI, Normal-inverse Gaussian, Value-at-Risk, Variance-gamma, Window size.

1. Introduction

An increasing number of studies in the ongoing literature has been dedicated to modelling the behaviour and characteristics of financial time series. Noticeably, a significant portion of these studies also includes contributions toward the estimation of financial risk measures. To adequately estimate financial risk measures, a robust methodology that can unequivocally describe the continuous movements of the time series needs to be identified at the onset. Subsequently, a procedure is implemented to accurately estimate the respective risk measures. Such a procedure typically involves specifying a sample period size (or window size) to employ the historical data for model training and the estimation of model parameters. This is usually imposed through a rule-of-thumb method instead of an adequate optimisation approach. However, errors in the estimation of model parameters may be exacerbated through an opaque choice of window sizes, leading to inferior risk measure estimates.

Corresponding author: Chun-Sung Huang (chun-sung.huang@uct.ac.za)
MSC2020 subject classifications: 62G32, 62M10, 91G70

Notwithstanding the above, there is a shortfall in the current literature on the identification of an optimal window size to effectively estimate model parameters when forecasting risk measures (such as VaR or ES). In practice, window sizes are often arbitrarily selected without any clear consensus or robust methodology. However, evidence from a number of prior research papers suggest that most estimation procedures (parametric or non-parametric) are sensitive to changes in window size (see, for example, Chen and Spokoiny, 2009; Halbleib and Pohlmeier, 2012; Sharma, 2012; Laker et al., 2017). In particular, a larger window size often results in low variance of estimates but raises the risk of modelling bias. On the contrary, small window sizes produce estimates that react efficiently to changing market conditions, but suffers from larger variations. Hence, the identification of an optimal window size becomes a critical task. Related studies on improving parameter estimation includes, but is not limited to, exponential smoothing, structural breaks, regime switching and adaptive point-wise estimation (see Čížek et al., 2009).

A wealth of models and methods for risk measure estimation have already been proposed in the existing body of knowledge. Prominent methodologies include, among others, the use of extreme value analysis (McNeil and Frey, 2000), the generalised lambda distribution (Corlu and Corlu, 2015) and quantile regression (Engle and Manganelli, 2004). In this paper, we focus on another popular class of distributions for describing financial returns, namely the generalised hyperbolic distributions (GHDs), when estimating risk measures. Such family of distributions are particularly suitable for capturing stylised facts, such as asymmetric and varying tail behaviours (including semi-heavy and heavy tails), embedded within financial data. The novel work of Eberlein and Keller (1995) was among the first to apply these extreme value distributions to financial modelling. The successes of GHDs in modelling financial data were further advocated by various subsequent studies, such as Eberlein and Prause (2002), Aas and Haff (2006), Hu and Kercheval (2007), and Huang et al. (2014), among others.

In this paper, we first deploy a GARCH(1,1) model in describing the daily returns volatility of our chosen dataset, the MSCI All Country World Index (ACWI). Specifically, we allow the distribution of the resulting GARCH(1,1) innovations to follow different subclasses of the GHDs (namely, the hyperbolic (HYP), the normal-inverse Gaussian (NIG), the generalised hyperbolic skewed-t (GHSt) and the variance-gamma (VG) subclasses). We show that the resulting VaR estimates, using the above models, can change considerably across different window sizes on the same out-of-sample set. This challenges the common practice of utilising an arbitrary fixed window size, and motivates a need for determining optimal window sizes when estimating risk measures.

We contribute to the existing literature by proposing a method to identify the required optimal window size, and show that such a method may effectively improve the estimation of model parameters and the resulting VaR forecasts. Furthermore, we proceed with a method that follows a daily rolling window procedure to detect an optimal size for each iteration. Our findings demonstrate the importance of relaxing the usual fixed window size restriction, and allow for time-varying window sizes when forecasting VaR. To the best of the authors' knowledge, there exists no literature relating to window size optimisation in VaR estimation under the GHD framework. In addition, although prior research exists in identifying systematic breaks (or structural breaks) and the maximum period of stability (see, for example, Spokoiny, 2009; Härdle et al., 2003), very few have been applied under the GHD framework. Hence, our study also provides further insight towards the limited research on GHDs' benefits in financial risk modelling.

The remainder of this paper proceeds as follows. In Section 2, we present the VaR methodology. Discussions around the subclasses of GHDs and optimal window size derivations are provided in Sections 3 and 4, respectively. Finally, we reveal our empirical results in Section 5 and conclude the paper in Section 6.

2. Value-at-Risk

While there are criticisms on the use of VaR, it remains a popular benchmark risk measure among banks and financial institutions for evaluating and estimating financial risks. In particular, it is directly linked to the adequate amount of market risk capital that financial entities must set aside to compensate for unprecedented large losses, as recommended by the Basel Committee on Banking Supervision. Even with the ongoing migration towards the more sophisticated Expected Shortfall as a measure of risk, in accordance with Basel III, VaR continues to be widely utilised by market participants in conjunction. Hence, further research to improve the forecast of VaR may continue to bear fruit for the fragile financial sector.

Formally, VaR is defined as a threshold amount such that the probability of the realised loss on a portfolio, over a given time horizon, exceeding this value is equal to a pre-specified confidence level. For a sequence of daily log-returns, R_t, on an existing portfolio, we assume $R_t = \mu_t + \sigma_t Z_t$, where Z_t represents the innovation characterised by some marginal distribution $F_Z(z)$. The parameters μ_t and σ_t are measurable with respect to Ω_{t-1}, all information on the process up to time $t-1$. Furthermore, if $F_R(r)$ denotes the distribution of R_t, we can deduce that

$$F_{R_{t+1}|\Omega_t}(r) = P\left(\mu_{t+1} + \sigma_{t+1}Z_{t+1} \leq r \mid \Omega_t\right) = F_Z\left(\frac{r-\mu_{t+1}}{\sigma_{t+1}}\right). \tag{1}$$

Consequently, we can express VaR for day $t+1$, with probability of exceedance equal to $1-p$, as

$$\text{VaR}_p(t+1) = \mu_{t+1} + \sigma_{t+1} z_p, \tag{2}$$

where z_p denotes the lower p^{th} quantile of Z_t. For forecasting purposes, we need to first specify a model for the dynamics of the mean, μ_{t+1}, and volatility, σ_{t+1}. We utilise the celebrated GARCH(1,1) process for the volatility and the AR(1) process for the mean, i.e.,

$$\sigma^2_{t+1} = \alpha_0 + \alpha_1 \varepsilon^2_t + \beta\sigma^2_t \quad \text{and} \quad \mu_{t+1} = \phi R_t, \tag{3}$$

where $\varepsilon_t = \sigma_t Z_t, \alpha_0 > 0, \alpha_1 \geq 0, \beta \geq 0, \alpha_1 + \beta < 1$, and ϕ is the AR(1) coefficient.

Following McNeil and Frey (2000), we fit the GARCH(1,1) model using a pseudo maximum likelihood (PML) procedure, which minimises the assumptions about the distribution of innovations, and estimates μ_{t+1} and σ_{t+1} using standard one-day ahead forecasts. We further suggest this to be amalgamated with the assumption that the innovations are distributed according to a GHD subclass, and estimate the resulting z_p accordingly. This may then be implemented in a rolling window procedure to produce daily out-of-sample forecasts of VaR. Consequently, as per standard procedure, the resulting forecasts are then backtested against the realised daily returns observed. We utilised two widely-accepted backtests for VaR, namely, the Kupiec likelihood ratio test (Kupiec, 1995) and the Christoffersen conditional coverage test (Christoffersen et al., 2001). While the former tests for the unconditional coverage of the correct number of exceedances in our VaR estimates, the latter tests for the conditional coverage.

3. Generalised hyperbolic distributions (GHD)

GHDs, such as the HYP, NIG, VG and GHSt distributions, have the ability to cater for asymmetric, heavy and semi-heavy tailed datasets. They enable researchers to model data across a wide variety of disciplines, including finance and economics. By adequately capturing the above-mentioned stylised facts embedded in financial data, the resulting VaR estimates may also be greatly improved (see, for example, Huang et al., 2014). In this section, we shall introduce the full GHD and its range of subclasses.

3.1 The full GHD model

The probability density function (pdf) of the full GHD is given by

$$f_{GHD}(x) = \frac{\left(\alpha^2-\beta^2\right)^{\lambda/2}\left(\delta^2+(x-\mu)^2\right)^{(\lambda-1/2)/2}K_{\lambda-1/2}\left(\alpha\sqrt{\delta^2+(x-\mu)^2}\right)\exp(\beta(x-\mu))}{\sqrt{2\pi}\alpha^{\lambda-1/2}\delta^{\lambda}K_{\lambda}\left(\delta\sqrt{\alpha^2-\beta^2}\right)}, \quad (4)$$

where K_j is the modified Bessel function of the third kind with order j (Abramowitz and Stegun, 1972), and μ is the location parameter. It should also be noted that the domain of the parameters must satisfy the following conditions

$$\begin{aligned}\delta > 0, |\beta| < \alpha, \text{ if } \lambda = 0,\\ \delta > 0, |\beta| \leq \alpha, \text{ if } \lambda < 0,\\ \delta \geq 0, |\beta| < \alpha, \text{ if } \lambda > 0,\end{aligned}$$

where δ serves as a scaling factor, α determines the shape, β determines the skewness, and λ influences the kurtosis (Necula, 2009). We utilise the maximum likelihood estimation (MLE) for parameter estimates of all GHD subclasses. The various subclasses of the GHD can be obtained by considering different assumptions and asymptotic behaviours of the parameters above. We demonstrate this in the sequel.

3.2 The Hyperbolic (HYP) distribution

The HYP distribution (with $\lambda = 1$) allows us to determine the shape of the distribution by controlling both the gradient and skewness parameters. The HYP distribution is characterised by having a hyperbolic log-density function and exponential tails. A random variable follows the HYP distribution if its pdf is given by

$$f_{HYP}(x) = \frac{\sqrt{\alpha^2-\beta^2}}{2\alpha\delta K_1\left(\delta\sqrt{\alpha^2-\beta^2}\right)}e^{-\alpha\sqrt{\delta^2+(x-\mu)^2}+\beta(x-\mu)}, \quad (5)$$

where K_1 denotes the Bessel function of the third kind with order 1. The parameters α and β, with $\alpha > 0$ and $0 \leq |\beta| < \alpha$, represent the gradient and the skewness, respectively. Finally, $\delta > 0$ is the scale parameter and $\mu \in R$ is the location parameter.

3.3 The Normal-Inverse Gaussian (NIG) distribution

The NIG distribution is well-known for its ability to capture the asymmetric semi-heavy tails of financial returns (Andersson, 2001; Venter and de Jongh, 2002). In particular, the NIG distributions

are most appropriate when the two extreme tails of the returns distribution to be modelled are not too heavy (Aas and Haff, 2006). The pdf of the NIG, as a subclass of GHDs with $\lambda = -1/2$, can be expressed as

$$f_{NIG}(x) = \frac{\alpha\delta}{\pi} e^{\delta\sqrt{\alpha^2-\beta^2}+\beta(x-\mu)} \frac{K_1\left(\alpha\sqrt{\delta^2+(x-\mu)^2}\right)}{\sqrt{\delta^2+(x-\mu)^2}}, \tag{6}$$

where K_1 denotes the Bessel function of the third kind with order 1.

3.4 The Variance-Gamma (VG) distribution

The VG distribution has tails that decrease less rapidly than that of a Gaussian distribution. Such a characteristic makes the VG a suitable model for phenomena where extreme values are more probable than in the case of a Gaussian distribution, such as logarithmic returns from financial assets (Madan and Seneta, 1990). We attain the pdf of the VG distribution from the full GHD when $\lambda > 0$ and $\delta \to 0$. Hence, we have

$$f_{VG}(x) = \frac{\left(\alpha^2-\beta^2\right)^{\lambda} |x-\mu|^{\lambda-1/2} K_{\lambda-1/2}(\alpha|x-\mu|)}{\sqrt{\pi}\Gamma(\lambda)(2\alpha)^{\lambda-1/2}} e^{\beta(x-\mu)}, \tag{7}$$

where $K_{(\lambda-1/2)}$ denotes the Bessel function of the third kind with order $\lambda - 1/2$.

3.5 The Generalised Hyperbolic Skewed-t (GHSt) distribution

Finally, the pdf of the GHSt distribution is obtained by letting $\alpha \to |\beta|$ in the full GHD. This results in the following expression

$$f_{GHSt}(x) = \frac{2^{1/2+\lambda}\delta^{-2\lambda}|\beta|^{1/2-\lambda} K_{1/2-\lambda}\left(\sqrt{\beta^2\left(\delta^2+(x-\mu)^2\right)}\right)\exp(\beta(x-\mu))}{\Gamma(-\lambda)\sqrt{\pi}\left(\sqrt{\delta^2+(x-\mu)^2}\right)^{1/2-\lambda}}, \tag{8}$$

for $\beta \neq 0$ and $\lambda < 0$. If $\beta = 0$, we obtain the non-central (scaled) Student's t-distribution. Notably, the GHSt distribution exhibits one heavy polynomial tail and one semi-heavy exponential tail. This unique property makes the GHSt distribution particularly dissimilar to the range of subclasses mentioned above. More importantly, it allows the GHSt distribution to uniquely model skewed data with dissimilar tail behaviours, which are commonly observed in financial data (Aas and Haff, 2006).

4. Optimal window size

The choice of an appropriate window size can affect the resulting model parameter estimates, and consequently the accuracy of the final VaR forecasts. However, identifying an optimal window size remains a difficult task. In the current literature, most analyses are conducted by utilising a fixed window size that is arbitrarily chosen according to a rule-of-thumb, or is only tested against a few alternative choices in order to determine an appropriate size. The chosen window size is then used to perform a rolling window procedure to estimate VaR at each time step of the out-of-sample data. Even though such methods of window size selection are deemed reasonable by prior studies, it can produce biased parameter estimations and inadequate VaR forecasts as a result. To remedy such

drawback, a more effective procedure that can accommodate varying window sizes at each time step needs to be derived. Moreover, for robustness, the said procedure needs to optimise some criterion related to time homogeneity. In our current study, we deploy five different criteria for selecting an optimal window size at each rolling window iteration. The window sizes are chosen to either (i) minimise or maximise the standard deviation; (ii) minimise or maximise the kurtosis; or (iii) include a change-point with a fixed right-end point. While the reasons for our choice of (i) and (ii) are more apparent for risk measure focus, with the estimation of tail events, the justification for (iii) is more inline with the notion of structural breaks detection in a given dataset.

A change-point is defined as a location in the dataset in which the statistical properties of the sequence experiences a significant change. We identify change-points optimally using the at-most-one-change-point (AMOC) procedure (Silva and Teixeira, 2008) and the binary segmentation (BinSeg) procedure (Scott and Knott, 1974), with variance as the optimisation measure. The detection of a change-point can be viewed as a hypothesis test, whereby the null, H_0, corresponds to no change-point, and the alternative, H_1, advocates the existence of a change-point. The likelihood ratio method (i.e., AMOC) involves calculating the maximum likelihood under both hypotheses above. Subsequently, the ratio is maximised over all possible change-point locations. The BinSeg approach, on the other hand, is a generalisation of the AMOC, whereby the data sequence is segmented into two parts once a change-point is detected. Finally, each segment is then tested for change-points and the process continues until a pre-specified threshold is reached, or until no further change-points are identified.

5. Data and Empirical Results

In our study of optimal window size detection and the proposed varying window size approach, we use daily log-returns of the MSCI ACWI index over a 15-year period, ranging from 27 August 2001 to 25 August 2016. The MSCI ACWI is a flagship global index that aims to capture equity returns of large- and mid-cap stocks across 23 developed and 24 emerging markets. This offers investors a fully integrated view of exposure to all sources of equity returns using just a single index.

Table 1 shows the descriptive statistics of the original return series over the entire sample period, as well as the resulting innovations after fitting the GARCH(1,1) model to the same data. The large excess kurtosis of the original return series is a common characteristic found in financial data, which implies a vast tail deviation from that of the Gaussian distribution. In addition, we observe that the resulting GARCH(1,1) innovations still exhibit heavy tails, albeit to a lesser degree. These are both well-known stylised facts of financial time series (Cont, 2001).

Notably, the heavy-tails of the residuals are even more pronounced when we implement a rolling

Table 1. Summary statistics for MSCI ACWI and its GARCH innovations.

Data	Mean	Std. dev.	Min	Max	Excess kurtosis	Skewness
ACWI	−0.000130	0.010289	−0.089030	0.073713	8.172448	0.396315
Innovations	0.041017	0.999549	−3.841127	6.237750	1.311213	0.303292

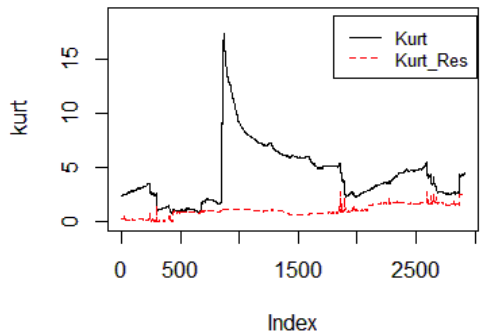


Figure 1. Rolling excess kurtosis for ACWI returns and its GARCH innovations (1000-day rolling window size).

window procedure to analyse the varying kurtosis over time. Figure 1 records the time-varying excess kurtosis of the original return series, as well as the corresponding innovations, when iterated at each time step through a 1000-day rolling window procedure. It is evident that the conditional kurtosis can deviate significantly from that of a Gaussian distribution at isolated time periods within the return series (as depicted by the sudden spikes and long periods of consistent non-zero values).

To encapsulate the effects of window size selection, we first conduct a VaR estimation procedure using a fixed window size approach. Our estimation procedure is then repeated across a range of different window sizes on the same dataset. Specifically, we will estimate the rolling window daily VaR at each time step of the out-of-sample period (from day 1501) using fixed window sizes ranging from 100 to 1500 days (at 25-day increments). For each window size, we forecast the daily VaR using the GARCH(1,1) filter with a conditional distribution following a GHD subclass. Finally, the sequence of VaR estimates are then backtested against the actual daily returns observed, and the respective p-values recorded.

Under both the Kupiec likelihood ratio and the Christoffersen conditional coverage tests, where the null hypothesis advocates for the model being 'correct' or well-specified, a higher p-value is desired. In Table 2, we present the mean, standard deviation, minimum, maximum and coefficient of variation (CV) of the different p-values obtained for both backtests across the various GHD subclasses. Interestingly, across all GHD subclasses evaluated, a range of 400-500 days appears to be the optimal choice when implementing a fixed window size. Figures 2 to 6 presents the changing p-values, for both the Kupiec and Christoffersen tests, across the range of fixed window sizes. These observations provide further empirical evidence that the performance of VaR models may depend heavily on the appropriate choice of window sizes. Apart from our explicit evidence to infer 400-500 days as an optimal range for window sizes, we observe that a larger window size tends to consistently produce inferior VaR estimates across all GHD subclasses. On the contrary, smaller window sizes, which allows more emphasis on recent market data, tends to provide more ideal VaR estimates.

To implement a varying window size selection process, within a rolling window procedure, an optimising criterion is needed to determine an adequate window size at each iteration. We shall utilise a wide range of different criteria and compare the resulting model performances through the two

Table 2. Summary statistics of VaR backtesting p-values from Kupiec and Christoffersen tests, using rolling fixed window sizes ranging from 100 to 1500 days, at the 97.5% VaR level.

GHD subclass	VaR test	Mean	Std. dev.	Min	Max	Window size for max	CV
GHD	Kupiec	0.006118	0.006207	0.000731	0.037242	400	1.014557
	Christoffersen	0.021915	0.018777	0.002515	0.107563	400	0.856805
HYP	Kupiec	0.006556	0.006174	0.000478	0.027580	400/475	0.941610
	Christoffersen	0.023240	0.019076	0.001644	0.084333	400/475	0.820814
NIG	Kupiec	0.006569	0.006794	0.000310	0.037242	400	1.034228
	Christoffersen	0.023146	0.020220	0.001061	0.107563	400	0.873588
VG	Kupiec	0.003027	0.003771	0.000198	0.027580	400	1.245834
	Christoffersen	0.011702	0.011861	0.000676	0.084333	400	1.013565
GHSt	Kupiec	0.005125	0.005726	0.000478	0.037242	400	1.117432
	Christoffersen	0.018682	0.017402	0.001644	0.107563	400	0.931458

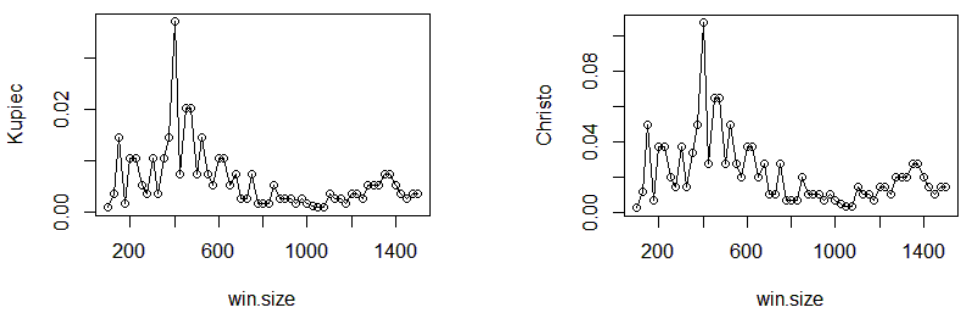


Figure 2. p-values of Kupiec and Christoffersen tests for rolling fixed window sizes ranging from 100 to 1500 days when using GHD.

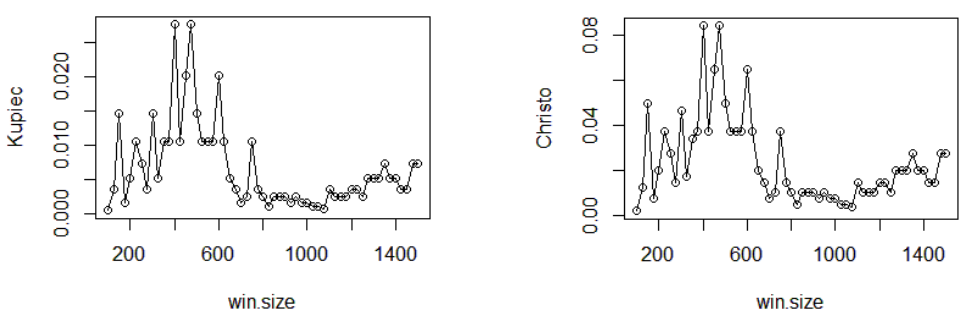


Figure 3. p-values of Kupiec and Christoffersen tests for rolling fixed window sizes ranging from 100 to 1500 days when using HYP.

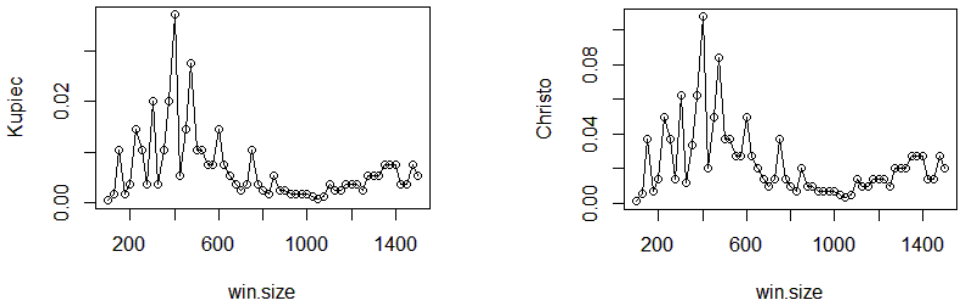


Figure 4. p-values of Kupiec and Christoffersen tests for rolling fixed window sizes ranging from 100 to 1500 days when using NIG.

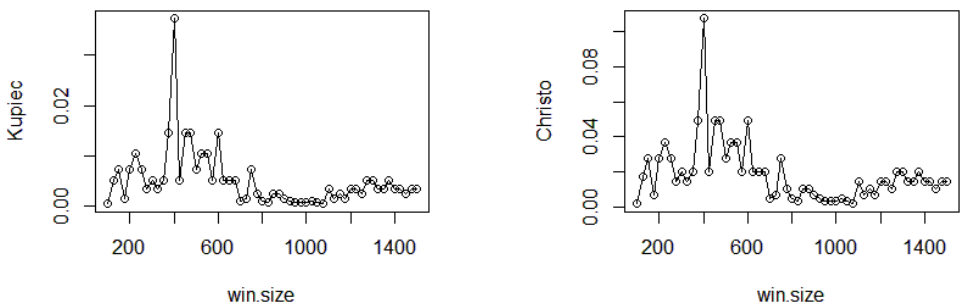


Figure 5. p-values of Kupiec and Christoffersen tests for rolling fixed window sizes ranging from 100 to 1500 days when using VG.

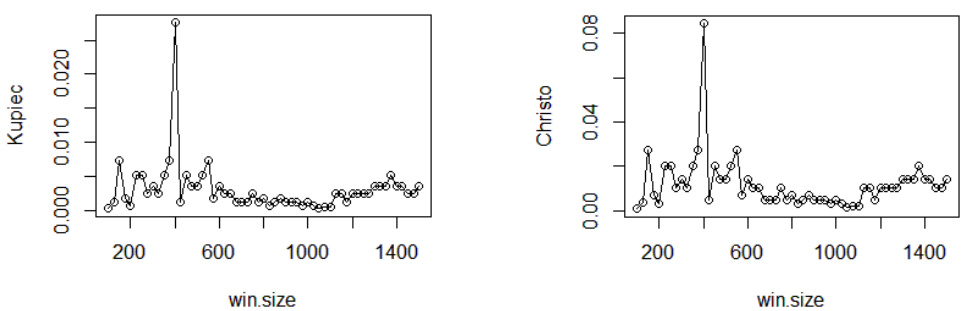


Figure 6. p-values of Kupiec and Christoffersen tests for rolling fixed window sizes ranging from 100 to 1500 days when using GHSt.

Table 3. VaR backtesting p-values from Kupiec and Christoffersen tests under varying window sizes across different optimising criteria.

GHD subclass	VaR test	Max. std. dev.	Min. std. dev.	Max Kurtosis	Min Kurtosis	Change-point AMOC	Change-point BinSeg
GHD	Kupiec	0.176607	< 0.000001	0.002444	0.005169	0.020189	0.027580
	Christoffersen	0.332875	0.000004	0.010141	0.019968	0.065124	0.084333
HYP	Kupiec	0.176607	< 0.000001	0.002444	0.005169	0.020189	0.010451
	Christoffersen	0.332875	0.000001	0.010140	0.019968	0.065124	0.037132
NIG	Kupiec	0.140315	< 0.000001	0.003574	0.003574	0.010451	0.014609
	Christoffersen	0.287102	0.000001	0.014331	0.014331	0.037132	0.049541
VG	Kupiec	0.176607	< 0.000001	0.001652	0.005169	0.020189	0.014609
	Christoffersen	0.332875	< 0.000001	0.007076	0.019968	0.065124	0.049541
GHSt	Kupiec	0.140315	< 0.000001	0.001652	0.002444	0.010451	0.014609
	Christoffersen	0.287102	< 0.000001	0.007076	0.010141	0.037132	0.049541

standard backtests. Firstly, we select an optimal window size for each iteration of the rolling window according to a maximum standard deviation, minimum standard deviation, maximum kurtosis and minimum kurtosis, over the different window sizes ranging from 100 to 1500 days (at 25-day increments). Secondly, we utilise two change-points procedures, namely, AMOC and BinSeg, for identifying the change-point(s) within each rolling window (with the base window size set to 1500 days). For AMOC, the period between the change-point and the most right-end point in a given rolling window is selected. For BinSeg, the period between the largest change-point and the most right-end point of the rolling window is selected instead. Finally, the different VaR estimates are obtained through the various optimal window sizes detected per criteria and backtested accordingly.

Table 3 presents the Kupiec test and Christoffersen test p-values for the varying window size procedure using the different optimisation methods mentioned above. The minimum standard deviation, maximum kurtosis and minimum kurtosis appears to be inadequate as optimising criteria, each exhibiting poorer results in comparison to the average performance of the alternative fixed window size approach (as shown in Table 2). Surprisingly, while producing superior results to that of the above-mentioned trio, the two change-point procedures seem to be marginally better or on par with the average performance of using fixed window sizes (at a 5% confidence level). However, both AMOC and BinSeg are still less robust than using an optimal fixed window of 400 days (when such a window size may be determined a priori). The selection of varying window sizes through maximum standard deviation overwhelmingly outperforms the alternative criteria, as well as the fixed window approach. It also consistently produces the highest p-values among all criteria across the various GHD subclasses. Overall, our results also demonstrate that a GARCH(1,1) with a conditional distribution of either the GHD, HYP or VG is the most robust model for forecasting VaR in MSCI ACWI returns.

Figure 7 shows the changing window sizes at each rolling window iteration for the maximum standard deviation, AMOC and BinSeg optimising criteria. Notably, the changes in optimal varying

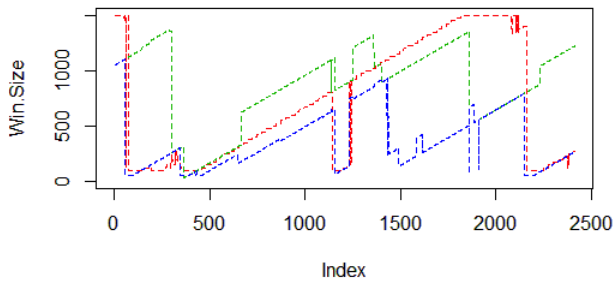


Figure 7. Varying window sizes over time using various selection criteria (red = max standard deviation, green=AMOC, blue=BinSeg).

window sizes over time also reflect the market conditions that ensued. For instance, the recommended optimal window size contracts sharply during periods of market distress, which adequately allows rolling windows to capture more recent data that better represents the prevailing market downturn. This is clearly exemplified by the infamous 2008 Global Financial Crisis, the subsequent Eurozone crisis, as well as the 2015-2016 selloff triggered by the Chinese stock market turbulence (as depicted in Figure 7). While both the maximum standard deviation and BinSeg criteria are efficient in following market trends, the AMOC, with the restriction of at most one change-point detection, tend to suffer from excessive lags in its response.

6. Limitations and concluding remarks

In this paper, we examine how the accuracy of VaR estimates, under conditional GHD subclasses, may vary depending on the choice of an appropriate window size when estimating model parameters. Forecasting performances were measured according to the widely-accepted Kupiec likelihood ratio and the Christoffersen conditional coverage tests. Evidently, our analyses showed that the robustness of VaR models rely heavily on the appropriate selection of window sizes for parameter estimation. In order to identify an optimal window size (for each rolling window iteration), we investigated several possible optimisation methods to enable a time-varying window size procedure, and compared our results to that of the classical fixed window implementation. The optimising criteria employed to select a suitable varying window size were given by either maximising or minimising the standard deviation, maximising or minimising the kurtosis, using an AMOC procedure, or using a BinSeg procedure. It is worthwhile noting that the AMOC and BinSeg procedures appeared to be only as good as the average performance of the fixed window size approach, and worse off when an optimal fixed window size is utilised. Maximising the standard deviation under the varying window size approach seemed to produce the best risk forecasting results under the GHD framework. Our findings advocate the critical need to optimise window sizes prior to parameter estimation when forecasting VaR. Moreover, it is necessary to relax the usual restriction of a fixed window size, and allow for time-varying window sizes instead. Lastly, it is necessary to evaluate a range of optimising criteria in order to identify the most appropriate criterion to deploy.

An important caveat to our study is the limited number of criteria investigated for optimal window

size selection. Further research may include implementing other attractive methods, such as the adaptive pointwise estimation (see Čížek et al., 2009) or segment neighbourhood procedure for change-point identification (see Auger and Lawrence, 1989), and analysing the accuracy of resulting VaR forecasts. Additionally, it may be worthwhile to explore whether the suitability of selection criteria, or procedure, may change significantly under different market conditions (when certain stylised facts may become extreme), or when different distributional assumptions for the data series are implemented. Finally, with the recommended migration towards Expected Shortfall (as per the latest Basel Accords), further studies of optimal window sizes detection to improve ES estimation is paramount.

References

Aas, K. and Haff, I. H. (2006). The generalized hyperbolic skew Student's t-distribution. *Journal of Financial Econometrics*, **4**, 275–309.

Abramowitz, M. and Stegun, I. A. (1972). *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables. National Bureau of Standards Applied Mathematics Series 55. Tenth Printing*. ERIC.

Andersson, J. (2001). On the normal inverse Gaussian stochastic volatility model. *Journal of Business & Economic Statistics*, **19**, 44–54.

Auger, I. E. and Lawrence, C. E. (1989). Algorithms for the optimal identification of segment neighborhoods. *Bulletin of Mathematical Biology*, **51**, 39–54.

Chen, Y. and Spokoiny, V. (2009). Modeling and estimation for nonstationary time series with applications to robust risk management.
URL: *https://www.academia.edu/22109590*

Christoffersen, P., Hahn, J., and Inoue, A. (2001). Testing and comparing value-at-risk measures. *Journal of Empirical Finance*, **8**, 325–342.

Čížek, P., Härdle, W., and Spokoiny, V. (2009). Adaptive pointwise estimation in time-inhomogeneous conditional heteroscedasticity models. *The Econometrics Journal*, **12**, 248–271.

Cont, R. (2001). Empirical properties of asset returns: stylized facts and statistical issues. *Quantitative Finance*, **1**, 223.

Corlu, C. G. and Corlu, A. (2015). Modelling exchange rate returns: which flexible distribution to use? *Quantitative Finance*, **15**, 1851–1864.

Eberlein, E. and Keller, U. (1995). Hyperbolic distributions in finance. *Bernoulli*, **1**, 281–299.

Eberlein, E. and Prause, K. (2002). The generalized hyperbolic model: financial derivatives and risk measures. *In Mathematical Finance—Bachelier Congress 2000*. Springer, 245–267.

Engle, R. F. and Manganelli, S. (2004). CAViaR: Conditional autoregressive value at risk by regression quantiles. *Journal of Business & Economic Statistics*, **22**, 367–381.

Halbleib, R. and Pohlmeier, W. (2012). Improving the value at risk forecasts: Theory and evidence from the financial crisis. *Journal of Economic Dynamics and Control*, **36**, 1212–1228.

Härdle, W., Herwartz, H., and Spokoiny, V. (2003). Time inhomogeneous multiple volatility modeling. *Journal of Financial Econometrics*, **1**, 55–95.

Hu, W. and Kercheval, A. (2007). Risk management with generalized hyperbolic distributions. *In Proceedings of the fourth IASTED international conference on financial engineering and applications*. ACTA Press, 19–24.

Huang, C.-S., Huang, C.-K., and Chinhamu, K. (2014). Assessing the relative performance of heavy-tailed distributions: Empirical evidence from the Johannesburg Stock Exchange. *Journal of Applied Business Research*, **30**, 1263–1286.

Kupiec, P. (1995). Techniques for verifying the accuracy of risk measurement models. *Journal of Derivatives*, **3**, 73–84.

Laker, I., Huang, C.-K., and Clark, A. E. (2017). Dependent bootstrapping for value-at-risk and expected shortfall. *Risk Management*, **19**, 301–322.

Madan, D. B. and Seneta, E. (1990). The variance gamma (VG) model for share market returns. *Journal of Business*, **63**, 511–524.

McNeil, A. J. and Frey, R. (2000). Estimation of tail-related risk measures for heteroscedastic financial time series: an extreme value approach. *Journal of Empirical Finance*, **7**, 271–300.

Necula, C. (2009). Modeling heavy-tailed stock index returns using the generalized hyperbolic distribution. *Romanian Journal of Economic Forecasting*, **10**, 118–131.

Scott, A. J. and Knott, M. (1974). A cluster analysis method for grouping means in the analysis of variance. *Biometrics*, **30**, 507–512.

Sharma, M. (2012). Evaluation of Basel III revision of quantitative standards for implementation of internal models for market risk. *IIMB Management Review*, **24**, 234–244.

Silva, E. G. and Teixeira, A. A. (2008). Surveying structural change: Seminal contributions and a bibliometric account. *Structural Change and Economic Dynamics*, **19**, 273–300.

Spokoiny, V. (2009). Multiscale local change point detection with applications to value-at-risk. *The Annals of Statistics*, **37**, 1405–1436.

Venter, J. H. and de Jongh, P. J. (2002). Risk estimation using the normal inverse Gaussian distribution. *Journal of Risk*, **4**, 1–24.

Proceedings of the 63rd Annual Conference of the South African Statistical Association

An investigation on the use of Bernstein polynomials in entropy estimation

Shawn C. Liebenberg

North-West University

Entropy estimation has become an important component in many fields of research. Among the many developed procedures for estimating entropy, spacing and kernel density based procedures have become the most prominent. Kernel density estimation is plagued by boundary bias that potentially carry over to the corresponding entropy estimators. This study introduces two new Bernstein based entropy estimators and aims to investigate Bernstein polynomial density estimation in entropy estimation as a remedy to the boundary bias problem. It was found that the Bernstein based entropy estimators performed very well against the spacing and kernel density estimators used in this study.

Keywords: Bernstein polynomials, Bias reduction, Density estimation, Entropy estimation.

1. Introduction

Entropy is generally thought of as a measure of uncertainty in the outcome of a random process. Since the introduction of entropy by Shannon (1948), it has become a building block of information theory. It has further found use in areas such as the study of languages (see Ratnaparkhi, 1997), Biology (see Adami, 2004), economics (see Maasoumi and Racine, 2002) and astronomy (Cincotta et al., 1999). From a data analysis point of view, the estimation of entropy is of increasing importance. An abundance of literature exists on this topic ranging from the nonparametric estimation of differential entropy of a continuous random variable (see Beirlant et al., 1997) to more generalised entropy measures used in data analysis (see Pompe, 1994). In this paper, our interest lies in the estimation of the entropy for a random variable X and specifically in estimating the entropy with the use of Bernstein polynomial density estimators. To formally proceed, let $X_1, X_2, \ldots X_n$ be independent, identically distributed (i.i.d.) valued random variables of size n with absolutely continuous density function $f(x)$ and distribution function $F(x)$. Furthermore, let $X_{(1)}, X_{(2)}, \ldots X_{(n)}$ be order statistics such that $X_{(1)} < X_{(2)} < \cdots < X_{(n)}$. The entropy of the random variable X is then defined as

$$\mathcal{H}_f = \mathbb{E}\left[-\log f(x)\right] = -\int_{-\infty}^{\infty} f(x)\log f(x)\mathrm{d}x. \tag{1}$$

Vasicek (1976) considered the nonparametric estimation of this quantity by noting that (1) can be expressed in the form

$$\mathcal{H}_f = \int_0^1 \log\left\{\frac{\mathrm{d}}{\mathrm{d}p}F^{-1}(p)\right\}\mathrm{d}p = \int_0^1 \log\left\{\frac{1}{f(F^{-1}(p))}\right\}\mathrm{d}p. \tag{2}$$

Corresponding author: Shawn C. Liebenberg (shawn.liebenberg@nwu.ac.za)

MSC2020 subject classifications: 94A17, 62G05, 62G07

Consequently, by replacing the distribution function $F(x)$ by its empirical counterpart $F_n(x) = n^{-1}\sum_{i=1}^{n} I(X_i \leq x)$ and using a difference operator rather than the differential operator, the slope can be estimated by $\mathrm{d}/\mathrm{d}p \;\; F^{-1}(p) = (n/2r)\{X_{(i+r)} - X_{(i-r)}\}$. This ultimately leads to the entropy estimator

$$H_{n,r}^{V} = H_r\left(X_1, X_2, \ldots, X_n\right) = \frac{1}{n}\sum_{i=1}^{n} \log\left\{\frac{n}{2r}\left(X_{(i+r)} - X_{(i-r)}\right)\right\}, \tag{3}$$

where r is a positive, integer valued window-width such that $r \leq n/2$, and $X_{(i)} = X_{(1)}$, if $i < 1$ and $X_{(i)} = X_{(n)}$, if $i > n$. Since the inception of the Vasicek entropy estimator, many adjusted or modified versions have been suggested. Van Es (1992) proposed another spacing based entropy estimator derived from estimation of the functionals of the density

$$H_{n,r}^{VE} = \frac{1}{n-r}\sum_{j=1}^{n-r} \log\left\{\frac{n+1}{r}\left(X_{(j+r)} - X_{(j)}\right)\right\} + \sum_{k=r}^{n}\frac{1}{k} + \log(r) - \log(n+1).$$

Ebrahimi et al. (1994) proposed a weighted version of Vasicek's estimator that take the truncation around the smallest and the largest data points into account:

$$H_{n,r}^{E} = \frac{1}{n}\sum_{i=1}^{n} \log\left\{\frac{n}{c_i r}\left(X_{(i+r)} - X_{(i-r)}\right)\right\},$$

where

$$c_i = \begin{cases} 1 + \frac{i-1}{r}, & 1 \leq i \leq r, \\ 2, & r+1 \leq i \leq n-r, \\ 1 + \frac{n-i}{r}, & n-r+1 \leq i \leq n. \end{cases}$$

Furthermore, Correa (1995) proposed a modification of the Vasicek estimator by using a local linear model approach in the interval $(X_{(i+r)}, X_{(i-r)})$, and Zamanzade and Arghami (2012) noted that the Ebrahimi estimator underestimates the entropy for small sample sizes and proposed a further modification. In addition, entropy estimates based on kernel density estimation were introduced by Dmitriev and Tarasenko (1974) and later studied by Ahmad and Lin (1976), Hall and Morton (1993), and Bouzebda and Elhattab (2014). However, it is well known that kernel density estimation has difficulty estimating the boundaries of a density. This is especially evident when dealing with non-negative variable support in which case the kernel density estimate introduces large bias at the boundaries. The bias is caused by the kernel density estimate giving weight to the area of the bandwidth that falls outside of the data range (see Wand and Jones, 1994). This, in turn, could affect the entropy estimation based on the kernel density estimate.

Many statistical procedures have been developed that make use of entropy and consequently, entropy estimators. For example, Robinson (1991) constructed an entropy based test for independence in time series, Dudewicz et al. (1995) used an entropy based random number evaluation technique in an effort to assess commonly used random number generators, Mudholkar and Tian (2002) presented an entropy characterisation of the inverse Gaussian distribution and a subsequent goodness-of-fit test, and Park and Park (2003) derived a piece-wise uniform distribution function of the sample entropy and developed goodness-of-fit tests based on this nonparametric distribution functions.

The rest of the paper is outlined as follows. Section 2 will investigate the different Bernstein polynomial entropy estimators and will present a theoretical motivation based on reduction of boundary

bias for the use of these estimators. In Section 3, the finite sample performance of different entropy estimators are investigated and compared to the Bernstein based estimators. Finally, Section 4 presents the conclusion of the study and provides some avenues for future research.

2. Entropy estimators based on Bernstein polynomials

In this section the entropy estimator of Vasicek (1976) is discussed, followed by the kernel density based entropy estimator of Ahmad and Lin (1976). Finally, existing Bernstein polynomial density estimates are given and eventually two new entropy estimates are constructed.

As starting point, Vasicek (1976) remarks that the estimator proposed in (3) can be rewritten as the three-part summation

$$\begin{aligned} H_{n,r}^{V} := & -n^{-1}\sum_{i=1}^{n}\log f\left(X_i\right)+n^{-1}\sum_{i=1}^{n}\log\left[\frac{F\left(X_{(i+r)}\right)-F\left(X_{(i-r)}\right)}{f\left(X_{(i)}\right)\left\{X_{(i+r)}-X_{(i-r)}\right\}}\right] \\ & +n^{-1}\sum_{i=1}^{n}\log\left[\frac{n}{2r}\left\{F\left(X_{(i+r)}\right)-F\left(X_{(i-r)}\right)\right\}\right], \end{aligned} \tag{4}$$

where the first term represents the sample mean estimate of $\mathcal{H}_f$ as defined in (1), the second term represents the error brought on by estimation of f by finite differences and the third term represents the error brought on by estimating increments of the distribution F by increments of the empirical cumulative distribution function, F_n.

If it were possible to eliminate the error introduced by term two and three in (4), the only term left to estimate would be $n^{-1}\sum_{i=1}^{n}\log f\left(X_i\right)$, which turns out to be the direct sample estimate of the expectation in (1) with $f(x)$ replaced by an appropriate empirical counterpart, $\widehat{f_n}(x)$. Indeed, Ahmad and Lin (1976) considered the kernel density estimate of Parzen (1962) and Rosenblatt (1956) for $\widehat{f_n}(x)$ which lead to the entropy estimator

$$\widehat{H}_{n,h}^{K}=-\frac{1}{n}\sum_{i=1}^{n}\log\widehat{f_h}(X_i), \tag{5}$$

where $\widehat{f_h}(x)=n^{-1}\sum_{i=1}^{n}k_h\left(x-X_i\right)$, $k_h(u)=h^{-1}k(u/h)$ and h is the bandwidth. This estimator exhibits desirable properties but potentially suffers from the same boundary bias complications inherent in kernel density estimation. A natural progression therefore, would be to use an estimator $\widehat{f_n}(x)$ which reduces the boundary bias. In point of fact, a large variety of boundary bias reduction techniques for kernel density estimation have been developed. See instances such as Schuster (1985), Jones (1993) and Dai and Sperlich (2010). However, an interesting approach is density estimators that are free of boundary bias such as Bernstein polynomial based density estimators.

The Bernstein polynomial density estimator was introduced by Vitale (1975) and further investigated, for example, by Babu et al. (2002) and Kakizawa (2004). The estimator is given by

$$\widehat{f_{n,m}}(x)=m\sum_{j=0}^{m-1}\left[F_n\left(\frac{j+1}{m}\right)-F_n\left(\frac{j}{m}\right)\right]B_{j,m-1}(x), \tag{6}$$

where F_n is the empirical cumulative distribution function, $B_{j,m}(x)=\binom{m}{j}x^j(1-x)^{m-j}$, $m\in\mathbb{N}$, are binomial probabilities, and it is assumed that the underlying density f has compact support (throughout

taken to be [0,1]). It was shown by Leblanc (2010) that the bias of the Bernstein polynomial density estimator is given by

$$\mathbb{B}\left[\widehat{f}_{n,m}(x)\right] = \frac{\frac{1}{2}\left[(1-2x)f'(x) + x(1-x)f''(x)\right]}{m} + o\left(m^{-1}\right), \quad x \in [0,1],$$

and the variance by

$$\mathbb{V}\left[\hat{f}_{n,m}(x)\right] = \begin{cases} \frac{m^{1/2}}{n} f(x)[4\pi x(1-x)]^{-1/2} + o\left(\frac{m^{1/2}}{n}\right), & \text{for } x \in (0,1) \\ \frac{m}{n} f(x) + o\left(\frac{m}{n}\right), & \text{for } x = 0, 1, \end{cases}$$

if $m, n \to \infty$ such that $m/n \to 0$. Taking the relation to the bandwidth as $m = 1/h$ (see, Leblanc, 2010), it can be ascertained from the bias result that the density estimator has uniform bias throughout its support and therefore does not have the typical boundary bias problem found in kernel density estimators. However, this may come at the cost of higher bias than its kernel density counterpart in general. In fact, Leblanc (2010) states that the bias is an increase from $O(h^2)$ to $O(h)$. On the other hand, the variance is of order $O(1/nh^{1/2})$ compared to the kernel estimation's $O\left(1/nh\right)$. Ultimately, the Bernstein estimator is shown by Leblanc (2010) to have a smaller mean integrated square error (MISE) than the kernel density estimator. The aforementioned reduction in MISE and the absence of the boundary bias problem should result in an improved entropy estimate when the Bernstein estimator is used. To introduce the new Bernstein based entropy estimator, we write

$$\widehat{H}^{B}_{n,m} = -\frac{1}{n}\sum_{i=1}^{n} \log \widehat{f}_{n,m}(X_i),$$

where $\widehat{f}_{n,m}(x)$ is defined in (6). In an extension, bias reduced Bernstein estimators exist. An additive Bernstein estimator was introduced by Leblanc (2010) and a multiplicitive estimator by Igarashi and Kakizawa (2014). The discussion here will be restricted to the additive bias reduction case. The additive bias reduced estimator takes the simple form

$$\widehat{f}_{n,m,M}(x) = \frac{m}{m-M}\widehat{f}_{n,m}(x) - \frac{M}{m-M}\widehat{f}_{n,M}(x), \quad m > M.$$

This density estimator once again has uniform bias throughout its support and therefore is free of boundary bias. Moreover, it improves the bias to $O(h^2)$ which is on par with that of the kernel density estimator while still maintaining a variance of $O(1/nh^{1/2})$. The corresponding entropy estimator can then be written as,

$$\widehat{H}^{C}_{n,m,M} = -\frac{1}{n}\sum_{i=1}^{n} \log \widehat{f}_{n,m,M}(X_i).$$

Another approach using Bernstein estimators was given in Chaubey and Vu (2021) and is based on a quantile density estimator using Bernstein polynomials. That is, using a sample version of the formulation in (2), the quantile density $(\mathrm{d}/\mathrm{d}p \;\; F^{-1}(p) = \mathrm{d}/\mathrm{d}p \;\; Q(p))$ is estimated by

$$\widehat{q}_{n,m}(p) = \frac{\mathrm{d}}{\mathrm{d}p}\widehat{Q}_{n,m}(p) = \sum_{j=1}^{m} X_{(j)} \frac{j-mp}{p(1-p)} B_{j,m}(p), \quad p \in (0,1),$$

and $B_{j,m}(p)$ are again binomial probabilities. Interestingly, the estimator $\widehat{q}_{m,n}(p)$ is a special case of the generalised estimator developed by Cheng and Parzen (1997). The entropy estimator based on the quantile density estimator can be defined as

$$\widehat{H}^{Q}_{n,m} = \int_0^1 \log \widehat{q}_{n,m}(p)\, \mathrm{d}p.$$

Moreover, following the framework for the generalised estimator, a framework for an entropy estimator based on the quantile density function can be formulated as

$$\widehat{H}^{G}_{n} = \int_0^1 \log \widehat{q}^{G}_{n}(p)\mathrm{d}p,$$

where

$$\widehat{q}^{G}_{n}(u) = \mathrm{d}/\mathrm{d}u\ \widehat{Q}^{G}_{n}(u) = \mathrm{d}/\mathrm{d}u \int_0^1 \tilde{Q}_n(t)\mathrm{d}_t K_n(u,t),$$

$\tilde{Q}_n(t)$ is a natural estimator of the quantile function, and $K_n(u,t)$ is a cumulative distribution function on [0,1]. However, Cheng and Parzen (1997) elucidate certain aspects on the choice of kernel and the preservation of monotonicity for the practical implementation of the generalised estimator. It is accepted by Theorem 3.1 in Cheng and Parzen (1997) that similar considerations do not carry over to the Bernstein estimator, $\widehat{q}_{n,m}(p)$, when $m \geq 3$.

2.1 Practical considerations

In this section a few practical considerations for the implementation of the entropy estimators are discussed. Special attention is given to the choice of tuning parameter for the different entropy estimators as it influences the performance of the estimator:

- Estimators using spacings ($\widehat{H}^{V}_{n,r}$, $\widehat{H}^{VE}_{n,r}$ and $\widehat{H}^{E}_{n,r}$): The choice of the window-width, r, in these estimators is still an open problem, but some simple choices do exist. The most widely used methods are either the heuristic formula of Crzcgorzewski and Wirczorkowski (1999) given by $r = [\sqrt{n} + 0.5]$ (rounded to the nearest integer) or a grid search approach.
- Estimator using kernel density estimators ($\widehat{H}^{K}_{n,h}$): The choice of bandwidth, h, is critical for these estimators and have been extensively studied in literature. In particular, commonly used choices of the bandwidth include the choice based on cross-validation, which has been shown to produce accurate estimators (see Heidenreich et al., 2013), as well as the bandwidth choice of Silverman (2018).
- Estimators using Bernstein polynomial density estimates ($\widehat{H}^{B}_{n,m}$, $\widehat{H}^{Q}_{n,m}$, $\widehat{H}^{C}_{n,m,M}$): The polynomial order m of the Bernstein density based entropy estimators can be chosen through the cross-validation method discussed in Leblanc (2010) (see Kakizawa, 2004). The bias reduced estimator has an extra parameter M which is taken as $M = m/2$ and leads to improved mean integrated square error results, see Leblanc (2010). A guideline for the range of values that m can assume is $2 \leq m \leq n/\log n$, as motivated in Babu et al. (2002).

The bias reduced estimator, $\widehat{f}_{n,m,M}(x)$, always integrates to one, but can take on negative values which prove problematic in the entropy estimator. To correct for this, a slight modification of the approach of Glad et al. (2003) can be implemented. Instead of using $\tilde{f}_n(x) = \max(0, \widehat{f}_n(x))$, the modification $\tilde{f}_n(x) = \max(\epsilon, \widehat{f}_n(x))$ is suggested where $\epsilon > 0$ is a value arbitrarily close to zero.

3. Simulations and results

This section will investigate the finite sample performance of the Bernstein based entropy estimators against existing sample entropy estimators. In this limited study, seven different distributions are used and the simulation setup follows similarly to that of Chaubey and Vu (2021). The selected distributions have support $(0, \infty)$ and include the standard exponential distribution, $E(1)$ ($\mathcal{H}_f = 1$), the lognormal distribution, $LN(0, .5)$ ($\mathcal{H}_f = 0.7258$), the gamma distribution, $G(2, 2)$ ($\mathcal{H}_f = 0.8841$), the Weibull distribution, $W(2, 2)$ ($\mathcal{H}_f = 1.2886$), and support $[0, 1]$ for the beta distribution, $B(2, 2)$ ($\mathcal{H}_f = -0.1251$), $B(.5, .5)$ ($\mathcal{H}_f = -0.2416$), as well as the standard uniform distribution, $U(0, 1)$ ($\mathcal{H}_f = 0$). For each distribution, 1000 samples were generated. Since the Bernstein based estimators require the distribution to have support on $[0, 1]$, the transformation $Y_i = (X_i - X_{(1)})/(X_{(n)} - X_{(1)})$ was applied to each sample. For each transformed sample each of the Bernstein polynomial based entropy estimators, $\widehat{H}^B_{n,m}$, $\widehat{H}^Q_{n,m}$ and $\widehat{H}^C_{n,m,M}$ were calculated, as well as the spacing based estimators of Vasicek (1976), $\widehat{H}^V_{n,r}$, the Van Es (1992) estimator, $\widehat{H}^{VE}_{n,r}$, and the Ebrahimi et al. (1994) estimator, $\widehat{H}^E_{n,r}$. The estimator of Ahmad and Lin (1976), $\widehat{H}^K_{n,h}$, stated in (5), was also included as the kernel density based estimator. Since the samples were transformed, an adjustment of $\log(X_{(n)} - X_{(1)})$ was made to correct the resultant entropy estimate values. In order to compare the estimators, the variance, bias and root mean square error (RMSE) were calculated. This is repeated for sample sizes $n = 10$, $n = 50$, $n = 100$ and can be found in Tables 1, 2 and 3, respectively. For convenience, the smallest bias, variance and RMSE values are highlighted. The window-width value for the spacing estimators was obtained from the heuristic formula stated in Section 2.1. For the estimator $\widehat{H}^K_{n,h}$, the bandwidth was chosen with least-squares cross-validation and denoted h^*. In an effort to use the same order m across the Bernstein density based estimators, $\widehat{H}^B_{n,m}$ and $\widehat{H}^C_{n,m,M}$, the value m that exhibited acceptable results was chosen from a preliminary study based on each sample size. The Bernstein quantile density estimator, $\widehat{H}^Q_{n,m}$, had the imposed restriction $m \geq 3$ and so m was selected with the suggested value of $m = \lfloor n/\log(n) \rfloor$ where $\lfloor z \rfloor$ denotes the greatest integer smaller or equal to z (see, Chaubey and Vu (2021)). It should be noted that preliminary studies showed that the estimation improved in terms of the RMSE when a grid search method was used for the order m in all cases. All calculations and simulation were performed using the statistical computing environment R (R Core Team, 2021) with the bde package (Santafe et al., 2015).

From the output presented in Tables 1 to 3, it is evident that there is no estimator that outright performed the best in terms of bias, variance or RMSE. Considering the spacing estimators in isolation, it is clear that the Ebrahimi et al. (1994) estimator, $\widehat{H}^E_{n,r}$, displayed the lowest RMSE values for this class of estimators using the given choice of window-width. Turning attention to the Bernstein based estimators, it is found that new estimators, $\widehat{H}^B_{n,m}$ and $\widehat{H}^C_{n,m}$, generally performed better than the quantile density estimator $\widehat{H}^Q_{n,m}$. The $\widehat{H}^C_{n,m}$ estimator often exhibited the smallest bias, whereas $\widehat{H}^Q_{n,m}$ in many cases had the smallest variance. The $\widehat{H}^B_{n,m}$ estimator followed closely both in bias and variance performance. The Bernstein based estimators generally outperformed the kernel density based estimator, $\widehat{H}^K_{n,h}$, in terms of the RMSE for large sample sizes ($n = 50$ and $n = 100$). The kernel density based estimator performed better for $n = 10$, but still fell short of $\widehat{H}^B_{n,m}$ in many cases. Overall, the Bernstein based estimators tended to outperform the spacing based estimators and the kernel density based estimator for bias, variance and RMSE. Furthermore, it is noted that the Ebrahimi et al. (1994) estimator was a close competitor and even performed the best in some cases. The $\widehat{H}^E_{n,r}$

estimator especially showed good performance in terms of the variance. For small sample sizes ($n = 10$), the best performing test was $\widehat{H}^B_{n,m}$. This estimator even outperformed its bias-corrected counterpart $\widehat{H}^C_{n,m}$ which possibly can be ascribed to the better performance in variance of $\widehat{H}^B_{n,m}$ that lead to a better RMSE. For large sample sizes ($n = 100$), the best performing estimators was a mix of $\widehat{H}^B_{n,m}$, $\widehat{H}^C_{n,m}$ and $\widehat{H}^E_{n,r}$.

In an attempt at a more in-depth look at the kernel density based estimator compared to the Bernstein based estimators, the RMSE was calculated for samples $n \in \{10, 20, 30, 40, 50, 60, 70, 80, 90, 100\}$ and plotted in Figure 1 for the $E(1)$ and $G(2, 2)$ distributions. In the two cases, it can be concluded that the $\widehat{H}^B_{n,m}$ and $\widehat{H}^C_{n,m}$ estimators (red solid line and green dotted line) consistently exhibited smaller RMSE values than the kernel density based estimator, $\widehat{H}^K_{n,h}$ (purple dash-dotted line), as the sample size increases.

Table 1. Bias, variance and RMSE results of the entropy estimators ($n = 10$).

		$\widehat{H}^V_{n,4}$	$\widehat{H}^{VE}_{n,4}$	$\widehat{H}^E_{n,4}$	$\widehat{H}^K_{n,h^*}$	$\widehat{H}^B_{n,2}$	$\widehat{H}^Q_{n,4}$	$\widehat{H}^C_{n,2,1}$
E(1)	bias	0.4513	0.1221	0.1345	0.1647	**0.0470**	0.3325	0.1583
	variance	**0.1443**	0.1510	**0.1443**	0.2188	0.1647	0.1532	0.2538
	RMSE	0.5900	0.5960	**0.4030**	0.4960	0.4090	0.5140	0.5280
LN(0,.5)	bias	0.5859	0.2532	0.2690	0.2075	0.2086	0.4137	**0.0173**
	variance	**0.1065**	0.1079	**0.1065**	0.1413	0.1171	0.1166	0.2063
	RMSE	0.6710	0.6720	0.4230	0.4290	**0.4010**	0.5360	0.4550
G(2,2)	bias	0.5465	0.1995	0.2297	0.1810	0.1714	0.3864	**0.0165**
	variance	**0.0958**	0.1029	**0.0958**	0.1440	0.1034	0.1051	0.1837
	RMSE	0.6280	0.6340	0.3850	0.4200	**0.3640**	0.5040	0.4290
W(2,2)	bias	0.5867	0.2128	0.2699	0.1704	0.2263	0.4139	**0.0966**
	variance	**0.0668**	0.0778	**0.0668**	0.1120	0.0706	0.0800	0.1203
	RMSE	0.6410	0.6500	0.3740	0.3760	**0.3490**	0.5010	0.3600
B(2,2)	bias	0.5570	0.1401	0.2401	0.1149	0.0989	0.4020	**0.0977**
	variance	0.0412	0.0599	0.0412	0.0855	**0.0019**	0.0513	0.0082
	RMSE	0.5930	0.6080	0.3140	0.3140	**0.1080**	0.4610	0.1330
B(.5,.5)	bias	**0.0802**	0.4247	0.2366	0.1719	0.2023	0.0940	0.2398
	variance	0.0217	0.0532	0.0217	0.1848	**0.0039**	0.0307	0.0219
	RMSE	**0.1680**	0.2440	0.2790	0.4630	0.2120	0.1990	0.2820
U(0,1)	bias	0.4585	**0.0011**	0.1416	0.0341	0.0345	0.3508	0.0134
	variance	0.0278	0.0509	0.0278	0.0773	**0.0030**	0.0316	0.0189
	RMSE	0.4880	0.5110	0.2190	0.2800	**0.0647**	0.3930	0.1380

4. Conclusion and outlook

In this paper, the aim was to investigate the performance of entropy estimators based on a Bernstein polynomial density estimator approach. The Bernstein based estimators are similar to the Ahmad and Lin (1976) kernel density based estimator, where the density in the formulation is estimated by

Table 2. Bias, variance and RMSE results of the entropy estimators ($n = 50$).

		$\widehat{H}^{V}_{n,8}$	$\widehat{H}^{VE}_{n,8}$	$\widehat{H}^{E}_{n,8}$	$\widehat{H}^{K}_{n,h^*}$	$\widehat{H}^{B}_{n,3}$	$\widehat{H}^{Q}_{n,12}$	$\widehat{H}^{C}_{n,3,2}$
E(1)	bias	0.1216	0.0960	**0.0093**	0.0766	0.0869	0.0695	0.0320
	variance	**0.0223**	0.0231	**0.0223**	0.0337	0.0272	0.0225	0.0267
	RMSE	0.1920	0.1950	**0.1500**	0.1990	0.1860	0.1650	0.1660
LN(0,.5)	bias	0.1517	0.1887	0.0394	0.0683	0.0408	0.0750	**0.0301**
	variance	0.0183	**0.0178**	0.0183	0.0224	0.0218	0.0181	0.0199
	RMSE	0.2030	0.2020	**0.1410**	0.1640	0.1530	0.1540	0.1440
G(2,2)	bias	0.1632	0.1626	0.0509	0.0596	0.0079	0.0887	**0.0030**
	variance	**0.0157**	**0.0157**	**0.0157**	0.0208	0.0178	**0.0157**	0.0181
	RMSE	0.2060	0.2060	0.1350	0.1560	0.1350	0.1530	**0.1340**
W(2,2)	bias	0.1750	0.1593	0.0627	0.0382	**0.0047**	0.0927	0.0374
	variance	**0.0097**	0.0110	**0.0097**	0.0136	0.0111	0.0100	0.0154
	RMSE	0.2010	0.2040	0.1170	0.1230	**0.1060**	0.1370	0.1300
B(2,2)	bias	0.1865	0.1131	0.0742	**0.0229**	0.0755	0.1025	0.0324
	variance	0.0040	0.0072	0.0040	0.0106	**0.0008**	0.0046	0.0049
	RMSE	0.1970	0.2050	0.0977	0.1060	0.0805	0.1230	**0.0774**
B(.5,.5)	bias	**0.0348**	0.2993	0.1471	0.0644	0.1894	0.0366	0.1387
	variance	0.0053	0.0041	0.0053	0.0376	**0.0010**	0.0123	0.0041
	RMSE	0.0809	**0.0731**	0.1640	0.2040	0.1920	0.1170	0.1530
U(0,1)	bias	0.1598	**0.0015**	0.0475	0.0328	0.0127	0.1078	0.0019
	variance	0.0014	0.0037	0.0014	0.0109	**0.0002**	0.0020	0.0017
	RMSE	0.1640	0.1710	0.0605	0.1100	**0.0200**	0.1170	0.0412

Table 3. Bias, variance and RMSE results of the entropy estimators ($n = 100$).

		$\widehat{H}^{V}_{n,10}$	$\widehat{H}^{VE}_{n,10}$	$\widehat{H}^{E}_{n,10}$	$\widehat{H}^{K}_{n,h^*}$	$\widehat{H}^{B}_{n,5}$	$\widehat{H}^{Q}_{n,21}$	$\widehat{H}^{C}_{n,5,3}$
E(1)	bias	0.0757	0.0873	**0.0073**	0.0757	0.0432	0.0435	0.0087
	variance	0.0111	0.0110	0.0111	0.0163	0.0121	**0.0109**	0.0126
	RMSE	0.1300	0.1290	**0.1060**	0.1480	0.1180	0.1130	0.1120
LN(0,.5)	bias	0.0819	0.1500	**0.0136**	0.0450	0.0571	0.0400	0.0278
	variance	0.0086	**0.0081**	0.0086	0.0094	0.0106	0.0083	0.0092
	RMSE	0.1240	0.1220	**0.0937**	0.1070	0.1180	0.0994	0.0997
G(2,2)	bias	0.0911	0.1273	0.0227	0.0377	0.0210	0.0468	**0.0005**
	variance	0.0073	0.0077	0.0073	0.0093	0.0074	**0.0071**	0.0073
	RMSE	0.1250	0.1260	0.0884	0.1040	0.0885	0.0966	**0.0852**
W(2,2)	bias	0.1015	0.1262	0.0331	0.0217	**0.0114**	0.0516	0.0127
	variance	0.0047	0.0054	0.0047	0.0065	0.0049	**0.0047**	0.0052
	RMSE	0.1230	0.1250	0.0764	0.0835	**0.0706**	0.0860	0.0734
B(2,2)	bias	0.1121	0.0856	0.0437	**0.0015**	0.0383	0.0580	0.0042
	variance	0.0018	0.0031	0.0018	0.0040	**0.0009**	0.0020	0.0017
	RMSE	0.1200	0.1250	0.0613	0.0630	0.0483	0.0733	**0.0414**
B(.5,.5)	bias	0.0287	0.2312	0.0971	0.0452	0.1459	**0.0242**	0.1165
	variance	0.0039	0.0028	0.0039	0.0096	**0.0013**	0.0062	0.0022
	RMSE	0.0691	**0.0603**	0.1160	0.1080	0.1500	0.0823	0.1260
U(0,1)	bias	0.1015	**0.0003**	0.0331	0.0315	0.0101	0.0645	0.0115
	variance	0.0004	0.0011	0.0004	0.0036	**0.0001**	0.0005	0.0001
	RMSE	0.1030	0.1070	0.0385	0.0678	**0.0135**	0.0683	0.0157

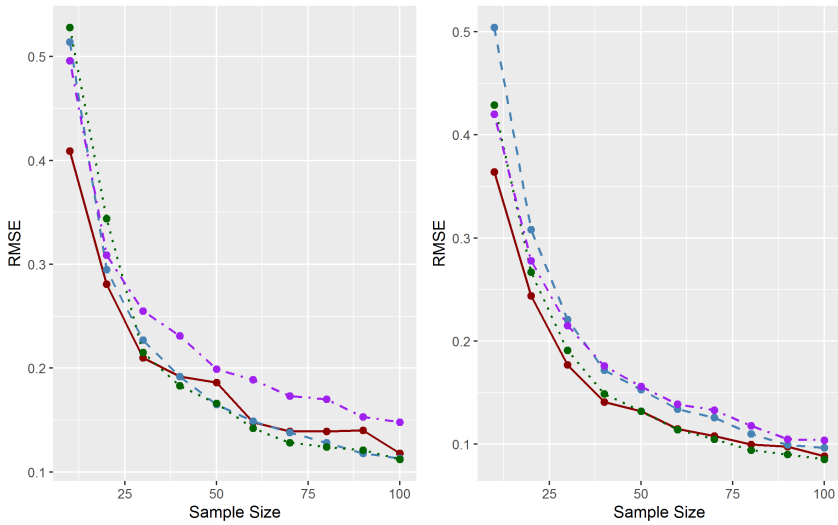


Figure 1. RMSE results for $\widehat{H}^{K}_{n,h^*}$ (purple dashed-dotted line), $\widehat{H}^{B}_{n,m}$ (red solid line), $\widehat{H}^{Q}_{n,m}$ (blue dashed line) and $\widehat{H}^{C}_{n,m}$ (green dotted line) with $n \in \{10, 20, 30, 40, 50, 60, 70, 80, 90, 100\}$ for $E(1)$ (left) and $G(2, 2)$ (right).

Bernstein polynomial methods in order to remedy the problem of boundary bias which commonly plagues kernel density estimation. It was found that the boundary bias free estimation approaches lead to good entropy estimators. The Bernstein entropy estimators were also compared to existing spacing based estimators such as the widely used Vasicek (1976) estimator and its modification. The Bernstein based entropy estimators exhibited the smallest bias and RMSE values in many cases which suggests that they are competitive and, in some cases, are the superior entropy estimator choice.

The choices for the tuning parameter values were not necessarily optimal; an avenue for future research is to find methods of choosing the optimal value not only for the Bernstein based estimators, but also the window-width of the Vasicek (1976) estimator and its modifications. It is well documented that this is still an open problem. Furthermore, the consistency and other asymptotic results of the Bernstein based estimators require investigation. In a further extension motivated by the work of Hall and Morton (1993), it may be prudent to consider a leave-one-out version of the Bernstein density entropy estimator. The motivation being parallel to the kernel density case where the estimator is dependent on each observation X_i and may result in a notable difference between $\mathbb{E}[\widehat{H}^{B}_{n,m}(x)]$ and $\int f(x)(\log \mathbb{E}[\widehat{f}_{n,m}(x)])$.

References

Adami, C. (2004). Information theory in molecular biology. *Physics of Life Reviews*, **1**, 3–22.

Ahmad, I. and Lin, P.-E. (1976). A nonparametric estimation of the entropy for absolutely continuous distributions (corresp.). *IEEE Transactions on Information Theory*, **22**, 372–375.

Babu, G. J., Canty, A. J., and Chaubey, Y. P. (2002). Application of Bernstein polynomials for smooth estimation of a distribution and density function. *Journal of Statistical Planning and*

Inference, **105**, 377–392.

Beirlant, J., Dudewicz, E. J., Györfi, L., and Van der Meulen, E. C. (1997). Nonparametric entropy estimation: An overview. *International Journal of Mathematical and Statistical Sciences*, **6**, 17–39.

Bouzebda, S. and Elhattab, I. (2014). New kernel-type estimator of Shanonn's entropy. *Comptes Rendus Mathematique*, **352**, 75–80.

Chaubey, Y. P. and Vu, N. L. (2021). A numerical study of entropy and residual entropy estimators based on smooth density estimators for non-negative random variables. *Journal of Statistical Research*, **54**, 99–121.

Cheng, C. and Parzen, E. (1997). Unified estimators of smooth quantile and quantile density functions. *Journal of Statistical Planning and Inference*, **59**, 291–307.

Cincotta, P. M., Helmi, A., Méndez, M., Núñez, J. A., and Vucetich, H. (1999). Astronomical time-series analysis—ii. a search for periodicity using the Shannon entropy. *Monthly Notices of the Royal Astronomical Society*, **302**, 582–586.

Correa, J. C. (1995). A new estimator of entropy. *Communications in Statistics-Theory and Methods*, **24**, 2439–2449.

Crzcgorzewski, P. and Wirczorkowski, R. (1999). Entropy-based goodness-of-fit test for exponentiality. *Communications in Statistics-Theory and Methods*, **28**, 1183–1202.

Dai, J. and Sperlich, S. (2010). Simple and effective boundary correction for kernel densities and regression with an application to the world income and engel curve estimation. *Computational Statistics & Data Analysis*, **54**, 2487–2497.

Dmitriev, Y. G. and Tarasenko, F. P. (1974). On the estimation of functionals of the probability density and its derivatives. *Theory of Probability & Its Applications*, **18**, 628–633.

Dudewicz, E. J., van der Meulen, E. C., SriRam, M. G., and Teoh, N. K. W. (1995). Entropy-based random number evaluation. *American Journal of Mathematical and Management Sciences*, **15**, 115–153.

Ebrahimi, N., Pflughoeft, K., and Soofi, E. S. (1994). Two measures of sample entropy. *Statistics & Probability Letters*, **20**, 225–234.

Glad, I. K., Hjort, N. L., and Ushakov, N. G. (2003). Correction of density estimators that are not densities. *Scandinavian Journal of Statistics*, **30**, 415–427.

Hall, P. and Morton, S. C. (1993). On the estimation of entropy. *Annals of the Institute of Statistical Mathematics*, **45**, 69–88.

Heidenreich, N.-B., Schindler, A., and Sperlich, S. (2013). Bandwidth selection for kernel density estimation: a review of fully automatic selectors. *AStA Advances in Statistical Analysis*, **97**, 403–433.

Igarashi, G. and Kakizawa, Y. (2014). On improving convergence rate of Bernstein polynomial density estimator. *Journal of Nonparametric Statistics*, **26**, 61–84.

Jones, M. C. (1993). Simple boundary correction for kernel density estimation. *Statistics and Computing*, **3**, 135–146.

Kakizawa, Y. (2004). Bernstein polynomial probability density estimation. *Journal of Nonparametric*

Statistics, **16**, 709–729.

Leblanc, A. (2010). A bias-reduced approach to density estimation using Bernstein polynomials. *Journal of Nonparametric Statistics*, **22**, 459–475.

Maasoumi, E. and Racine, J. (2002). Entropy and predictability of stock market returns. *Journal of Econometrics*, **107**, 291–312.

Mudholkar, G. S. and Tian, L. (2002). An entropy characterization of the inverse Gaussian distribution and related goodness-of-fit test. *Journal of Statistical Planning and Inference*, **102**, 211–221.

Park, S. and Park, D. (2003). Correcting moments for goodness of fit tests based on two entropy estimates. *Journal of Statistical Computation and Simulation*, **73**, 685–694.

Parzen, E. (1962). On estimation of a probability density function and mode. *The Annals of Mathematical Statistics*, **33**, 1065–1076.

Pompe, B. (1994). On some entropy methods in data analysis. *Chaos, Solitons & Fractals*, **4**, 83–96.

R Core Team (2021). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.
URL: *https://www.R-project.org/*

Ratnaparkhi, A. (1997). A simple introduction to maximum entropy models for natural language processing. Technical report, IRCS Technical Reports Series, 81, University of Pennsylvania.

Robinson, P. M. (1991). Consistent nonparametric entropy-based testing. *The Review of Economic Studies*, **58**, 437–453.

Rosenblatt, M. (1956). A central limit theorem and a strong mixing condition. *Proceedings of the National Academy of Sciences*, **42**, 43–47.

Santafe, G., Calvo, B., Perez, A., and Lozano, J. A. (2015). *bde: Bounded Density Estimation*. R package version 1.0.1.
URL: *https://CRAN.R-project.org/package=bde*

Schuster, E. F. (1985). Incorporating support constraints into nonparametric estimators of densities. *Communications in Statistics-Theory and methods*, **14**, 1123–1136.

Shannon, C. E. (1948). A mathematical theory of communication. *The Bell System Technical Journal*, **27**, 379–423.

Silverman, B. W. (2018). *Density estimation for statistics and data analysis*. Routledge.

Van Es, B. (1992). Estimating functionals related to a density by a class of statistics based on spacings. *Scandinavian Journal of Statistics*, **19**, 61–72.

Vasicek, O. (1976). A test for normality based on sample entropy. *Journal of the Royal Statistical Society. Series B (Methodological)*, **38**, 54–59.

Vitale, R. A. (1975). A bernstein polynomial approach to density function estimation. *In Statistical Inference and Related Topics*. Elsevier, 87–99.

Wand, M. P. and Jones, M. C. (1994). *Kernel smoothing*. CRC Press.

Zamanzade, E. and Arghami, N. R. (2012). Testing normality based on new entropy estimators. *Journal of Statistical Computation and Simulation*, **82**, 1701–1713.

Proceedings of the 63rd Annual Conference of the South African Statistical Association

A Möbius-transformed toroidal distribution for dihedral angles modelling in protein structure

Thasmika Mohan[1], *Najmeh Nakhaei Rad*[1,2] *and Ding-Geng (Din) Chen*[1]

[1]Department of Statistics, University of Pretoria, Pretoria 0002, South Africa
[2]DSI-NRF Centre of Excellence in Mathematical and Statistical Sciences (CoE-MaSS)

In this paper we propose a novel distribution on the torus by applying a Möbius transformation to an existing distribution. This distribution can then be used to model dihedral angles in protein structure and also as a proposal distribution for Markov-Chain Monte-Carlo sampling to predict the 3-D structure of protein molecules. We discuss the related properties of the proposed model and substantiate our contribution using two real datasets and a simulation study in the performance assessment of the estimating approach.

Keywords: Dihedral angles, Metropolis-Hastings algorithm, Möbius Transformation, Proposal distribution, Ramachandran plot, Toroidal data, Torus.

1. Introduction

One of the fundamental problems in molecular biology is to predict the 3-D structure of proteins. Advances in this field would lead to extensive results in drug discovery, biotechnology and evolutionary biology (Ley and Verdebout, 2017). Protein molecules are required for the structure, function and regulation of the body's tissues and organs. Amino acids are the building blocks of proteins. There are 20 different amino acids in naturally occurring proteins. Natural amino acids have an amino group or nitrogen atom N ($–NH_2$), a carboxylic acid group C ($–COOH$), a hydrogen atom attached to a central carbon atom C_α and a side-chain which is attached to C_α. The backbone is identical across all 20 amino acids, while the side chains are different which gives the amino acids different biochemical properties. Amino acids are joined by covalent peptide bonds to form a single macromolecule. In fact, proteins are biopolymers consisting of linear sequences of amino acids:

$$\cdots\!-\!N^{(i-1)}-C_\alpha^{(i-1)}-C^{(i-1)}\!-\!N^{(i)}-C_\alpha^{(i)}-C^{(i)}\!-\!N^{(i+1)}-C_\alpha^{(i+1)}-C^{(i+1)}\!-\cdots\cdot$$

For each amino acid i, there are three dihedral angles: $\phi^{(i)}$, $\psi^{(i)}$ and $\omega^{(i)}$. ω is usually close to 180° or occasionally 0° and can be modelled with a discrete two-state variable, while ϕ and ψ play a vital role in the protein structure as they define the backbone of a protein (See Figure 1). Ramachandran and Sasisekharan (1968) explained how the pair of dihedral angles ϕ and ψ can be represented on a scatterplot. In their fundamental work, they plotted (ϕ, ψ) and found their empirical distribution.

Corresponding author: Najmeh Nakhaei Rad (najmeh.nakhaeirad@up.ac.za)
MSC2020 subject classifications: 62P10

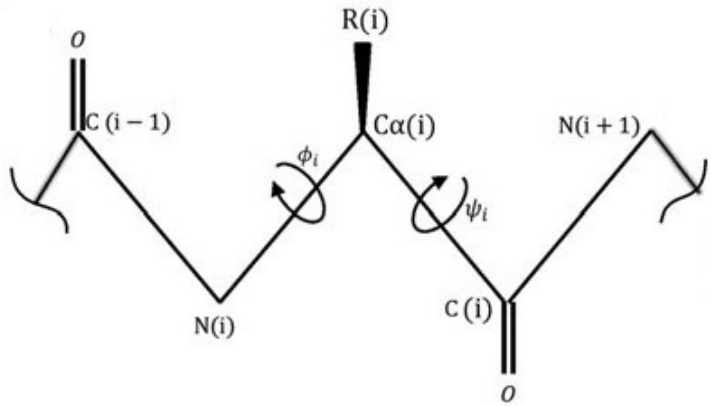


Figure 1. The 3-D structure of proteins.

From the directional statistics point of view, dihedral angles lie on the circumference of a torus. A torus is the product of two circles, $(-\pi, \pi] \times (-\pi, \pi]$. A circular random variable Θ is measured in degrees from 0 to 360° or radians in $(-\pi, \pi]$. In general, (θ_1, θ_2) refers to any observation on the torus where $\theta_1 \in (-\pi, \pi]$ and $\theta_2 \in (-\pi, \pi]$. Circular and toroidal variables cannot be modelled using standard univariate and bivariate statistical distributions and periodical models are needed.

Dihedral angles (ϕ, ψ), $\phi \in (-\pi, \pi]$ and $\psi \in (-\pi, \pi]$ are classified as toroidal data. Toroidal distributions are used to model dihedral angles for visualisation and prediction of the 3-D structure of proteins. To predict the 3-D structure of proteins a Markov chain is constructed using the Metropolis-Hastings algorithm with the Boltzmann distribution as the stationary distribution and a symmetric proposal distribution (Ley and Verdebout, 2017). Toroidal distributions can be used as proposal distributions in Markov Chain Monte Carlo (MCMC) sampling of proteins. Choosing a good proposal distribution is one of the challenges in MCMC sampling of proteins. Concentrated Gaussian perturbations are the most straightforward proposal distributions to use. When the proposal distribution is closer to the stationary distribution, the results are more accurate. Therefore, protein structural information such as dihedral angles should be incorporate into proposal distributions.

Thus, this paper contributes to this field by proposing a new flexible toroidal model that is an alternate candidate for modelling of dihedral angles.

To pave the way for the foundation of the new model, the existing toroidal models will be briefly reviewed. Let (Θ_1, Θ_2) be jointly continuous random variables on the torus $(-\pi, \pi] \times (-\pi, \pi]$ with the joint probability density function (pdf) $f(\theta_1, \theta_2)$. The bivariate von Mises (BvM) distribution, presented in Mardia (1975), was the first footprint to model toroidal data with pdf

$$\begin{aligned} f(\theta_1, \theta_2) = \exp\{&\kappa_1 \cos(\theta_1 - \mu_1) + \kappa_2 \cos(\theta_2 - \mu_2) + (\cos(\theta_1 - \mu_1) \sin(\theta_1 - \mu_1)) \\ &\times \mathbf{A}(\cos(\theta_2 - \mu_2, \sin(\theta_2 - \mu_2))'\}, \end{aligned} \tag{1}$$

where μ_1,$\mu_2 \in$ [-π,π) are the circular location parameters, $\kappa_1, \kappa_2 \geq 0$ are the concentration parameters and the circular-circular dependence parameter $\mathbf{A}$, is a 2× 2 matrix. This BvM distribution is over-parametrised (Ley and Verdebout, 2017) due to eight parameters, thus Rivest (1988) proposed a

subclass with pdf as

$$f(\theta_1,\theta_2) \propto \exp\{\kappa_1\cos(\theta_1-\mu_1)+\kappa_2\cos(\theta_2-\mu_2) + \alpha\cos(\theta_1-\mu_1)\cos(\theta_2-\mu_2)+\beta\sin(\theta_1-\mu_1)\sin(\theta_2-\mu_2)\}, \tag{2}$$

where $\alpha, \beta \in \mathbb{R}$. To achieve better parametrisation, Singh et al. (2002) used (2) setting $\alpha = 0$, which paves the way to the *Sine model* with pdf

$$f(\theta_1,\theta_2) = C\exp\{\kappa_1\cos(\theta_1-\mu_1)+\kappa_2\cos(\theta_2-\mu_2)+\beta\sin(\theta_1-\mu_1)\sin(\theta_2-\mu_2)\}, \tag{3}$$

where $C^{-1} = 4\pi^2\sum_{i=0}^{\infty}\binom{2i}{i}(\beta^2/4\kappa_1\kappa_2)^i I_i(\kappa_1)I_i(\kappa_2)$ ($I_\alpha(z)$ is the modified Bessel function of the first kind of order α).

Mardia et al. (2007) proposed a special case of (2) with $\alpha = \beta = -\kappa_3$. This was called the *Cosine model* with pdf

$$f(\theta_1,\theta_2) = C\exp\{\kappa_1\cos(\theta_1-\mu_1)+\kappa_2\cos(\theta_2-\mu_2)-\kappa_3\cos(\theta_1-\mu_1-\theta_2+\mu_2)\}, \tag{4}$$

where κ_1,$\kappa_2 \geq 0$ and $C^{-1} = 4\pi^2\left[I_0(\kappa_1)I_0(\kappa_2)I_0(\kappa_3)+2\sum_{i=1}^{\infty}I_i(\kappa_1)I_i(\kappa_2)I_i(\kappa_3)\right]$.

The departure for this paper will be the model introduced by Sengupta and Ong (2014). They implemented a mixture approach to construct a bivariate circular-circular distribution on the torus with pdf

$$f(\theta_1,\theta_2) = \left(\frac{1}{2\pi}\right)^2\Big[1+2\gamma_1\{\delta_1\cos(\theta_1-\mu_1)+\delta_2\cos(\theta_2-\mu_2)\} + 4\gamma_2\delta_1\delta_2\cos(\theta_1-\mu_1)\cos(\theta_2-\mu_2)\Big], \tag{5}$$

where $\delta_1, \delta_2 \in (-1/2, 1/2)$. For $\tau > 0$, $\gamma_1 = \gamma(2,\tau)/\tau(1-e^{-\tau})$ and $\gamma_2 = \gamma(3,\tau)/\tau^2(1-e^{-\tau})$ such that $\gamma_1, \gamma_2 \in (0,1)$. The marginals are cardioid distributions on the circle.

The interested reader is referred to the works of the following authors for further reading, see amongst others, Johnson and Wehrly, 1977; Jones et al., 2015; Pertsemlidis et al., 2005; Fernández-Durán and Gregorio-Domínguez, 2014.

The rest of the paper is structured as follows. Section 2 outlines the contribution of this paper with the novel distribution and the associated statistical properties. In Section 3, a simulation study is conducted to assess the performance of the DEoptim package in R software to obtain the estimates of parameters. We discuss the application of this novel distribution to model the dihedral angles in the protein structure in Section 4 and we conclude this paper with a discussion in Section 5.

2. Möbius-transformed bivariate distribution

In this section, a new distribution is introduced on the torus by applying a Möbius transformation to (5). The Möbius transformation of $\tilde{\theta}$ to θ is mapped by (Kato and Pewsey, 2015):

$$\theta = \mathcal{M}(\tilde{\theta},\mu,\upsilon,\xi) = \mu+\upsilon+2\arctan\left\{\frac{1-\xi}{1+\xi}\tan(\frac{\tilde{\theta}-\upsilon}{2})\right\}, \tag{6}$$

where $-\pi < \theta, \tilde{\theta} \leq \pi$, $-\pi < \mu, \upsilon \leq \pi$ and $\xi \in [0,1)$. If $\xi = 0$, the transformation is an identity mapping and when $\xi \to 1$, then $\mathcal{M}(\tilde{\theta},\mu,\upsilon,\xi) \to \upsilon$. The reader is referred to the work of the

following contributors to show the usefulness of the Möbius transformation, Kato and Pewsey (2015) and Arashi et al. (2021).

Theorem 1. *Let $f(\tilde{\theta_1},\tilde{\theta_2})$ be the bivariate distribution given by* (5)*, by applying the Möbius transformation* (6) *on $\tilde{\theta_1}$ and $\tilde{\theta_2}$ with $\upsilon = 0$ (without loss of generality), the new Möbius-transformed bivariate model has pdf*

$$f(\theta_1,\theta_2) = \left(\frac{1}{2\pi}\right)^2 \frac{\left(1-\xi_1^2\right)\left(1-\xi_2^2\right)}{(1+\xi_1^2-2\xi_1\cos(\theta_1-\mu_1))^2(1+\xi_2^2-2\xi_2\cos(\theta_2-\mu_2))^2} \times \{c_0 + c_1\cos(\theta_1-\mu_1) + c_2\cos(\theta_2-\mu_2) + c_3\cos(\theta_1-\mu_1)\cos(\theta_2-\mu_2)\}, \tag{7}$$

where $0 \le \xi_1, \xi_2 \le 1$ *and* $c_0 = (1+\xi_1^2)(1+\xi_2^2) - 4\gamma_1\delta_1\xi_1(1+\xi_2^2) - 4\gamma_1\delta_2\xi_2(1+\xi_1^2) + 16\gamma_2\delta_1\delta_2\xi_1\xi_2$, $c_1 = 2\gamma_1\delta_1(1+\xi_1^2)(1+\xi_2^2 - 2\xi_1(1+\xi_2^2) - 8\gamma_2\delta_1\delta_2\xi_2(1+\xi_1^2) - 8\gamma_1\delta_2\xi_1\xi_2$, $c_2 = 2\gamma_1\delta_2(1+\xi_1^2)(1+\xi_2^2) - 2\xi_2(1+\xi_1^2) - 8\gamma_2\delta_1\delta_2\xi_1(1+\xi_2^2) + 8\gamma_1\delta_1\xi_1\xi_2$, $c_3 = 4\xi_1\xi_2 - 4\gamma_1\delta_1\xi_2(1+\xi_1^2) - 4\gamma_1\delta_2\xi_1(1+\xi_2^2) + 4\gamma_2\delta_1\delta_2(1+\xi_1^2)(1+\xi_2^2)$.

Proof. See Appendix. ∎

Figure 2 shows the contour plots of (7) for different values of the parameters to show the flexibility. As can be seen in Figure 2, the bivariate probability model (7) has both unimodal and bimodal shapes. The marginals of (5) are cardioid distributions therefore the marginals of (7) are Möbius-transformed cardioid distributions which is studied by Wang and Shimizu (2012). The marginal pdf of θ_2 is,

$$f_{\Theta_2}(\theta_2) = \frac{1}{2\pi}\frac{1-\xi_2^2}{(1+\xi_2^2-2\xi_2\cos(\theta_2-\mu_2))}\left(1 + 2\delta_2\gamma_1\frac{(1+\xi_2^2)\cos(\theta_2-\mu_2) - 2\xi_2}{1+\xi_2^2-2\xi_2\cos(\theta_2-\mu_2)}\right). \tag{8}$$

If $\xi_2 = 0$, the cardioid distribution is obtained. Figure 3 shows the marginal density (8) for different parameter values.

2.1 Asymmetric form

The Möbius-transformed bivariate distribution (7) is symmetric as well as the marginals, because $f(\mu_1-\theta_1,\mu_2-\theta_2) = f(\mu_1+\theta_1,\mu_2+\theta_2)$ and $f(\mu-\theta) = f(\mu+\theta)$, respectively. In this section, using the approach from Ameijeiras-Alonso and Ley (2022) and also Abe and Pewsey (2011), we introduce the sine-skewed (SS) version of (7) and (8). The SS version of the Möbius-transformed bivariate distribution (7) is

$$f_{ss}(\theta_1,\theta_2) = f(\theta_1,\theta_2)(1+\lambda_1\sin(\theta_1-\mu_1) + \lambda_2\sin(\theta_2-\mu_2)), \tag{9}$$

where $f(\cdot,\cdot)$ is from (7) and λ_1, λ_2 are skewness parameters. Note that $0 \le |\lambda_1| + |\lambda_2| \le 1, -1 \le \lambda_1, \lambda_2 \le 1$.

Figure 4 shows the skew density (9) for different values of λ_1 and λ_2. The marginal of the SS version of the Möbius-transformed bivariate distribution (7) is,

$$f_{\Theta_{2_{ss}}}(\theta_2) = f_{\Theta_2}(\theta_2)(1+\lambda\sin(\theta_2-\mu_2)), \tag{10}$$

where $-1 \le \lambda \le 1$ is the skewness parameter. If $\lambda = 0$ the symmetric form is obtained. $\lambda > 0$ provides the left skewed and $\lambda < 0$ provides the right skewed distribution. Figure 5 illustrates the density (10) for different values of λ.

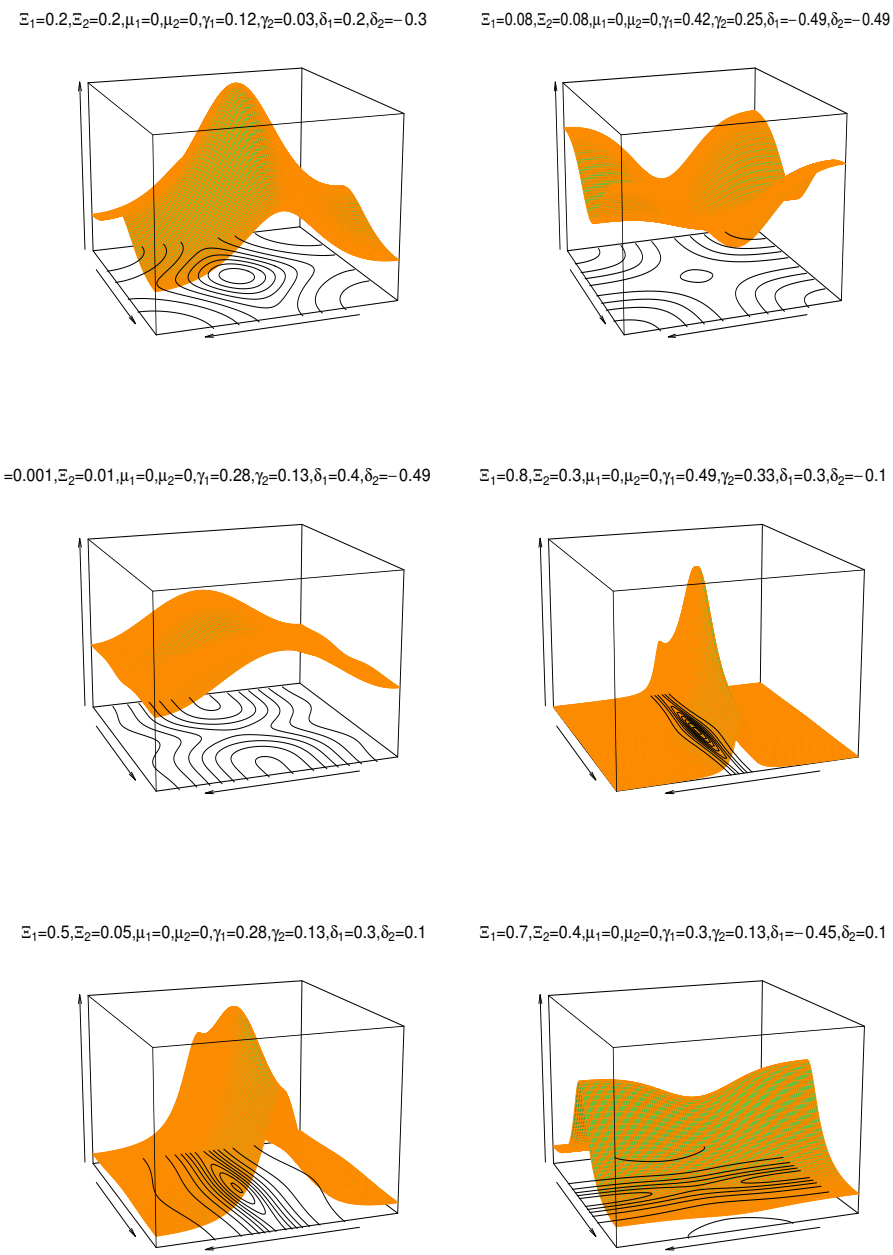


Figure 2. Pdf plots and contour plots of the Möbius-transformed bivariate distribution (7).

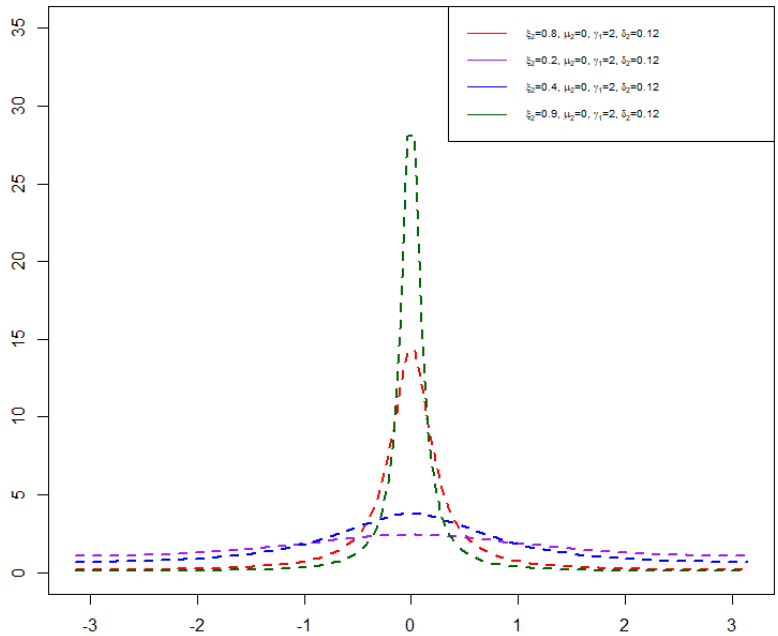


Figure 3. Pdf plots of the marginal distribution (8).

2.2 Maximum likelihood estimation

The maximum likelihood method is discussed below, to obtain the estimates parameters for the Möbius-transformed bivariate distribution (7). Suppose $\mathbf{\Gamma} = (\mu_1, \mu_2, \xi_1, \xi_2, \gamma_1, \gamma_2, \delta_1, \delta_2)^T$ are the parameters of (7) and $(\theta_{1i}, \theta_{2i})$, $i = 1, 2, 3, ..., n$ is a sample of size n from (7). The log-likelihood function of the model is represented as follows:

$$
\begin{aligned}
l(\mathbf{\Gamma}) = {} & 2n \log\left(\frac{1}{2\pi}\right) + n \log\left(1 - \xi_1^2\right) + n \log\left(1 - \xi_2^2\right) - 2 \sum_{i=1}^{n} \log(1 + \xi_1^2 - 2\xi_1 \cos(\theta_{1i} - \mu_1)) \\
& - 2 \sum_{i=1}^{n} \log(1 + \xi_2^2 - 2\xi_2 \cos(\theta_{2i} - \mu_2)) + \sum_{i=1}^{n} \{c_0 + c_1 \cos(\theta_{1i} - \mu_1) + c_2 \cos(\theta_{2i} - \mu_2) \\
& + c_3 \cos(\theta_{1i} - \mu_1) \cos(\theta_{2i} - \mu_2)\},
\end{aligned}
\tag{11}
$$

where c_i for $i = 0, 1, 2, 3$ is as defined in (7). The maximum likelihood estimates (MLE) of the parameters, $\hat{\mathbf{\Gamma}} = (\hat{\mu_1}, \hat{\mu_2}, \hat{\xi_1}, \hat{\xi_2}, \hat{\gamma_1}, \hat{\gamma_2}, \hat{\delta_1}, \hat{\delta_2})^T$ can be determined by maximising (11) with respect to $\mathbf{\Gamma} = (\mu_1, \mu_2, \xi_1, \xi_2, \gamma_1, \gamma_2, \delta_1, \delta_2)^T$. By setting the partial derivatives of the log-likelihood functions in (11) with respect to $\mathbf{\Gamma}$ to zero, the MLEs of $\mathbf{\Gamma} = (\mu_1, \mu_2, \xi_1, \xi_2, \gamma_1, \gamma_2, \delta_1, \delta_2)^T$ for the Möbius-transformed bivariate distribution (7) can be obtained. The details of derivations are avilable upon request from the authors. The DEoptim package in R which is based on the Differential Evolution (DE) algorithm (Ardia et al., 2011) has been used to obtain the MLEs of parameters. Extensive

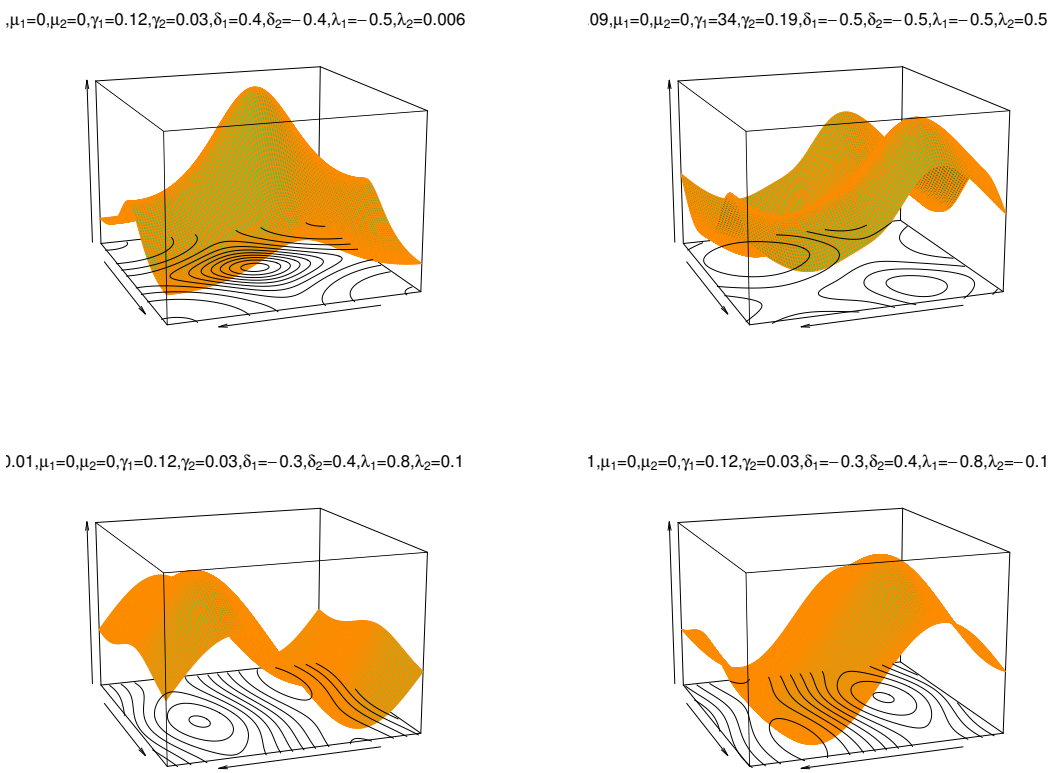


Figure 4. Pdf plots and contour plots of the sine-skewed Möbius-transformed bivariate distribution (9).

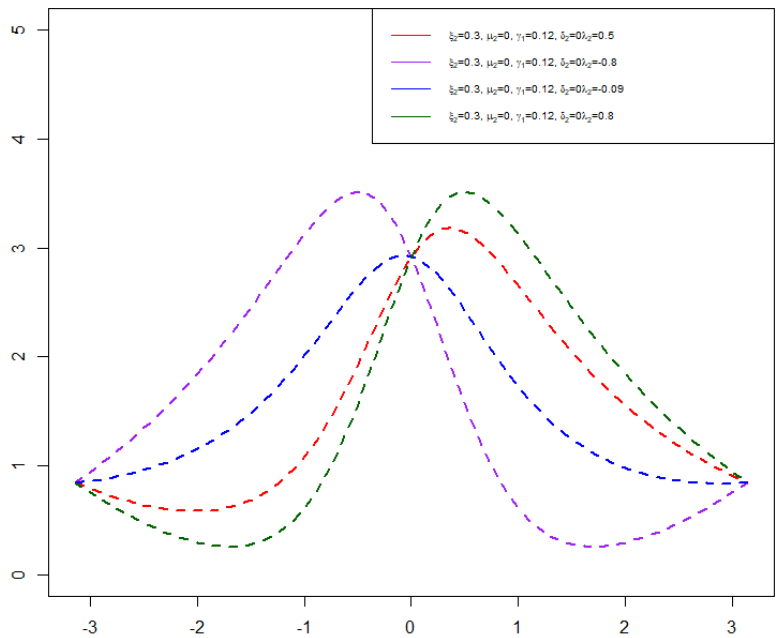


Figure 5. Pdf plots of the skew marginal distribution (10).

studies have been undertaken of the DE algorithm's significant performance as a global optimisation algorithm on continuous numerical minimisation problems (**?**).

3. Simulation study

In this section a simulation study is conducted to asses the performance of the DEoptim package in estimating the parameters of the Möbius-transformed bivariate distribution (7). In the first part of the simulation study, two samples of size 1000 was generated from (7) with parameters ($\mu_1 = -1.5$, $\mu_2 = 1$, $\gamma_1 = 0.5$, $\gamma_1 = 0.3$, $\delta_1 = 0.45$, $\delta_2 = 0.45$, $\xi_1 = 0.6$,$\xi_2 = 0.06$) and ($\mu_1 = 0$, $\mu_2 = 0$, $\gamma_1 = 0.42$, $\gamma_1 = 0.25$, $\delta_1 = 0.1$, $\delta_2 = 0.4$, $\xi_1 = 0.2$,$\xi_2 = 0.1$) using the Metropolis-Hastings algorithm (Robert and Casella, 1999). Mardia (2014) explored the suitable methods for generating samples from toroidal models and they found that the rejection sampling approaches are more efficient (also see Arashi et al. (2021).

The scatterplots of the simulated data along with the contour plots of the target distributions are shown in Figure 6. The traceplots, compare-partial and running mean plots are shown in Figure 7. These plots provide evidence that the simulated data comes from the target distribution and also indicates that the initial and final parts of the chain were sampled from the same distribution.

In the second part of the study, 500 samples of size 1000 were generated from the Möbius-transformed bivariate distribution (7) and the Monte Carlo method was used to assess the performance of DEoptim package for estimating the parameters. The MLEs, standard errors, biases, mean-squared errors (MSEs) and the coverage probabilities (CP) are given in Table 1. The columns under the

denomination of MLE and estimated standard deviation include an average of the 1000 maximum likelihood estimates obtained and an average of the 1000 standard errors of obtained estimates. The values of the standard deviations in Table 1 show the standard deviation of 1000 maximum likelihood estimates obtained. As can be seen in Table 1, the MSEs and biases are very small indicating that the parameter estimates are close to the true parameter values and therefore these estimates are unbiased. The coverage probabilities are close to 95% indicating that the variance estimation is consistent and valid. Hence, we conclude that the DEoptim package is suitable and accurate for estimating the parameters of (7) and the Metropolis-Hastings algorithm is suitable for generating samples from (7).

Table 1. The MLEs, standard errors, biases, mean-squared errors (MSEs) and the coverage probabilities (CP).

Parameter	Value	MLE	Est. std. dev.	Bias	Std. dev.	MSE	CP
ξ_1	0.60	0.6263	0.0326	0.0293	0.0304	0.0044	0.936
ξ_2	0.50	0.5004	0.0436	0.0004	0.0482	0.0023	0.924
μ_1	−1.50	−1.5023	0.0452	−0.0023	0.0464	0.0022	0.952
μ_2	1.00	0.9975	0.0691	−0.0025	0.0652	0.0045	0.957
γ_1	0.45	0.4514	0.0594	0.0015	0.0570	0.0028	0.938
γ_2	0.30	0.2986	0.0552	−0.0013	0.0510	0.0027	0.942
δ_1	0.30	0.3047	0.0599	0.0048	0.0538	0.0029	0.923
δ_2	0.20	0.2068	0.0438	0.0070	0.0492	0.0024	0.939
ξ_1	0.20	0.1197	0.0527	−0.0002	0.0519	0.0027	0.929
ξ_2	0.10	0.1158	0.0281	0.0156	0.0275	0.0010	0.907
μ_1	0.00	−0.0224	0.2381	−0.0226	0.2357	0.0561	0.926
μ_2	0.00	−0.0012	0.2345	−0.0013	0.2319	0.0538	0.935
γ_1	0.42	0.3892	0.0538	−0.0288	0.0516	0.0035	0.938
γ_2	0.25	0.2276	0.0574	−0.0264	0.0557	0.0038	0.939
δ_1	0.10	0.1167	0.0927	0.0167	0.0917	0.0087	0.942
δ_2	0.40	0.4022	0.0445	0.0022	0.0435	0.0019	0.948

4. Protein structure application

To demonstrate the performance of the Möbius-transformed bivariate distribution (7) in modelling dihedral angles, we used two datasets available at http://scop.mrc-lmb.cam.ac.uk/scop/. The datasets are TCBIG.CYS.left and TCBIG.ASN.right which consist of 823 and 3467 dihedral angles, respectively. The Ramachandran plots of these two datasets are shown in Figure 8. According to the Ramachandran plots in Figure 8, it is evident that the datasets are bimodal and hence, a mixture distribution with two components is required. The results of the bivariate circular-circular distribution (5) including the MLEs of the parameters, log-likelihood, the Akaike information criterion (AIC) and the Bayesian information criterion (BIC) are shown in Table 2 for each dataset. The same results for the Möbius-transformed bivariate distribution (7) are given in Table 2. The scatter plots of data and the contour plots of fitted distributions are given in Figure 9.

As can be seen in Table 2, the Möbius-transformed bivariate distribution (7) has smaller AIC and

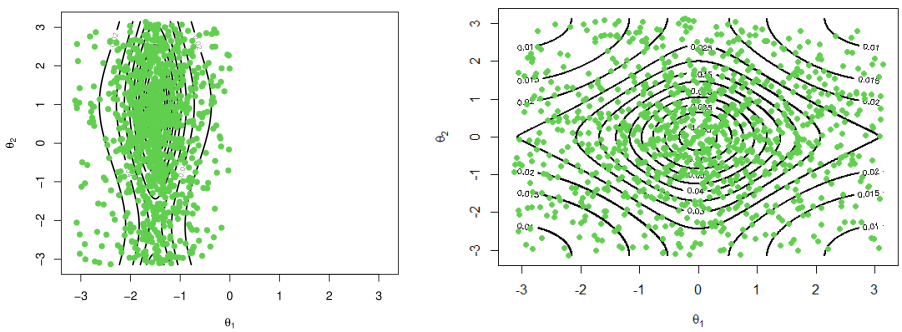


Figure 6. The scatter plot of the simulated data and the contour plot of the target distribution.

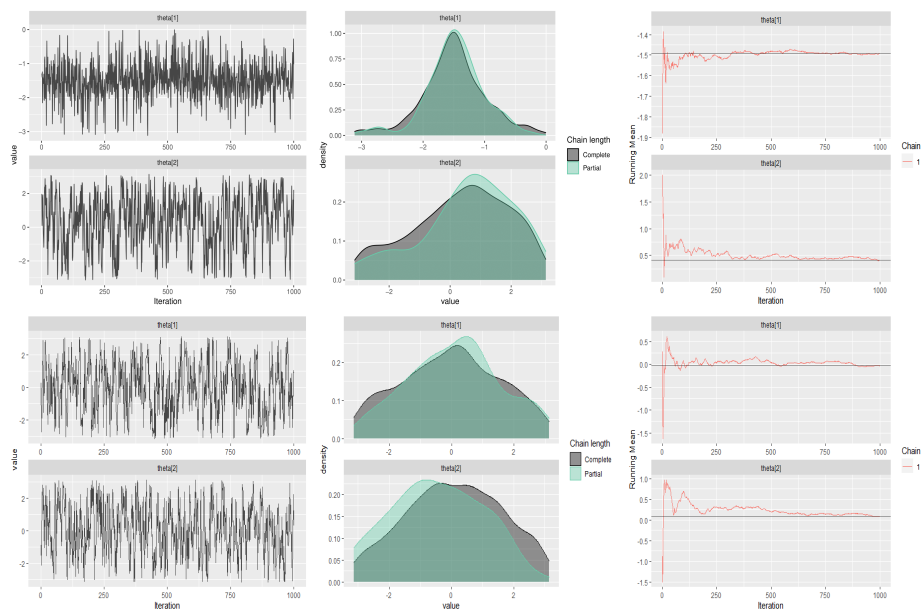


Figure 7. The traceplots, compare-partial plots and running mean plots of the simulated data.

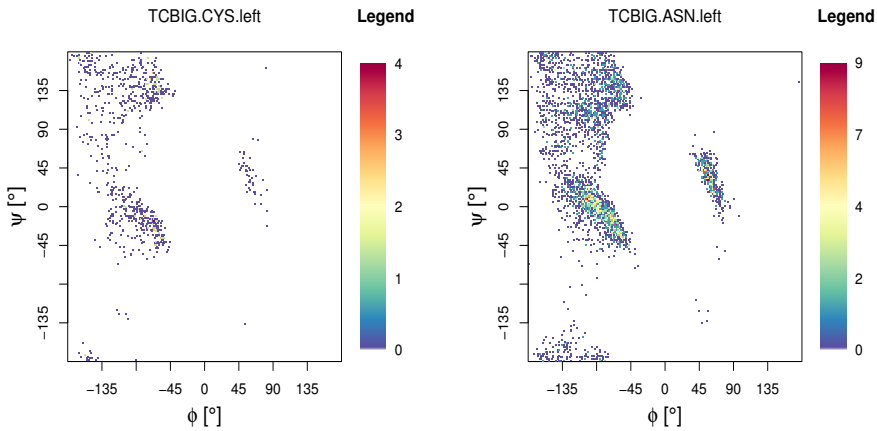


Figure 8. Ramachandran plots for the datasets.

BIC values and thus we choose this as the best model for the given datasets. Looking at Figure 9, we see that the fitted contour plots suit the spread of the data well. Thus, the Möbius-transformed bivariate distribution (7), proves to be a flexible toroidal model for future research and use.

Table 2. MLEs and corresponding log-likelihood, AIC and BIC.

	TCBIG.CYS.left		TCBIG.ASN.left	
	circular-circular model	**Möbius-transformed model**	**circular-circular model**	**Möbius-transformed model**
$\hat{\mu_1}$	-1.5899	-1.5348	-1.4449	-1.5977
$\hat{\mu_2}$	1.2299	1.2002	1.1016	1.0177
$\hat{\gamma_1}$	0.4999	0.5011	0.5090	0.5102
$\hat{\gamma_2}$	0.3333	0.3209	0.3367	0.3266
$\hat{\delta_1}$	0.4887	0.4509	0.4777	0.2393
$\hat{\delta_2}$	0.4776	0.4408	0.4800	0.4699
$\hat{\xi_1}$	-	0.5093	-	0.5908
$\hat{\xi_2}$	-	0.0005	-	0.0647
Log-lik.	-2693.92	-2360.33	-11348.69	-10118.61
AIC	5397.83	4734.65	22707.37	20251.21
BIC	5421.39	4767.64	22738.13	20251.21

5. Conclusion

This paper contributed to protein dihedral angles modelling with proposing a flexible model for toroidal data. More specifically, we constructed a novel bivariate distribution on the torus by applying a Möbius transformation to a pre-existing bivariate circular distribution. The obtained distribution

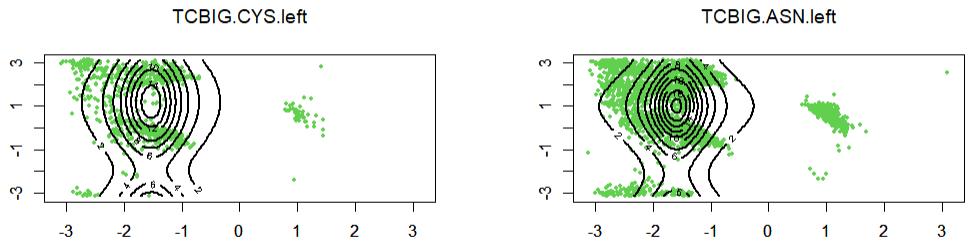


Figure 9. The scatter plots of the datasets and the contour plots of the fitted Möbius-transformed bivariate distribution (7).

proved to be an alternate model to the existing proposal distributions for the MCMC sampling of proteins. Substantiated by real data, the proposed model outperformed the existing original model. Finally, the Metropolis-Hastings algorithm proved to be accurate in generating samples from the new Möbius-transformed bivariate distribution.

6. Acknowledgement

This work was based upon research supported in part by the South African DST-NRF-MRC SARChI Research Chair in Biostatistics (Grant No. 114613); the National Research Foundation (NRF) of South Africa, Ref.: RA210106581084, grant No. 150170 and DSI-NRF Centre of Excellence in Mathematical and Statistical Sciences (CoE-MaSS), South Africa. The opinions expressed and conclusions arrived at are those of the authors and are not necessarily to be attributed to the CoE-MaSS or the NRF.

Appendix

Proof of Theorem 1:
Let $\theta_1 = \mathcal{M}(\tilde{\theta_1}; \mu_1, \upsilon_1, \xi_1)$ and $\theta_2 = \mathcal{M}(\tilde{\theta_2}; \mu_2, \upsilon_2, \xi_2)$. Then, it follows that $\tilde{\theta_1} = \mathcal{M}^{-1}(\theta_1, \mu_1, \upsilon_1, \xi_1)$ and $\tilde{\theta_2} = \mathcal{M}^{-1}(\theta_2, \mu_2, \upsilon_2, \xi_2)$, where

$$\tilde{\theta} = \mathcal{M}^{-1}(\theta, \mu, \upsilon, \xi) = \upsilon + 2\arctan\left\{\frac{1+\xi}{1-\xi}\tan(\frac{\theta-\mu-\upsilon}{2})\right\}. \tag{12}$$

Therefore,

$$f(\theta_1, \theta_2) = |J| f\left(\mathcal{M}^{-1}(\theta_1, \mu_1, \upsilon_1, \xi_1), \mathcal{M}^{-1}(\theta_2, \mu_2, \upsilon_2, \xi_2)\right), \tag{13}$$

with the Jacobian matrix as

$$J = \begin{bmatrix} \frac{\partial \mathcal{M}^{-1}(\theta_1, \mu_1, \upsilon_1, \xi_1)}{\partial \theta_1} & 0 \\ 0 & \frac{\partial \mathcal{M}^{-1}(\theta_2, \mu_2, \upsilon_2, \xi_2)}{\partial \theta_2} \end{bmatrix}.$$

Without loss of generality, set $\mu_1 = \mu 2 = 0$ in (5) and $\upsilon_1 = \upsilon_2 = 0$ in (12). Then, from (12), we have

$$\frac{\partial \mathcal{M}^{-1}(\theta_1;\mu_1,\xi_1)}{\partial \theta_1} = \frac{1-\xi_1^2}{1+\xi_1^2-2\xi_1\cos(\theta_1-\mu_1)}.$$

Similarly,

$$\frac{\partial \mathcal{M}^{-1}(\theta_2;\mu_2,\xi_2)}{\partial \theta_2} = \frac{1-\xi_2^2}{1+\xi_2^2-2\xi_2\cos(\theta_2-\mu_2)}.$$

Then, from (13),

$$\begin{aligned} f(\theta_1,\theta_2) = &\left(\frac{1}{2\pi}\right)^2 \frac{\left(1-\xi_1^2\right)\left(1-\xi_2^2\right)}{(1+\xi_1^2-2\xi_1\cos(\theta_1-\mu_1))(1+\xi_2^2-2\xi_2\cos(\theta_2-\mu_2))}\,[1+2\gamma_1 \\ &\left\{\frac{\delta_1(1+\xi_1^2)\cos(\theta_1-\mu_1)-2\delta_1\xi_1}{1+\xi_1^2-2\xi_1\cos(\theta_1-\mu_1)}+\frac{\delta_2(1+\xi_2^2)\cos(\theta_2-\mu_2)-2\delta_2\xi_2}{1+\xi_2^2-2\xi_2\cos(\theta_2-\mu_2)}\right\}+4\gamma_2\delta_1\delta_2 \\ &\frac{\left((1+\xi_1^2)\cos(\theta_1-\mu_1)-2\xi_1\right)\left((1+\xi_2^2)\cos(\theta_2-\mu_2)-2\xi_2\right)}{\left(1+\xi_1^2-2\xi_1\cos(\theta_1-\mu_1)\right)\left(1+\xi_2^2-2\xi_2\cos(\theta_2-\mu_2)\right)}\Bigg], \end{aligned} \tag{14}$$

where in (14),

$$\begin{aligned} &1+2\gamma_1\left\{\frac{\delta_1(1+\xi_1^2)\cos(\theta_1-\mu_1)-2\delta_1\xi_1}{1+\xi_1^2-2\xi_1\cos(\theta_1-\mu_1)}+\frac{\delta_2(1+\xi_2^2)\cos(\theta_2-\mu_2)-2\delta_2\xi_2}{1+\xi_2^2-2\xi_2\cos(\theta_2-\mu_2)}\right\} \\ &+4\gamma_2\delta_1\delta_2\frac{\left((1+\xi_1^2)\cos(\theta_1-\mu_1)-2\xi_1\right)\left((1+\xi_2^2)\cos(\theta_2-\mu_2)-2\xi_2\right)}{\left(1+\xi_1^2-2\xi_1\cos(\theta_1-\mu_1)\right)\left(1+\xi_2^2-2\xi_2\cos(\theta_2-\mu_2)\right)} \\ =&(1+\xi_1^2-2\xi_1\cos(\theta_1-\mu_1))(1+\xi_2^2-2\xi_2\cos(\theta_2-\mu_2))+\left(2\gamma_1\delta_1(1+\xi_1^2)\cos(\theta_1-\mu_1)-4\gamma_1\delta_1\xi_1\right) \\ &\times(1+\xi_2^2-2\xi_2\cos(\theta_2-\mu_2))+\left(2\gamma_1\delta_2(1+\xi_2^2)\cos(\theta_2-\mu_2)-4\gamma_1\delta_2\xi_2\right)(1+\xi_1^2-2\xi_1\cos(\theta_1-\mu_1)) \\ &+4\gamma_2\delta_1\delta_2((1+\xi_1^2)\cos(\theta_1-\mu_1)-2\xi_1)((1+\xi_2^2)\cos(\theta_2-\mu_2)-2\xi_2) \\ =&(1+\xi_1^2)(1+\xi_2^2)-2\xi_2(1+\xi_1^2)\cos(\theta_2-\mu_2)-2\xi_1(1+\xi_2^2)\cos(\theta_1-\mu_1) \\ &+4\xi_1\xi_2\cos(\theta_1-\mu_1)\cos(\theta_2-\mu_2)+2\gamma_1\delta_1(1+\xi_1^2)(1+\xi_2^2)\cos(\theta_1-\mu_1) \\ &-4\gamma_1\delta_1\xi_2(1+\xi_1^2)\cos(\theta_1-\mu_1)\cos(\theta_2-\mu_2)-4\gamma_1\delta_1\xi_1(1+\xi_2^2)+8\gamma_1\delta_1\xi_1\xi_2\cos(\theta_2-\mu_2) \\ &+2\gamma_1\delta_2(1+\xi_1^2)(1+\xi_2^2)\cos(\theta_2-\mu_2)-4\gamma_1\delta_2\xi_1(1+\xi_2^2)\cos(\theta_1-\mu_1)\cos(\theta_2-\mu_2) \\ &-4\gamma_1\delta_2\xi_2(1+\xi_1^2)+8\gamma_1\delta_2\xi_1\xi_2\cos(\theta_1-\mu_1)+4\gamma_2\delta_1\delta_2(1+\xi_1^2)(1+\xi_2^2)\cos(\theta_1-\mu_1)\cos(\theta_2-\mu_2) \\ &-8\gamma_2\delta_1\delta_2\xi_2(1+\xi_1^2)\cos(\theta_1-\mu_1)-8\gamma_2\delta_1\delta_2\xi_1(1+\xi_2^2)\cos(\theta_2-\mu_2)+16\gamma_2\delta_1\delta_2\xi_1\xi_2. \end{aligned}$$

Hence,

$$\begin{aligned} f(\theta_1,\theta_2) = &\left(\frac{1}{2\pi}\right)^2 \frac{\left(1-\xi_1^2\right)\left(1-\xi_2^2\right)}{(1+\xi_1^2-2\xi_1\cos(\theta_1-\mu_1))^2(1+\xi_2^2-2\xi_2\cos(\theta_2-\mu_2))^2}\,\{c_0+c_1\cos(\theta_1-\mu_1) \\ &+c_2\cos(\theta_2-\mu_2)+c_3\cos(\theta_1-\mu_1)\cos(\theta_2-\mu_2)\}, \end{aligned}$$

where $c_0 = (1+\xi_1^2)(1+\xi_2^2)-4\gamma_1\delta_1\xi_1(1+\xi_2^2)-4\gamma_1\delta_2\xi_2(1+\xi_1^2)$, $c_1 = 2\gamma_1\delta_1(1+\xi_1^2)(1+\xi_2^2-2\xi_1(1+\xi_2^2)-8\gamma_2\delta_1\delta_2\xi_2(1+\xi_1^2)-8\gamma_1\delta_2\xi_1\xi_2$, $c_2 = 2\gamma_1\delta_2(1+\xi_1^2)(1+\xi_2^2)-2\xi_2(1+\xi_1^2)-8\gamma_2\delta_1\delta_2\xi_1(1+\xi_2^2)+8\gamma_1\delta_1\xi_1\xi_2$, $c_3 = 4\xi_1\xi_2 - 4\gamma_1\delta_1\xi_2(1+\xi_1^2) - 4\gamma_1\delta_2\xi_1(1+\xi_2^2) + 4\gamma_2\delta_1\delta_2(1+\xi_1^2)(1+\xi_2^2)$.

References

Abe, T. and Pewsey, A. (2011). Sine-skewed circular distributions. *Statistical Papers*, **52**, 683–707.

Ameijeiras-Alonso, J. and Ley, C. (2022). Sine-skewed toroidal distributions and their application in protein bioinformatics. *Biostatistics*, **23**, 685–704.

Arashi, M., Nakhaei Rad, N., Bekker, A., and Schubert, W. D. (2021). Möbius transformation-induced distributions provide better modelling for protein architecture. *Mathematics*, **9**, 2749.

Ardia, D., Boudt, K., Carl, P., Mullen, K. M., and Peterson, B. G. (2011). Differential Evolution with DEoptim: An application to non-convex portfolio optimization. *R Journal*, **3**, 27–34. doi:10.32614/RJ-2011-005.

Fernández-Durán, J. J. and Gregorio-Domínguez, M. M. (2014). Modeling angles in proteins and circular genomes using multivariate angular distributions based on multiple nonnegative trigonometric sums. *Statistical Applications in Genetics and Molecular Biology*, **13**, 1–18.

Johnson, R. A. and Wehrly, T. (1977). Measures and models for angular correlation and angular–linear correlation. *Journal of the Royal Statistical Society: Series B (Methodological)*, **39**, 222–229.

Kato, S. and Pewsey, A. (2015). A Möbius transformation-induced distribution on the torus. *Biometrika*, **102**, 359–370.

Ley, C. and Verdebout, T. (2017). *Modern Directional Statistics*. CRC Press.

Mardia, K. V. (1975). Statistics of directional data. *Journal of the Royal Statistical Society: Series B (Methodological)*, **37**, 349–371.

Mardia, K. V. (2014). *Statistics of Directional Data*. Academic Press.

Ramachandran, G. N. and Sasisekharan, V. (1968). Conformation of polypeptides and proteins. *Advances in Protein Chemistry*, **23**, 283–437.

Rivest, L.-P. (1988). A distribution for dependent unit vectors. *Communications in Statistics-Theory and Methods*, **17**, 461–483.

Robert, C. P. and Casella, G. (1999). The metropolis—hastings algorithm. *In Monte Carlo Statistical Methods*. Springer, 231–283.

Sengupta, A. and Ong, S. H. (2014). A unified approach for construction of probability models for bivariate linear and directional data. *Communications in Statistics-Theory and Methods*, **43**, 2563–2569.

Wang, M.-Z. and Shimizu, K. (2012). On applying Möbius transformation to cardioid random variables. *Statistical Methodology*, **9**, 604–614.

Proceedings of the 63rd Annual Conference of the South African Statistical Association

Multivariate big data sampling for crop area coverage

Tshepiso Selaelo Rangongo, Inger Fabris-Rotelli and Renate Thiede

Department of Statistics, University of Pretoria, Pretoria, South Africa

Big data can result in more than sufficient information if used efficiently and effectively. Big data poses challenges in storage, management, processing, analysis and visualisation. Techniques for handling big data, specifically geospatial data, have advanced over the years. However, most require high computational power and time. The use of metadata is a solution. Metadata provides a descriptive, administrative, structural and statistical summary of data. This paper constructs metadata of a remote sensing image dataset for crop classification, and proposes a novel multivariate stratified sampling algorithm which selects the most informative images to minimise the number of images used for training. The proposed sampling algorithm performs effectively on a big spatial image dataset of crop types. The results are assessed by measuring the number of images sampled and as well as matching the proportionality of the population crop percentages.

Keywords: Metadata, Remote sensing, Sampling.

1. Introduction

The amount of data produced increases exponentially over time. In 2018, there were 33 trillion gigabytes of data produced in the world and this will grow to 175 trillion gigabytes of data in 2025[1]. Data is defined as individual facts, items of information or statistics. As much as the terms data and information have been used interchangeably, they are not necessarily the same. Data can be transformed to information when viewed in context or post-analysis[2]. Data is gradually increasing as it can now be collected by an increasing number of ways such as surveys and devices such as mobile devices, aerial devices, cameras, microphones and wireless sensor networks. The continuously increasing collection of data has led to what is known as big data. Big data is defined as data sets that are large and complex to deal with using traditional data processing software[3]. Big data can be considered as a bond that acts as an integration between human society, the physical world and cyberspace (Jin et al., 2015). Big data can be divided into two categories, namely data from the physical world, which can be obtained through scientific experiments and observations or sensors, and

Corresponding author: Inger Fabris-Rotelli (inger.fabris-rotelli@up.ac.za)

MSC2020 subject classifications: 62H11

[1] The Conversation, Science + Technology, The world's data explained,
https://theconversation.com/the-worlds-data-explained-how-much-were-producing-and-where-its-all-stored-159964

[2] Data vs. Information – Difference and Comparison, 2022,
https://www.diffen.com/difference/Data_vs_Information#google_vignette

[3] Big data, Wikipedia, The Free Encyclopedia, 2014, https://en.wikipedia.org/wiki/Big_data

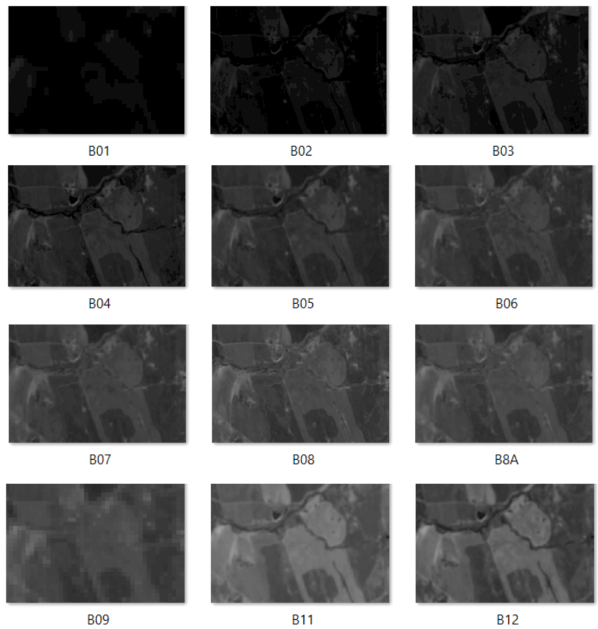


Figure 1. A single image tile from the data shown over 12 different bands.

data from human society which is acquired through human-computer interfaces and brain-computer interfaces (Cheng et al., 2014).

Big geospatial data falls under the category of data from the physical world. Geospatial data is information that describes events, objects or features associated with a location on or near the surface of the earth. Geospatial data can be obtained by remote sensing, ground surveying, laser scanning, mobile mapping, geo-tagged web contents and many more techniques. Geospatial data grows as the machinery used to capture it increases yearly. Big data was characterised by the three initial V's in 2001 by Laney (2001), namely volume, variety and velocity. Other V's were added as time went on such as value, veracity, variability and visualisation. Karimi (2014) showed that big geospatial data exhibits at least one of the 3 initial V's along with the other V's introduced later in time.

Two strategies that have been introduced and implemented for geospatial big data handling include parallel and distributed programming (Lee et al., 2014; Shekhar et al., 2012a,b) whereas others have suggested the use of functional programming concepts or languages. One suggested way of dealing with this is using metadata which is mentioned to be useful in cases of classification procedures (Li et al., 2016).

The data used herein is a big crop dataset intended for crop classification in South Africa (Fig. 1). As a developing country, most big data handling techniques are not implementable due to lack of resources such as computational power. This paper proposes the use of data handling techniques that are easily implementable. These include metadata which will be used to sample from. This paper proposes a multivariate sampling algorithm that uses crop area coverages, which is a novel concept in the literature. The sampling algorithm aims to minimise the number of images while maximising information obtained from those images. The multivariate aspect is a result of the variables taken

into account, namely the crop types, field areas and percentage cloud coverage per image.

Metadata is data that provides information about other data. Metadata provides content, type, quality, and creation information about another data set. For geospatial data, the metadata additionally contains a spatial component such as the extent of the surface of the earth that the data covers. Types of metadata include descriptive, structural, administrative, statistic, legal and reference metadata. Metadata makes data easier to document and discover, and reduces data duplication (Coulondre et al., 1998).

Big data tends to require a lot of storage, which makes it hard for people without enough computational power to work with. Metadata can be a solution that helps alleviate the storage requirement of big data, as it enables the navigation and summarisation of big data without needing to load an entire big dataset into memory.

An alternative used to alleviate having to read in all data is obtaining a good representation of the data through sampling. A sample is considered a good representation if there are certain characteristics of interest of the population that can be estimated with known accuracy. Examples of sampling techniques include random, systematic, stratified and clustering sampling. Since this research focuses on crop classification, stratified sampling works best because it requires that each unit must belong to only one stratum. Applications of stratified sampling in remote sensing include the detection of spatial variability amongst peach orchids to classify trees into homogenous strata with the aim of decreasing sampling size (Miranda et al., 2018) and estimation of crop area using stratified sampling in remote sensing (Jiao et al., 2006; Zhu and Zhang, 2013).

This paper proposes an algorithm that makes use of multivariate stratified sampling to obtain a sample that gives the best representation of the population. The multivariate population under consideration consists of a large database of remote sensing images of crop fields, for which each image has a varying number of fields, crop types and field sizes. First, the data summary is obtained in the form of a metadata dataframe, as explained in Section 2. The metadata itself is used to obtain an informative sample using the algorithm developed in Section 3. The aim of the algorithm is to achieve similar proportionality of crop types between the sample and the population. Section 4 evaluates the usefulness of the proposed algorithm and the effect of parameter choices. Section 5 discusses the results and Section 6 provides a conclusion.

2. The data

2.1 Data summary

The dataset to be used in this research is the Sentinel-2 time series data for the Western Cape province in South Africa. This dataset is freely accessible on the Radiant MLHub website and was generated by Radiant Earth Foundation and the Western Cape department of Agriculture in 2021[5]. Radiant MLHub is a cloud-based open library of Earth Observation data including land cover, wildfire, floods, tropical storms, building footprints and crop datasets. The dataset to be used is a crop dataset that has 12 bands in the near infrared, short wave infrared and visible part of the spectrum and the 13th image type (CLM) which gives the cloud coverage on a tile image. The time series is provided every

[5] Radiant Earth Foundation, Crop Type Classification Dataset for Western Cape, South Africa, 2021, https://doi.org/10.34911/rdnt.j0co8q

five days from the 1st of April until the 27th of November (48 dates) of 2017. Figure 1 shows the 12 bands images of one area of land with tile ID 1114 taken by the Sentinel-2 satellite on the 28th of October 2017.

From Figure 1, B01 is the coastal aerosol band with a resolution of 60m. B02, B03 and B04 are the blue, green and red bands all with the same resolution of 10m. B05, B06, B07 and B8A are the vegetation red edge bands with the same resolution of 20m. B08 is the near infrared band with resolution of 10m. B09 is the water vapour with resolution of 60m whereas B11 and B12 are the short wave infrared bands with resolution of 20m.

Each image is an area of land made up of crop fields. Each field contains only one type of crop. The dataset consists of 9 types of crops, namely fallow, canola, wheat, wine grapes, weeds, small grain gazing, lucerne/medics, planted pastures (perennial) and rooibos. Figure 2 shows the different fields and labels of the area covered in tile ID 1114. This area is made up of 25 fields that contain 6 different crop types, namely lucerne, planted pastures, fallows, small grain grazing, wheat and canola.

Each area of land (2650 locations) was captured every five days (×48) through 12 bands of the electromagnetic spectrum and the cloud coverage image (×13), such that the whole data set is made up of 1 653 000 images. The area of interest is 17 367 040 km of land of which 6 581 396 km (roughly 38%) has been labelled. The labelled images constitute the portions that will be considered in assessing the accuracy of sampling. The area coverages of the crop types in each image and field has also been calculated and this will help calculate the proportion of the crop types in the population. Summing all the area coverages in each image gives the overall area coverage of each crop type, which helps to determine the proportions of the crop types in the population. Figure 3 shows the proportions of the crop types using their area coverage, so that the one with the highest proportion is the one with the highest crop coverage. As can be seen from Figure 3, wheat has the highest proportion with 23.08% followed by small grain grazing with 14.146% with the least being canola with a percentage of 3.405%.

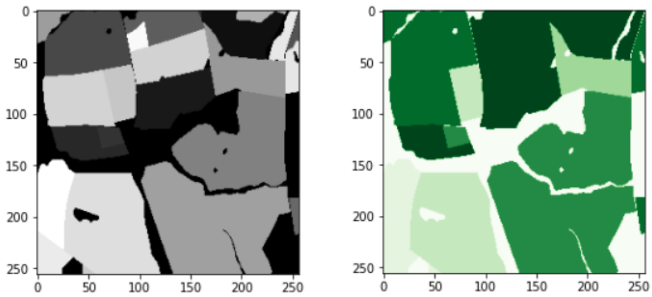
2.2 Metadata construction

To avoid loading all 1 653 000 images of data (approximately 45.15GB) we construct metadata so that only relevant images are loaded into memory. The metadata construction was performed in Python with the code accessible at Figshare[6]. The structure of the metadata consists of three categories, namely general information, tile ID information and image information.

General information includes properties that all images share regardless of location or date captured, namely the satellite used to capture the image, the type of image, data licence, data providers and image size since all images are the same size. This information is given in the images STAC (SpatioTemporal Asset Catalogs) files. Figure 4 is an illustration of what the general information is for each image.

Tile ID information is information that has been used to differentiate between the different areas of land such as tile ID, the spatial extent of the area captured, and the number of fields along with the crop types they contain. Figure 5 shows this. An image of another area of land, i.e. with a different tile ID, will not have the same information as the one in Figure 5. The spatial extent, also referred to

[6] Metadata construction, Figshare, Python code, https://doi.org/10.25403/UPresearchdata.20349426



(a) Fields (b) Labels

Figure 2. An illustration of the fields and crop types of the area with tile ID 1114.

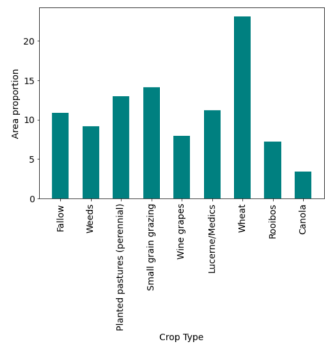


Figure 3. Proportions of the crop types using area coverage.

Itype	Image\tif
Satellite	Sentinel-2
Coverage	Western Cape, South Africa
License	CC-BY-4.0
Providers	Name: Radiant Earth Foundation url: https://radiant.earth'
Temporal	Interval: [2017-04-01 T00:00:00+00Z, 2017-11-30 T00:00:00+00Z]
Size	256*256

Figure 4. General information that applies on all images.

Tile ID	1114
Bounding box	[18.514546656, 33.6848043291, 18.542797031, -33.6611841904]
Number of fields	25
Number of crop types	6
Crop type proportion	['No Data: 17.278%', 'Lucerne/Medics: 4.997%', 'Planted pastures: 12.914%', 'Fallow: 4.218%', 'Small grain grazing: 27.017%', 'Wheat: 13.843%', 'Canola: 19.734%']

Figure 5. Tile ID information that applies to all images of the same area of land.

as the bounding box, number of fields and crop proportions will also differ.

Information associated with each image is unique for each image, such as the date and time and cloud coverage on that date. With the three categories brought together, metadata in the form of a database can be created. The database is a Pandas (Pandas Development Team, 2020) dataframe where the rows are indexed by the tile ID and the date the images were captured. From the database itself, one can obtain the structure of the data, the description of the data as well as the administration involved in publishing the data. The database is useful because performing procedures such as sampling and classification will not require loading and reading all the images into memory.

3. A multivariate stratified sampling algorithm

This section covers the proposed sampling algorithm that makes use of multivariate stratification.

Let N be the number of images in the population and M be the number of different crop types. Let n be the sample size of images and N_i be the number of images that contain crop type i in the population. We notate A^i_{pop} and A^i_{samp} as the area coverages of crop type i in the population and sample respectively. Thus making $\mathbf{A}_{pop}$ and $\mathbf{A}_{samp}$ vectors of area coverages of the M crop types in the population and the sample respectively, and $\mathbf{V}_{pop}$ and $\mathbf{V}_{samp}$ vectors containing the proportions of the M crop types in terms of area coverage in the population and sample respectively:

$$\mathbf{V}_{pop}=\begin{bmatrix} V^1_{pop} \\ V^2_{pop} \\ \vdots \\ V^M_{pop} \end{bmatrix}, \mathbf{V}_{samp}=\begin{bmatrix} V^1_{samp} \\ V^2_{samp} \\ \vdots \\ V^M_{samp} \end{bmatrix}, \mathbf{A}_{pop}=\begin{bmatrix} A^1_{pop} \\ A^2_{pop} \\ \vdots \\ A^M_{pop} \end{bmatrix} \text{ and } \mathbf{A}_{samp}=\begin{bmatrix} A^1_{samp} \\ A^2_{samp} \\ \vdots \\ A^M_{samp} \end{bmatrix}.$$

We propose an algorithm to obtain a sample from the population ensuring that the proportion between the population and the sample are similar while minimising the number of images sampled. The proportions are calculated in terms of area coverage. The area coverages of the M crop types in the population ($\mathbf{A}_{pop}$) should be directly proportional to the area coverages of the crop types in the sample ($\mathbf{A}_{samp}$). Ideally, the desired area coverages in the sample, $\mathbf{A}_{samp}$ should be $\frac{n}{N} \times \mathbf{A}_{pop}$. Mathematically, the aim is to show the following equation holds for some small ϵ:

$$||\mathbf{V}_{pop} - \mathbf{V}_{samp}|| \leq \epsilon. \tag{1}$$

The algorithm is separated into two main steps. The first step samples by looking at the most represented crop type in the population. The second uses the partial sample from the first stop and focuses on the least represented crop type. This is done iteratively until all crop types are represented, while satisfying equation (1).

1. Calculate $\mathbf{A}_{pop}$, the area coverages of the M crop types in the population. From this, compute $\mathbf{V}_{pop}$ the proportions of the crop types in the population.

2. Let $cropA$ be the crop type in $\mathbf{V}_{pop}$ with the highest proportion, such that $cropA = \underset{i}{\text{argmax}}(V^i_{pop})$.

3. Extract a sub-dataframe $newA\_df$ from the dataframe containing metadata such that $newA\_df$ only contains images that have $cropA$ such that N_A is the length of $newA\_df$.

4. Order the images $I_{(A,1)}, I_{(A,2)}, ..., I_{(A,N_A)}$ in $newA\_df$ in descending order according to the area coverage of $cropA$ in each image. The new order will now be $I'_{(A,1)}, I'_{(A,2)}, ..., I'_{(A,N_A)}$.

5. Introduce parameter $cropAmax$. This ensures that when other crop types are considered, the desired area coverage A^i_{samp} of the previously considered crop type is not exceeded.

6. Include images $I'_{(A,1)}, I'_{(A,2)}, ..., I'_{(A,n_A)}$ such that the area coverage of $cropA$ in the $n_A \leq N_A$ images is $cropAmax$ of the desired $cropA$ sample area coverage. These n_A images are included in the sample.

7. From thus far n_A sampled images, the area coverages of the other crop types are also captured and stored in $\mathbf{A}_{samp}$ as the corresponding $\mathbf{A}^i_{samp}$.

8. Considering the current $\mathbf{V}_{samp}$, let $cropB$ be the least crop type currently represented by the sample, such that $cropB = \underset{i}{\text{argmin}}(V^i_{pop})$. Extract another sub-dataframe $newB\_df$ that contains images with $cropB$ in them but excluding the n_A already in the current sample. Let $N_B{}^*$ denote the number of these images which may be less or equal to N_B depending on whether or not the n_A sampled images contain $cropB$.

9. Rearrange the images $I_{(B,1)}, I_{(B,2)}, ..., I_{(B,N_B{}^*)}$ in new_df in descending order according to the area coverage of $cropB$ in each image such that the new order is $I'_{(B,1)}, I'_{(B,2)}, ..., I'_{(B,N_B{}^*)}$.

10. Introduce another parameter $cropBmax$ which works similar to $cropAmax$ except now it is imposed on the desired $cropB$ area coverage in the sample. Denote the number of images that make up $cropBmax$ of the remaining desired $cropB$ area coverage in the sample by n_B.

11. Capture the area coverages of all the crop types in the n_B images and add to the ones from the previously sampled images, in $\mathbf{A}_{samp}$. The total $n_A + n_B$ now become the updated sample size with images $I'_{(A,1)}, I'_{(A,2)}, ..., I'_{(A,n_A)}, I'_{(B,1)}, I'_{(B,2)}, ..., I'_{(B,n_B)}$ being the sample.

12. Repeat Step (8)-(9) for the next least represented crop type. Step (10) is modified to (10*) such that the $crop(i)max$ parameter is not included any longer i.e. we want to make up the remaining desired area coverage. Iterate step (8), (9) and (10*) $M-2$ times to account for the remaining crop types. Each time an iteration occurs, the previously n_i selected images are not considered in the next iteration as they already been added to the sample.

13. After the iterations, the final sample will now be the images $I'_{(A,1)}, I'_{(A,2)}, ..., I'_{(A,n_A)}, I'_{(B,1)}, I'_{(B,2)}, ..., I'_{(B,n_B)}, .., I'_{(M,1)}, I'_{(M,2)}, ..., I'_{(M,n_M)}$ and from the final $\mathbf{A}_{samp}$, compute $\mathbf{V}_{samp}$, the proportions of the crop types in the sample.

4. Results

This section provides results of the effect of the different values of $cropAmax$ and $cropBmax$ on the sample area coverages, the number of images sampled and the Euclidean norm.

The implementation of the sampling algorithm is done in Python and the notebook containing the code for the algorithm is available on Figshare[7]. We investigate first the role of the parameters $cropAmax$ and $cropBmax$ on the values of n_A and n_B. $cropA$ from our dataset is wheat and the length of $newaA\_df$ is N_A=106. The values of $cropAmax$ used are [0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1]. To illustrate the effect of $cropAmax$ on n_A, we compare the different values. For example, when $cropAmax$=0.4, then n_A is 12, and when $cropAmax$=1, then n_A is 33. Table 1 gives an example of the percentages of the desired $\mathbf{A}_{samp}$ area coverages achieved from the n_A image.

Using $cropAmax$ of 0.4, the partial sample is made up of n_A=12 images. This is the selected number of images containing $cropA$ in the first iteration. $cropB$, which is the least represented crop in the partial sample, is weeds. The number of images that contain $cropB$ is N_B=N_B^*=1428 images because the previously sampled images do not include weeds. Comparing two values of $cropBmax$, 0.4 and 1, in combination with the $cropAmax$ of 0.4, we found n_B to be 16 and 38 respectively. This is the selected number of images with $cropB$ added to the partial sample. Table 2 shows how different values of $cropBmax$ increases the area coverages of each respective crop.

Using $cropAmax$=$cropBmax$=0.4, it can be seen from Table 2 that the next crop to be considered is wine grapes followed by planted pastures with the last iteration (M^{th}) focusing on canola. Table 3 illustrates how the area coverages increase over the iterations. From Table 3, the overall sample size relative to the population is 10.277%.

[7] Sampling algorithm, Figshare, Python code, https://doi.org/10.25403/UPresearchdata.20444061

Table 1. Achieved area coverage percentage using two different $cropAmax$ parameters.

	Achieved area coverage (%)	
Crop Type	$cropAmax = 40\%$	$cropAmax = 100\%$
Wheat	4.06	10.1
Weeds	0.0	0.14
Canola	1.71	4.15
Wine grapes	0.12	0.15
Fallow	0.11	0.26
Rooibos	0.01	0.01
Planted Pastured (perennial)	0.057	0.15
Lucerne/Medics	0.24	1.99
Small grain grazing	0.34	1.54

Table 2. Achieved area coverage percentage using two different $cropBmax$ parameters.

	Achieved area coverage (%)			
	$cropAmax = 40\%$		$cropAmax = 100\%$	
Crop Type	$cropBmax = 40\%$	$cropBmax = 100\%$	$cropBmax = 40\%$	$cropBmax = 100\%$
Wheat	4.11	4.159	10.1	10.164
Weeds	4.286	10.3789	0.246	0.893
Canola	1.71	1.714	4.15	4.15
Wine grapes	0.12	0.174	0.15	0.152
Fallow	0.53	2.187	0.597	1.05
Rooibos	0.325	0.524	4.391	10.64
Planted Pastured (perennial)	0.135	0.290	0.231	0.277
Lucerne/Medics	0.24	0.248	1.99	1.99
Small grain grazing	0.446	0.599	1.54	1.639

Table 6 consists of the sample size achieved given different values of $cropAmax$ and desired sample sizes. Note that the sample sizes are calculated using area coverages and not number of images, and this is before introducing $cropBmax$, which is the same as taking $cropBmax$=1.

Table 4 contains the number of images sampled given the different values of $cropAmax$ and the different desired sample size (area-wise). This is similar to Table 6 where the parameter imposed on the second considered crop type is not included.

Table 5 contains results of achieved sample sizes given different desired sample sizes and values of $cropBmax$. Now the parameter imposed on $cropAmax$ is kept constant ($cropAmax$=1) to illustrate the effect of $cropBmax$.

The Euclidean norm is computed to quantify the difference between the area coverages of the crop types in the sample to those in the population. Table 7 gives the Euclidean norm between the different samples for different values of $cropAmax$ not considering the effect of $cropBmax$ (i.e. setting $cropBmax$=1).

Figures 6 and 7 gives the effect of the different values of $cropAmax$ and $cropBmax$ using the Euclidean norm calculated from the different desired sample sizes and the population. The lighter

Table 3. Achieved area coverage percentages over the remaining iterations.

	Achieved area coverage (%)						
Crop Type	3rd iteration (Wine grapes)	4th iteration (Planted Pastures)	5th iteration (Rooibos)	6th iteration (Lucerne Medics)	7th iteration (Fallow)	8th iteration (Small grain)	9th iteration (Canola)
Wheat	4.11	4.54	4.60	7.73	7.8	8.79	8.83
Weeds	4.34	4.6	5.20	5.20	6.22	6.63	6.63
Canola	1.71	2.68	2.68	4.4	4.42	4.80	10.92
Wine grapes	10.73	11.43	11.43	11.44	11.44	11.66	11.66
Fallow	0.85	0.96	1.75	1.8	10.26	10.55	10.61
Rooibos	0.33	0.33	10.31	10.31	10.1	10.34	10.79
Planted Pastured (perennial)	0.19	10.24	10.37	10.66	10.80	11.65	11.9
Lucerne Medics	0.38	1.20	1.2	10.2	10.25	10.53	11.02
Small grain grazing	0.45	1.49	1.59	2.18	2.2	10.32	10.38

Table 4. Number of images per desired sample size and $cropAmax$.

	$cropAmax$									
Sample size	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
10%	129	131	132	134	137	138	140	143	145	146
20%	269	273	271	275	278	284	286	290	294	297
30%	430	435	436	440	444	447	450	458	462	463
40%	620	617	613	616	621	627	629	630	634	635
50%	813	798	800	810	814	822	820	822	828	839
60%	1026	1033	1006	1009	1020	1019	1022	1045	1040	1037
70%	1248	1253	1238	1240	1258	1259	1263	1251	1249	1247
80%	1484	1486	1489	1496	1490	1513	1508	1501	1496	1471
90%	1799	1801	1803	1800	1816	1826	1824	1814	1785	1798
100%	2646	2646	2646	2646	2646	2646	2646	2646	2646	2650

Table 5. Achieved sample size per desired sample size and $cropBmax$.

	$cropBmax$									
Sample size	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
10%	10.3	10.4	10.6	10.7	10.8	10.9	11.0	11.1	11.1	11.3
20%	20.3	20.6	20.8	20.9	21.2	21.4	21.6	21.8	22.0	22.2
30%	30.1	30.5	30.9	31.3	31.7	31.9	32.1	32.5	32.7	32.9
40%	39.9	40.3	40.7	41.2	41.5	41.3	42.2	42.6	42.9	43.4
50%	50.7	51.1	51.6	52.0	52.4	52.8	52.9	53.2	53.4	53.6
60%	61.1	61.5	61.9	62.5	62.7	62.8	63.0	63.4	63.6	63.8
70%	70.6	71.1	71.6	72.0	72.2	72.5	72.6	72.9	73.3	73.6
80%	78.5	78.3	78.9	79.6	80.2	81.0	81.5	82.2	82.7	83.3
90%	90.0	90.4	90.2	90.6	90.9	91.1	91.3	91.5	91.8	91.9
100%	99.2	99.2	99.2	99.3	99.4	99.4	99.5	99.6	99.8	100.0

Table 6. Achieved sample size per desired sample size and $cropAmax$.

	$cropAmax$									
Sample size	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
10%	10.6	10.7	10.7	10.8	10.9	10.9	11.1	11.2	11.3	11.3
20%	21.0	21.2	21.0	21.2	21.3	21.6	21.7	21.8	22.0	22.2
30%	31.9	32.3	32.2	32.3	32.4	32.3	32.3	32.8	32.9	32.9
40%	43.1	43.3	43.0	43.1	43.3	43.5	43.4	43.3	43.4	43.4
50%	54.0	53.4	53.3	54.4	54.6	54.5	53.9	53.8	53.9	53.6
60%	64.4	64.7	63.6	64.9	64.4	64.0	64.0	63.9	63.6	63.8
70%	74.6	74.9	75.8	75.4	74.0	74.2	73.8	73.5	73.4	73.6
80%	83.9	84.0	84.1	83.8	83.2	83.2	82.9	82.8	82.8	83.3
90%	91.9	92.0	92.0	92.1	92.1	91.5	91.5	91.4	92.0	91.9
100%	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0

Table 7. Euclidean norm between population and sample proportions per desired sample size and $cropAmax$.

	$cropAmax$									
Sample size	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
10%	5.18	4.62	4.16	3.83	3.38	3.23	3.61	4.04	4.68	4.99
20%	9.33	8.22	6.39	5.67	4.86	5.28	5.6	6.44	7.44	8.72
30%	13.33	12.01	10.24	9.25	8.72	8.06	8.05	9.42	10.32	11.44
40%	15.99	14.35	12.47	11.81	11.24	11.55	11.41	11.8	12.54	13.88
50%	18.3	17.45	16.45	15.88	15.8	15.24	13.86	14.21	15.17	15.49
60%	20.02	19.32	17.28	17.84	16.58	15.67	15.41	14.08	14.77	16.64
70%	20.44	20.42	21.15	19.46	16.02	16.26	13.71	13.57	14.59	16.52
80%	17.93	17.83	17.25	15.42	14.28	12.36	11.75	12.14	12.99	16.68
90%	10.43	10.42	10.11	10.48	10.04	5.73	6.0	6.11	10.1	9.81
100%	0.02	0.02	0.02	0.02	0.02	0.02	0.03	0.03	0.03	0.02

and larger the dots, the higher the Euclidean norm, and the darker and smaller the dots, the smaller the Euclidean norm.

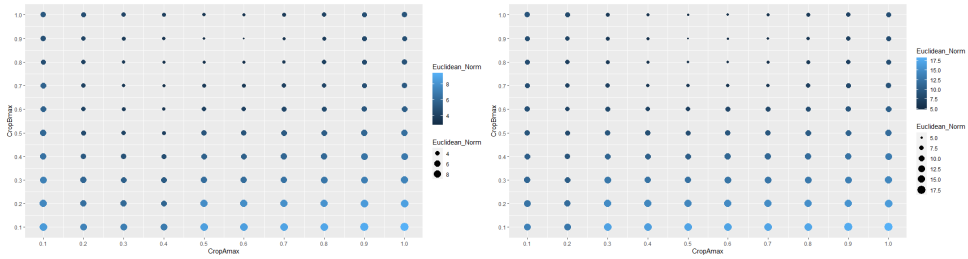
Figure 8 illustrates how the highest and lowest Euclidean norms change with sample size. The averages of the Euclidean norms in each sample are also plotted.

5. Discussion

The algorithm works in such a way that the proportions of the crop types in the sample should be similar to that of the population, while minimising the number of images sampled.

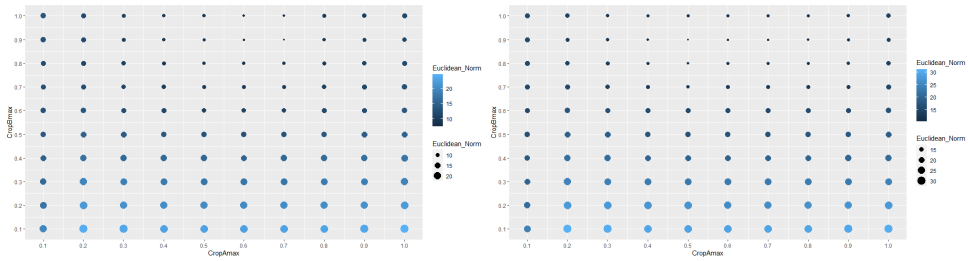
Table 1, illustrates how the parameter $cropAmax$ works, with wheat being the crop of interest. If $cropAmax$ is set to 100%, then the achieved area coverage is already at the desired level of 10% in the first iteration. The area coverage will increase when considering other crop types. The same is observed for the parameter $cropBmax$ in Table 2. The desired sample size of weeds (crop B) is already achieved at the second iteration if $cropBmax$ is set to 100%. Using this information, one can deduce that using high values of $cropAmax$ and $cropBmax$ will lead to over-sampling. From Table 2, we see that if a $cropAmax$ of 40% is chosen, then the least represented in the sample is weeds, but if a $cropAmax$ of 100% is chosen, rooibos is the least represented in the sample and will be considered in the second iteration.

Table 3 shows how the area coverages increase as other crop types are considered throughout the remaining iterations. Most of the crop types achieved the desired 10% area coverage. Only wheat and weeds were under-sampled. The reason for this might be the choice of $cropAmax$ (imposed on wheat) and $cropBmax$ (imposed on weeds); one might argue that 40% is small and a higher value such as 60% might avoid under-sampling of the first two crops. Table 6 shows the effect of the parameter $cropAmax$ on the achieved sample area coverage given certain sample sizes. For smaller sample sizes, the achieved sample area coverages seem to increase with higher values of $cropAmax$. Note that the effect of $cropBmax$ is not included in this instance. If one assessed the level of accuracy by comparing the desired sample sizes to the achieved sample area coverages, then the algorithm would pass this accuracy test. This is because the highest difference between the two



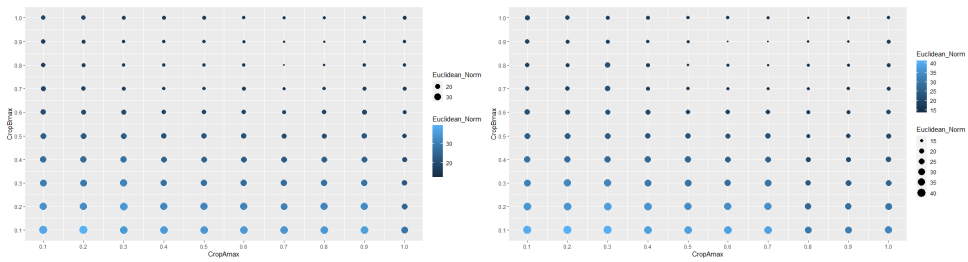
(a) 10% sample size

(b) 20% sample size



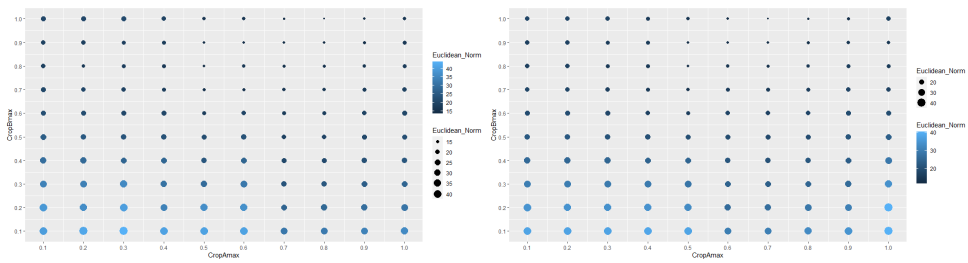
(c) 30% sample size

(d) 40% sample size



(e) 50% sample size

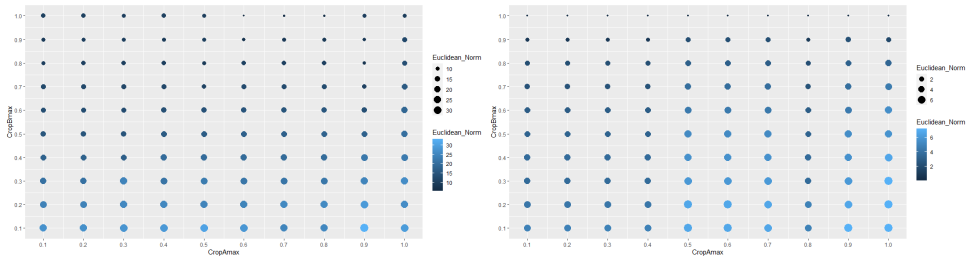
(f) 60% sample size



(g) 70% sample size

(h) 80% sample size

Figure 6. Euclidean norm between (10%–80%) sample and population.



(a) 90% sample size (b) 100% sample size

Figure 7. Euclidean norm between (90%–100%) sample and population.

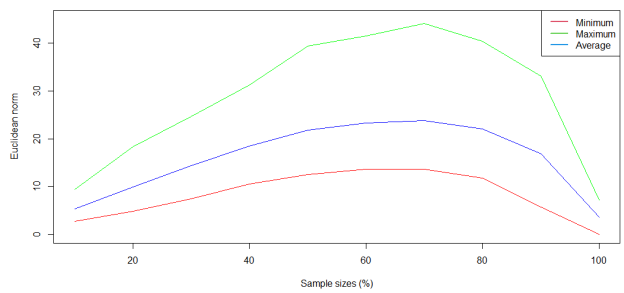


Figure 8. The bounds and averages of Euclidean norms per sample size.

is 5.8, and this is for a sample size of 70%. Looking at the smaller samples, which is ideally what we want to work with, the difference can be considered trivial. The achieved sample area coverages are always higher than the desired sample sizes in this instance. This is because, as much as we are considering area coverages, we consider a whole image and not individual fields. If from previously selected images we have a sample area coverage of 9.1% for a specific crop type, and an additional image were selected containing 1.6% area coverage for that crop type during an iteration to meet the desired 10% sample size, then all the area will be considered and not just the required (10%–9.1%). This will result in 10.5% area coverage being achieved.

Since the aim of the algorithm is to minimise the number of images sampled, Table 4 gives a summary of how the number of images sampled changes according to desired sample size and the parameter $cropAmax$. For lower sample sizes, the number of images sampled increases as the $cropAmax$ parameter increases, which corresponds to the result from Table 6. For a 10% desired sample size, an average of 138 images is sampled, which makes up 5% of the total number of images (2650). While simple random sample (SRS) would result in 10% of all the images, the proposed algorithm would get the same information area-wise using 5%. The algorithm ensures that the most information is achieved with less images. For lower sample sizes (10%–40%), roughly (48%–60%) of the images that would be selected using SRS are obtained using the proposed algorithm. Even for a sample size of 90%, roughly 75% of images selected using a SRS approach are selected using this algorithm. This is due to the fact that some of the fields in the images were not labelled (62% of area

is not labelled), so having more images does not necessarily mean having more information.

Table 5 shows how the area coverages achieved changes with $cropBmax$, when the effect of $cropAmax$ is nullified by setting it to 100%. In this instance, we have that the achieved area coverages increase with increasing values of $cropBmax$ for all different sample sizes. This is different to when we were considering the effect of $cropAmax$ only, where this is was true for only smaller sample sizes. We see that the differences in this instance are lower compared to Table 4. The highest difference decreased from 5.8 to 3.6 with the lowest being 0.0. The highest difference, similar to when only considering $cropAmax$ (Table 4), is from a high sample size of 70%.

To compute the difference between the proportions of the crop types between sample and the population, the Euclidean norm is considered. Only the $cropAmax$ parameter is considered in this instance. The Euclidean norm is at its smallest at 100% sample size as we expect it to be. For lower values of sample size, the Euclidean norm is at its lowest when $cropAmax$ is at 50% and 60%. This might be that choosing lower values of $cropAmax$ leads to under-sampling and higher values of $cropAmax$ lead to oversampling. Figures 6 and 7, looking at the 10% sample size, higher values of $cropAmax$ and lower values of $cropBmax$ give high Euclidean norm values, whereas low and high values of $cropAmax$ give high Euclidean norm values. As $cropBmax$ increases, the Euclidean norm decreases. The lowest Euclidean norms are achieved when $cropAmax$ is contained in $(0.4, 0.7)$ and $cropBmax$ is in $(0.6, 0.9)$. Higher values of $cropBmax$ and middle values of $cropAmax$ tend to give the smallest Euclidean values in the smaller sample sizes. The Euclidean norm is at its lowest when $cropBmax$=0.9 and $cropAmax$ is between 0.5 and 0.7.

Note that the lowest value of the Euclidean norm in a sample size of 10% and 20% are not the same, but are 2.79 and 4.82 respectively. From Figure 8, the lowest Euclidean norm is achieved when the sample size is 100%, with the second lowest being at a 10% sample size followed by 20%. As the sample sizes increases, so does the range of Euclidean norms (this is true until after sample size of 70%). For smaller sample sizes, the Euclidean norm range is quite small. The smallest range is (2.79,8.4) when the sample size is 10% and the largest is achieved when the sample size is 70% with (13.57,44.06). A sample size of 70% does not only give the highest Euclidean norms, but also the highest difference in terms of sample size and achieved area coverage as already discussed.

6. Conclusion

This paper proposes a novel multivariate stratified sampling algorithm that selects the most informative images in order to minimise the number of images sampled. Two parameters were introduced to control the number of images sampled. The algorithm samples using metadata of the real dataset which minimises the computational power and time. The data used is a crop dataset intended for classification purposes. The algorithm gave good results on this dataset, and was able to extract the same information from half the number of images as simple random sampling. Depending on the choice of the parameters, the achieved sample size area wise is often slightly higher than the choice of sample size. The Euclidean norm was computed to investigate how the proportionality changes from the population to the sample. For smaller sample sizes, the Euclidean norm is lower for medium values of the parameter imposed on the first considered crop and for high values of the parameter imposed on the second considered crop. The proposed sampling algorithm for multivariate image sampling provides a solution to the problem of handling big geospatial data. The effect of training

a machine learning algorithm on a sample of data can be investigated. This will be considered as future research.

Acknowledgements. This work is based on the research supported in part by the National Research Foundation of South Africa (Grant Number: 137785). Opinions expressed and conclusions arrived at, are those of the authors and are not necessarily to be attributed to the NRF. Funding received towards Masters in eScience from the NEPTTP Bursary programme is also acknowledged.

References

Cheng, X.-Q., Jin, X. L., Wang, Y., Guo, J., Zhang, T., and Li, G. (2014). Survey on big data system and analytic technology. *Journal of Software*, **25**, 1889–1908.

Coulondre, S., Libourel, T., and Spéry, L. (1998). Metadata and GIS. *In Proceedings of GIS PlaNET'98, the 1st International Conference and Exhibition on Geographic Information*.

Jiao, X. F., Yang, B. J., and Pei, Z. Y. (2006). Paddy rice area estimation using a stratified sampling method with remote sensing in china. *Transactions of the Chinese Society of Agricultural Engineering*, **22**, 105–110.

Jin, X., Wah, B. W., Cheng, X., and Wang, Y. (2015). Significance and challenges of big data research. *Big Data Research*, **2**, 59–64.

Karimi, H. A. (2014). *Big Data: Techniques and Technologies in Geoinformatics*. CRC Press.

Laney, D. (2001). 3D data management: Controlling data volume, velocity and variety. URL: *https://blogs.gartner.com/doug-laney/files/2012/01/ad949-3D-Data-Management-Controlling-Data-Volume-Velocity-and-Variety*

Lee, K., Ganti, R. K., Srivatsa, M., and Liu, L. (2014). Efficient spatial query processing for big data. *In Proceedings of the 22nd ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. 469–472.

Li, S., Dragicevic, S., Castro, F. A., Sester, M., Winter, S., Coltekin, A., Pettit, C., Jiang, B., Haworth, J., Stein, A., and Cheng, T. (2016). Geospatial big data handling theory and methods: A review and research challenges. *ISPRS Journal of Photogrammetry and Remote Sensing*, **115**, 119–133.

Miranda, C., Santesteban, L. G., Urrestarazu, J., Loidi, M., and Royo, J. B. (2018). Sampling stratification using aerial imagery to estimate fruit load in peach tree orchards. *Agriculture*, **8**, 78.

Pandas Development Team (2020). pandas-dev/pandas: Pandas. doi:10.5281/zenodo.3509134.

Shekhar, S., Evans, M. R., Gunturi, V., Yang, K., and Cugler, D. C. (2012a). Benchmarking spatial big data. *In Specifying Big Data Benchmarks*. Springer, 81–93.

Shekhar, S., Gunturi, V., Evans, M. R., and Yang, K. (2012b). Spatial big-data challenges intersecting mobility and cloud computing. *In Proceedings of the Eleventh ACM International Workshop on Data Engineering for Wireless and Mobile Access*. 1–6.

Zhu, S. and Zhang, J. (2013). Provincial agricultural stratification method for crop area estimation by remote sensing. *Transactions of the Chinese Society of Agricultural Engineering*, **29**, 184–191.