Simulated Annealing (Recozimento Simulado)

UFMG

10/11/2025

Sumário

Recozimento Simulado - Introdução

Também chamado de método meta-heurístico, este tem como intuito resolver problemas de otimização de grande complexidade. Este algoritmo foi introduzido por volta de 1980 por três pesquisadores, sendo eles: Kirkpatrick, Gelatt e Vecchi. O método traz soluções sub-ótimas, ou seja, eficientes e sem grande esforço computacional.
Tem como objetivo encontrar resultados satisfatórios, ainda que não exatos, sendo que muitas vezes estes resultados são inviáveis e/ou difíceis de serem alcançados.
Se enquadra na classe *Markov chain Monte Carlo* (MCMC) e, por ter caráter estocástico alinhado à sua capacidade de escapar de mínimos locais, este é um ótimo candidato para resolução de problemas complexos de otimização.

Processo de Recozimento

O conceito foi introduzido da ideia de recozimento, onde um sólido é levado para um estado de baixa energia após aumentar sua temperatura.
O processo consiste em duas etapas, sendo elas:

- "Derretimento" da sua estrutura ao levá-lo a uma temperatura muito elevada.
- Resfriamento esquematizado buscando atingir um estado sólido de energia mínima.

Uma vez em estado líquido as partículas do material exposto são distribuídas de forma aleatória.
O estado mínimo de energia só é alcançado com uma temperatura inicial que seja suficientemente alta e um tempo de resfriamento que seja logo o suficiente.

Aplicabilidade

O algoritmo **Simulated Annealing** tem aplicabilidade em várias áreas e aqui serão expostas algumas delas:

- Engenharia de software
- *Machine Learning*
- Teoria de filas
- Processos de manufatura
- logística e transporte com otimização

Vantagens e Desvantagens

Vantagens

- Pode lidar com modelos não lineares complexos, dados caóticos e desordenados, sendo uma técnica robusta no geral;
- É flexível para aproximar do ótimo global;
- O algoritmo é considerado versátil por não se prender às propriedades do modelo;
- O método é facilmente ajustado ("tunado"), ou seja, buscando melhorar seu desempenho.

Desvantagens

- Necessita de muitas escolhas (e parâmetros) para torná-lo um algoritmo real;
- O trabalho para adaptar(tailoring) e ajustar os parâmetros do algoritmo pode ser bastante delicado;
- A precisão dos número usados na implementação do SA pode ter um efeito significativo na qualidade do resultado do método;
- Não é um algoritmo simples e convencional a ser implementado.

Dist. Weibull Tri-Paramétrica

O estudo será centrado em estimar os parâmetros primeiramente de uma distribuição Weibull com a seguinte função densidade:

$$f(x) = \frac{\beta}{\eta}\left(\frac{x-\gamma}{\eta}\right)^{\beta-1} e^{\left(-\frac{x-\gamma}{\eta}\right)^{\beta}};\ \beta > 0,\ \eta > \gamma \geq 0. \tag{1}$$

Tenha ciência também que sua acumulada tem o seguinte resultado:

$$F(x) = 1 - e^{\left(-\frac{x-\gamma}{\eta}\right)^{\beta}},$$

sendo $\beta,\ \eta,\ \gamma$ os parâmetros de forma, escala e locação, respectivamente.

A distribuição Weibull tri-paramétrica é raramente usada devido à dificuldade de se estimar estes parâmetros. Portanto, o uso de SA será feito para tentar estimar estes parâmetros. Aqui o algoritmo será fundamental para maximizar a função de verossimilhança.

Estimação por EMV

Temos ciência que a função de máxima verossimilhança é descrita da seguinte forma:

$$L(x) = \prod_{i=1}^{n} f_{x_i}(x_i\theta)$$

Com isso, a log-verossimilhança é a seguinte:

$$\text{Ln}(L(x_1, \ldots, x_n, \beta, \eta, \gamma)) = n\text{Ln}\left(\frac{\beta}{\eta}\right) + \sum_{i=1}^{n}\left(-\left(\frac{x_i - \gamma}{\eta}\right)^{\beta} + (\beta - 1)\text{Ln}\left(\frac{x_i - \gamma}{\eta}\right)\right). \tag{2}$$

Para encontrar as estimativas dos parâmetros devemos realizar as derivadas parciais e igualar a zero, da seguinte forma:

$$\frac{\partial}{\partial\theta} Ln(L(x)) = 0$$

Fica evidente que esta é uma função bem difícil de derivar, necessitando derivar gradiente ou mesmo aplicar métodos numéricos, portanto prosseguiremos para a ideia geral do algoritmo SA

Algoritmo

A principal vantagem deste método perante os demais é fato de que este, ao se empregar uma busca aleatória, **não se prende a pontos de mínimos locais**. O algoritmo aceita mudanças que aceitam a diminuição e também o aumento da função objetivo f. O seu aumento é aceito com probabilidade: $p = e^{-\frac{\Delta}{T}}$ sendo θ o aumento e T o parâmetro de controle, analogamente conhecido como temperatura do sistema. A sua implementação é simples:

- Uma representação de soluções possíveis,
- Um gerador de mudanças aleatórias nas soluções,
- Um meio de avaliar as funções do problema, e
- Um esquema de resfriamento (annealing schedule) – uma temperatura inicial e regras para diminuí-la à medida que a busca progride.

Por ser um algoritmo meta-heurístico, urge a importância de o mesmo ter a possibilidade de aceitar uma solução pior, evitando que o algoritmo caia em um máximo local e permaneça por lá. Claramente a probabilidade de aceitar uma pior solução diminui na mesma medida que a temperatura cai em cada um dos outros ciclos. A performance do algoritmo portanto é definida dentro de parâmetros de controle, sendo eles:

- A temperatura inicial (T_0) deve ser grande o suficiente para que na primeira iteração do algoritmo,a probabilidade de aceitação de um cenário ruim seja, pelo menos 80%;
- É utilizada a função de redução, sendo esta geométrica comprrendida por: $T_i = CT_{i-1}$, onde $C < 1$, constante e se encontra usualmente entre 0.75 e 0.95;
- O tamanho de cada nível de temperatura determina o número de soluções geradas em uma certa temperatura T;
- O critério de parada define quando o sistema encontrou o nível de energia desejado. Em suma, isso define:
  - O total de números de soluções geradas.
  - A temperatura a cada nível de energia desejado.
  - A razão de aceitação (entre o número de soluções aceitadas e geradas).

Algoritmo SA para Maximização da Verossimilhança

Estimativa de β, η, γ

Passo	Ação / Descrição
1	**Obter Amostra** aleatória (tamanho grande o suficiente).
2	**Determinar Parâmetros de Controle** (T_0, T_f, C, I).
3	**Gerar Solução Inicial** aleatória (a, b, c).
4	**Computar Verossimilhança Inicial** (L) na solução (a, b, c).
	Loop Externo (Resfriamento)
5	**Enquanto** $T > T_f$: $T \leftarrow CT$
	Loop Interno (Equilíbrio)
6	**Para** $i = 1$ a I:
6.1	**Gerar Vizinho:** Gerar a_1, b_1, c_1 vizinhos de a, b, c.
6.2	**Calcular Verossimilhança Vizinha** (L_0) em (a_1, b_1, c_1).
6.3	**Avaliar Parâmetros (Critério de Metropolis):**
6.3.1	**Se** $L_0 > L$ (Melhora): $a \leftarrow a_1, b \leftarrow b_1, c \leftarrow c_1,$ e $L \leftarrow L_0$
6.3.2	**Senão** (Piora): Gerar $u \sim \text{Uni}(0, 1)$.

6.3.2.1	**Se** $u < e^{\frac{(L_0 - L)}{T}}$ (Aceitação Probabilística): $a \leftarrow a_1, b \leftarrow b_1, c \leftarrow c_1$
(Fim do Loop Interno)	
(Fim do Loop Externo)	
7	**Resultado:** Imprimir a, b, c e L.
Nota: a, b, c são estimativas de β, η, γ. F é uma função de custo/energia (ex: $-\ln(L)$).	

Código implementado

```
loglik_weibull3 <- function(params, x) {
  b <- params[1] # beta (forma)
  g <- params[2] # eta (escala)
  c <- params[3] # gamma (locação)

  if (b <= 0 || g <= 0 || any(x <= c)) {
    return(-Inf)
  }

  z <- (x - c) / g
  # Fórmula Log-Verossimilhança
  ll <- length(x) * (log(b) - log(g)) + sum((b - 1) * log(z) - (z^b))
  return(ll)
}
```

```
propose_neighbor <- function(curr, x, sds = c(0.1, 0.1, 0.1)) {
  attempt <- 0
  min_x <- min(x)

  repeat {
    attempt <- attempt + 1

    b1 <- curr[1] + rnorm(1, mean = 0, sd = sds[1])
    g1 <- curr[2] + rnorm(1, mean = 0, sd = sds[2])
    c1 <- curr[3] + rnorm(1, mean = 0, sd = sds[3])

    if (b1 <= 0) {
      b1 <- abs(b1) + 1e-6
    }
    if (g1 <= 0) {
      g1 <- abs(g1) + 1e-6
    }


    if (c1 < min_x - 1e-8) {
      return(c(b1, g1, c1))
    }
```

```
    if (attempt > 50) {
      sds[3] <- sds[3] * 1.5
    }

    if (attempt > 1000) {
      return(c(b1, g1, min_x - 1e-6))
    }
  }
}

# Algoritmo Simulated Annealing para Maximização
sa_weibull3 <- function(
  x,
  init = NULL,
  T0 = 100,
  Tf = 0.001,
  cooling_rate = 0.99,
  L = 5,
  sds = c(0.1, 0.1, 0.1)
) {
  n <- length(x)
```

```r
if (is.null(init)) {

  init <- c(1, sd(x), min(x) - 0.1)
}

curr <- init
curr_ll <- loglik_weibull3(curr, x)
best <- curr
best_ll <- curr_ll
T <- T0

#
while (curr_ll == -Inf) {
  curr <- propose_neighbor(curr, x, sds = sds * 10)
  curr_ll <- loglik_weibull3(curr, x)
  best <- curr
  best_ll <- curr_ll
}

history <- data.frame(eval = 1, ll = curr_ll)
eval_count <- 1
```

```
# Loop SA
while (T > Tf) {
  for (i in 1:L) {
    prop <- propose_neighbor(curr, x, sds = sds)
    prop_ll <- loglik_weibull3(prop, x)

    if (prop_ll > curr_ll) {
      curr <- prop
      curr_ll <- prop_ll
    } else {
      delta_ll <- prop_ll - curr_ll

      if (runif(1) < exp(delta_ll / T)) {
        curr <- prop
        curr_ll <- prop_ll
      }
    }

    if (curr_ll > best_ll) {
      best <- curr
      best_ll <- curr_ll
    }
```

```
      eval_count <- eval_count + 1
      history <- rbind(history, data.frame(eval = eval_count, ll = curr_ll))
    }
    T <- cooling_rate * T
  }

  return(list(
    best_params = best,
    best_ll = best_ll,
    history = history,
    evals = eval_count
  ))
}

# Função de geração SA
run_and_save_example <- function(example_num, real_params) {
  sample_sizes <- c(2500, 1000, 500, 100)
  T0_val <- 100
  Tf_val <- 0.001
  cooling_rate_val <- 0.99
  L_val <- 5

  results_table <- data.frame(
```

```
  "N" = integer(),
  "beta_est" = numeric(),
  "eta_est" = numeric(),
  "gamma_est" = numeric(),
  "LL_real" = numeric(),
  "LL_est" = numeric()
)

for (N in sample_sizes) {

  data_x <- rweibull(N, shape = real_params[1], scale = real_params[2])
  + real_params[3]


  ll_real <- loglik_weibull3(real_params, data_x)

  # Inicialização
  init_params <- c(1.5, sd(data_x), min(data_x) - 0.1)

  sa_result <- sa_weibull3(
    x = data_x,
    init = init_params,
    T0 = T0_val,
```

```
  Tf = Tf_val,
  cooling_rate = cooling_rate_val,
  L = L_val
)

best_p <- sa_result$best_params
best_ll <- sa_result$best_ll

new_row <- data.frame(
  "N" = N,
  "beta_est" = best_p[1],
  "eta_est" = best_p[2],
  "gamma_est" = best_p[3],
  "LL_real" = ll_real,
  "LL_est" = best_ll,

)
results_table <- rbind(results_table, new_row) #{...}
}}
```

Resultados da Estimação por Recozimento Simulado

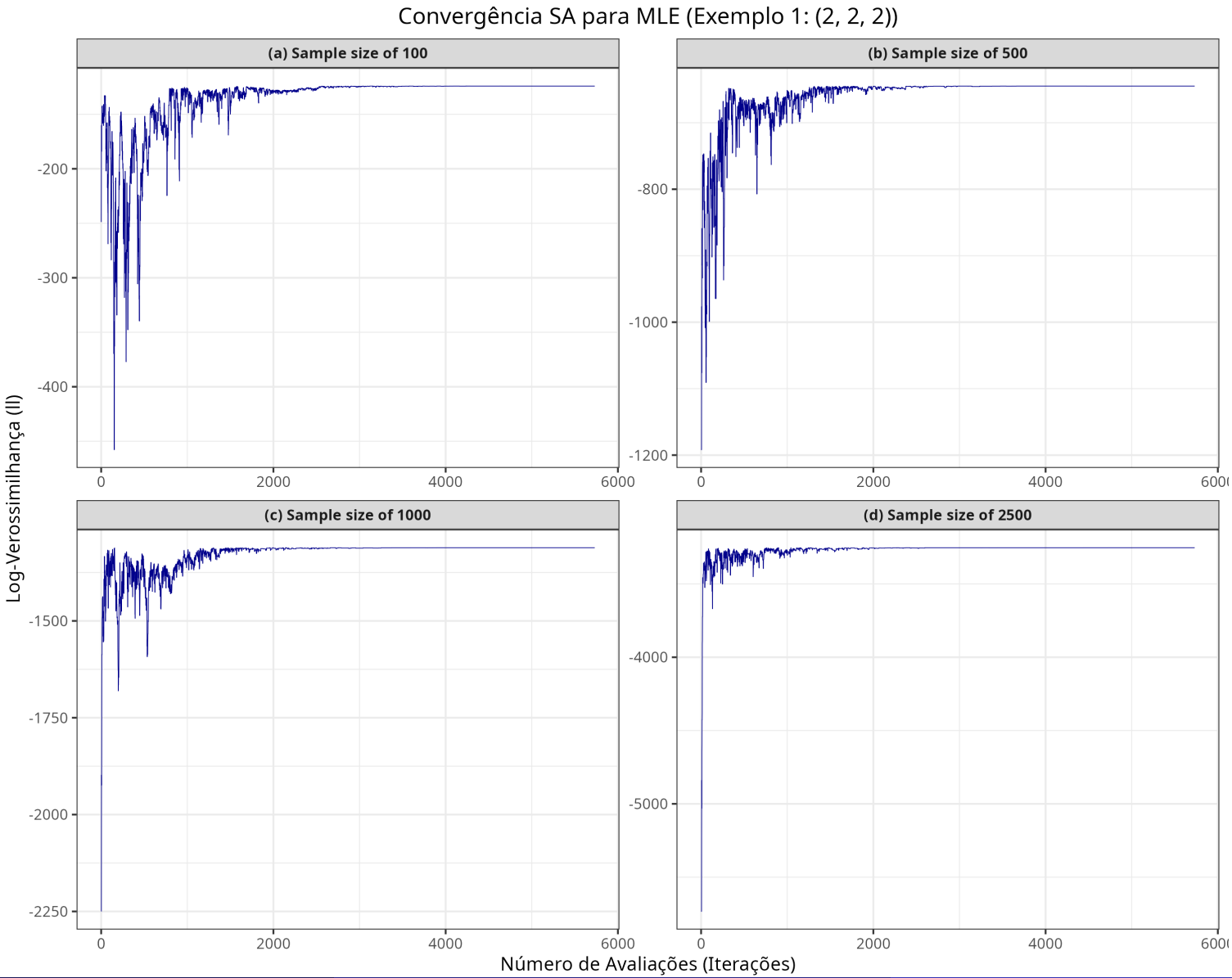
Weibull ($\beta = 2,\ \eta = 2,\ \gamma = 2$)

Weibull parameters			**(2, 2, 2)**	
Measure	**2500**	**1000**	**500**	**100**
Estimated parameters (β, η, γ)	(1.9991, 2.0158, 2.0138)	(1.9777, 2.0351, 1.9797)	(2.1459, 2.0908, 1.9071)	(2.3615, 2.1445, 1.8567)
Likelihood function at the real values	-3254.7346	-1311.5570	-646.9332	-125.4270
Likelihood function at the estimated value	-3252.9872	-1311.0612	-644.8159	-124.0860
Run time (s)	4.7858	2.9836	2.3574	2.1606

Table: Tabela de Resultados para o Exemplo 1: Weibull (2, 2, 2) via Simulated Annealing

Resultados da Estimação por Recozimento Simulado

Weibull ($\beta = 2,\ \eta = 2,\ \gamma = 2$)



Resultados da Estimação por Recozimento Simulado

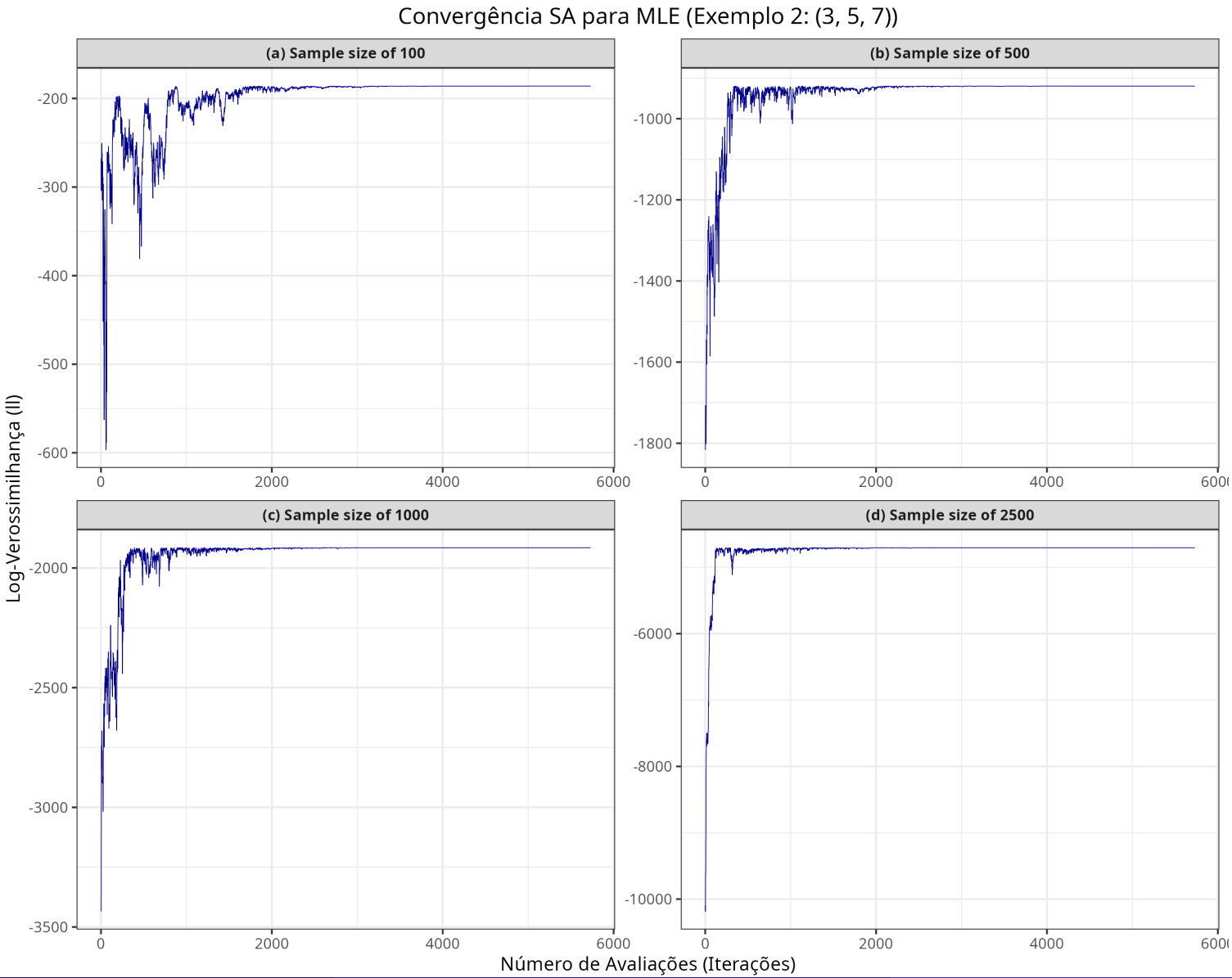
Weibull ($\beta = 3,\ \eta = 5,\ \gamma = 7$)

Weibull parameters			**(3, 5, 7)**	
Measure	**2500**	**1000**	**500**	**100**
Estimated parameters (β, η, γ)	(2.9692, 4.8821, 7.0640)	(2.6933, 4.6690, 7.2710)	(2.8394, 4.5196, 7.3546)	(2.1266, 3.6834, 8.1256)
Likelihood function at the real values	-4705.6693	-1917.4969	-922.3498	-188.0940
Likelihood function at the estimated value	-4704.4097	-1915.2503	-919.5619	-185.9989
Run time (s)	3.5846	2.8512	3.0746	2.8736

Table: Tabela de Resultados para o Exemplo 2: Weibull (3, 5, 7) via Simulated Annealing

Resultados da Estimação por Recozimento Simulado

Weibull ($\beta = 3,\ \eta = 5,\ \gamma = 7$)



Resultados da Estimação por Recozimento Simulado

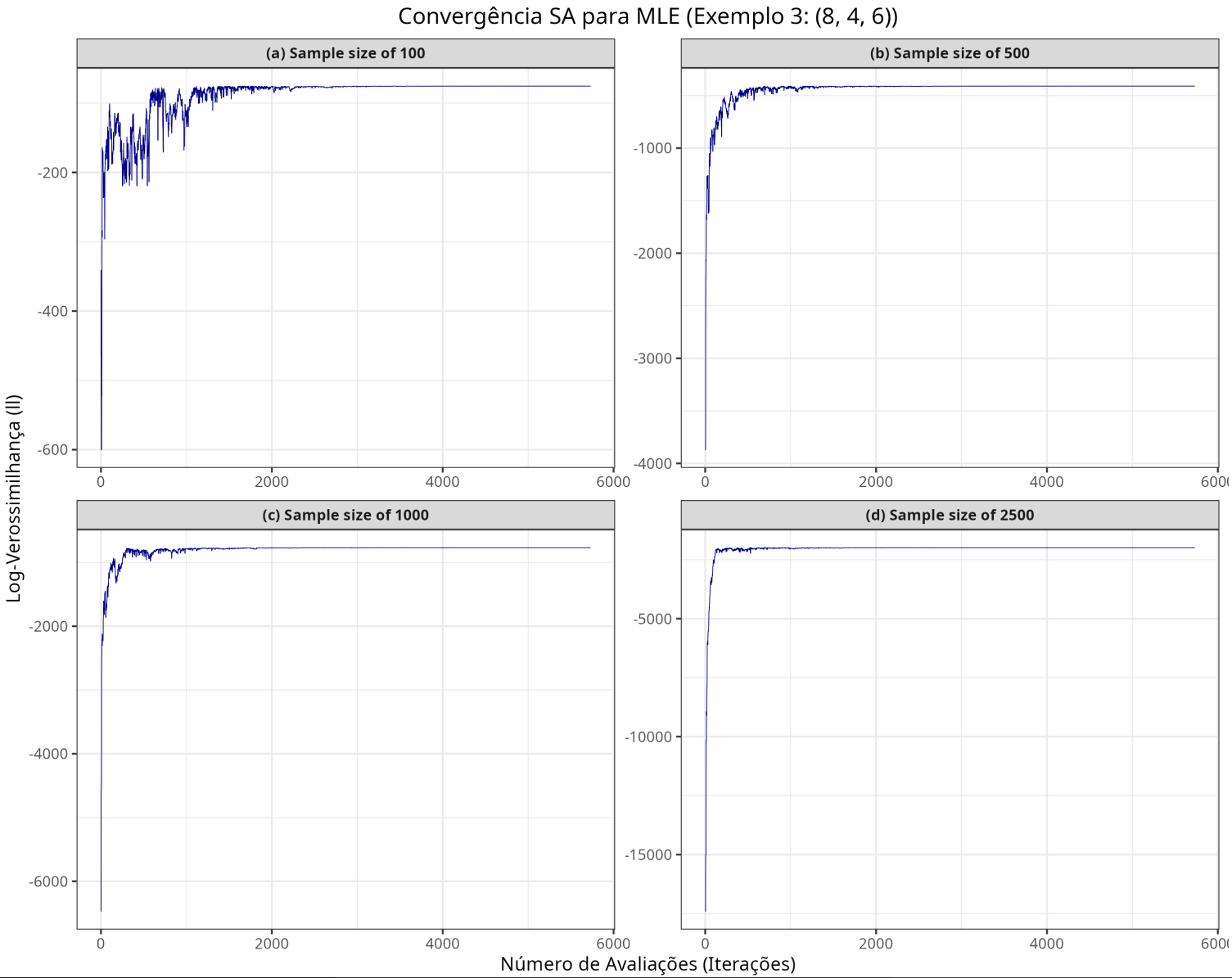
Weibull ($\beta = 8,\ \eta = 4,\ \gamma = 6$)

Weibull parameters			**(8, 4, 6)**	
Measure	**2500**	**1000**	**500**	**100**
Estimated parameters (β, η, γ)	(7.5211, 3.7231, 6.2683)	(8.5488, 4.0806, 5.9406)	(7.6652, 3.8755, 6.1333)	(4.3496, 2.1853, 7.8769)
Likelihood function at the real values	-1993.8436	-770.9186	-409.9350	-78.5454
Likelihood function at the estimated value	-1992.6522	-768.5618	-409.7919	-75.4730
Run time (s)	3.6161	3.0264	2.5860	2.3906

Table: Tabela de Resultados para o Exemplo 3: Weibull (8, 4, 6) via Simulated Annealing

Resultados da Estimação por Recozimento Simulado

Weibull ($\beta = 8,\ \eta = 4,\ \gamma = 6$)



Estudo Monte Carlo

Visando compreender a robustez do modelo, será feito um estudo Monte Carlo, baseando-se na computação da média amostral para um número m de réplicas suficientemente grande da distribuição em questão.
Para uma função monótona $g(\mathbf{X})$, com $\mathbf{X} = (X_1, \ldots, X_n)$ representando os elementos amostrais, seja

$$\mathbf{X}^{(j)} = \{X_1^{(j)}, X_2^{(j)}, \ldots, X_n^{(j)}\}, \quad j = 1, \ldots, m.$$

Sendo $X_i^{(j)}$, $i = 1, \ldots, n$, vetores de v.a. i.i.d de $\mathbf{X}$.
Computamos as respectivas réplicas por:

$$Y_j = g(\mathbf{X}_1^{(j)}, \ldots, \mathbf{X}_n^{(j)}).$$

Então $Y_1, \ldots, Y_m$ são independentes e identicamente distribuídas de $Y = g(\mathbf{X})$.
Considere, portanto:

$$\bar{Y} = \frac{1}{m} \sum_{j=1}^{m} Y_j$$

```
run_example <- function(real_params) {
  sample_sizes <- 2500
  T0_val <- 100
  Tf_val <- 0.001
  cooling_rate_val <- 0.99
  L_val <- 5

  results_table <- data.frame(
    "N" = integer(),
    "beta_est" = numeric(),
    "eta_est" = numeric(),
    "gamma_est" = numeric(),
    "LL_real" = numeric(),
    "LL_est" = numeric()
  )

  for (N in sample_sizes) {
    data_x <- rweibull(N, shape = real_params[1], scale = real_params[2])
    + real_params[3]

    ll_real <- loglik_weibull3(real_params, data_x)
    init_params <- c(1.5, sd(data_x), min(data_x) - 0.1)
```

```
sa_result <- sa_weibull3(
  x = data_x,
  init = init_params,
  T0 = T0_val,
  Tf = Tf_val,
  cooling_rate = cooling_rate_val,
  L = L_val
)
best_p <- sa_result$best_params
best_ll <- sa_result$best_ll

new_row <- data.frame(
  "N" = N,
  "beta_est" = best_p[1],
  "eta_est" = best_p[2],
  "gamma_est" = best_p[3],
  "LL_real" = ll_real,
  "LL_est" = best_ll,
  "Time_s" = run_time
)
results_table <- rbind(results_table, new_row)
return(results_table)
```

```
  }}

estudo_mc <- function(n = 2500, real_params, m = 500) {
  beta_est <- numeric(m)
  eta_est <- numeric(m)
  gamma_est <- numeric(m)

  for(i in 1:m){
    est <- run_example(real_params)

    beta_est[i] <- tail(est$beta, 1)
    eta_est[i] <- tail(est$eta, 1)
    gamma_est[i] <- tail(est$gamma, 1)
  }

  return(data.frame(beta_est, eta_est, gamma_est))
}
av1 <- estudo_mc(real_params = c(2,2,2))
av2 <- estudo_mc(real_params = c(3,5,7))
av3 <- estudo_mc(real_params = c(8,4,6))
```

Resultados da Estimação por SA - MC

Avaliação dos Estimadores - Método de Monte Carlo								
β **Verdadeiro**	β **Médio**	β **DP**	η **Verdadeiro**	η **Médio**	η **DP**	γ **Verdadeiro**	γ **Médio**	γ **DP**
2	1.9997	0.0426	2	1.9908	0.0268	2	2.0075	0.0158
3	3.0038	0.1035	5	5.0028	0.1293	7	7.0001	0.1155
8	7.8918	0.6235	4	3.9459	0.3046	6	6.0540	0.3015

Resultados da Estimação por SA - MC

Weibull ($\beta = 2,\ \eta = 2,\ \gamma = 2$)

Tamanho da Amostra	Parâmetro	Valor Real	Estimativa Média	Desvio Padrão	Viés Relativo
50	β	2	1.967	0.472	-1.648
50	η	2	1.908	0.334	-4.580
50	γ	2	2.074	0.257	3.684
100	β	2	1.955	0.266	-2.274
100	η	2	1.929	0.193	-3.531
100	γ	2	2.058	0.140	2.890
500	β	2	1.982	0.098	-0.883
500	η	2	1.979	0.073	-1.040
500	γ	2	2.017	0.047	0.859
1000	β	2	1.986	0.067	-0.682
1000	η	2	1.986	0.050	-0.722
1000	γ	2	2.011	0.031	0.552

Table: Tabela de Estimações via SA - Weibull(2,2,2)

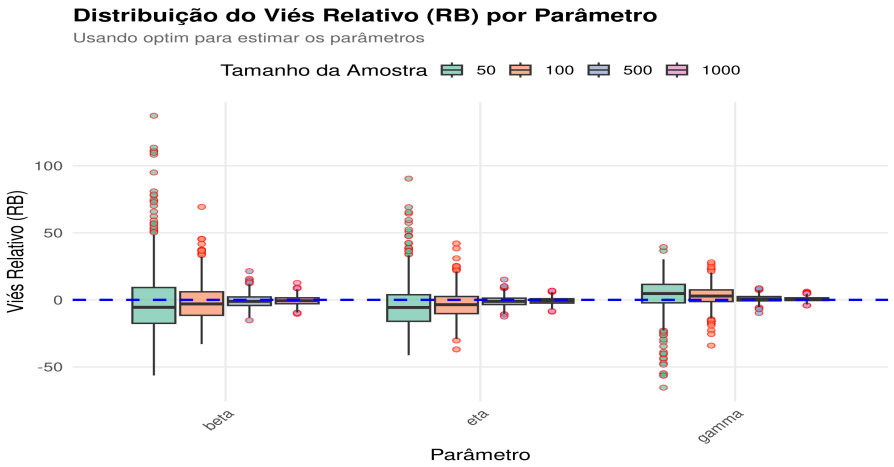
Resultados da Estimação por Optim - MC

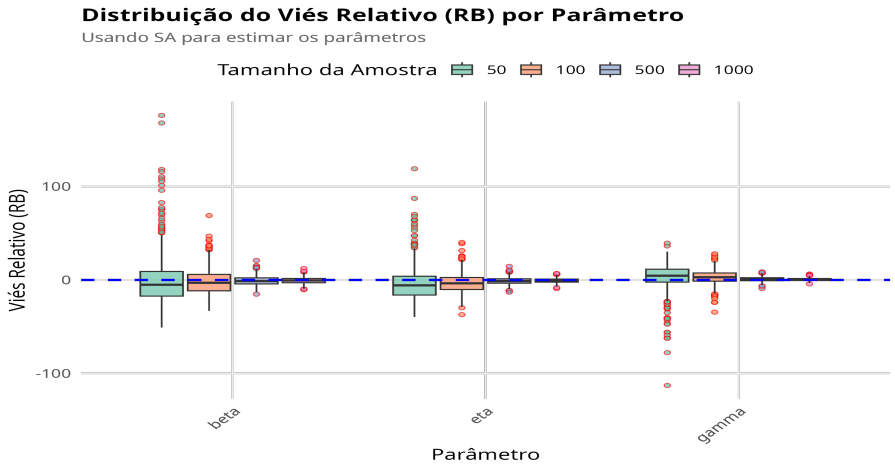
Weibull ($\beta = 2,\ \eta = 2,\ \gamma = 2$)

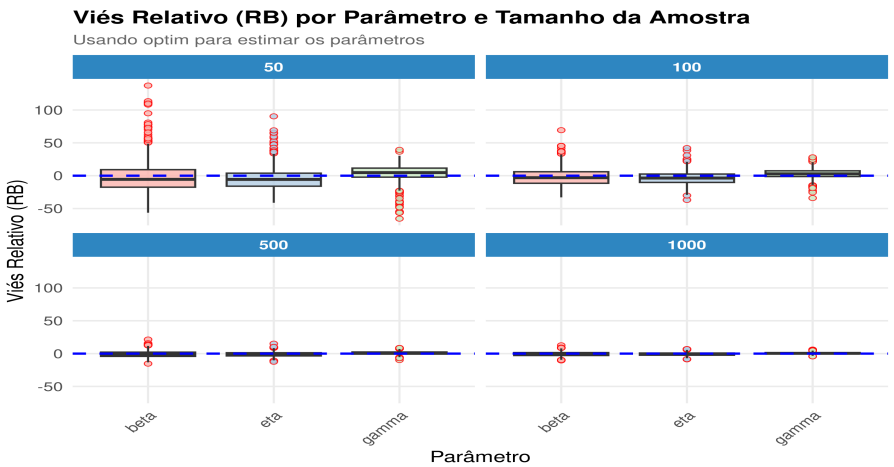
Tamanho da Amostra	Parâmetro	Valor Real	Estimativa Média	Desvio Padrão	Viés Relativo
50	β	2	1.961	0.451	-1.965
50	η	2	1.904	0.320	-4.812
50	γ	2	2.078	0.241	3.892
100	β	2	1.954	0.265	-2.297
100	η	2	1.929	0.193	-3.568
100	γ	2	2.058	0.140	2.915
500	β	2	1.981	0.098	-0.928
500	η	2	1.978	0.073	-1.077
500	γ	2	2.018	0.047	0.887
1000	β	2	1.986	0.066	-0.678
1000	η	2	1.986	0.049	-0.720
1000	γ	2	2.011	0.030	0.560

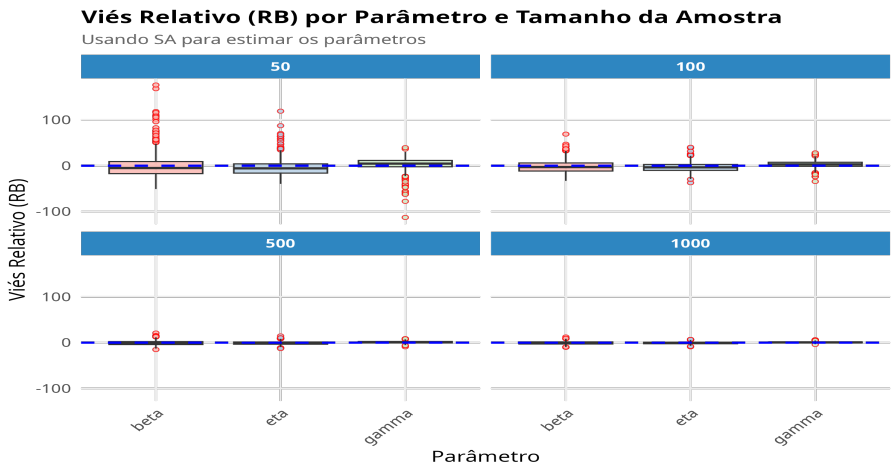
Table: Tabela de Estimações via Optim - Weibull(2,2,2)

Gráficos


Distribuição do Viés Relativo (RB) por Parâmetro
Usando optim para estimar os parâmetros
Tamanho da Amostra 50 100 500 1000
Viés Relativo (RB)
100
50
0
-50
beta
eta
gamma
Parâmetro


Distribuição do Viés Relativo (RB) por Parâmetro
Usando SA para estimar os parâmetros
Tamanho da Amostra 50 100 500 1000
Viés Relativo (RB)
100
0
-100
beta
eta
gamma
Parâmetro


Viés Relativo (RB) por Parâmetro e Tamanho da Amostra
Usando optim para estimar os parâmetros
50
100
500
1000
Viés Relativo (RB)
100
50
0
-50
beta
eta
gamma
Parâmetro


Viés Relativo (RB) por Parâmetro e Tamanho da Amostra
Usando SA para estimar os parâmetros
50
100
500
1000
Viés Relativo (RB)
100
0
-100
beta
eta
gamma
Parâmetro

Referências

- ABBasi, B.; EshrAsh Jahromi, A. H.; ARKAT, J.; Hosseinkouchack, M. **Estimating the parameters of Weibull distribution using simulated annealing algorithm**. Applied Mathematics and Computation, v. 183, p. 85-93, 2006.
- ALBERT, C.; KÜNSCH, H. R.; SCHEIDEGGER, A. **A simulated annealing approach to approximate Bayes computations**. Stat Comput, v. 25, n. 6, p. 1217–1232, 2015.
- LEIJOTO, H. D. **Avaliação de Mecanismo Probabilístico de Cooling Schedule para o Algoritmo Simulated Annealing**. Monografia de Graduação, Departamento de Estatística, Universidade Federal de Ouro Preto (UFOP), 2024.
- LUNA, A. H. **Introdução aos Métodos de Monte Carlo Avançados**. Relatório Técnico RTE_01_2019, Departamento de Estatística-UFMG.
- Lee, Sangmin e Seoung Bum Kim: **Parallel simulated annealing with a greedy algorithmfor Bayesian network structure learning**. IEEE Transactions on Knowledge and Data Engineering,2019.
- Shen, Yadi, Yingchao Dong, Xiaoxia Han, Jinde Wu, Kun Xue, Meizhu Jin, GangXie e Xinying Xu: **Prediction model for methanation reaction conditions based on a state transition simulated annealing algorithm optimized extreme learning machine**. International Journal of Hydrogen Energy.
- Spinellis, D, C Papadopoulos e J M Smith: **Large production line optimization using simulated annealing**. International Journal of Production Research, 2000.