Use of metaheuristic methods in the estimation of indices of non-normal processes

Research practise 1: Progress presentation

Andrea Molina-Alonso[1] Myladis Rocio Cogollo-Flórez[2]

[1]Responsible
amolin13@eafit.edu.co

[2]Tutor
mcogollo@eafit.edu.co

EAFIT University
Mathematical Engineering
Department of Mathematical Sciences
October 20, 2015

Log-likelihood function

The probability density function of the Burr XII distribution is defined by:

$$f(x) = \frac{k\ c\ x^{c-1}}{(1+x^c)^{k+1}}; \quad x > 0,\ c > 0,\ k > 0$$

k: shape parameter
c: scale parameter

The logarithm of the likelihood function for this distribution is:

$$\underbrace{\ln(L) = n(\ln(c) + \ln(k)) + (c-1)\sum_{i=1}^{n} \ln(x_i) - (k+1)\left(\sum_{i=1}^{n} \ln(1 + x_i^c)\right)}_{\text{Objective function to maximize}}$$

Meta-heuristics used

- A heuristic is a technique (consisting of a rule or a set of rules) which seeks (and hopefully finds) good solutions at a reasonable computational cost.
- A meta-heuristic refers to a master strategy that guides and modifies other heuristics to produce solutions beyond those that are normally generated in a quest for local optimality.

- Particle Swarm Optimization (PSO)
- Particle Swarm Optimization Modified (MPSO)
- Artificial Bee Colony (ABC)

[Voß, 2001]

Algorithm 1 PSO pseudocode

Initialize the position $x_i(0) \quad \forall i \in 1:N$
Initialize the particle's best position $p_i(0) = x_i(0)$
Calculate the fitness of each particle and if $f(x_j(0)) \geq f(x_i(0)) \quad \forall i \neq j$ initialize the global best as $g = x_j(0)$
Until a stopping criterion is met, repeat:
for $i = 1$ to N **do**
    $v_i(t+1) = v_i(t) + c_1(p_i - x_i(t))R_1 + c_2(g - x_i(t))R_2$
    $x_i(t+1) = x_i(t) + v_i(t+1)$
    Evaluate the fitness of the particle $f(x_i(t+1))$
    if $f(x_i(t+1)) \geq f(p_i)$ **then**
        $p_i = x_i(t+1)$
    end if
    if $f(x_i(t+1)) \geq f(g)$ **then**
        $g = x_i(t+1)$
    end if
end for
return g

Taken from [Marini and Walczak, 2015]

Algorithm 2 MPSO pseudocode

Initialize the position $x_i(0) \quad \forall i \in 1:N$
Initialize the particle's best position $p_i(0) = x_i(0)$
Calculate the fitness of each particle and if $f(x_j(0)) \geq f(x_i(0)) \quad \forall i \neq j$ initialize the global best as $g = x_j(0)$
Until a stopping criterion is met, repeat:
for $i = 1$ to N **do**
  if $f(x_{id}(t)) \geq f(p_{id}(t))$ **then**
    $p_{id}(t) = x_{id}(t)$
  end if
  if $f(x_{id}(t)) \geq f(g)$ **then**
    $g = x_{id}(t)$
  end if
  $A_i(t) = \dfrac{fit_i(t) - Maxfit(t)}{Medfit(t) - Maxfit(t)}$
end for
for $i = 1$ to N **do**
  $a_i(t) = \dfrac{A_i(t)}{\sum_{j=1}^{N} A_j(t)}$
  $M_{id}(t) = a_i(t)[rand * (p_{id}(t) - p_{md}(t) - x_{id}(t)) + rand * (g - p_{md}(t) - x_{id}(t))]$
  $v_{id}(t+1) = v_i(t) + M_{id}(t)$
  $x_{id}(t+1) = x_{id}(t) + v_{id}(t+1) + \frac{1}{2} * [rand * (p_{id}(t) - x_{id}(t)) + rand * (g - x_{id}(t))]$
end for
return g

Taken from [Beheshti et al., 2013]

Algorithm 3 ABC pseudocode

```
Randomly generate SN points in the search space to find an initial population
Evaluate the objective function values of the population
Until a stopping criterion is met, repeat:
Move the employed bees onto their food sources and determine their nectar amounts
Move the onlookers onto the food sources and determine their nectar amounts
Move the scouts for searching new food sources
Memorize the best food source found so far
return  best food source
```

Taken from [Karaboga and Basturk, 2008, Gao et al., 2015]

Values of the method's parameters

The methods used have a number of parameters that determine its behavior and efficacy in optimizing a given problem.

Problem Dimensions	Fitness Evaluations	PSO Parameters			
		S	ω	ϕ_p	ϕ_g
2	400	25	0.3925	2.5586	1.3358
		29	-0.4349	-0.6504	2.2073
2	4,000	156	0.4091	2.1304	1.0575
		237	-0.2887	0.4862	2.5067
5	1,000	63	-0.3593	-0.7238	2.0289
		47	-0.1832	0.5287	3.1913
5	10,000	223	-0.3699	-0.1207	3.3657
		203	0.5069	2.5524	1.0056
10	2,000	63	0.6571	1.6319	0.6239
		204	-0.2134	-0.3344	2.3259
10	20,000	53	-0.3488	-0.2746	4.8976
20	40,000	69	-0.4438	-0.2699	3.3950
20	400,000	149	-0.3236	-0.1136	3.9789
		60	-0.4736	-0.9700	3.7904
		256	-0.3499	-0.0513	4.9087
30	600,000	95	-0.6031	-0.6485	2.6475
50	100,000	106	-0.2256	-0.1564	3.8876
100	200,000	161	-0.2089	-0.0787	3.7637

Taken from [Pedersen, 2010]

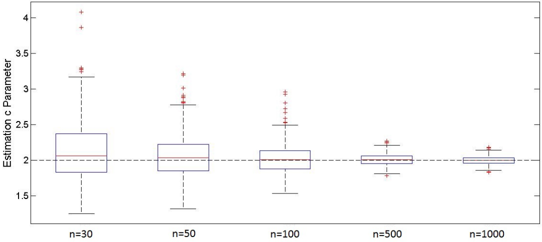
Results: Implementation of metaheuristic

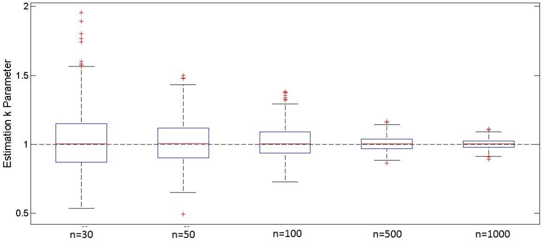
Algorithm 4 pseudocode generation of random variables

```
Select the sample size n
Select the values of the parameters c and k
for i = 1 to n do
    y = unif(0, 1)
    x_i = [ (1 / (1 - y))^(1/k) - 1 ]^(1/c)
end for
Evaluation of the function
Implementation of the heuristic method
```

$$x_i = \left[\left(\frac{1}{1-y} \right)^{1/k} - 1 \right]^{1/c}$$

Results: PSO

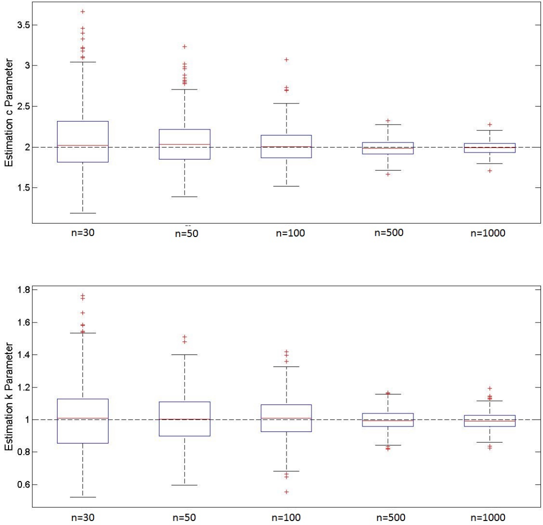




Note that..

The estimators are consistent

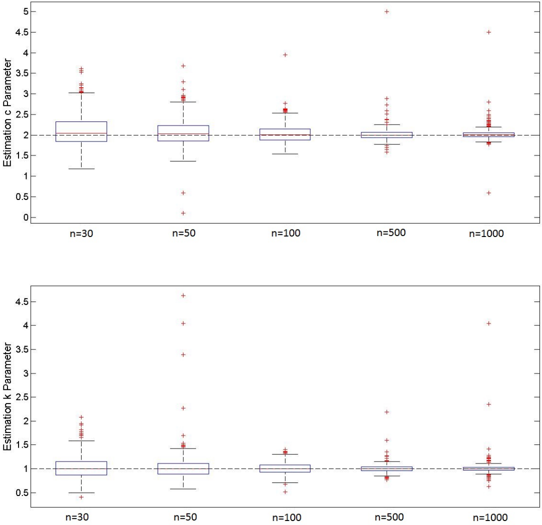
Results: MPSO



Note that..

The estimators are consistent

Results: ABC

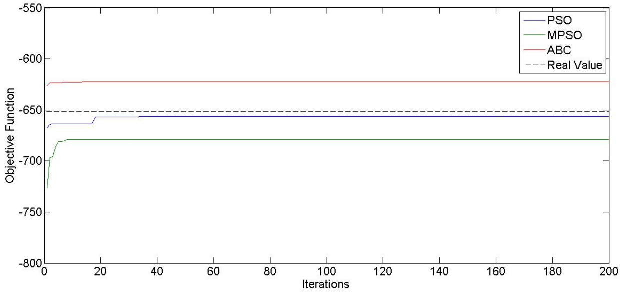


Note that..

The consistency of the estimators is affected by outliers

Convergence and computing time

For a sample data of $n = 500$ with parameters $c = 2$ and $k = 1$, we obtained the following results:



- PSO: 2,228106 s
- MPSO: 1,833951 s
- ABC: 0,963984 s

Parameters	n	[Abbasi et al., 2010]	PSO	MPSO	ABC	[Liu and Chen, 2006]
c=4,14224 k=9,13497	100	c=3,9113 k=10,1661	c=4,1007 k=8,3162	c=4,0761 k=8,2304	c=4,1007 k=8,3164	c=4,87371 k=6,15756
	1000	c=4,0531 k=9,0436	c=4,1620 k=9,4552	c=4,1625 k=9,4310	c=4,1620 k=9,4458	c=4,87371 k=6,15756
	2500	c=4,0758 k=9,0610	c=4,1422 k=9,3333	c=3,8042 k=8,0458	c=4,1270 k=9,2339	c=4,87371 k=6,15756
	10000	c=4,0998 k=9,0781	c=4,1264 k=9,1480	c=4,0712 k=8,6973	c=4,1278 k=9,1531	c=3,93893 k=19,864823
c=2 k=5	100	c=1,6068 k=4,2142	c=1,8809 k=4,6006	c=1,8442 k=4,6556	c=2,0498 k=5,5355	c=-8,83754 k=0,09995
	1000	c=1,8145 k=4,5983	c=2,0456 k=5,1508	c=2,0242 k=5,1657	c=2,2792 k=5,7089	c=2,55182 k=3,68994
	2500	c=1,8841 k=4,7489	c=1,9975 k=5,2704	c=1,9681 k=5,0915	c=2,0669 k=5,6685	c=3,58714 k=2,19903
	10000	c=1,9586 k=4,9288	c=1,9931 k=4,9063	c=1,9878 k=4,9445	c=2,0829 k=5,7299	c=2,770828 k=3,17098
c=3 k=4	100	c=2,5043 k=3,5291	c=2,7973 k=3,7460	c=2,7887 k=3,7531	c=2,9279 k=4,4239	c=3,22768 k=2,46376
	1000	c=2,8966 k=3,8751	c=2,9872 k=4,2007	c=2,9919 k=4,3154	c=3,1187 k=5,2343	c=2,53777 k=12,52234
	2500	c=2,8988 k=3,8987	c=3,0094 k=4,0019	c=3,0129 k=3,9275	c=3,0707 k=4,2874	c=-7,17600 k=0,07865
	10000	c=2,9576 k=3,9662	c=2,9870 k=4,0126	c=3,0044 k=3,8206	c=2,9870 k=4,0126	c=-7,17600 k=0,07865

Table: Comparison of the results of methods with other proposed in the literature

Project objectives

Objective	Percentage
Identify which heuristics methods are appropriate to find a general way to estimate the parameters of the Burr XII distribution	100%
Establish the found estimation method in a programming language	100%
Compare the results obtained with the estimation method with other proposed in the literature, using experimental data	30%

Table: Objectives

References I

- Abbasi, B., Hosseinifard, S. Z., and Coit, D. W. (2010).
  A neural network applied to estimate Burr XII distribution parameters.
  Reliability Engineering & System Safety, 95(6):647–654.
- Beheshti, Z., Shamsuddin, S. M. H., and Hasan, S. (2013).
  MPSO: Median-oriented Particle Swarm Optimization.
  Applied Mathematics and Computation, 219(11):5817–5836.
- Gao, W., Chan, F. T., Huang, L., and Liu, S. (2015).
  Bare bones artificial bee colony algorithm with parameter adaptation and fitness-based neighborhood.
  Information Sciences, 316:180–200.

References II

Karaboga, D. and Basturk, B. (2008).
On the performance of artificial bee colony (ABC) algorithm.
Applied soft computing, 8(1):687–697.

Liu, P.-H. and Chen, F.-L. (2006).
Process capability analysis of non-normal process data using the Burr XII distribution.
The International Journal of Advanced Manufacturing Technology, 27(9-10):975–984.

Marini, F. and Walczak, B. (2015).
Particle swarm optimization (PSO). A tutorial.
Chemometrics and Intelligent Laboratory Systems.

References III

Pedersen, M. E. H. (2010).
Good Parameters for Particle Swarm Optimization.
Technical report, Hvass Laboratories.

Voß, S. (2001).
Meta-heuristics: The state of the art.
In *Local Search for Planning and Scheduling*, volume 2148 of *Lecture Notes in Computer Science*, pages 1–23. Springer Berlin Heidelberg.

Thanks for your attention!!

Use of metaheuristic methods in the estimation of indices of non-normal processes.

Estimation with tabulated values

Skewness	Kurtosis	Clements's			Burr				
a_3	a_4	$ZP_{.00135}$	$ZP_{.5}$	$ZP_{.99865}$	$BZP_{.00135}$	$BZP_{.5}$	$BZP_{.99865}$	c	k
0	2	−1.966	0.000	1.966	−1.843	0.022	2.396	−18.148445	0.062932
0	2.2	−2.210	0.000	2.210	−1.959	0.037	2.697	−13.840637	0.093482
0	2.4	−2.442	0.000	2.442	−2.076	0.047	2.911	−12.134081	0.120321
0	2.6	−2.663	0.000	2.663	−2.197	0.053	3.078	−11.251863	0.146295
0	2.8	−2.839	0.000	2.839	−2.735	0.008	2.914	3.938938	19.864823
0	3	−3.000	0.000	3.000	−2.884	0.010	3.081	4.873717	6.157568
0	3.2	−3.140	0.000	3.140	−3.020	0.011	3.221	6.065153	3.745010
0	3.4	−3.261	0.000	3.261	−3.148	0.011	3.340	7.695948	2.700685
0	3.6	−3.366	0.000	3.366	−3.269	0.011	3.442	10.182078	2.089559
0	3.8	−3.458	0.000	3.458	−3.388	0.009	3.529	14.723762	1.664480
0	4	−3.539	0.000	3.539	−3.509	0.015	3.609	27.068908	1.325754
0	4.2	−3.611	0.000	3.611	−3.642	0.001	3.659	−195.260000	0.959315

Taken from [Liu and Chen, 2006]