An Acad Bras Cienc (2022) 94(3): e20201542 DOI 10.1590/0001-3765202220201542
Anais da Academia Brasileira de Ciências | *Annals of the Brazilian Academy of Sciences*
Printed ISSN 0001-3765 I Online ISSN 1678-2690
www.scielo.br/aabc | www.fb.com/aabcjournal

MATHEMATICAL SCIENCES

Continuous Probability Distributions generated by the PIPE Algorithm

LUIS G.B. PINHO, JUVÊNCIO S. NOBRE & GAUSS M. CORDEIRO

Abstract: We investigate the use of the Probabilistic Incremental Programming Evolution (PIPE) algorithm as a tool to construct continuous cumulative distribution functions to model given data sets. The PIPE algorithm can generate several candidate functions to fit the empirical distribution of data. These candidates are generated by following a set of probability rules. The set of rules is then evolved over a number of iterations to generate better candidates regarding some optimality criteria. This approach rivals that of generated distribution, obtained by adding parameters to existing probability distributions. There are two main advantages for this method. The first is that it is possible to explicitly control the complexity of the candidate functions, by specifying which mathematical functions and operators can be used and how lengthy the mathematical expression of the candidate can be. The second advantage is that this approach deals with model selection and estimation at the same time. The overall performance in both simulated and real data was very satisfying. For the real data applications, the PIPE algorithm obtained better likelihoods for the data when compared to existing models, but with remarkably simpler mathematical expressions.

Key words: PIPE, continuous probability distributions, function regression, generated distributions.

1 - INTRODUCTION

In this paper we describe an approach to probability modeling based on evolutionary algorithms. Consider a data set of observations of an absolutely continuous random variable representing a given experiment. Suppose that there is not enough understanding of the problem to allow for the construction of a specific cumulative distribution function (cdf) or to suggest the use of an existing one. Given such a data set we attempt to find the "best possible" continuous probability distribution to model the data regarding some fitness or optimality criteria. This is achieved by exploring a search space of candidate functions in a way described in the next sections. We shall first briefly review the most popular techniques in the field of generated distributions presently.

A major effort has been devoted to the creation of new probability models over the last few decades. These models have been obtained by composing two already existing models and creating a new one that inherits parameters from both models. This has been motivated by need and curiosity and has given interesting results as the new models have a very wide range of successful applications. A recent paper (Tahir & Nadarajah 2014) reviews several of these new ways of obtaining new models. Each technique defines a usually very large class of probability distributions. For each class they list

several papers on particular families in that class. For a quick introduction to the subject the reader can refer to Tahir & Nadarajah 2014.

Two increasingly popular classes are the beta-G class from Jones 2004 and gamma-G class from Zografos & Balakrishnan 2009. The first adds two parameters to a given cdf $G(x)$. Usually, $G(x)$ is referred to as baseline distribution in this context. The resulting distribution can be defined by its cdf as

$$F(x) = \frac{1}{B(a,b)} \int_0^{G(x)} t^{a-1}(1-t)^{b-1} dt, \ a > 0, \ b > 0,$$

which is the beta cdf calculated at the point $G(x)$. The second generator, gamma-G, adds one parameter to the baseline distribution in a similar way:

$$F(x) = \frac{1}{\Gamma(a)} \int_0^{-\log(1-G(x))} t^{a-1} e^{-t} dt, \ a > 0,$$

which uses the gamma cdf, instead of the beta cdf, at $-\log(1-G(x))$.

See, for example, Aly & Benkherouf 2011 or Alzaatreh et al. 2013 for several general results on the composition of existing probability models which may be used to generated several others class of distribution families. Some papers showcase applications of the new probability distribution families. Some examples of applications can be seen in, for instance, Nadarajah & Gupta 2004, Nadarajah & Kotz 2004, 2006, Fischer & Vaughan 2010, Paranaíba et al. 2011.

This paper presents an alternative to these classes of distributions as well as to classical non-parametric methods. It has two main motivations: 1) there is an increasing difficulty in choosing a model for a given data set because of the raise in the number of new distributions; 2) most of the new models use formulas that are written with elementary functions such as logarithms and exponentials and arithmetic operators. Given a data set and an optimality criterion, our goal is to obtain a cdf suitable for the data using only these elementary functions and arithmetic operators for at most a certain number of times. This may lead to models with simpler mathematical expressions. Thus, we consider model selection and model generation in a single approach. The main advantage of this method over the other two commented above is that we are able to explicitly control the complexity of the resulting cdf. In this text, we use complexity in its most basic meaning, not as in computational complexity.

This paper is divided as follows. In Section 2 we describe the Probabilistic Incremental Program Evolution (PIPE) algorithm developed in Salustowicz & Schmidhuber 1997, which is the base technique employed in this paper. It is slightly modified for our purpose. Section 3 describes and exemplifies the new proposal. Simulation studies are presented in Section 4 to illustrate the performance of the method. In Section 5 we consider data sets from other papers on new distributions, comparing the fitness of the distributions in those papers to the fitness of the distribution obtained by the method proposed here. Issues and final remarks are addressed in Section 6. The source code for our implementations of the method described here are available, as well as instructions and examples, at https://github.com/lgbpinho/pipeForFunctionalRegression.

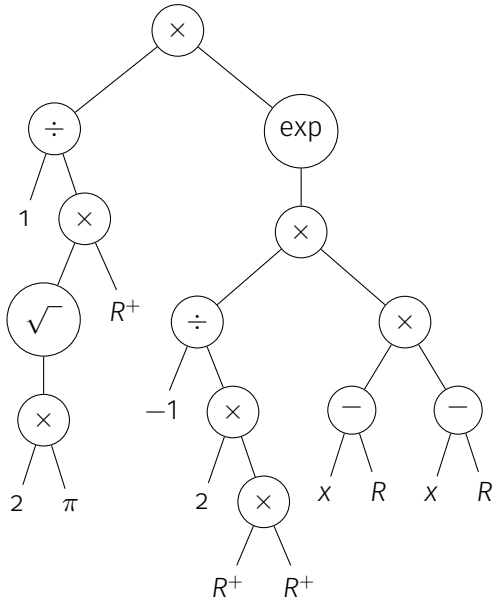

Figure 1. Tree representation of the normal distribution pdf.

2 - THE PIPE ALGORITHM

The PIPE algorithm was presented in Salustowicz & Schmidhuber 1997 and is capable of producing programs according to a set of probability rules. These rules are improved over iterations so that the generated programs are more likely to solve a given problem. In this section we follow closely the explanation in the original paper.

A *program* is a set of instructions given in a certain order. Each of these instructions may depend on a (possibly empty) set of terminal symbols, which usually denote constants or user inputs. Let $F = \{f_1, f_2, \dots, f_k\}$ be a set of k functions and $T = \{t_1, t_2, \dots, t_l\}$ be a set of l terminals. For instance, to write a program that calculates the value of the probability density function (pdf) for the normal or exponential distributions at a point x and a given set of parameters, it is sufficient to take $F = \{-, \times, \div, exp, \sqrt{}\}$ and $T = \{x, \pi, 2, 1, -1, R, R^+\}$, where $\div$ the protected division (does not allow division by zero), x represents an user input, R represents a real constant and R^+ represents a positive real constant. The normal distribution pdf can be described as

$$\left(1 \div (\sqrt{2 * \pi} * R^+) * \exp((-1 \div (2 * R^+ * R^+)) * (x - R) * (x - R)\right),$$

for example.

Each program can be represented by an n-ary tree, where n is the maximum possible of arguments for a function in F. For the normal distribution example we may use the tree in Figure 1. The tree representing a program is not unique unless we specify a set of rules for parsing a program to a tree, however this is negligible for our purpose.

Programs can be created randomly by traversing a structure called *Probabilistic Prototype Tree* (PPT). The PPT is a n-ary tree with n, again, representing the maximum arity of an instruction in F. The

node at depth $d \geq 0$ and horizontal position $w \geq 0$ (width) is represented by $N_{d,w}$. Each node contains a probability vector $\mathbf{P}_{d,w}$ whose entries are $\mathbf{P}_{d,w}(I)$ for each $I \in F \cup T$ such that

$$\sum_{I \in F \cup T} P_{d,w}(I) = 1, \ \forall N_{d,w}.$$

That is, each node has the probability distribution of the possible instructions in the programs at the respective node of their tree representation. The PPT is traversed in a depth first fashion from left to right, starting at $N_{0,0}$. For each accessed node, an instruction I is selected with probability $P_{d,w}(I)$ and denoted $I_{d,w}$. If $I_{d,w} \in F$, then a subtree is created for each argument of $I_{d,w}$. If $I_{d,w} \in T$ then it is replaced by an instance $V_{d,w}(I_{d,w})$ of that terminal. This instance equals $I_{d,w}$ if $P_{d,w}(I_{d,w})$ is greater than a certain threshold T_I and equals a randomly generated number $R_{d,w}$ otherwise. For each terminal instruction $I \in T$ there corresponds a threshold T_I and these are not changed throughout the iterations. The authors in Salustowicz & Schmidhuber 1997 also consider the *growing* and *pruning* of the PPT to reduce the memory requirements of the algorithm. Initially there is only the node $N_{0,0}$. If $I_{d,w} \in F$ is chosen and the subtree for its arguments are missing in the PPT, then additional nodes are created (growing). Conversely, if the probability of accessing a certain node in the PPT is too small, the node is deleted from the PPT (pruning).

PIPE has two learning mechanics: elitist learning and generation-based learning. These two mechanics alternate until a stopping criterion is met. Generation-based learning comprises five distinct phases.

1. **Creation of a program population.** A population of programs is created according to the rules mentioned earlier. These programs are enumerated as PROG_j, $0 < j \leq PS$, with PS denoting the population size. Probabilities in each node are initialized in a random way but maintaining their sum equal to 1.

2. **Population evaluation.** Each PROG_j in the population is evaluated regarding a certain fitness function. This is a numeric value assigned by a function $\text{FIT}(\text{PROG}_j)$. The programs are ranked in ascending order of those values. The best program in the current population is denoted PROG_b while the best program found so far is denoted PROG^{el}.

3. **Learning from the population.** The probabilities in each node of the PPT are modified as to increase the likelihood of PROG_b being generated. The following steps can be stored as the content of an `adaptPPTtowards`(PROG_b) routine at the time of the implementation by the reader. First the probability $P(PROG_b)$ is obtained as $\prod P_{d,w}(I_{d,w})$ for each instruction $I_{d,w}$ used in the production of the candidate PROG_b. A target probability is calculated as

$$P_{TARGET} = P(PROG_b) + lr[1 - P(PROG_b)] \times \frac{\varepsilon + FIT(PROG^{el})}{\varepsilon + FIT(PROG_b)},$$

in which the constant lr denotes the learning rate of the algorithm, and ε is a user-defined positive real constant. The fraction in the right hand side of the equation implements the fitness-dependent-learning (fdl). If ε is chosen such that $\varepsilon \ll FIT(PROG^{el})$ then generations with lower quality (higher fitness values) programs do not influence much the learning process, allowing for the use of smaller populations. Once P_{TARGET} is obtained, all the probabilities

$P_{d,w}(I_{d,w})$ for the instructions used in $PROG_b$ are increased iteractively as seen in algorithm 1, where c^{lr} denotes a constant that influences the number of iterations and the precision. The choice of this constant is subjective. Lower values will imply more iterations and more precision while higher values will do the opposite. Then, each terminal used in the construction of $PROB_b$ is stored in the respective node of the PPT, that is, $I_{d,w} := V_{d,w}(I_{d,w})$ for each terminal instruction $I_{d,w}$ used in $PROG_b$.

4. **Mutation of the PPT.** In this step, the nodes accessed during the production of $PROG_b$ are mutated with a probability P_{M_p} given by

$$P_{M_p} = \frac{P_M}{(l+k)\sqrt{|PROG_b|}},$$

where P_M is a user defined parameter controlling the overall probability of mutation. The previous formula is empirically justified in Salustowicz & Schmidhuber 1997. If a node is to be mutated, the probability $P_{d,w}(I_{d,w})$ is changed to $P_{d,w}(I_{d,w}) + mr \cdot (1 - P_{d,w}(I_{d,w}))$, in which mr represents a mutation rate. Notice that this change is small if $P_{d,w}(I_{d,w})$ is already large. After the mutation step every modified node is normalized to keep the sum of probabilities equal to 1.

5. **PPT pruning.** If $P_{d,w}(I_{d,w})$ becomes too small for a certain node $N_{d,w}$ and instruction $I_{d,w} \in F$ then the subtrees corresponding to the possible arguments of $I_{d,w}$ are deleted from the PPT.

After the generation-based learning, elitist learning takes place by repeating the previous procedure using $PROG^{el}$ instead of $PROG_b$. However, during the elitist learning mutation is not performed. The PPT is then pruned accordingly.

3 - THE NEW PROPOSAL

Given a data set, we propose the use of the PIPE method, with minor modifications, to generate a continuous function that resembles the empirical cumulative distribution of the data. We argue that, for a suitable choice of the sets F and T, it is possible to achieve a good fit of the data while controlling the complexity of the model by limiting the maximum height of the PPT. The minor modifications considered are:

1) During the generation of a program, if a node $N_{d,w}$ is to receive an instruction $I_{d,w} \in$ representing an user input variable (a data point), we set the respective threshold to 1.

2) The maximum size of the program, measured by the height of the tree representing it, is controlled. Programs are not allowed to grow indeterminately. To achieve this, we modify the

Algorithm 1: Updating the PPT.

```
repeat
    forall I_{d,w} in PROG_b do
        P_{d,w}(I_{d,w}) := P_{d,w}(I_{d,w}) + c^{lr} · lr · (1 − P_{d,w}(I_{d,w}));
until P(PROG_b)>P_TARGET;
```

nodes at the maximum depth of the PPT by forcing $P_{d,w}(I) = 0$ for every I∈F and normalizing the distribution $\mathbf{P}_{d,w}$ again.

3) No pruning or growing is performed, since memory consumption for small PPTs is not of concern in this context.

4) At each generation we randomly choose to adapt the PPT towards $PROG_{el}$ or towards $PROG_b$. Mutation occurs regardless of the choice.

These modifications simplified the code while yielding good results in our investigations. Also, they help to incorporate some aspects of the problem at hand. In this paper we choose the functions and terminal sets to be $F = \{\times, \div, +, pow(\cdot,\cdot), \exp(-\cdot), \log(\cdot)\}$ and $I = \{x, R\}$, where $\div$ represents the protected division, a^b is represented by $pow(a, b)$, $\exp(-x)$ is the usual e^{-x}, $\log(x)$ is the protected logarithm (does not accept negative arguments), x represents an user input and R represents a random number between 0 and a specified maximum value. This is the set of functions and terminals we use throughout the paper unless stated otherwise. The fitness function if given by:

$$FIT(PROG) = -\frac{1}{n}\sum_{i=1}^{n}(PROG(x_i) - \hat{F}_n(x_i))^2,$$

where x_i is ith observation of the data set, n is size of the data set, and $\hat{F}_n$ is the empirical distribution function. Our implementation will run for a number of generations and then stop. The overall sequence of an implementation for the PIPE algorithm is as follows.

- Set the following parameters: the elements in F∪T, size of the program population, number of generations, maximum size of the programs, learning rate, ε, mutation rate, maximum value of a generated random number and the probability of adapting the PPT towards the elite program.
- Initialize the PPT and assign probabilities to the vectors in every node. Let the PPT be as high as the maximum size of a program. Remember to set the probability of non-terminal symbols to zero for the leaf nodes.
- Read the data and obtain the empirical distribution function.
- For every generation:
  - Generate a random population of functions from the PPT.
  - Evaluate the fitness of every function.
  - Found the best program in the current generation and obtain $PROB(PROG_b)$. Replace $PROG^{el}$ if needed.
  - Adapt the PPT towards $PROG_b$ or $PROG_e l$ randomly.
  - Mutate the PPT, if it was adapted towards $PROG_b$.
  - Normalize $P_{d,w}$ for every node of the PPT.

4 - SIMULATION STUDIES

In this section we investigate how well the models proposed by the PIPE algorithm fit the data when compared to the true distribution of the data. For several distributions with cdf F and several sample sizes n, we generate an artificial data set of size n from F and use the proposed method to generate a cdf for the data. We compare the proposed distribution to the distribution F by means of the logarithm of the likelihood function based on the data (log-likelihood). The Anderson-Darling and Kolmogorov-Smirnov tests (see Stephens 1974) are also performed to verify the adequacy of the model to the data. The simulation can be described as follows.

- Generate a data set S from F.
- Estimate the parameters of F using the data and the maximum likelihood method.
- Run the proposed method and obtain a cdf G.
- Perform both the Anderson-Darling and Kolmogorov-Smirnov tests, under a significance level of 0.10.
- Calculate the log-likelihood of G based on the data and compare to that of F.

Results of the simulation are presented in Table I. The Kolmogorov-Smirnov (KS) test considers the overall fit of the distribution, whereas the Anderson-Darling (AD) test is built to emphasize the fit in the tails of the distribution (see Stephens 1974). The columns of Table I represent, respectively, the sample size, the distribution from which the data was simulated, the logarithm of the maximized likelihood for the model in the previous column with its parameters being estimated by the maximum likelihood method, the logarithm of the likelihood of the PIPE model and the p-values for the KS and AD tests. Criteria such as the Akaike information criterion (AIC) and Bayesian information criterion (BIC) are not used since we do not have a solid understanding of how would the comparison apply to the PIPE generated distributions. For all of the simulation runs, the parameters of the PIPE algorithm were set as follows: 100 candidates per generation, 3000 generations, $T_l = 0.2$ for all terminal numbers, 100 as the maximum random number generated, height of the PPT set to 3, learning rate of 0.001, $c^{lr} = 0.01$, $\varepsilon = 0.01$, $P_M = 0.1$, mutation rate of 0.1 and the probability of adapting the PPT towards the elite program was set to 0.5. The column LL represents the logarithm of the maximized likelihood function.

We chose the normal distribution because it is widely used in practical situations, the exponential and gamma distributions for being popular lifetime distributions, the Cauchy distribution for its heavy tails, a skew-t distribution from Fernandez & Steel 1998 for the heavy tails and skewness, the Pareto distribution for its shift, the beta distribution for being confined in the $(0, 1)$ interval and a mixture of two normal distributions for bimodal data. Certainly it was not expected that the PIPE distributions would outperform the distributions that originated the data, even though it happened for a few runs. During this simulation we noticed that the data generated from the normal, exponential, gamma, normal mixture, skew-t and Pareto distributions were easily modeled by the proposed distributions from the PIPE algorithm. One run of the algorithm was most of the times enough to obtain a distribution that would achieve a log-likelihood close to or better than the original model. We ran the

Table I. Results support the PIPE method distribution.

		ll		p-value	
n	Distribution	Fitted	PIPE	KS	AD
10	Normal(0,1)	−9.41	−11.34	0.5360	0.6047
	Cauchy(0,1)	−22.60	−21.77	0.8633	0.6579
	Exponential(1)	−15.62	−14.55	0.8282	0.6536
	Gamma(3,1)	−19.94	−18.68	0.6581	0.9247
	Beta(3,5)	6.20	5.30	0.9448	0.8216
	Pareto (1,3)	−9.99	−9.28	0.8101	0.8564
	MN(0,5,1,1,0.25)	−12.14	−12.80	0.7953	0.9348
	Skew-t(2,2)	−23.90	−22.94	0.9585	0.9699
30	Normal(0,1)	−49.13	−49.16	0.8098	0.6490
	Cauchy(0,1)	−77.36	−93.89	0.2368	0.0111
	Exponential(1)	−15.61	−32.99	0.1538	0.2048
	Gamma(3,1)	−59.53	−61.58	0.6300	0.6858
	Beta(3,5)	18.96	17.09	0.8287	0.7421
	Pareto(1,3)	−81.87	−83.96	0.8036	0.8051
	MN(0,5,1,1,0.25)	−36.13	−40.15	0.6238	0.6667
	Skew-t(2,2)	−91.50	−79.11	0.4857	0.2997
50	Normal(0,1)	−70.32	−70.35	0.4798	0.2208
	Cauchy(0,1)	−114.37	−133.22	0.7690	0.5053
	Exponential(1)	−56.80	−60.99	0.7635	0.6453
	Gamma(3,1)	−92.53	−94.34	0.7588	0.7505
	Beta(3,5)	24.16	24.33	0.8891	0.9453
	Pareto(1,3)	−152.28	−152.32	0.9630	0.9471
	MN(0,5,1,1,0.25)	−59.16	−65.44	0.7566	0.5720
	Skew-t	−104.19	−97.11	0.3121	0.1469

algorithm at most five times with different seeds for the random number generation and presented the best results in here. The Cauchy distribution proved to be the more challenging distribution. Better levels of log-likelihood than those shown in Table I for the Cauchy distribution were achieved but at the expense of failing the AD test at the 0.1 significance level. Solutions for this problem would possibly include the addition or removal of elements in the *F* set or the possibility of applying different weights to the data points in the tails of the data set when calculating the fitness values, if the user knows beforehand that the data supplied is heavy tailed. We noticed no increasing nor decreasing of the performance of the algorithm regarding the sample size.

5 - REAL DATA APPLICATION AND COMPARISONS

In this section we apply the PIPE method to a pair of data sets that were modeled in already published papers on probability distribution families. For each re-visited paper we compare the likelihood of the distributions proposed in that paper to the one suggested by the PIPE method. To verify the adequacy of the model proposed by the PIPE method to the data we resort to the KS and AD tests, the visual inspection of the theoretical quantiles plotted against the empirical quantiles and the plot of the probability density function overlapping the data histogram. To the quantile plot we add a simulated 95% envelope for the empirical quantiles obtained by bootstrapping the original sample. This is done by generating a high number B of pseudo-samples by sampling, with replacement, the original data set. Order each pseudo-sample to obtain its empirical quantiles. There will be, after the simulation, B samples of each quantile. For each quantile q find the quantiles 2.5% and 97.5% of its respective sample and use them to plot the envelope.

The results suggest that good fitness levels for real data can be attained by the distributions proposed by the PIPE method, in agreement to what was suggested by the simulations. These distributions presented simpler mathematical expressions for the cdf and pdf when compared to the distributions in the previous paper on those data sets. Whenever estimation was necessary, it was done by the `fitdistr()` routine from the `MASS` package in `R`. For some models, however, we reported the estimates of the parameters and the respective logarithm of the likelihood from the original papers.

The parameters for the PIPE algorithm in this section are the same used in the simulation section.

Wheaton river data set

The following data consist of 72 observations of the exceedances of flood peaks, measured in m^3/s, of the Wheaton River located near Carcross in the Yukon Territory, Canada. These data were analyzed by several authors, amongst which we cite a few. In Akinsete et al. 2008 the four parameter beta-Pareto (BP) distribution was used to model the data. Its density is given by

$$f(x) = \frac{k}{\theta B(\alpha,\beta)} \left\{1 - \left(\frac{x}{\theta}\right)^{-k}\right\}^{\alpha-1} \left(\frac{x}{\theta}\right)^{-k\beta-1},$$

with $x \geq \theta$ and $\alpha, \beta, \theta, k > 0$. A better fit, regarding the most common criteria such as Akaike information criterion (AIC), was found by Alshawarbeh et al. 2012 using the beta-Cauchy (BC) distribution. The pdf for the beta-Cauchy distribution is given by

$$f(x) = \frac{\lambda}{\pi B(\alpha,\beta)} \left\{\frac{1}{2} + \frac{1}{\pi} arctan\left(\frac{x-\theta}{\lambda}\right)\right\}^{\alpha-1} \left\{\frac{1}{2} - \frac{1}{\pi} arctan\left(\frac{x-\theta}{\lambda}\right)\right\}^{\beta-1} \frac{1}{\lambda^2 + (x-\theta)^2},$$

where $-\infty < x < \infty$, $0 < \alpha, \beta, \lambda < \infty$ and $-\infty < \theta < \infty$. Finally, Cordeiro et al. 2013 used the exponentiated generalized Gumbel (EGGU) distribution to achieve an even better fit for the data. The pdf for the EGGU distribution is

$$\begin{aligned} f(x) =& \alpha\beta\sigma^{-1} \left\{1 - \left[1 - \exp\left\{-\exp\left(-\frac{x-\mu}{\sigma}\right)\right\}\right]^{\alpha}\right\}^{\beta-1} \\ & \times \left[1 - \exp\left\{\exp\left(-\frac{x-\mu}{\sigma}\right)\right\}\right]^{\alpha-1} \exp\left\{-\exp\left(-\frac{x-\mu}{\sigma}\right)\right\} \exp\left(-\frac{x-\mu}{\sigma}\right), \end{aligned}$$

with $-\infty < x < \infty$, $0 < \alpha, \beta, \sigma < \infty$ and $-\infty < \mu < \infty$.

These data were also analyzed in Bourguignon et al. 2013, but they used the Kumaraswamy-Pareto (KWP) distribution with pdf

$$f(x) = \frac{abk\beta^k}{x^{k+1}} \left[1 - \left(\frac{\beta}{x}\right)^k\right]^{a-1} \left\{1 - \left[1 - \left(\frac{\beta}{x}\right)^k\right]^a\right\}^{b-1},$$

for $x \geq \beta$ and $0 < a,\ b\,, k\,, \beta < \infty$. The histogram for this data set is very skewed to the right as seen in Figure 2.

We obtained, using the PIPE method, for this data set the following cdf:

$$F(x) = \frac{1.17258x}{x + 11.6991},\ 0 \leq x < x_1,$$

with $F(x_1) = 1$, $x_1 = 67.7894$. The corresponding pdf is given by

$$f(x) = \frac{17.7181}{(x + 11.6991)^2},\ 0 \leq x < x_1.$$

For this example we removed $\exp(-\cdot)$ from F, to search for simpler alternatives. Table II shows the information of the fitting of the models in the previous papers. We used the estimates and information available from the original papers. The information on the likelihood of the BP model was not available in Akinsete et al. 2008 and was omitted here as well.

Table II. Estimation for the Wheaton river data set.

	ϕ	$\hat{\phi}$	LL
BP	$(\alpha, \beta, k, \theta)$	$(7.69, 85.75, 0.02, 0.10)$	–
KWP	(a, b, k, β)	$(2.86, 85.85, 0.05, 0.10)$	−271.20
BC	$(\alpha, \beta, \lambda, \theta)$	$(387.65, 1.46, 2.05, 0.08)$	−260.48
EGGU	(a, b, μ, σ)	$(0.11, 0.48, 2.63, 1.63)$	−256.90
PIPE	–	–	−235.83

Figure 2 shows the pdfs of the EGGU and the PIPE model overlapping the data's histogram and the quantile plot with 95% bootstrap simulated envelope for the PIPE model. The histogram suggests that the proposed distribution is suitable for the data. The quantile plot shows a good agreement between the model and the data. The largest observation in this data set is 64.0, which is much larger than the second largest one, 39.0, considering the sample standard-deviation of 12.41m^3/s. The PIPE model seems to capture the behavior of this possibly extreme value very well. The KS test returned a p-value equal to 0.3197 and AS test returned 0.3341 suggesting that there is no evidence of a bad fit. As in the previous example, the PIPE distribution achieved the highest likelihood. We emphasize that the distribution obtained from the PIPE method has a much simpler analytic expression for its cdf and cdf than the other candidates.

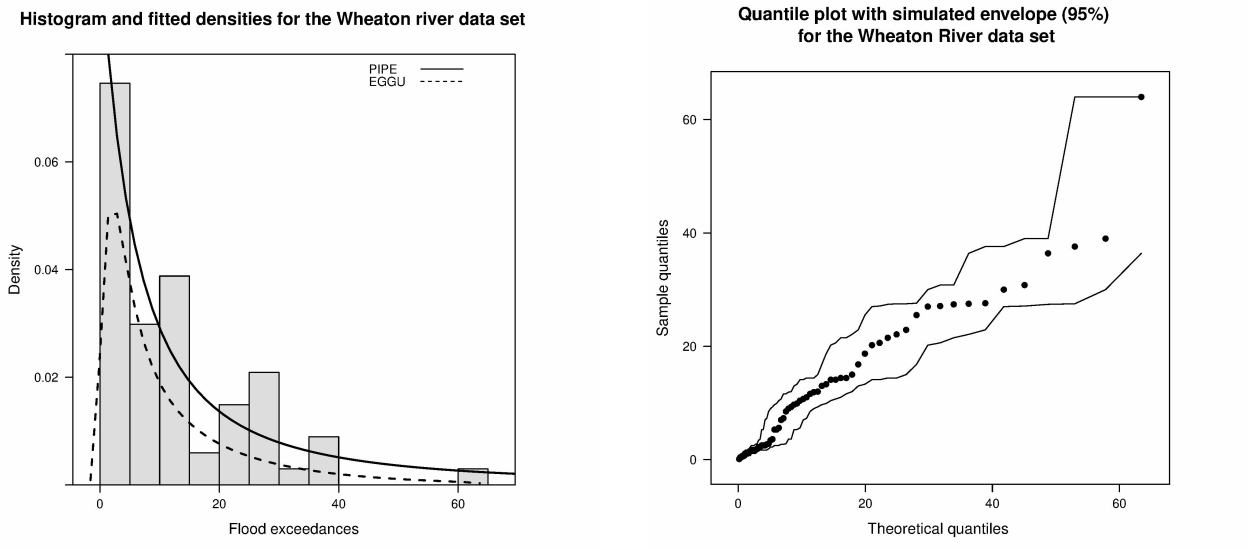

Figure 2. Histogram and quantile plot for the Wheaton river data set.

Ball bearing data set

The next data set is used in many papers on lifetime distributions. It consists of 23 observations of the fatigue failure times, measured in millions of revolutions, of ball bearings. In Nassar & Nada 2012 the authors proposed to use the beta-exponential-geometric (BEG) distribution, which they presented, and it fitted the data nicely. Its pdf is given by

$$f(x) = \frac{1}{B(a,b)} \left(\frac{1-e^{-\beta x}}{1-pe^{-\beta x}} \right)^{a-1} \frac{\beta(1-p)^b e^{-b\beta x}}{(1-pe^{-\beta x})^{b+1}},$$

for $x > 0$, $p \in (0,1)$, a, b, $\beta > 0$. This is the result of the composition of the beta distribution and the exponential geometric distribution (see Adamidis & Loukas 1998). The gamma and exponentiated Weibull (EW) distribution from Mudholkar & Srivastava 1993, with cdf $F(x) = \{1-exp[-(x/\lambda)^k]\}^\alpha$, $\lambda > 0$, $\alpha > 0$, $x > 0$, distributions are also considered here. The EW distribution is a good benchmark since it is known to be able to fit a very wide variety of data. The data are:

17.23, 28.92, 33.00, 41.52, 42.12, 45.60, 48.80, 51.84, 51.96, 54.12, 55.56, 67.80, 68.64, 68.64, 68.88, 84.12, 93.12, 98.64, 105.12, 105.84, 127.92, 128.04, 173.4.

Table III shows the fitted distributions and the logarithm of the maximized distribution.

The PIPE method proposed the function

$$F(x) = 0.000847396^{e^{-x/26.272}}, \ -\infty < x < \infty$$

as cdf. The Kolmogorov-Smirnov test for the PIPE distribution returned a p-value of 0.8974, while for the Anderson-Darling test the value was 0.8007. Figure 3 suggests the PIPE distribution is adequate for the data and this agrees with the previous tests. The logarithm of the likelihood for this model was −113.5005, which is slightly lower than the one from the BEG distribution. However, it is computationally much simpler to obtain values for $P(X \leq x)$ under the PIPE model than under the BEG model — even a handheld calculator suffices. There is the inconvenience of $P(X < 0) \neq 0$, however it is a small value for this particular application.

Table III. Estimation for the ball bearings data set.

	ϕ	$\hat{\phi}$	LL
BEG	(p, a, b, β)	$(0.35, 5.38, 2.58, 0.01)$	−113.06
EW	(α, k, λ)	$(4.49, 1.06, 34.83)$	−113.06
Weibull	(k, λ)	$(2.10, 81.88)$	−113.73
Gamma	(α, β)	$(3.99, 0.06)$	−113.10
PIPE	—	—	−113.50

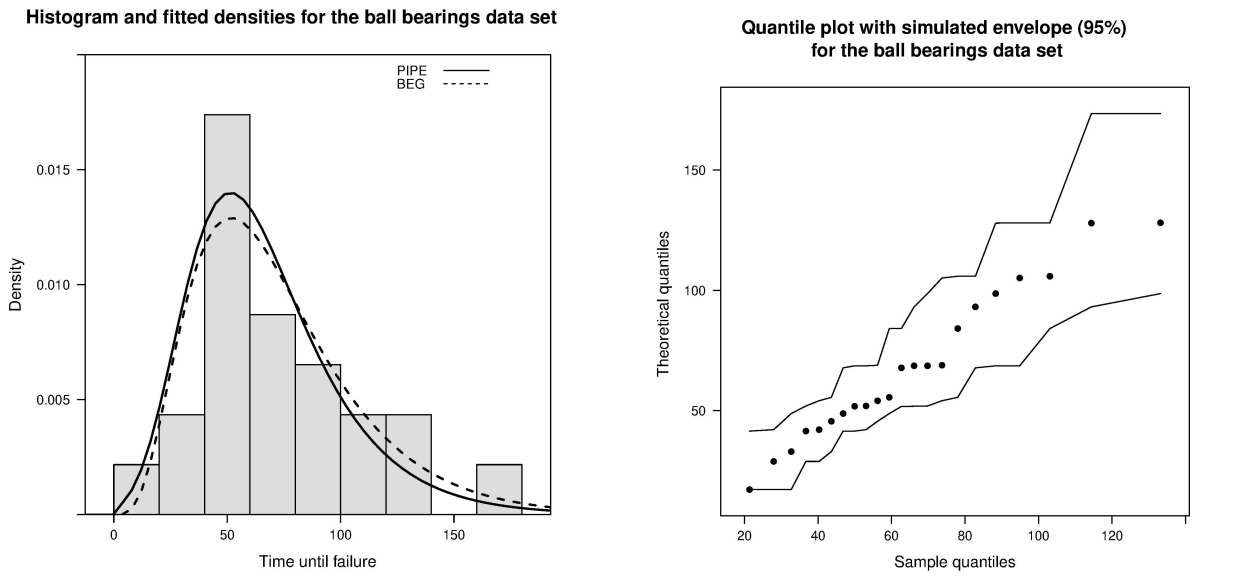

Figure 3. Histogram and quantile plot for the ball bearings data set.

6 - ADDRESSING ISSUES AND FINAL COMMENTS

In this paper we suggested the use of the PIPE method for generating possibly new continuous probability distributions from a given data set. The method described here allows the development of relatively simple distributions that performed better than several more complicated available distributions in two data sets. Its use was also illustrated by several runs of simulations with artificial data. The simulation studies suggest that the PIPE algorithm can properly handle data modeling from a wide variety, including heavy tailed, bimodal and skewed data.

We recommend the use of this method if there is no evidence in the problem being studied that may lead to the use of a specific distribution, existing or new. The main advantage over the use of distributions in the so called G-classes (as the beta-G and gamma-G classes) is that it is possible to control the complexity of the new distribution using the PIPE method. The cdf of the G-classes distributions tend to have mathematically complicated formulae.

There are also situations where we do not recommend the use of this method. For instance, if it is possible to develop an specific distribution from the underlying physical properties of the

problem, we do not recommend using the PIPE method unless there is a clear advantage in doing so. Another situation where we do not recommend its use is when the problem satisfies all the theoretical criteria for the use of an existing distribution, such was the waiting times in Poisson processes or small measurement errors that are easily handled by the normal distribution. Adequacy measures and the likelihood criteria should never replace proper mathematical analysis of the problem.

Future efforts may be able to describe the mathematical properties of the search in the space of the candidates distributions. A useful refinement that we were not able to provide is to obtain an algorithm for generating a function that integrates up to a constant. If there was such an algorithm we would be able to find distributions with support on the whole real line or in the $(0, \infty)$ half of the line in a much easier way. However, finding a distribution that has support on bounded intervals is also reasonable.

We observed many other interesting events during the development of this paper, while working on other examples. Changing the numbers in the cdfs found by the PIPE method for unknown parameters and estimating them by maximum likelihood led to values very close to those proposed by the algorithm for many data sets (real and simulated). In some data sets we changed the seed for the random number generators or modified some of the initial conditions, such as the values in $\mathbf{P}_{d,w}$. These runs of the algorithm led to different solutions (cdfs) since it is a non-deterministic algorithm. However, these solutions were usually very similar to each other. For the ball bearings and the Wheaton river data sets there were two groups of very similar solutions that were suggested by the PIPE method depending on the initial settings and the method did not find anything outside those groups. Last, the output of the algorithm seems to depend heavily on the choice of *F*. Adding or removing elements to *F* changes drastically the behavior of the algorithm in our experience. We suggest starting from very few elements and adding more as needed. We also suggest keeping the mutation probability high enough. It seems to play a major role in the final result.

Overall, our opinion is that the PIPE algorithm is an interesting and promising alternative in the field of data modeling.

Acknowledgments

The first and third authors are thankful to Fundação de Amparo a Ciência e Tecnologia do Estado de Pernambuco - FACEP (grant IBPG-0441-1.02/12) for the partial financial support during this research. The second author was partially supported by Conselho Nacional de Desenvolvimento Científico e Tecnológico - CNPq (305336/2017-7).
The content of this paper is derived from Pinho 2015, which is the PhD thesis of the first author on the use of some machine learning algorithms in tasks typical of statistical analysis and on the building of new probability distribution based on already existing ones.

REFERENCES

ADAMIDIS K & LOUKAS S. 1998. A lifetime distribution with decreasing failure rate. Stat Probab Lett 39: 35-42.

AKINSETE A, FAMOYE F & LEE C. 2008. The beta-Pareto distribution. Statistics 42: 547-563.

ALSHAWRBEH E, LEE C & FAMOYE F. 2012. The beta-Cauchy distribution. JPSS 10: 41-57.

ALY E & BENKHEROUF L. 2011. A new family of distributions based on probability generating functions. Sankhya B - App and Interd Stat 73: 70-80.

ALZAATREH A, LEE C & FAMOYE F. 2013. A new method for generating families of continuous distributions. METRON 71: 63-79.

BOURGUIGNON M, SILVA RB, ZEA LM & CORDEIRO GM. 2013. The Kumaraswamy Pareto distribution. JSTA 12: 129-144.

CORDEIRO GM, ORTEGA EMM & CUNHA DCC. 2013. The Exponentiated Generalized Class of Distributions. Data Sci J 11: 777-803.

FERNANDEZ C & STEEL MFJ. 1998. On Bayesian modeling of fat tails and skewness. JASA 93: 359-371.

FISCHER M & VAUGHAN DC. 2010. The Beta-hyperbolic secant (BHS) distribution. Austrian J Stat 39: 245-258.

JONES MC. 2004. Families of distributions arising from distributions of order statistics. TEST 13: 1-43.

MUDHOLKAR GS & SRIVASTAVA DK. 1993. Exponentiated Weibull family for analyzing bathtub failure real data. IEEE Trans Reliab 42: 299-302.

NADARAJAH S & GUPTA AK. 2004. The beta Fréchet distribution. Far East J Theor Stat 14: 15-24.

NADARAJAH S & KOTZ S. 2004. The beta Gumbel distribution. Math Probl Eng 4: 323-332.

NADARAJAH S & KOTZ S. 2006. The beta exponential distribution. Reliab Eng Syst 91: 689-697.

NASSAR M & NADA N. 2012. A new generalization of the exponential-geometric distribution. Journal of Statistics: Adv Group Theory Appl 7: 25-48.

PARANAÍBA PF, ORTEGA EMM, CORDEIRO GM & PESCIM RR. 2011. The beta Burr XII distribution with application to lifetime data. CSDA 55: 1118-1136.

PINHO LGB. 2015. Building new probability distributions: the composition method and a computer based approach. Universidade Federal de Pernambuco, Recife, Brazil, 105 p.

SALUSTOWICZ R & SCHMIDHUBER J. 1997. Probabilistic incremental program evolution. Evol Comput 5: 123-141.

STEPHENS MA. 1974. EDF Statistics for Goodness of Fit and Some Comparisons. JASA 69: 730-737.

TAHIR MH & NADARAJAH S. 2014. Parameter induction in continuous univariate distributions: Well-established G families. An Acad Bras Cienc 87: 539-568.

ZOGRAFOS K & BALAKRISHNAN N. 2009. On families of beta- and generalized gamma-generated distributions and associated inference. Stat Methodol 6: 344-362.

How to cite

PINHO LGB, NOBRE JS & CORDEIRO GM. 2022. Continuous Probability Distributions generated by the PIPE Algorithm. An Acad Bras Cienc 94: e20201542. DOI 10.1590/0001-3765202220201542.

Manuscript received on January 7, 2020;
accepted for publication on November 4, 2021

LUIS G.B. PINHO[1]
https://orcid.org/0000-0002-1680-6303

JUVÊNCIO S. NOBRE[1]
https://orcid.org/0000-0002-7321-3221

GAUSS M. CORDEIRO[2]
https://orcid.org/0000-0002-3052-6551

[1]Universidade Federal do Ceará (UFC), Centro de Ciências, Campus do Pici, Av. Mister Hull, s/n, Bloco 910, 60440-900 Fortaleza, CE, Brazil
[2]Universidade Federal de Pernambuco, (CCEN), Departamento de Estatística, Av. Jorn. Aníbal Fernandes, 497-629, Cidade Universitária, 50740-540 Recife, PE, Brazil

Correspondence to: **Luis Gustavo Bastos Pinho**

E-mail: lgbpinho@gmail.com

Author contributions

Luis Gustavo Bastos Pinho: conceived the main idea of the manuscript, coded the algorithm, ran the experiments and examples and wrote the manuscript with the support of the other authors. Juvêncio Santos Nobre: theoretical formalism, checked analytical calculations, revised the manuscript and experiments, interpreted the results of the experiments, helped supervise the production of the manuscript. Gauss Moutinho Cordeiro: theoretical formalism, checked analytical calculations, selected adversarial models, revised the manuscript, supervised the project.