

Estimação por Máxima Verossimilhança para manter o padrão dos parâmetros da distribuição Weibull via BFGS com a linguagem de programação Ox

Estimation by Maximum Likelihood to maintain the pattern of Weibull distribution parameters via BFGS with the programming language Ox

Marcelo dos Santos

Instituto Federal de Educação, Ciência e Tecnologia da Bahia (IFBA), *Campus* Eunápolis, Eunápolis, BA, Brasil

Universidade Federal de Pernambuco (UFPE), Departamento de Estatística (DE), Recife, PE, Brasil
marcelo20xjg@hotmail.com

Informações do Artigo



Histórico do Artigo

Submissão: 18 de julho de 2018.

Aceite: 24 de setembro de 2018.

Palavras-chave

Máxima Verossimilhança
Monte Carlo
BFGS
Ox

Keywords

Maximum Likelihood
Monte Carlo
BFGS
Ox

Resumo

No presente trabalho foi realizada uma avaliação numérica do desempenho do método iterativo BFGS, usando a Estimação de Máxima Verossimilhança para os parâmetros da distribuição de Weibull biparamétrica sem censura. Para se obterem os resultados numéricos, implementamos o código computacional na linguagem de programação Ox, utilizando a função MaxBFS disponível na linguagem, simulação Monte Carlo e rotinas estatísticas disponíveis na biblioteca da própria linguagem. O código utilizado encontra-se disponível no Apêndice 6. Os resultados numéricos apontam que a linguagem Ox é eficiente para problemas de maximização, além de validarem a aproximação assintótica normal para as distribuições marginais dos estimadores de Máxima Verossimilhança dos parâmetros da Weibull sem a presença de censura.

Abstract

In the present work a performance numerical evaluation of BFGS iterative method was performed, using the Maximum Likelihood Estimation for the parameters of the uncensored bivariate Weibull distribution. In order to get the numerical results, we implemented the computational code in the programming language Ox, using the MaxBFS function available in the language, Monte Carlo simulation and statistical routines available in the language library. The code used is available in Appendix 6. The numerical results indicate that the Ox language is efficient for maximization problems, besides validating the normal asymptotic approximation for the marginal distributions of the Maximum Likelihood estimators of the Weibull parameters without the presence of censorship.

1 Introdução

A distribuição de Weibull, gozando das distribuições exponencial e Rayleigh como seus casos particulares, é amplamente popular no que diz respeito à modelagem de sobrevivência. Frente a isso, diversos pesquisadores vêm propondo modificações da célebre distribuição para lidar com taxas de falhas. PERCONTINI em [10] propõe uma mistura das distribuições Weibull e Poisson

para análise de dados de sobrevivência. Com o mesmo objetivo, BARROS [3] propõe uma classe de Weibull generalizada a qual mostraram ser muito flexível para modelagem de dados de sobrevivência. Por outro lado, GUSMÃO [8] apresentaram uma Weibull Inversa para análise de taxas de falhas em dados médicos com censuras. Em nosso caso particular, assumiremos a distribuição de Weibull tradicional, e desta feita, o objetivo é avaliar o desempenho do método de otimização não linear, pertencente a família dos métodos iterativos Quasi-Newton, BFGS (Broyden-Fletcher-Goldfarb-Shanno).

Substancialmente, o método BFGS é implementado com o objetivo de melhorar a convergência do algoritmo de Newton a partir da construção de uma matriz hessiana aproximada, da função a ser maximizada, em geral uma função de log-verossimilhança, mediante sucessivas observações e análise de vetores gradientes.

Diante disso, o objetivo do presente trabalho é estudar o comportamento das estimativas de Máxima Verossimilhança, dos parâmetros que indexam a distribuição Weibull, quando se executa o método BFGS disponível na linguagem de programação **Ox**. A versão utilizada para implementar e realizar as simulações foi *Ox Console version 7.20 (Windows/U) (C) J.A. Doornik, 1994-2017*, desenvolvida por DOORNIK [7], disponível em www.doornik.com. Usamos a função *MaxBFGS* integrada com outras rotinas matemática e estatística disponíveis na linguagem.

2 Distribuição de Weibull

Segundo ACTION em [1], a distribuição de Weibull foi proposta originalmente por Waloddi Weibull em 1951 em estudos relacionados ao tempo de falha devido à fadiga de metais. Ela é frequentemente usada para descrever o tempo de vida de produtos industriais, isto é, uma distribuição proposta para análise de confiabilidade. Devido a sua flexibilidade vem sendo amplamente utilizada para modelar dados de sobrevida; usualmente, suas aplicações visam a determinação da vida média e da taxa de falhas em função do tempo da população observada.

Nos últimos anos vêm-se se utilizando diversas variações para função densidade da distribuição de Weibull, a exemplo de PERCONTINI [10], BARROS [2], BARROS [3], dentre outros. Apesar disso, em nosso trabalho tomaremos a função densidade de probabilidade usual definida em REIS [11] como sendo:

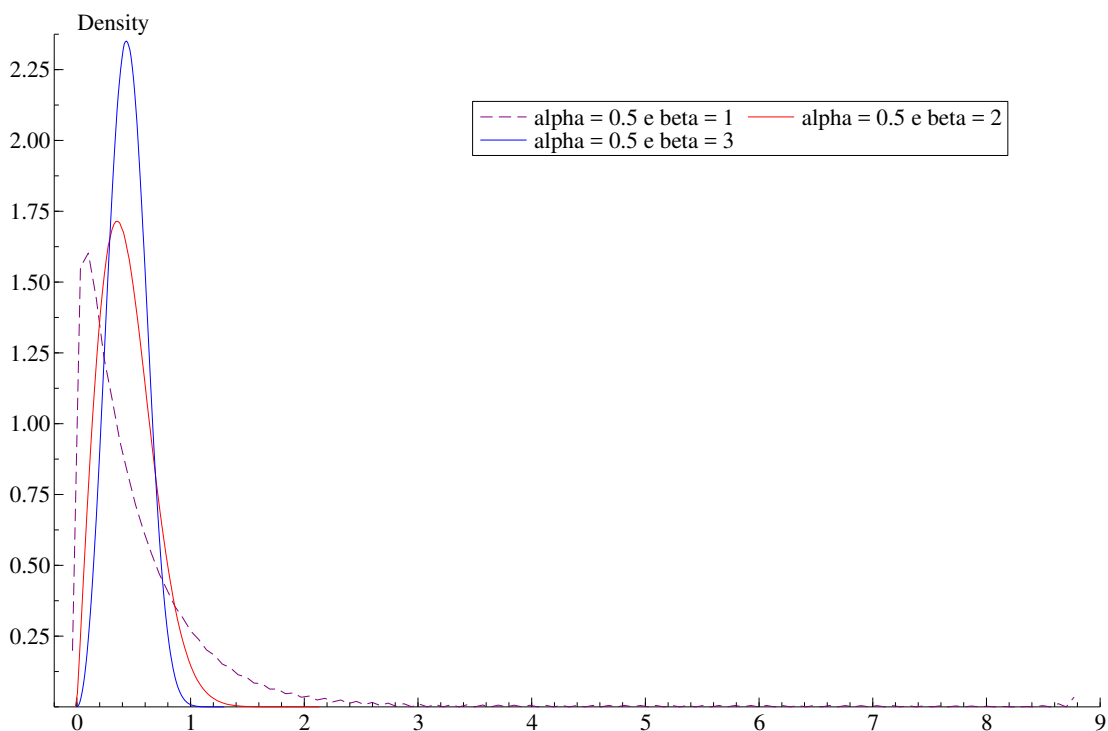
$$f(t) = \frac{\beta}{\alpha^\beta} t^{\beta-1} \exp \left[- \left(\frac{t}{\alpha} \right)^\beta \right], \quad t > 0, \quad (1)$$

onde $\beta > 0$ e $\alpha > 0$ são os parâmetros de forma e escala respectivamente. Contudo, a linguagem de programação **Ox**, a qual é utilizada para obter todos os resultados no presente trabalho, inclusive os gráficos, considera a distribuição padrão de Weibull como sendo da forma da equação:

$$f(t) = abt^{b-1} \exp \left[-at^b \right], \quad t > 0. \quad (2)$$

Dessa maneira, para que os resultados obtidos não sejam discrepantes com relação ao modelo (1) é necessário fazer uma mudança/ajuste no parâmetro a do modelo (2), o qual será utilizado para as simulações. Por exemplo, sem perda de generalidade, se fixarmos em nosso modelo (1) o parâmetro de escala $\alpha = 0.5$, então, no modelo (2) o parâmetro de escala será $a = 2$ para $\beta = 1$, $a = 4$ para $\beta = 2$ e $a = 8$ para $\beta = 3$. Note que, para $\beta = 1$ os modelos (1) e (2) reduzem-se a densidade da exponencial com parâmetros α e a , respectivamente. A figura 1 mostra algumas variações da função densidade (1).

Figura 1 – Variações da função densidade da distribuição de Weibull.



Fonte: Elaboração do autor.

O código utilizado para gerar a figura 1 na linguagem *Ox Console version 7.20 (Windows/U)* (C) J.A. Doornik, 1994-2017, está descrito abaixo:

```

#include<oxstd.h>// Biblioteca padrão
#include<oxdraw.h>    // rotinas de gráficos
#include<oxprob.h>    // rotinas de probabilidade

main(){

// Gerando as amostras
decl amostra1 = ranweibull(2000000000,1,2,1) ;
decl amostra2 = ranweibull(2000000000,1,4,2) ;
decl amostra3 = ranweibull(2000000000,1,8,3) ;

//Construção dos gráficos das estimativas
DrawDensity(0,amostra1', "alpha = 0.5 e beta = 1", TRUE,0,0,0,0,0,8);
DrawDensity(0,amostra2', "alpha = 0.5 e beta = 2", TRUE,0,0,0,0,0,2);
DrawDensity(0,amostra3', "alpha = 0.5 e beta = 3", TRUE,0,0,0,0,0,3);
DrawLegend (0, 400, 100,0);    // determina a localização da Legenda
SaveDrawWindow("weibull.pdf");    // Salva o gráfico em pdf

}

```

A função densidade de probabilidade definida em (1) adequa-se às formas unimodal e monótona decrescente variando-se o parâmetro de forma β , sendo que, para $\beta \leq 1$ a função é monótona decrescente e para $\beta > 1$ a função é unimodal, REIS [11].

A função de distribuição acumulada de Weibull é definida como, REIS [11]:

$$F(t) = 1 - e^{-(t/\alpha)^\beta}, \quad (3)$$

por outra lado, ACTION [1] define sua função quantil como :

$$t_p = \alpha(-\log(1 - p))^{1/\beta}, \quad \text{para } 0 < p < 1. \quad (4)$$

Na seção que segue apresentamos as expressões de verossimilhança e log-verossimilhança para os parâmetros da distribuição de Weibull, a partir de amostras de tamanho n não sujeitas à censura.

3 Estimação de Máxima Verossimilhança

Em conformidade com o dicionário online [9], verossimilhança tem o sentido de qualidade do que parece verdadeiro, do que não contraria a verdade. Dessa forma, sua definição é aplicada ao contexto da inferência estatística para estimar parâmetros que melhor expliquem a amostra observada, isto é, dada uma amostra quais os parâmetros, dentre os possíveis, nos retorna uma distribuição com maior probabilidade de ter gerado a amostra observada. Sem dúvida, neste âmbito existem na literatura diversas técnicas dedicadas à estimação de parâmetros, entretanto o método da Máxima Verossimilhança (EMV) é o mais utilizado. Consoante a BARROS [3], os estimadores de Máxima Verossimilhança desfrutam de propriedades desejáveis e podem ser usados na construção de intervalos de confiança e testes de hipóteses. A aproximação normal para estes estimadores em distribuições de grandes amostras é facilmente manipulada, seja analítica ou seja numericamente.

Nesta conjuntura, o conceito de verossimilhança enunciado aqui deriva do apresentado em BOLFARINE [4]. Suponhamos t_1, \dots, t_n , uma amostra aleatória de tamanho n caracterizada pela função densidade definida por (1). Isso posto, a função de verossimilhança para os parâmetros α e β é dada por:

$$L(\theta) = \prod_{i=1}^n f(t_i; \theta), \quad \text{onde } \theta = (\alpha, \beta). \quad (5)$$

Desta feita, para determinarmos a melhor distribuição para a amostra é preciso determinar o valor de θ que maximiza $L(\theta)$, isto é, o valor de θ que maximiza a probabilidade da amostra ocorrer. De acordo com BOLFARINE [4], o valor de θ que maximiza a função de verossimilhança $L(\theta)$, também maximiza $l(\theta)$, onde

$$l(\theta) = \log(L(\theta)), \quad (6)$$

sendo $\log(L(\theta))$ o logaritmo natural da função de verossimilhança e $l(\theta)$ é denominada função de log-verossimilhança. Então, $l(\theta)$ para o modelo descrito em (1) pode ser definida como:

$$l(\theta) = n \log(\beta) - n\beta \log(\alpha) + (\beta - 1) \sum_{i=1}^n \log(t_i) - \frac{1}{\alpha^\beta} \sum_{i=1}^n t_i^\beta \quad (7)$$

Os estimadores EMV de α e β podem ser obtidos determinando o ponto no espaço paramétrico que maximiza a função (7). Para tanto, temos que calcular as primeiras derivadas de (7) e

igualar a zero, isto é, teremos que resolver o sistema:

$$\begin{cases} \frac{\partial l(\theta)}{\partial \alpha} = \frac{\beta}{\alpha} \left(\frac{1}{\alpha^\beta} \sum_{i=1}^n t_i^\beta - n \right) = 0 \\ \frac{\partial l(\theta)}{\partial \beta} = \frac{n}{\beta} - n \log(\alpha) + \sum_{i=1}^n \log(t_i) - \frac{1}{\alpha^\beta} \sum_{i=1}^n t_i^\beta \log(\beta) = 0 \end{cases}$$

O sistema de equações precedente não admite forma fechada sendo necessário a utilização de métodos numéricos para resolvê-lo. Na literatura existem diversos algoritmos para maximização que envolvem uma estimativa inicial $\hat{\theta}_0$ para θ e um processo iterativo que constrói uma sequência de estimadores que convergem para θ . Em nosso caso particular, computaremos as estimativas dos parâmetros pela maximização numérica por meio do algoritmo de otimização não linear quasi-Newton, conhecido como BFGS (Broyden-Fletcher-Goldfarb-Shanno).

O algoritmo BFGS é similar ao método de Newton - Raphson, distinguindo - se apenas pelo fato de utilizar uma sequência de matrizes simétricas e positivas definidas B^k em vez da oposta da hessiana $-H^{-1}$, de tal forma que

$$\lim_{k \rightarrow \infty} B^k = -H^{-1}.$$

Via de regra, utiliza-se a matriz identidade de mesma ordem como matriz inicial B^0 , isso porque uma matriz identidade é sempre positiva definida e simétrica, propagando-se para aproximações B^k positivas definidas e simétricas. Para detalhes sobre este método veja DAI [6].

Conquanto, ao utilizarmos o método de Máxima Verossimilhança para estimar os parâmetros de alguma distribuição, na verdade estamos calculando estimadores pontuais. Por sua vez, também é possível obter intervalos de confiança para os parâmetros. Para tanto, faz-se necessário o uso de propriedades adequadas para grandes amostras dos estimadores. Frente a isso, e considerando a normalidade assintótica dos estimadores de Máxima verossimilhança, obtêm-se os intervalos de confiança para os parâmetros de escala e forma do modelo (1), como

$$IC(\alpha, 1 - p) = \left(\hat{\alpha} - z_{p/2} \sqrt{\text{var}(\hat{\alpha})}, \hat{\alpha} + z_{p/2} \sqrt{\text{var}(\hat{\alpha})} \right) \quad \text{para } 0 < p < 1.$$

$$IC(\beta, 1 - p) = \left(\hat{\beta} - z_{p/2} \sqrt{\text{var}(\hat{\beta})}, \hat{\beta} + z_{p/2} \sqrt{\text{var}(\hat{\beta})} \right) \quad \text{para } 0 < p < 1.$$

onde $z_{p/2}$ representa o $(1 - p)$ -ésimo quantil da distribuição normal, REIS [11].

Na próxima seção expomos de forma substancial o método Monte Carlo e dispomos dos resultados numéricos obtidos via simulações.

4 Simulação via Monte Carlo

Tal como define CORRAR [5], o método de Monte Carlo é um tipo especial de simulação utilizada em modelos envolvendo eventos probabilísticos. À vista disso, essa técnica viabiliza analisar o comportamento de processos que dependem de fatores aleatórios. Desta feita, é incontável a gama de aplicações para os métodos Monte Carlo, sobretudo quando o processo de simulação envolve riscos ou custos muito altos.

Em nosso caso peculiar, estaremos utilizando essa técnica para avaliarmos o desempenho de estimadores via Máxima Verossimilhança (EMV) para distribuição de Weibull apresentada em (1). Para verificarmos a convergência dos estimadores empregaremos o método BFGS disponível na biblioteca da linguagem de programação **Ox**, com a função *MaxBFGS*. Essa função realiza maximização de funções com a possibilidade de escolha entre derivadas analíticas ou numéricas, os critérios de convergência, número máximo de iterações, dentre outras funcionalidades. Os números pseudo-aleatórios foram obtidos com o gerador de *MWC_32* com a semente 1985, sendo geradas amostras com a função *ranweibull* (*const r*, *const c*, *const a*, *const b*) disponível na biblioteca *ox-prob.h*. O código utilizado encontra-se no Apêndice 6.

4.1 Resultados

Na tabela 1 estão apresentados os resultados para estimativas do parâmetro de escala α . Para este cenário fixamos o valor de forma em $\beta = 1.0$, e arbitramos 10000 réplicas Monte Carlo em todas as simulações. Note que quando aumentamos os valores de, α a variância, o quantil e o desvio padrão assintótico estão crescendo, e à medida que a amostra aumenta, a variância vai pra zero; ver tabela 2.

Tabela 1 – Estimativa para variações de α e $\beta = 1.0$ (fixo) com 10000 réplicas $N = 100$.

α	AMOSTRA DE TAMANHO $N = 100$				
	$\hat{\alpha}$	Viés relativo	Variância (α e β)	Erro padrão assintótico (α e β)	IC (α)
0.5	0.501	-0.192058	0.003 ; 0.006	0.051804 ; 0.076854	(0.496; 0.506)
1.5	1.503	-0.192058	0.025 ; 0.006	0.15541 ; 0.076854	(1.454; 1.552)
3.5	3.507	-0.192057	0.136 ; 0.006	0.36263 ; 0.076854	(3.241; 3.772)

Fonte: Dados da pesquisa.

Percebe-se, a partir da tabela 1 que todos os parâmetros estimados estão acima dos valores estipulados como verdadeiros para α , indicando que os parâmetros estão sendo sobre-estimados. Por outro lado, os vieses relativos tendem a diminuir. Por outro lado, quando aumentamos o tamanho da amostra é possível obter os parâmetros estimados, com três casas decimais, idênticos aos valores tidos como verdadeiros, no entanto, seus vieses sofreram pequenos aumentos; ver tabela 2.

Tabela 2 – Estimativa para variações de α e $\beta = 1.0$ (fixo) com 10000 réplicas amostra $N = 250$.

α	AMOSTRA DE TAMANHO $N = 250$				
	$\hat{\alpha}$	Viés relativo	Variância (α e β)	Erro padrão assintótico (α e β)	IC (α)
0.5	0.500	-0.00242124	0.001; 0.002	0.031147; 0.051643	(0.498; 0.502)
1.5	1.500	-0.00242131	0.010; 0.002	0.093441; 0.051643	(1.480; 1.520)
3.5	3.500	-0.00242244	0.055; 0.002	0.21803; 0.051643	(3.392; 3.608)

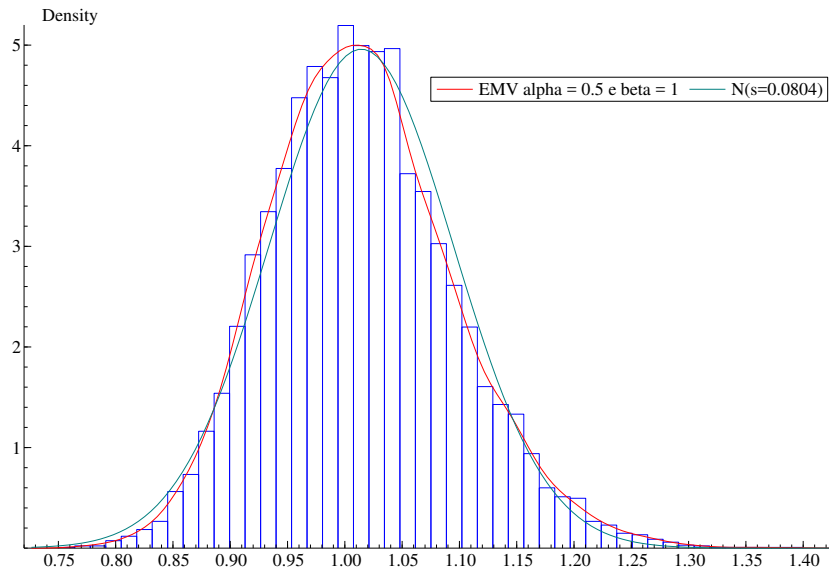
Fonte: Dados da pesquisa.

Para cada estimação dos parâmetros de forma e escala, foram construídos intervalos de confiança (ver coluna 6 das tabelas 1 e 2 para o parâmetro de escala e coluna 6 das tabelas 3 e 4 para o parâmetro de forma), cuja obtenção advém do pressuposto da normalidade assintótica do estimador de Máxima Verossimilhança, sendo calculados a partir da expressão que segue:

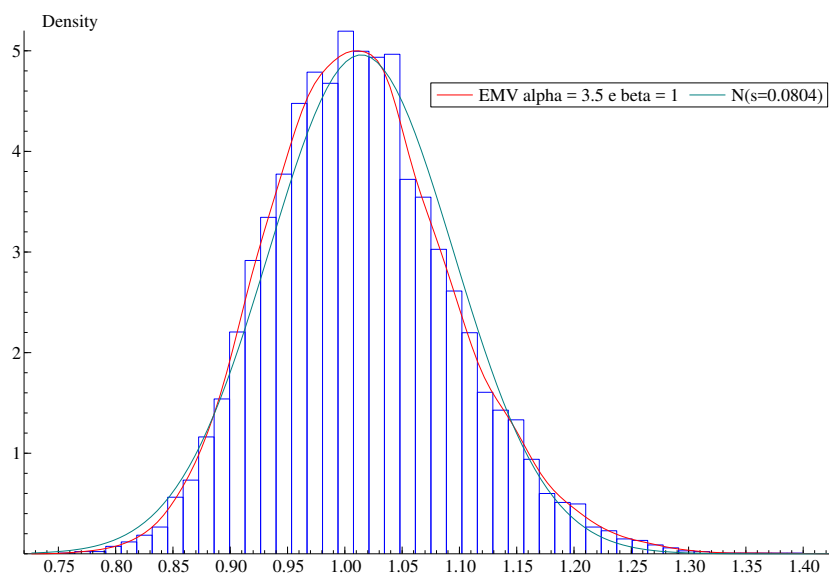
$$\hat{\theta} \pm 1.96\sqrt{\text{var}(\hat{\theta})}, \quad \text{onde } \hat{\theta} = (\hat{\alpha}, \hat{\beta}).$$

A proporção de cada intervalo que contenham o valor do parâmetro verdadeiro caracteriza a probabilidade de cobertura do mesmo. Além dos intervalos de confiança, foram obtidos os valores do erro padrão assintótico tanto para os parâmetro de escala, quanto para o de forma, sendo seus resultados aceitáveis dentro dos cenários propostos.

As figuras 2 e 3 mostram o histograma com o ajuste dos parâmetros e da função de log-verossimilhança da distribuição de Weibull para os casos das linhas 1 e 3 da tabela 1; é perceptível que a curva das estimativas ajustou-se melhor do que a curva da normal ao cenário proposto.

Figura 2 – Histograma, densidade Normal e densidade estimada com $N = 100$ e $\alpha = 0.5$.

Fonte: Elaboração do autor.

Figura 3 – Histograma, densidade Normal e densidade estimada com $N = 100$ e $\alpha = 3.5$.

Fonte: Elaboração do autor.

Agora supomos um cenário em que o parâmetro escala é fixo, enquanto se varia o parâmetro de forma. Os resultados estão nas tabelas 3 e 4. Da mesma forma que nos ensaios com o parâmetro de forma fixo e variando o parâmetro de escala, quando fixamos o parâmetro de escala e variamos o parâmetro de forma, ainda continuamos com as estimativas dos parâmetros sobre-estimadas. Note, ainda, que mesmo quando aumentamos o tamanho da amostra os vieses permanecem quase constantes.

Tabela 3 – Estimativa para variações de β e $\alpha = 2.0$ (fixo) com 10000 réplicas amostra $N = 100$.

β	AMOSTRA DE TAMANHO $N = 100$				
	$\hat{\beta}$	Viés relativo	Variância (α e β)	Erro padrão assintótico (α e β)	IC (β)
0.5	0.507	-1.42966	0.183 ; 0.002	0.39167; 0.038427	(0.504; 0.510)
1.0	1.014	-1.42966	0.044 ; 0.006	0.20722 ; 0.076854	(1.002; 1.027)
1.5	1.521	-1.42966	0.020 ; 0.015	0.14077; 0.11528	(1.493; 1.550)

Fonte: Dados da pesquisa.

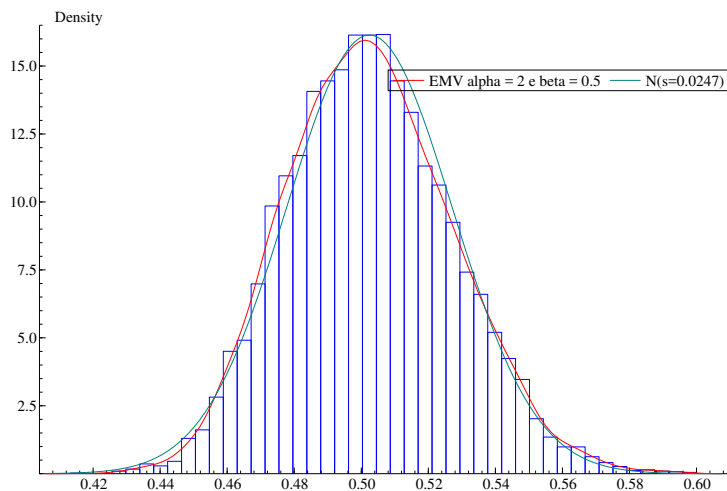
Tabela 4 – Estimativa para variações de β e $\alpha = 2.0$ (fixo) com 10000 réplicas amostra $N = 250$.

β	AMOSTRA DE TAMANHO $N = 250$				
	$\hat{\beta}$	Viés relativo	Variância (α e β)	Erro padrão assintótico (α e β)	IC (β)
0.5	0.502	-0.486099	0.073 ; 0.001	0.24388 ; 0.025821	(0.501; 0.504)
1.0	1.005	-0.486098	0.018 ; 0.002	0.12459 ; 0.051643	(1.000; 1.010)
1.5	1.507	-0.486096	0.008 ; 0.005	0.083656 ; 0.077464	(1.497; 1.518)

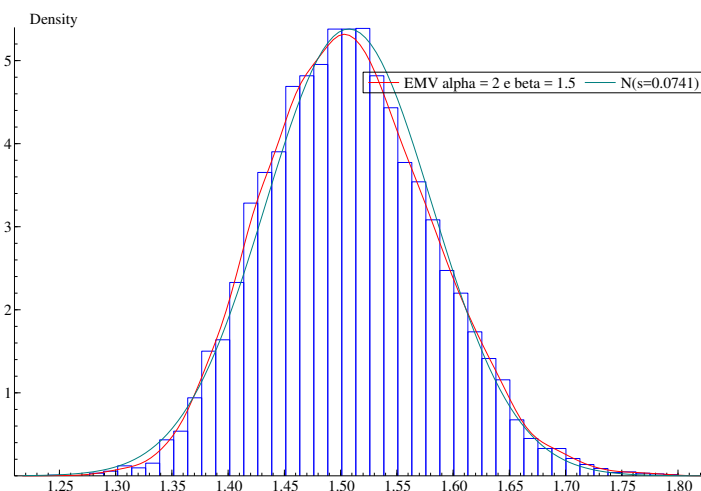
Fonte: Dados da pesquisa.

As Figuras 4 e 5 apresentam a função densidade de probabilidade estimada com o parâmetro β variando e α fixo. Nota-se que a função densidade estimada por Máxima Verossimilhança continua melhor ajustada à amostra com relação à normal quando se varia o parâmetro de forma.

Figura 4 – Histograma, densidade Normal e densidade estimada com $N = 250$ e $\beta = 0.5$.



Fonte: Elaboração do autor.

Figura 5 – Histograma, densidade Normal e densidade estimada com $N = 250$ e $\beta = 1.5$.

Fonte: Elaboração do autor.

5 Considerações finais

Com a análise dos resultados obtidos da distribuição assintótica dos estimadores de Máxima Verossimilhança, foi possível perceber que as estimativas adequaram-se aos dados amostrais. Além do mais, como a proposta do presente trabalho foi a de avaliar o desempenho do método BFGS, quando implementado na linguagem Ox, podemos dizer que alcançamos nosso objetivo, tendo em vista que houve convergência do método proposto, além de resultados, no que diz respeito a viés, variância e erro padrão assintótico aceitáveis em todos os cenários.

Como proposta de trabalhos futuros, é interessante que se compare o desempenho do método em uma outra linguagem de programação, a exemplo de C integrada com a biblioteca GSL, além de as estatísticas da razão de máxima verossimilhança RV para os submodelos da distribuição.

Referências

- [1] PA. **Portal Action**. 2005. Disponível em: <http://www.portalaction.com.br/probabilidades/613-distribuicao-weibull>. Acesso em: 01 jun. 2018.
- [2] BARROS, P. S. N.; BARROS, Kleber N.; BRITO, Cícero C. R.; JUNIOR, Wilson R. O. A distribuição weibull exponencial de zografos. **Proceeding Series of the Brazilian Society of Computational and Applied Mathematics**. v. 3, n. 1, 2015.

- [3] BARROS, P. S. N.; BRITO, Cícero C. R.; JUNIOR, Wilson R. O.; RÊGO, Leandro C.; GOMES-SILVA, Frank S. **Uma nova classe de distribuições weibull generalizada**. 2017.
- [4] BOLFARINE, H.; SANDOVAL, M. C. **Introdução à inferência estatística**. v. 2. Rio de Janeiro: SBM, 2001.
- [5] CORRAR, L. J. O modelo econômico da empresa em condições de incerteza aplicação do método de simulação de monte carlo. **Caderno de Estudos**, SciELO Brasil, n. 8, p. 01 – 11, 1993.
- [6] DAI, Y.-H. Convergence properties of the bfgs algoritm. **SIAM Journal on Optimization**. v. 13, n. 3, p. 693 – 701, 2002.
- [7] DOORNIK, J. **Object-oriented matrix programming using ox 5th edition**. London: Timberlake consultants ltd, 2006.
- [8] GUSMÃO, F. R. S.; JUNIOR, Erinaldo L. S.; SANTOS, Eufrázio de S.; ORTEGA, Edwin M. M. **Distribuição weibull inversa: uma aplicação a dados médicos com censuras**. VIII Encontro Regional de Matemática Aplicada e Computacional, Natal, nov. 2008.
- [9] INFORMAL, D. Dicionário online. **Paciente/Paciência**. v. 18. Disponível em: <<http://www.dicionarioinformal.com.br/>>. Acesso em: 2014.
- [10] PERCONTINI, A; GOMES-SILVA, Frank S.; M. W. A. Ramos; R. Venancio; G. M. Cordeiro. A distribuição gama weibull poisson aplicada a dados de sobrevivência. **TEMA**, São Carlos, v. 15, n. 2, p. 165 – 176, 2014.
- [11] REIS, T. C. S. dos. **Extensões da distribuição weibull aplicadas na análise de séries climatológicas**. Dissertação (Mestrado em Matemática Aplicada e Computacional) – Programa de Pós-Graduação em Matemática Aplicada e Computacional, Universidade Estadual Paulista (UNESP), 2017.

APÊNDICE

6 Código na linguagem Ox para EMV e simulação de Monte Carlo

```

/*****
PROGRAMA DE MAXIMIZAÇÃO E SIMULAÇÃO DE MONTE CARLO
PARA DISTRIBUIÇÃO WEIBULL

AUTOR: MARCELO DOS SANTOS

DATA: JUNHO 2018

*/ *****/

```

```

#include<oxstd.h> // Biblioteca padrão
#include<oxdraw.h> // Biblioteca de gráficos
#include<oxfloat.h> // permite usar pi=3.141516...
#import<maximize> // causa #include<maximize.h>
//e marcação de maximização.oxo para ligação
#include<oxprob.h> // rotinas de probabilidade

decl N = 250; // Numero de observacoes
decl Nrep = 10000; // Numero de replicas

//vetor para armazenar os valores da amostra aleatoria
//da distribuição weibull
static decl vec_weibull;

weibull(const vP,const adFunc, const avScore,const amHessian)

{
decl t,cont;
decl a1=vP[0]; // parametro alpha
decl a2=vP[1]; // paramentro beta

//Matriz linha unitaria de N colunuas assim podemos multiplicar pelo
// vetor da amostra Nx1 que retornará a soma do vetor
decl vone = ones(1,N);

// funcao de log verossimilhanca
adFunc[0]=N*log(a2)-N*a2*log(a1)+(a2-1)*(vone*(log(vec_weibull))) -
(1/(a1^a2))*(vone*(vec_weibull.^a2));

if(avScore){
(avScore[0])[0]=-N*(a2/a1)+a2*(a1^(-(a2+1)))*(vone*(vec_weibull.^a2));
(avScore[0])[1]=(N/a2)-N*log(a1)+vone*(log(vec_weibull)) -
(vone*(vec_weibull.^a2))*(log(a2)/(a1^a2));
}

```

```
if(isnan(adFunc[0])||isdotinf(adFunc[0]))
    return 0;
else
return 1; // indica que houve sucesso
}

main ( )
{

decl ir,i, var1, variancia, vp, dfunc, mhess, vies;

// valores verdadeiro dos parâmetros
decl beta=1, alpha=2.0, a = double(2^(-beta));
decl v=<2.0;1>;
decl cont = 0; // Contador de falhas
decl exectime; // tempo de execucao
decl ep ; // Erro padrao assintotico

// vetor de estimativas. Iniciamos com 0,0,...,0 onde
//os zeros serão substituídos pelas respectivas
decl v_estimativa = zeros(Nrep,2);  estivativas
decl media;
exectime = timer(); // iniciar o tempo de execução

ranseed("MWC_32"); // gerador de numeros aleatorios
ranseed(1985); // Semente

// descomentar para limitar o número máximo de iterações
//MaxControl(20,-1);

// Inicio do Laço de Monte Carlo
```

```
for(i=0; i<Nrep; i++){

// Amostra aleatoria da distribuicao Weibull
vec_weibull = ranweibull(N,1,a,beta) ;

vp=<1.4 ;2.0>; // Chute inicial

// função de maximização via BFGS
ir = MaxBFGS(weibull, &vp, &dfunc, 0, TRUE);

if(ir==MAX_CONV || ir==MAX_WEAK_CONV) // Checando a convergência
{
  v_estimativa[i] []= vp'; // atualiza as estimativas
media = meanc(v_estimativa); // media das estimativas
variancia = varc(v_estimativa); // variancia das estimativas
}

else {
i--;
cont++;
}

} // fim do laço de Monte Carlo

// Valores numericos da matriz hessiana
Num2Derivative(weibull,vp,&mhess);
vies = v' - media;

// Elementos do viés relativo
decl a1=(vies[0]/v'[0])*100, a2=(vies[1]/v'[1])*100;

var1=invertsym((-1)*mhess);
ep = sqrt(diagonal(var1));
decl quantil = double(dropc(media,1)*(-log(0.95))^(1/dropc(media,0)));
```

```
// limite inferior do intervalo de confiança para alpha
decl Linf = double(dropc(media,1) - (1.96)*dropc(variancia,1));

// limite superior do intervalo de confiança para alpha
decl Lsup = double(dropc(media,1) + (1.96)*dropc(variancia,1));

// limite inferior do intervalo de confiança para beta
decl Linfbeta = double(dropc(media,0) - (1.96)*dropc(variancia,0));

// limite superior do intervalo de confiança para beta
decl Lsupbeta = double(dropc(media,0) + (1.96)*dropc(variancia,0));

//Construção do grafio da estimativa

DrawDensity(-1,v_estimativa', "EMV alpha = 2.0 e beta = 1",
TRUE,TRUE,TRUE,0,0,0,3);
DrawLegend (0, 10, 100,0); // Ajuste da legenda
SaveDrawWindow("weibull.pdf"); // Salvando o grafico em pdf

//Impressão de mensagem de convergência
println("CONVERGENCIA: ", MaxConvergenceMsg(ir) );

//imprimir a matriz hessiana numérica
println("Os valores da mhess \n ", mhess);
println("Numero de observações ",N);
println("Numero de replicas ",Nrep);
println("Numero de falhas ", cont);
println("\nParametros Verdadeiros ", "%3.1f |__|", alpha, "%3.1f", beta);
println("\n Media das Estimativas", "%15.3f", media);
println("Vies \n", a1, "|__|", a2);
println("Quantil:\n ",quantil);
println("Limite inferior para o intervalo de confinca de alpha",
"%10.3f \n",Linf);
```



```
println("Limite superior para o intervalo de confinca de alpha",
"%10.3f\n",Lsup);
println("Limite inferior para o intervalo de confinca de beta",
"%10.3f\n",Linfbeta);
println("Limite superior para o intervalo de confinca de beta",
"%10.3f\n",Lsupbeta);
println("\nVariancia das Estimativas", "%15.3f",  variancia);
println("Erro padrao assintótico:  \n",ep);
println("Tempo de execucao: ", timespan(exectime));

// SALVANDO OS DADOS EM UM ARQUIVO TEXTO
decl estimativa=fopen("Estimativas_Weibull.txt","w");
fprintf(estimativa, "ESTIMATIVAS",  v_estimativa, "media", media,
"VARIANCIA", variancia,      "TEMPO DE EXECUCAO", timespan(exectime));

}
```