A new family of distributions: properties and applications

Gauss M. Cordeiro[a] , Maria do Carmo S. Lima[a] and Pedro Rafael D. Marinho[b]

Abstract

This article introduces a new family by combining the Marshall and Olkin-G and Gamma-G classes. The family has only two extra shape parameters and can be a better model than other existing classes of distributions. Simulations are performed to verify the consistency of the estimators. Its flexibility is shown using two real data sets.

Key Words: Distribution family; mathematical properties; simulations; applications.

[a] Department of Statistics, Federal University of Pernambuco, Recife, Brazil.
[b] Department of Statistics, Federal University of Paraíba, João Pessoa, Brazil.
E-mail:*gausscordeiro@de.ufpe.br, maria@de.ufpe.br, pedro.rafael.marinho@gmail.com*

1 Introduction

The mechanism by adding shape parameters to a baseline distribution has proved to be useful to make the generated distributions more flexible especially for studying tail properties than existing distributions and for improving their goodness-of-fit statistics to the data under study. Many special distributions in these families are discussed by Tahir and Nadarajah (2015).

Let $G(x)$ be the cumulative distribution function (CDF) of a baseline distribution and $g(x) = dG(x)/dx$ be the corresponding probability density function (PDF) depending on a parameter vectot η. A generalized family are presented with two additional shape parameters by transforming the CDF $G(x)$ according to two sequential important generators. These families are important for modeling data in several engineering areas.

The CDF of the Marshall and Olkin's (1997) (MO-G) family (for $\theta > 0$) is

$$F_{\text{MO-G}}(x) = \frac{G(x)}{\theta + (1-\theta)G(x)} = \frac{G(x)}{1-(1-\theta)[1-G(x)]}, \quad x \in \mathbb{R}. \tag{1}$$

The density function corresponding to (1) has the form

$$f_{\text{MO-G}}(x) = \frac{\theta\, g(x)}{[\theta + (1-\theta)G(x)]^2}. \tag{2}$$

For $\theta = 1$, $f_{\text{MO-G}}(x)$ is equal to $g(x)$. Equation (2) represents the PDF of the minimum of n iid random variables having density $g(x)$, say $T_1, \cdots, T_N$, where N

has a geometric distribution with probability parameters θ and θ^{-1} if $0 < \theta < 1$ and $\theta > 1$, respectively.

Tahir and Nadarajah (2015, Table 2) presented thirty distributions belonging to this family. It is easily generated from the baseline quantile function (QF) by $Q_{\text{MO-G}}(u) = Q_G\left(\theta u\left[\theta u + 1 - u\right]\right)$ for $u \in (0, 1)$.

Marshall and Olkin considered the exponential and Weibull distributions for the baseline G and derived some structural properties of the generated distributions. The special case that G is an exponential distribution refers to a two-parameter competitive model to the Weibull and gamma distributions.

The CDF of the gamma-G (Γ-G) family (Zografos and Balakrishnan, 2009) is

$$F_{\Gamma\text{-G}}(x) = \gamma_1\left(a, -\log\left[1 - G(x)\right]\right), \quad x \in \mathbb{R}, \tag{3}$$

where $a > 0$ is an extra shape parameter, $\gamma_1(a, z) = \gamma(a, z)/\Gamma(a)$ is the incomplete gamma function ratio and $\gamma(a, z) = \int_0^z t^{a-1}\,\mathrm{e}^{-\mathrm{t}}\mathrm{dt}$.

Then, the PDF of the Γ-G family can be expressed as

$$f_{\Gamma\text{-G}}(x) = \frac{1}{\Gamma(a)}\left\{-\log[1 - G(x)]\right\}^{a-1}\, g(x). \tag{4}$$

Each new Γ-G distribution follows from a given baseline G. For $a = 1$, the Γ-G family reduces to G. If Z is a gamma random variable with unit scale parameter and shape parameter $a > 0$, then $W = Q_G(1 - \mathrm{e}^{\mathrm{Z}})$ has density (4). So, the Γ-G distribution is easily generated from the gamma distribution and the QF of G.

The remaining of the paper is addressed as follows. Section 2 introduces the *Marshall and Olkin-Gamma-G* (MO-Γ-G) family and presents some special models. The maximum likelihood estimates (MLEs) of the parameters of the new family is addressed in Section 3. Some simulations ate performed in Section 4 to estimate the biases of the MLEs. Two empirical applications illustrate the potentiality of the proposed family in Section 5. A variety of theoretical properties are derived in Section 6. Some conclusions remarks are offered in Section 7.

2 The New Family

By combining Equations (1) and (3), the CDF of the random variable $X \sim$MO-Γ-G representing the new family is defined by

$$F_X(x) = \frac{\gamma_1\left(a, -\log\left[1 - G(x)\right]\right)}{\theta + (1-\theta)\gamma_1\left(a, -\log\left[1 - G(x)\right]\right)}, \quad x \in \mathbb{R}. \tag{5}$$

By differentiating (5), the PDF of X follows as

$$f_X(x) = \frac{\theta\left\{-\log[1 - G(x)]\right\}^{a-1}\, g(x)}{\Gamma(a)\,\left\{\theta + (1-\theta)\gamma_1\left(a, -\log\left[1 - G(x)\right]\right)\right\}^2}. \tag{6}$$

The density (6) can be interpreted from a sequence of N iid random variables, say $Z_1, \cdots, Z_N$, each one having a gamma density unit scale and shape $a > 0$, assuming that N (is not fixed) has a geometric distribution with probabilities θ and θ^{-1} for $0 < \theta < 1$ and $\theta > 1$, respectively. By transforming the Z_i's via the baseline QF by $W_i = Q_G(1 - \mathrm{e}^{\mathrm{Z_i}})$ (for $i - 1, \ldots, N$), Equation (2) is defines the PDF of the minimum $W_1, \cdots, W_n$. Making this double composition of the two generators, the proposed family absorbs the impacts of two different flexibilities on applications.

Table 2 provides some special cases of (6), where $\Phi(x)$ and $\phi(x)$ are the CDF and PDF of the standard normal distribution. The density and hazard functions of the Marshall-Olkin-Γ-Weibull (MO-Γ-W) defined by $h(x) = f(x)/(1 - F(x))$ are displayed in Figure 1, which provide more flexibility for these functions in relation to the baseline ones.

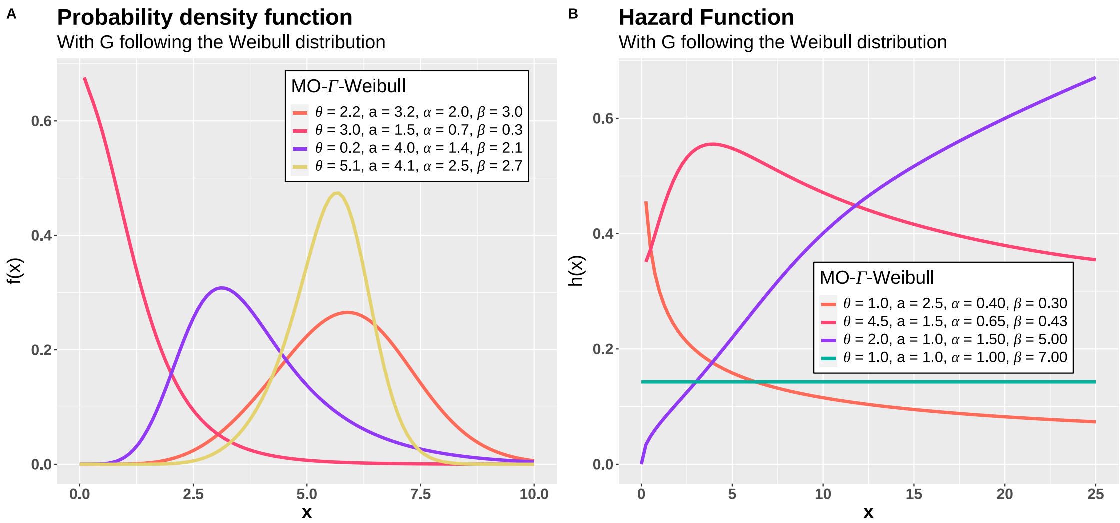

Figure 1: (A) MO-Γ-W density. (B) MO-Γ-W hazard.

The CDF (5) can be easily inverted to calculate the QF of the MO-Γ-G distribution, say $x = Q_X(u) = F_X^{-1}(u)$ for $u \in (0, 10$, in terms of the baseline QF $Q_G(\cdot)$. The inverse of $F_X(x) = u$, where u is a uniform number in $(0, 1)$ is easily obtained. By combining the inverses of Equations (1)and (5), $F_X(x) = u$ leads to $z = z(u) = \theta u/[1 - (1 - \theta)u]$ and $\gamma_1(a, -\log[1 - G(x)]) = z(u)$. Then, the QF of X can be expressed as

$$x = Q_G(v(u)),$$

where

$$v(u) = 1 - \exp\left[-\gamma_1^{-1}(a, z(u))\right],$$

and $\gamma_1^{-1}(a, w) = Q^{-1}(a, 1 - w)$ is the inverse function of $\gamma_1(a, w)$. Some formulae for $Q^{-1}(a, 1 - w)$ are given in http://functions.wolfram.com/GammaBetaErf/InverseGammaRegularized/.

Distribution	Baseline CDF	Generated PDF
Normal	$G(x) = \Phi(x)$	$f_X(x) = \frac{\theta\{-\log[1-\Phi(x)]\}^{a-1}\,\phi(x)}{\Gamma(a)\,\{\theta+(1-\theta)\gamma_1(a,-\log[1-\Phi(x)])\}^2}$
Logistic	$G(x) = \frac{1}{1+\mathrm{e}^{-\mathrm{x}}}$	$f_X(x) = \frac{\theta\,\mathrm{e}^{-\mathrm{x}}\left\{-\log[1-(1+\mathrm{e}^{-\mathrm{x}})^{-1}]\right\}^{\mathrm{a}-1}}{\Gamma(a)\,(1+\mathrm{e}^{-\mathrm{x}})^2\,\{\theta+(1-\theta)\gamma_1(\mathrm{a},-\log[1-(1+\mathrm{e}^{-\mathrm{x}})^{-1}])\}^2}$
Gumbel	$G(x) = 1 - \exp(-\mathrm{e}^{\mathrm{x}})$	$f_X(x) = \frac{\theta\ \exp(a\,x-\mathrm{e}^{\mathrm{x}})}{\Gamma(a)\,\{\theta+(1-\theta)\gamma_1(a,\mathrm{e}^{\mathrm{x}})\}^2}$
Log-Normal	$G(x) = \Phi(\log x)$	$f_X(x) = \frac{\theta\,\phi(\log x)\,\{-\log[1-\Phi(\log x)]\}^{a-1}}{\Gamma(a)\,x\,\{\theta+(1-\theta)\gamma_1(a,-\log[1-\Phi(\log x)])\}^2}$
Exponential	$G(x) = 1 - \exp(-\lambda x),\ \lambda > 0$	$f_X(x) = \frac{\theta\,\lambda^a\,x^{(a-1)}}{\Gamma(a)\,\{\theta+(1-\theta)\gamma_1(a,\lambda x)\}^2}$
Weibull	$G(x) = 1 - \exp(-(\lambda x)^\gamma),\ \lambda, \gamma > 0$	$f_X(x) = \frac{\theta\,\gamma\lambda^{a\,\gamma}x^{a\,\gamma-1}\exp\{-(\lambda\,\gamma)^\gamma\}}{\Gamma(a)\{\theta+(1-\theta)\gamma_1[a,(\lambda\,x)^\gamma]\}^2}$
Gamma	$G(x) = \gamma_1(\alpha, \beta x),\ \alpha,\ \beta > 0$	$f_X(x) = \frac{\theta\,\beta^\alpha\,x^{\alpha-1}\,\mathrm{e}^{-\beta\mathrm{x}}\,\{-\log[1-\gamma_1(\alpha,\beta\mathrm{x})]\}^{\mathrm{a}-1}}{\Gamma(a)\,\{\theta+(1-\theta)\gamma_1(a,-\log[1-\gamma_1(\alpha,\beta x)])\}^2}$
Pareto	$G(x) = 1 - \frac{1}{(1+x)^\nu},\ \nu > 0$	$f_X(x) = \frac{\theta\,\mathrm{e}^{-\mathrm{x}}\,[\nu\log(1+\mathrm{x})]^{\mathrm{a}-1}\,\mathrm{g(x)}}{\Gamma(a)\,(1+\mathrm{e}^{-\mathrm{x}})^2\,\{\theta+(1-\theta)\gamma_1(\mathrm{a},\nu\log[1+\mathrm{x}])\}^2}$

Table 1: Special Distributions in the MO-Γ-G family.

3 Estimation

The MO-Γ-G family can be fitted to real data using the **AdequacyModel** package in the R software. This package does not require to define the log-likelihood function and it computes the MLEs, their standard errors (SEs) and the formal statistics defined in Section 5. It is necessary to provide the PDF and CDF of the distribution to be fitted to a data set.

For example, if x_i is one observation from (6) and $\boldsymbol{\eta}$ is a q-parameter vector specifying $G(\cdot)$, the log-likelihood function for $\boldsymbol{\theta}^\top = (a, \theta, \boldsymbol{\eta}^\top)$ from n observations is

$$\begin{aligned}\ell(\boldsymbol{\theta}) =& \, n\log(\theta) + n\log(\gamma) + n\,a\,\gamma\log(\lambda) + (a\gamma-1)\sum_{i=1}^{n}\log(x_i) - \lambda^{\gamma}\sum_{i=1}^{n}\log(x_i)\\ &- n\log[\Gamma(a)] - 2\sum_{i=1}^{n}\log\{\theta + (1-\theta)\gamma_1[a,(\lambda\,x_i)^{\gamma}]\}. \qquad (7)\end{aligned}$$

Due to the impossibility of obtaining the MLEs in closed form, numerical methods to obtain the estimates that maximize $\ell(\cdot)$ are necessary. Several programming languages and statistical software distributes functions and routines that make it easy to obtain numerical estimates by various interactive methods. In practice, obtaining the MLEs for the parameters that index a probability distribution are commonly obtained in this way, since the Newton and quasi-Newton methods produce satisfactory results under reasonable conditions of the object function, that is, when they do not impose conditions that disturb the convergence of the algorithms.

To obtain the MLEs, the package **AdequacyModel** of the programming language R was used, see R Core Team (2020). This library, created and maintained by one of the authors of this paper, is widely cited by several papers in the field of statistics and serves as a basis for other library implementations available on the Comprehensive R Archive Network - CRAN. With it, in particular using the goodness.fit function, it is possible to provide an implementation R of (6), being in charge of this function, obtain $\ell(\cdot)$ by returning several measures of fit adequacy as well as the MLEs. Further details regarding this package can be obtained from Marinho et al. (2019).

4 Simulations

Due to the probable absence of MLEs in closed-form for distributions belonging to the MO-Γ-G family, it is necessary to examine the precision of the estimates calculated numerically. For doing that, the biases of the estimators of the parameters of the MO-Γ-Dagum(θ, a, α, β, p) distribution are determined, where $G \sim \text{Dagum}(\alpha, \beta, p)$ is the baseline distribution. All parameters are taken equal to one for different sample sizes reported in Table 2.

Ten thousand Monte Carlo simulations are performed for each sample size to examine the numerical estimates calculated by the BFGS method. The figures in Table 2 indicate that this method behaves well when the sample size increases. This is theoretically expected. However, in practice, difficulties can be faced in other families of distributions due to the flatness of the log-likelihood function.

All simulations can be reproduced using the script in Appendix A. The simulations are parallelized and able to use all threads available by a multicore processor, thus making them more computationally efficient and consequently requiring less time to complete. The simulations are performed on a computer with an Intel Core i5-8265U processor with 8 threads working at a maximum frequency of 3.90 GHz, requiring, on these hardware, a time of 14.36 hours to perform all simulations. The figures in Table 2 reveal that the average biases of the MLEs could be very reduced only for $n > 2,000$.

To generate observations from the random variable X with density f, the well-known Acceptance-Rejection Algorithm for continuous random variables, which is very useful when the quantile function involves complex functions that can lead to some numerical inaccuracies. For doing this, another random variable Y is chosen such that it can generate observations from a PDF h with the same support as f. Then, the acceptance and rejection algorithm is defined by the following steps:

1. Generate an outcome y from Y;
2. Generate an observation u from a random variable $U \sim \mathcal{U}(0,1)$;
3. If $u < \frac{f(y)}{c\,g(y)}$, where c is a real constant, accept $x = y$; otherwise reject y as an outcome from X and return to 1.

The constant c must be chosen in such a way that $\frac{f(y)}{c\,g(y)} \leq 1$. Thus, to minimize the computational cost of generating observations from X through the generated observations from Y, c is chosen as the lowest possible value to maximize the likelihood of acceptance. Further details of this method can be found in Rizzo (2019).

Table 2: Mean biases of the MLEs obtained using the BFGS method calculated from the Monte-Carlo simulation.

n	$B(\hat{\theta})$	$B(\hat{a})$	$B(\hat{\alpha})$	$B(\hat{\beta})$	$B(\hat{p})$	Time (mins)
10	0.2376	2.1635	2.7557	1.6282	1.3057	1.1430
20	0.4154	2.4639	1.5728	1.8082	0.7383	1.6248
60	0.7214	2.2432	0.5667	1.8815	0.2872	3.3954
100	0.6146	1.9579	0.3253	1.6148	0.2651	4.9628
200	0.3838	1.3894	0.1827	1.1773	0.3701	8.1457
400	0.2166	0.9635	0.1076	0.6181	0.3957	13.7370
600	0.1269	0.7242	0.0772	0.3968	0.3637	17.9310
1,000	0.0553	0.4885	0.0521	0.2328	0.2636	22.8784
2,000	0.0456	0.3087	0.0334	0.0990	0.1722	38.4593
5,000	-0.0058	0.1307	0.0146	0.0117	0.0171	52.8098
10,000	-0.0146	0.0842	0.0095	0.0095	0.0031	95.8380
20,000	-0.0090	0.0330	0.0038	0.0005	-0.0099	126.4260
30,000	-0.0028	0.0183	0.0012	-0.0036	-0.0029	182.0760
50,000	-0.0057	0.0124	0.0015	0.0016	-0.0021	291.9300

5 Applications

Consider the Weibull baseline. Two applications are provided to compare the new generated model with seven extended Weibull distributions, namely the beta-Weibull (β-W) (Famoye *et al.*, 2005), Kumaraswamy Weibull (Kw-W) (Cordeiro and

Nadarajah, 2010), Marshall-Olkin Weibull (MO-W) (Ahmed *et al.*, 2017), Marshall-Olkin Extended Weibull (MOE-W) (Cordeiro *et al.*, 2019), exponentiated Weibull (exp-W) (Mudholkar and Srivastava, 1993), gamma Weibull (Γ-W) Cordeiro *et al.*, 2016) and exponentiated generalized Weibull (EG-W) (Oguntunde *et al.*, 2015) (with $a = 1$). Some of these distributions are widely used in practice.

The log-likelihood for the Marshall-Olkin-Gamma-Weibull (MO-Γ-W) from one observation is

$$\begin{aligned}\ell(\boldsymbol{\theta}) =& \log(\theta) + \log(\gamma) + (a\,\gamma)\log(\lambda) + (a\,\gamma - 1)\log(x) - (\gamma\,x)^{\gamma} - \log[\Gamma(a)] \\ & - 2\log\{\theta + (1-\theta)\gamma_1[a, (\lambda\,x)^{\gamma}]\},\end{aligned} \tag{8}$$

where $\boldsymbol{\theta} = (a, \theta, \lambda, \gamma)^{\top}$. The components of the score function are

$$U_a(\boldsymbol{\theta}) = \gamma\log(\lambda) + \gamma\log(x) - \psi^{(0)}(a) - \frac{2\left\{(1-\theta)A - (1-\theta)\psi^{(0)}(a)\gamma_1\left[a, (x\lambda)^{\gamma}\right]\right\}}{\theta\,\Gamma(a) + (1-\theta)\gamma_1\left[a, (x\lambda)^{\gamma}\right]},$$

$$U_{\theta}(\boldsymbol{\theta}) = \frac{1}{\theta} - \frac{2\left\{\Gamma(a) - \gamma_1\left[a, (\lambda\,x)^{\gamma}\right]\right\}}{\theta\,\Gamma(a) + (1-\theta)\gamma_1\left[a, (\lambda\,x)^{\gamma}\right]},$$

$$U_{\lambda}(\boldsymbol{\theta}) = \frac{\gamma}{\lambda}\left[a - (\lambda\,x)^{\gamma}\right] + \frac{2\gamma\,\lambda^{-1}(\lambda\,x)^{a\,\gamma}(1-\theta)\exp\{-(\lambda\,x)^{\gamma}\}}{\theta\,\Gamma(a) + (1-\theta)\gamma_1\left[a, (x\lambda)^{\gamma}\right]}$$

and

$$U_{\gamma}\boldsymbol{\theta} = \frac{1}{\gamma} + a\log(\lambda) + a\log(x) - (\lambda x)^{\gamma}\log(\lambda x) + \frac{2(1-\theta)(\lambda x)^{\gamma\,a}\log(\lambda x)\exp\{-(\lambda x)^{\gamma}\}}{\theta\,\Gamma(a) + (1-\theta)\gamma_1\left[a, (x\lambda)^{\gamma}\right]},$$

where

$$A = G^{3,0}_{2,3}\left[(x\lambda)^{\gamma}\middle|\begin{array}{c}1,1\\0,0,a\end{array}\right] + \log\left[(\lambda x)^{\gamma}\right]\gamma_1\left[a, (x\lambda)^{\gamma}\right],$$

$\psi^{(n)}(x)$ is the n-th derivative of the digamma function,

$$A = \psi^{(0)}(a) - \log\left[(\lambda x)^{\gamma}\right] - G^{3,0}_{2,3}\left((\lambda x)^{\gamma}\middle|\,\overset{1,1}{0,0,a}\right),$$

and $G^{m,n}_{p,q}\left(z\middle|\,\overset{a_1,\ldots,a_p}{b_1,\ldots,b_q}\right)$ is the Meijer G function.

The **AdequacyModel** is used to fit the distributions cited before to two real data sets. The SANN method, which is a variant of simulated annealing (Belisle, 1992), is considered. The distributions are compared via the Anderson Darling (A*) and Cramér Von Mises (W*) statistics reported in the `goodness.fit` function.

For the first data set, a modification of the "FoodExpenditure" data from the **betareg** package is considered, which refers to the proportions of income spent on food for a random sample of 38 households in a large US city (according to the package information). Here, the household expenditures for food are considered and is given by

$$data = FoodExpenditure_{food}/\#(FoodExpenditure_{food}),$$

where $FoodExpenditure_{food}$ is the random variable corresponding to the household expenditures for food and $\#(\cdot)$ indicates the number of observations on this variable.

Table 3: Application 1

Model	a	θ	λ	γ	W^*	A^*
MO-Γ-W$(a, \theta, \lambda, \gamma)$	0.926134 (0.02626926)	1.379664 (0.22381086)	33.323073 (0.28530971)	25.398809 (0.08259864)	0.0339	0.2376
β−W$(a, \theta, \lambda, \gamma)$	9.92882 (0.02908181)	0.1700880 (0.02049663)	9.759469 (<0.0001)	1.530541 (<0.0001)	0.043567	0.2594618
KW-W$(a, \theta, \lambda, \gamma)$	0.04987575 (0.008090352)	99.99989793 (16.225905058)	1.07602954 (0.003156126)	23.40287099 (0.014646102)	1.330915	6.742609
MOE-W$(a, \theta, \lambda, \gamma)$	0.1366666 (0.1599182)	2.020436 (<0.0001)	62.72201 (<0.0001)	4.2956659 (0.7365535)	0.03541554	0.2579222
EGW(a, b, λ, γ)	5.6189861421 (0.0028147823)	6.1833138579 (0.0009807091)	1.2870816760 (0.1159810838)	1.3798562942 (0.1480747352)	0.03715255	0.2518879
MO-W(a, λ, γ)	0.15920715 (0.07170351)	- (-)	1.58609458 (0.13353716)	4.26713785 (0.16650667)	0.03456333	0.257339
exp-W(a, λ, γ)	6.1102948 (0.4222173)	- (-)	4.4680469 (0.3175876)	1.3858740 (0.1677722)	0.03726684	0.2523749
γ-W(a, λ, γ)	5.751546 (0.0015006)	- (-)	10.0000 (0.0001163185)	1.208773 (0.00822782)	0.0879	0.6599

The MLEs and their standard errors (SEs) (in parentheses) are listed in Table 3. The statistics W* and A* are also given in this table. The results indicate that the proposed model has better performance than the other seven fitted models.

As a second application, consider a data set collected in a pilot study about hypertension in the Dominican Republic in 1997 found in http://biostat.mc.vanderbilt.edu/wiki/Main/DataSets. The observations are the systolic blood pressure of persons who came to medical clinics in several villages for a variety of complaints. The MLEs of the parameters, their SEs and the values of the statistics are listed in Table 4 for the previous distributions. By comparing the measures of these formal statistics, we conclude that the proposed distribution outperforms the rest of them.

6 Mathematical properties

In this section, some main mathematical properties are presented for the MO-Γ-G family based on a general linear representation for its density function, which are important to determine its mathematical properties from those of exponentiated-G (exp-G) distributions.

6.1 Linear Representation

For an arbitrary CDF $G(x)$, the CDF and PDF of the exponentiated-G (exp-G) distribution with power parameter $a > 0$ are

$$\Pi_a(x) = G(x)^a \qquad \text{and} \qquad \pi_a(x) = a\, g(x)\, G(x)^{a-1},$$

respectively. This class of distributions is quite useful in several applications. In fact, Tahir and Nadarajah (2015) cited more than seventy papers on exponentiated distributions in their Table 1.

Table 4: Application 2

Model	a	θ	λ	γ	W^*	A^*
MO-Γ-W$(a,\theta,\lambda,\gamma)$	9.629304 (0.006217876)	3.640779 (0.182495785)	6.260826 (0.024446563)	12.823429 (0.007965221)	0.5093	2.8076
$\beta-$W$(a,\theta,\lambda,\gamma)$	31.08471 (0.01279570)	47.14636 (<0.0001)	0.01698383 (0.00014535)	2.054037 (<0.0001)	0.7540428	4.279418
KW-W$(a,\theta,\lambda,\gamma)$	7363.281 (0.04194304)	0.03925762 (0.0004194304)	1.467640 (<0.0001)	0.6146940 (<0.0001)	0.5351	2.9617
MOE-W$(a,\theta,\lambda,\gamma)$	101.13471834 (46.782882038)	0.42386507 (0.100832102)	0.03095935 (0.003644092)	1.70240332 (0.168142275)	1.14418	6.644617
EGW(a,b,λ,γ)	0.2351691 (0.002540837)	140.0000 (3.278565)	0.4576370 (<0.0001)	0.7425738 (<0.0001)	0.8925	5.3338
MO-W(a,λ,γ)	173.2139 (0.00016394)	- (-)	0.02125455 (0.000212794)	1.601970 (0.0001398109)	1.476088	8.58705
exp-W(a,λ,γ)	69.02916 (0.08389090)	- (-)	0.02405120 (0.0002249256)	1.345536 (<0.0001)	0.8899261	5.094141
Γ-W(a,λ,γ)	9.11229459 (0.96330688)	- (-)	0.02617729 (0.00291875)	1.74641178 (0.08030313)	0.6227098	3.488268

First, the MO-G cumulative distriution (2) admits the linear combination (Barreto-Souza *et al.*, 2013)

$$F_{\text{MO}-\Gamma}(x) \;=\; \sum_{i=0}^{\infty} w_i^{\text{MO-G}}\, \Pi_{i+1}(x) \;=\; \sum_{i=0}^{\infty} w_i^{\text{MO-G}}\, G(x)^{i+1}, \tag{9}$$

where the coefficients are (for $i = 0, 1, \dots$)

$$w_i^{\text{MO}-\Gamma} = w_i^{\text{MO}-\Gamma}(\theta) = \begin{cases} \dfrac{(-1)^i\,\theta}{(i+1)} \sum\limits_{j=i}^{\infty} (j+1) \dbinom{j}{i} \bar{\theta}^j, & \theta \in (0,1), \\ \theta^{-1}(1-\theta^{-1})^i, & \theta > 1, \end{cases}$$

and $\bar{\theta} = 1 - \theta$.

Second, the linear combination for the Γ-G cumulative distribution (4) follows from Castellares and Lemonte (2015) as

$$F_{\Gamma\text{-G}}(x) \;=\; \sum_{j=0}^{\infty} w_j^{\Gamma\text{-G}}\; \Pi_{a+j}(x). \tag{10}$$

Here,

$$w_j^{\Gamma\text{-G}} = w_j^{\Gamma\text{-G}}(a) = \frac{\varphi_j(a)}{(a+j)},$$

$$\varphi_0(a) = \frac{1}{\Gamma(a)}, \quad \varphi_j(a) = \frac{(a-1)}{\Gamma(a)}\, \psi_{j-1}(j+a-2), \quad j \geq 1,$$

and

$$\psi_{n-1}(x) = \frac{(-1)^{n-1}}{(n+1)!}\Bigg[H_n^{n-1} - \frac{x+2}{n+2} H_n^{n-2} + \frac{(x+2)(x+3)}{(n+2)(n+3)} H_n^{n-3} - \cdots$$
$$+ (-1)^{n-1}\frac{(x+2)(x+3)\cdots(x+n)}{(n+2)(n+3)\cdots(2n)} H_n^0\Bigg],$$

is the Stirling polynomial, $H_{n+1}^m = (2n+1-m)H_n^m + (n-m+1)H_n^{m-1}$ is a positive integer, $H_0^0 = 1$, $H_{n+1}^0 = 1 \times 3 \times 5 \times \cdots \times (2n+1)$ and $H_{n+1}^n = 1$.

By inserting (10) in Equation (9) and via a result for a power series raised to a positive integer (Gradshteyn and Ryzhik, 2000), the expansion for the cdf of the MO-Γ-G distribution reduces to

$$\begin{aligned} F_{\text{MO}-\Gamma\text{-G}}(x) &= \sum_{i=0}^{\infty} w_i^{\text{MO-}\Gamma}\, G(x)^{(i+1)a} \left[\sum_{j=0}^{\infty} w_j^{\Gamma\text{-G}}\, G(x)^j\right]^{i+1} \\ &= \sum_{i=0}^{\infty} w_i^{\text{MO-}\Gamma}\, G(x)^{(i+1)a} \sum_{j=0}^{\infty} c_{i+1,j}\, G(x)^j = \sum_{i,j=0}^{\infty} d_{i,j}\, \Pi_{(i+1)a+j}(x), \end{aligned}$$

where $d_{i,j} = d_{i,j}(a,\theta) = w_i^{\text{MO-G}}\, c_{i+1,j}(a)$, $c_{i+1,0}(a) = (w_0^{\Gamma\text{-G}})^{i+1}$ and, for $m \geq 1$, $c_{i+1,m}(a) = \frac{1}{m w_0^{\Gamma\text{-G}}} \sum_{r=1}^{m} [r(i+2)-m]\, w_r^{\Gamma\text{-G}}\, c_{i+1,m-r}(a)$.

By differentiating the last equation, the expansion for the MO-Γ-G density follows as

$$f_{\text{MO}-\Gamma\text{-G}}(x) = \sum_{i,j=0}^{\infty} d_{i,j}\, \pi_{(i+1)a+j}(x). \tag{11}$$

So, some structural properties of the proposed family can be determined from the double linear combination (11) and those properties of the exp-G distribution. In most applications, the indices i and j can vary up to five.

6.2 Some quantities

Hereafter, let $T_{i,j} \sim$exp-G$[(i+1)a+j]$. The nth moment of X can be obtained from (11) as

$$\mu_n' = E(X^n) = \sum_{i,j=0}^{\infty} d_{i,j}\, E(T_{i,j}) = \sum_{i,j=0}^{\infty} [(i+1)a+j-1]\, d_{i,j}\, \tau[n,(i+1)a+j-1], \tag{12}$$

where

$$\tau(n,a) = \int_{-\infty}^{\infty} x^n\, G(x)^a\, g(x)dx = \int_0^1 Q_G(u)^n\, u^a du.$$

Expressions for moments of several exponentiated distributions can be found in the papers cited in Tahir and Nadarajah (2015, Table 1). We give just one example from Equation (12) by taking the exponential distribution with rate $\lambda > 0$ for the baseline G. It follows easily as

$$\mu_n' = n!\, \lambda^n \sum_{i,j,m=0}^{\infty} \frac{(-1)^{n+m}\, [(i+1)a+j]\, d_{i,j}}{(m+1)^{n+1}} \binom{(i+1)a+j-1}{m}.$$

A general expansion for the moment generating function (MGF) $M(t) = E(\mathrm{e}^{\mathrm{t}\,\mathrm{X}})$ of X can be expressed from (11)

$$M(t) = \sum_{i,j=0}^{\infty} d_{i,j}\, M_{i,j}(t) = \sum_{i,j=0}^{\infty} [(i+1)a+j]\, d_{i,j}\, \rho(t,(i+1)a+j-1), \tag{13}$$

where $M_{i,j}(t)$ is the MGF of $Y_{i,j}$ and

$$\rho(t,a) = \int_{-\infty}^{\infty} \mathrm{e}^{\mathrm{tx}}\, \mathrm{G(x)}^{\mathrm{a}}\, \mathrm{g(x)dx} = \int_{0}^{1} \exp\left\{\mathrm{t}\, \mathrm{Q_G(u)}\right\}\, \mathrm{u^a du}.$$

The MGFs of many MO-Γ-G distributions can be determined from Equation (13). For example, the generating function of the MO-Γ-exponential with parameter λ (if $t < \lambda^{-1}$) is

$$M(t) = \sum_{i,j=0}^{\infty} [(i+1)a + j]\, d_{i,j}\; B((i+1)a + j, 1 - \lambda t).$$

For empirical purposes, the shape of many distributions can be usefully described by the incomplete moments. These moments play an important role for measuring inequality. For example, the mean deviations and Lorenz and Bonferroni curves depend upon the first incomplete moment of the distribution. The nth incomplete moment of X can be expressed as

$$m_n(y) = \int_{-\infty}^{y} x^n\, f_X(x)dx = \sum_{i,j=0}^{\infty} [(i+1)a + j]\, d_{i,j} \int_{0}^{G(y)} Q_G(u)^n\, u^{(i+1)a+j-1}\, du. \quad (14)$$

The definite integral in (14) can be evaluated for most baseline G distributions.

7 Conclusions

A new family of distributions called the Marshall and Olkin-Gamma-G family with two shape parameters is introduced. The estimation of the unknown parameters is done via the maximum likelihood method and a simulation study is conducted to verify its adequacy. Additionally, the usefulness of the proposed family is shown empirically by means of two applications to real data.

A Appendix: Simulation code

The script below is used to perform the simulations whose results are given in Table 2 of Section 4. It is developed a script using the programming language R to be used in other simulations or even to check the results presented in this paper.

Listing 1: Monte-Carlo simulations for different sample sizes.

```r
# Title: Marshall Olkin Gamma-G (MOGG)
# Author: Pedro Rafael D. Marinho

# Loading libraries. ------------------------------------------------
library(parallel)
library(tibble)
library(pbmcapply)
library(magrittr)
library(purrr)
library(xtable)

# Baseline functions. -----------------------------------------------
pdf_dagum <- function(x, alpha, beta, p)
  alpha * p / x * (x / beta) ^ (alpha * p) / ((x / beta) ^ alpha + 1) ^ (p + 1)
# integrate(f = pdf_dagum, lower = 0, upper = Inf, alpha = 1.2,
#           beta = 1.6, p = 2.2)

```

```r
cdf_dagum <- function(x, alpha, beta, p)
  (1 + (x / beta) ^ (-alpha)) ^ (-p)
# cdf_dagum(x = Inf, alpha = 1, beta = 4, p = 1)

# This function creates MOGG functions. ----------------------------------
pdf_mogg <- function(g, G) {
  # Using Closures.
  function(x, theta, a, ...) {
    if (theta <= 0 || a <= 0)
      warning("The \"a\" and \"theta\" parameters must be greater than zero.")
    num <-
      theta * (-log(1 - G(x = x, ...))) ^ (a - 1) * g(x = x, ...)
    den <-
      gamma(a) *
      (theta + (1 - theta) *
         pgamma(-log(1 - G(x = x, ...)), a, 1L)) ^ 2
    num / den
  }
}

rdagum <- function(n = 1L, alpha, beta, p) {
  beta * (runif(n = n, min = 0, max = 1) ^ (-1 / p) - 1) ^ (-1 / alpha)
}

pdf_mogdagum <- pdf_mogg(g = pdf_dagum, G = cdf_dagum)

# MOG-Dagum ---------------------------------------------------------------
rmogdagum <- function(n = 1L, theta, a, alpha, beta, p) {
  cond_c <- function(x, theta, a, alpha, beta, p) {
    num <- pdf_mogdagum(x, theta, a, alpha, beta, p)
    den <- pdf_dagum(x,
                     alpha = alpha,
                     beta = beta ,
                     p = p)
    - num / den
  }

  x_max <-
    optim(
      fn = cond_c,
      method = "BFGS",
      par = 1,
      theta = theta,
      a = a,
      alpha = alpha,
      beta = beta,
      p = p
    )$par

  c <-
    pdf_mogdagum(x_max, theta, a, alpha, beta, p) / pdf_dagum(x_max,
                                                              alpha = alpha,
                                                              beta = beta,
                                                              p = p)

  criterion <- function(y, u) {
    num <- pdf_mogdagum(y, theta, a, alpha, beta, p)
    den <- pdf_dagum(y,
                     alpha = alpha,
                     beta = beta,
                     p = p)
    u < num / (c * den)
  }

  values <- double(n)
  i <- 1L
  repeat {
    y <- rdagum(
      n = 1L,
      alpha = alpha,
      beta = beta,
      p = p
    )
    u <- runif(n = 1L, min = 0, max = 1)
```

```r

    if (criterion(y, u)) {
      values[i] <- y
      i <- i + 1L
    }
    if (i > n)
      break
  }
  values
}

# Testing the rmogw Function ---------------------------------------------------
theta = 5
a = 1
alpha = 5
beta = 1
p = 1
pdf_mogdagum <- pdf_mogg(g = pdf_dagum, G = cdf_dagum)
sample_data <- rmogdagum(n = 250L, theta, a, alpha, beta, p)
x <- seq(0, max(sample_data), length.out = 500L)
hist(sample_data,
     probability = TRUE,
     xlab = "",
     main = "")
lines(x, pdf_mogdagum(x, theta, a, alpha, beta, p))

# Monte Carlo simulations. ------------------------------------------------------
mc <- function(n = 250L,
               M = 1e3L,
               par_true,
               method = "BFGS") {
  theta <- par_true[1L]
  a <- par_true[2L]
  alpha <- par_true[3L]
  beta <- par_true[4L]
  p <- par_true[5L]

  # Log-likelihood function. ----------------------------------------------------
  pdf_mogw <- pdf_mogg(g = pdf_dagum, G = cdf_dagum)
  log_likelihood <- function(x, par) {
    theta <- par[1L]
    a <- par[2L]
    alpha <- par[3L]
    beta <- par[4L]
    p <- par[5L]

    - sum(log(
      pdf_mogdagum(
        x,
        theta = theta,
        a = a,
        alpha = alpha,
        beta = beta,
        p = p
      )
    ))
  }

  myoptim <-
    function(...)
      tryCatch(
        expr = optim(...),
        error = function(e)
          NA
      )

  one_step_mc <- function(i) {
    sample_data <- rmogdagum(n, theta, a, alpha, beta, p)

    result <- myoptim(
      fn = log_likelihood,
      par = c(1, 1, 1, 1, 1),
      x = sample_data,
      method = method
```

```
    )

    while (is.na(result) || result$convergence != 0) {
      sample_data <- rmogdagum(n, theta, a, alpha, beta, p)
      result <- myoptim(
        fn = log_likelihood,
        par = c(1, 1, 1, 1, 1),
        method = method,
        x = sample_data
      )
    }

    result$par
  }

  result_vector <-
    unlist(
      pbmcapply::pbmclapply(
        X = 1L:M,
        FUN = one_step_mc,
        mc.cores = parallel::detectCores()
      )
    )


  result <-
    tibble::as_tibble(matrix(result_vector, byrow = TRUE, ncol = 5L))

  names(result) <- c("theta", "a", "alpha", "beta", "p")

  result
}

bias_function <- function(x, par_true) {
  x - par_true
}

mse_function <- function(x, par_true) {
  (x - par_true) ^ 2
}

simulate <- function(n) {
  # True parameters (theta, a, alpha, beta and p) -----------------------
  true_parameters <- c(1, 1, 1, 1, 1)

  set.seed(1L, kind = "L'Ecuyer-CMRG")
  t0 <- Sys.time()
  result_mc <-
    mc(
      n = n,
      M = 1e4L,
      par_true = true_parameters,
      method = "BFGS"
    )
  total_time <- Sys.time() - t0

  mc.reset.stream()

  # Average Bias of Estimators ----------------------------------------
  eval(parse(
    text = glue(
      "bias_{n} <- apply(X = result_mc, MARGIN = 1L, FUN = bias_function,
                          par_true = true_parameters) %>%
                   apply(MARGIN = 1L, FUN = mean)"
    )
  ))
  eval(parse(text = glue(
    "save(file = \"bias_{n}.RData\", bias_{n})"
  )))

  # Mean Square Error -------------------------------------------------
  eval(parse(
    text = glue(
      "mse_{n} <- apply(X = result_mc, MARGIN = 1L, FUN = mse_function,
```

```r
                         par_true = true_parameters) %>%
                    apply(MARGIN = 1L, FUN = mean)"
    )
  ))
  eval(parse(text = glue(
    "save(file = \"mse_{n}.RData\", mse_{n})"
  )))

  # Total Time -----------------------------------------------------------
  eval(parse(text = glue("time_{n} <- total_time")))
  eval(parse(text = glue(
    "save(file = \"time_{n}.RData\", time_{n})"
  )))

  # Result MC
  eval(parse(text = glue("result_{n} <- result_mc")))
  eval(parse(text = glue(
    "save(file = \"result_{n}.RData\", result_{n})"
  )))

}

walk(
  .x = c(
    10,
    20,
    60,
    100,
    200,
    400,
    600,
    1000,
    2000,
    5000,
    10000,
    20000,
    30000,
    50000
  ),
  .f = simulate
)

first_col <-
  c(10, 20, 60, 100, 200, 400, 600, 1000, 5000, 10000, 20000, 30000, 50000)

tabela <-
  rbind(
    bias_10,
    bias_20,
    bias_60,
    bias_100,
    bias_200,
    bias_400,
    bias_600,
    bias_1000,
    bias_5000,
    bias_10000,
    bias_20000,
    bias_30000,
    bias_50000
  )
tabela <- cbind(n = first_col, tabela)
rownames(tabela) <- NULL

tabela <- tibble::as_tibble(tabela)

latex <-
  print.xtable(
    xtable(tabela,
           caption = "Mean bias of EMV obtained by the BFGS method in 10,000
           Monte Carlo repetitions.",
           digits = 4L),
    print.results = FALSE
  )
```

```
writeLines(
  c(
    "\\documentclass[12pt]{article}",
    "\\begin{document}",
    "\\thispagestyle{empty}",
    latex,
    "\\end{document}"
  ),
  "mc_simulation.tex"
)

tools::texi2pdf("mc_simulation.tex", clean = TRUE)
```

References

A. Alzaatreh, F. Famoye, C. Lee: A new method for generating families of continuous distributions. Metron **71**(1) 63-79 (2013).

W. Barreto-Souza, A.J. Lemonte, G.M. Cordeiro: General results for the Marshall and Olkin's family of distributions. Anais da Academia Brasileira de Ciências **85**(1) 3-21 (2013).

Z.W. Birnbaum, S.C. Saunders: A new family of life distributions. Journal of Applied Probability **6**(2) 319-327 (1969).

F. Castellares, A.J. Lemonte: A new generalized Weibull distribution generated by gamma random variables. Journal of the Egyptian Mathematical Society **23**(2) 382-390 (2015).

G.M. Cordeiro, M. de Castro: A new family of generalized distributions. Journal of Statistical Computation and Simulation **81**(7) 883-898 (2011).

N. Eugene, C. Lee, F. Famoye: Beta-normal distribution and its applications. Communications in Statistics – Theory and Methods **31**(4) 497-512 (2002).

Gradshteyn IS, Ryzhik IM: Table of Integrals, Series and Products. Sixth edition. San Diego: Academic Press (2000).

R.C. Gupta, R.D. Gupta, P.L. Gupta: Modeling failure time data by Lehmann alternatives. Communications in Statistics – Theory and Methods **27**(4) 887-904 (1998).

R.D. Gupta, D. Kundu: Exponentiated exponential family: an alternative to gamma and Weibull distributions. Biometrical Journal **43**(1) 117-130 (2001).

A.J. Lemonte, G.M. Cordeiro, G. Moreno-Arenas: A new useful three-parameter extension of the exponential distribution. Statistics **50**(2) 312-337 (2016).

A.W. Marshall, I. Olkin: A new method for adding a parameter to a family of distributions with application to the exponential and Weibull families. Biometrika **84**(3) 641-652 (1997).

G.S. Mudholkar, D.K. Srivastava: The exponentiated Weibull family: a reanalysis of the bus-motor-failure data. Technometrics **37**(4) 436-445 (1995).

P. R. D. Marinho, R. B. Silva, M. Bourguignon, G. M. Cordeiro, S. Nadarajah. AdequacyModel: An R package for probability distributions and general purpose optimization. PloS one, **14**(8), e0221487 (2019).

R Core Team. R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/ (2020).

S. Nadarajah, G.M. Cordeiro, E.M.M. Ortega: Some general results for the beta-modified Weibull distribution. Journal of Statistical Computation and Simulation **81**(10) 1211-1232 (2011).

S. Nadarajah, G.M. Cordeiro, E.M.M. Ortega: General results for the Kumaraswamy G distribution. Journal of Statistical Computation and Simulation **87**(7) 951-979 (2012).

S. Nadarajah, A.K. Gupta: The exponentiated gamma distribution with application to drought data. Calcutta Statistical Association Bulletin **59**(1-2) 29-54 (2007).

S. Nadarajah, F. Haghighi: An extension of the exponential distribution. Statistics **45**(6) 543-558 (2011).

S. Nadarajah, S. Kotz: The exponentiated-type distributions. Acta Applicandae Mathematicae **92**(2) 97-111 (2006).

M.D. Nichols, W.J. Padgett: A Bootstrap control chart for Weibull percentiles. Quality and Reliability Engineering International **22**(2) 141-151 (2006).

F. Proschan: Theoretical explanation of observed decreasing failure rate. Technometrics **5**(3) 375-383 (1963).

M.M. Ristić, N. Balakrishnan: The gamma-exponentiated exponential distribution. Journal of Statistical Computation and Simulation **82**(8) 1191-1206 (2012).

E.W. Stacy: A Generalization of the Gamma Distribution. Annals of Mathematical Statistics **33**(3) 1187-1192 (1962).

M.H. Tahir, S. Nadarajah: Parameter induction in continuous univariate distributions: Well-established G families. Anais da Academia Brasileira de Ciências **87**(2) 539-568 (2015).

K. Zografos, N. Balakrishnan: On families of beta-and generalized gamma-generated distribution and associate inference. Statistical Methodology **6**(4) 344-362 (2009).

Rizzo, Maria L. Statistical computing with R. CRC Press (2019).