

Using Genetic Algorithm to Estimate the Parameters of the Gompertz Distribution Function by Simulation

Salam Jasim Mohammed

Al-Karkh University of Science, Haifa Street, Baghdad, 10001, Iraq

salam.j@kus.edu.iq

Abstract

In this research, the estimation of Gompertz distribution parameters was studied using three methods: the maximum likelihood method, the moment method, and the genetic algorithm, Note the comparison was performed by calculating the mean square error and using language Python to find the results.

Keywords: Gompertz Distribution, likelihood Function, Moments Function, Genetic algorithm

استخدام الخوارزمية الجينية لتقدير معلمات دالة توزيع جومبيرتز عن طريق المحاكاة

م. سلام جاسم محمد

جامعة الكرخ للعلوم، شارع حيفا، بغداد، 10001، العراق

salam.j@kus.edu.iq

المستخلص

في هذا البحث تم التركيز على تقدير معلمات توزيع كومبيرتز باستخدام طريقة الامكان الاعظم وطريقة العزوم والخوارزمية الجينية وتمت المقارنة باستخدام متوسط مربعات الخطأ وباستخدام لغة بايثون لايجاد النتائج

1. Introduction

The Gompertz distribution [4] is a probability distribution that is commonly used in the analysis of life expectancy data. It is named after Benjamin Gompertz who used it to calculate mortality rates and has been applied in reliability studies, life tests, and biomedical studies

“Below are the probability density function (PDF) and cumulative distribution function (CDF) of the given distribution, respectively as follows”:

$$f(x) = \eta \beta e^{\beta x} e^{-\eta(e^{\beta x}-1)} \dots (1), \quad x \geq 0$$

Such:

η : shap parameter

β : scale parameter

$$F(x) = 1 - e^{-\eta(e^{\beta x}-1)} \dots (2), \quad x \geq 0$$

Such:

$\eta > 0$ and $\beta > 0$

1.1 Gompertz distribution

The Gompertz distribution [6] is one of the important distributions in statistical work and is used in several fields including (economics, medicine, engineering, and social sciences), where it can be

used in modeling due to the distribution's ability to represent unequal change with time. This property enables it to be applied in matters that require analysis of non-linear growth or deterioration with time.

2. Estimation methods

2.1 Method of Maximum likelihood (MLE)

In this method [3], [4], [7] we rely on maximizing the maximum possibility function through specific steps that aim to make the function at its highest limit, as follow:

$$L(x; \eta, \beta) = \prod_{i=1}^n f(x; \eta, \beta) \dots (3)$$

$$\log L = " \sum_{i=1}^n \log f(x; \eta, \beta) " \dots (4)$$

$$\log L = " n \log \eta + n \log \beta + \beta \sum_{i=1}^n x_i - \eta \sum_{i=1}^n e^{(\beta x_i - 1)} " \dots (5)$$

Derivative with respect to η :

$$\frac{\partial \log L}{\partial \eta} = \frac{n}{\eta} - \sum_{i=1}^n e^{(\beta x_i - 1)} \dots (6)$$

Setting the first derivative equal to zero gives:

$$\frac{n}{\eta} - \sum_{i=1}^n e^{(\beta x_i - 1)} = 0 \dots (7)$$

Hence,

$$\frac{n}{\eta} = \sum_{i=1}^n e^{(\beta x_i - 1)} \dots (8)$$

Solving for η , we obtain

$$\eta = \frac{n}{\sum_{i=1}^n e^{(\beta x_i - 1)}} \dots (9)$$

Derivative with respect to β :

$$\frac{\partial \log L}{\partial \beta} = " \frac{n}{\beta} + \sum_{i=1}^n x_i - \eta \sum_{i=1}^n x_i e^{(\beta x_i)} " \dots (10)$$

Equation (10) is nonlinear, so we need to use numerical methods to find the coefficients (η and β). We will use the Newton-Raphson algorithm [1] to solve these equations iteratively, Using the (Hessian matrix) [1], [7], we must find the second derivative of the parameters as follows:

$$\frac{\partial^2 \log L}{\partial \eta^2} = -\frac{n}{\eta^2} \dots (11)$$

$$\frac{\partial^2 \log L}{\partial \beta^2} = -\frac{n}{\beta^2} - \eta \sum_{i=1}^n x_i^2 e^{(\beta x_i)} \dots (12)$$

$$\frac{\partial^2 \log L}{\partial \eta \partial \beta} = -\sum_{i=1}^n x_i e^{(\beta x_i)} \dots (13)$$

The Hessian matrix is:

$$H = \begin{bmatrix} -\frac{n}{\eta^2} & -\sum_{i=1}^n x_i e^{(\beta x_i)} \\ -\sum_{i=1}^n x_i e^{(\beta x_i)} & -\frac{n}{\beta^2} - \eta \sum_{i=1}^n x_i^2 e^{(\beta x_i)} \end{bmatrix}$$

The Newton-Raphson update rule for the parameter vector $\theta = [\eta, \beta]^T$ is:

$$\theta^{(t+1)} = \theta^{(t)} - H^{-1}(\theta^{(t)}) \cdot \nabla \log L(\theta^{(t)}) \dots (14)$$

Where:

$$\nabla \log L = \left[\frac{\partial \log L}{\partial \eta}, \frac{\partial \log L}{\partial \beta} \right]^T$$

T is the gradient vector of the log-likelihood function. The superscript T denotes the transpose operator, which converts the row vector $\left[\frac{\partial \log L}{\partial \eta}, \frac{\partial \log L}{\partial \beta} \right]$ into the column vector $\begin{bmatrix} \frac{\partial \log L}{\partial \eta} \\ \frac{\partial \log L}{\partial \beta} \end{bmatrix}$

That maximize the log-likelihood function [5] Instead of maximizing the likelihood directly, we minimize the negative of the log-likelihood function because optimization tools like minimize in

scipy (Programming language (Python)) are designed to minimize functions, the function minimized in the program is the negative of the log-likelihood:

$$-\log L(x; \eta, \beta) = - \sum_{i=1}^n \log f(x; \eta, \beta) \dots (15)$$

$$-\log L(x; \eta, \beta) = -n \log \eta - n \log \beta - \beta \sum_{i=1}^n x_i + \eta \sum_{i=1}^n e^{(\beta x_i - 1)} \dots (16)$$

This is the function that is passed to the optimization algorithm (minimize), which attempts to find the optimal parameter values by minimizing this function.

2.2 Method of Moments (MOM)

To understand how to estimate the parameters η and β for the Gompertz model using moments theoretically [3], [4], [7] let's start by defining the required moments and their application.

Required Moments

We need the first and second moments for the Gompertz distribution:

- First Moment (Mean) given by the relation:

$$E(X) = \frac{1}{\beta} \left[-\ln \left(1 - \frac{\beta}{\eta} \right) \right] \dots (17)$$

This holds for small $\frac{\beta}{\eta} < 1$ for the second moment $E(X^2)$ the computation is even more complex, so we focus on the first moment and sample variance for estimation

▪ Sample Moments

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i : \text{sample mean}$$

$$s^2 = \frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2 : \text{sample variance (not used here due to complexity)}$$

▪ Equation Moments

Set the sample mean equal to the theoretical mean

$$\bar{x} = \frac{1}{\beta} \left[-\ln \left(1 - \frac{\beta}{\eta} \right) \right] \dots (18)$$

To estimate β and η , we need another equation but the second moment is complex instead we use a practical approach.

In the program, we compute.

$$\bar{x} = n p \text{ mean}(\text{data}) \dots (19)$$

$$\hat{\beta} = \frac{1}{\bar{x}} \dots (20)$$

$$\hat{\eta} = \frac{\beta}{1 - e^{-\beta \bar{x}}} \dots (21)$$

2.3 Genetic Algorithm(GA)

(GA) is one of the most important techniques used in the field of evolutionary computing, [5], [6] It can be said that it is an algorithm that simulates the process of natural evolution. This is due to the theory of natural selection invented by Charles Darwin. The algorithm aims to solve complex problems or find the best solutions through specific steps

How the Genetic Algorithm Works:

I. Creating the Initial Population

We start by generating a set of possible solutions called a population and each individual in the population is a solution that we call a possible solution and are referred to as a chromosome. After that, each chromosome is represented as a sequence of genes in the form of binary numbers (0s and 1s), or real numbers.

II. Calculating Fitness

The genetic algorithm has a function called the fitness function, which is used to evaluate the quality of each solution or chromosome, the higher the fitness value, the better the solution.

III. Selection Mechanism

The selection falls on individuals who have high levels fitness and the purpose of that is to be the "parents" of the next generation, knowing that there are several methods of selection, which are:

Roulette Wheel Selection: Each individual is assigned a portion based on fitness, and individuals are selected using a spinning wheel to determine the parents.

“Tournament Selection: A random subset of individuals is chosen, and the fittest among them is selected as a parent”.

Rank Selection: Individuals are ranked by fitness, and the highest-ranked individuals are chosen.

IV. Crossover

After selecting the parents, crossover occurs, where genes from the parents are combined to create new individuals (offspring). This step transfers beneficial traits across generations, improving solution quality. Types of crossover include:

One-Point Crossover: A single point on the gene sequence is chosen, and the segments are swapped between parents.

Two-Point Crossover: Two points are chosen to exchange segments between parents.

Uniform Crossover: Each gene is randomly selected from one of the parents.

V. Mutation

To maintain genetic diversity and avoid local optima, random mutations are introduced to some individuals. Mutation means altering one or more genes randomly in a chromosome, typically at a low rate to prevent major changes in solutions. Types of mutation include:

Bit Flip: A gene's value is flipped from 0 to 1 or vice versa (for binary sequences).

Gaussian Mutation: A random Gaussian-distributed value is added to the gene.

Swap Mutation: Two genes are swapped within the chromosome.

VI. Iteration

The steps of selection, crossover, and mutation are repeated for multiple generations until the algorithm meets the stopping condition. Stopping conditions include reaching a certain number of generations, achieving a target fitness level, or when no significant improvement occurs across generations.

VII. Termination

The algorithm stops when it reaches an optimal solution or meets the stopping condition, which may be a fixed number of generations or a specific fitness level.

The Experimental Aspect

By using simulation, complex problems can be solved that are difficult to work on in the external reality for various reasons, including the absence of data. Some researchers prefer to use it to save time and money, and it is as follows:

I. Samples:

Multiple samples were taken to ensure accuracy and evaluate the stability of the statistical estimates, as follows: (20, 50, 75, 150).

II. Generating random numbers:

The generation of random numbers U_i , which are uniformly distributed, was within a certain period $(0,t)$

$$U_i \sim U(0,1) \quad , i = 1, 2, \dots, n$$

Where:

U_i : denotes the i -th random variable generated from $U(0,1)$

n : represents the sample size considered in the simulation experiment.

III. Transfer data:

Through the last equation (20), we convert the data that follow the uniform distribution $U(0,1)$ into data that follow the required distribution, which is the Gompertz distribution, using distribution function (CDF) as follows:

Hence:

$$u = F(x) = 1 - e^{-\eta(e^{\beta x} - 1)} \dots (22)$$

Then:

$$x = \frac{1}{\beta} \log \left[1 - \frac{\log(1 - u)}{\eta} \right] \dots (23)$$

IV. Comparison Criterion.

There are several formulas for comparison [2], In our research, we relied on the mean square error (MSE) because it is a statistical measure that is characterized by ease and brevity, which facilitates its use and integration into calculations to determine the best model. Its mathematical formula is:

$$MSE = \frac{1}{R} \sum_{i=1}^n (\hat{\theta}_i - \theta_i)^2 \dots (24)$$

Since:

$\hat{\theta}_i$: "estimated model"

R : The number of replications for each experiment, which was equal to 100.

V. Results and Discussion:

The Python program was used to estimate the parameters in the three ways through simulation. For comparison, the mean square error (MSE) mentioned in the equation was taken, Table (1) represents the required results

Note: Below are the abbreviations mentioned in Table (1):

- MLE(N.R): Maximum likelihood (Newton Raphson)
- GA: Genetic Algorithm
- MOM: Method of Moments

VI. The initial values

"The initial values for each estimation method for the simulation in the Python program were specified as follows":

- (Maximum Likelihood Estimation) MLE : (Multiple initial guesses were used to ensure convergence to the maximum of the log-likelihood ,namely :
 $(\eta, \beta) = (0.5, 0.5)$ and $(\eta, \beta) = (0.3, 0.7)$

The best estimate was selected based on the lowest negative log-likelihood value ,using the Hessian matrix to iteratively update the parameters.

- (Method of Moments) MOM :(No initial values were required ,as estimates were computed directly using the sample mean (X) according to the equations $\hat{\beta} = \frac{1}{\bar{x}}$ and $\hat{\eta} = \frac{\beta}{1-e^{-\beta\bar{x}}}$ The condition $\frac{\beta}{\eta} < 1$ was verified to ensure valid estimates.
- (Genetic Algorithm) GA :(A random initial population of 20 candidate solutions for was generated within the range ,([2 ,0.1]) without specifying fixed initial values .Solutions were evaluated using the negative log-likelihood as the fitness function over 100 generation.

Table (1)

Sample size	Methods	$\hat{\eta}$	$\hat{\beta}$	MSE	Best
20	MLE(N.R)	0.6218	0.5758	0.0275	GA
	MOM	0.6265	0.5759	0.0280	
	GA	0.5523	0.5231	0.0062	
50	MLE(N.R)	0.5378	0.5265	0.0052	GA
	MOM	0.5395	0.5266	0.0053	
	GA	0.5264	0.5123	0.0028	
75	MLE(N.R)	0.5243	0.5157	0.0026	GA
	MOM	0.5259	0.5158	0.0027	
	GA	0.5187	0.5086	0.0019	
150	MLE(N.R)	0.5107	0.5061	0.0010	MLE(N.R)
	MOM	0.5122	0.5062	0.0011	
	GA	0.5092	0.5048	0.0012	

3. Conclusions

The results showed that with a sample size of (20, 50, 75), the genetic algorithm method was better as it showed the lowest mean square error, while with a large sample size of (150), the maximum likelihood method was better as it showed the lowest mean square error.

4. Future Work

- I. Future research could explore the application of the Genetic Algorithm method for parameter estimation in small-sample scenarios and compare its performance with the Maximum Likelihood method across a wider range of sample sizes.
- II. The scope of the current study can be extended by incorporating other estimation techniques, such as Particle Swarm Optimization, Differential Evolution, and Artificial Neural Networks, and evaluating their effectiveness using different statistical criteria.

References

1. Endre Suli and David F. Mayer (2003) . *An Introduction to Numerical Analysis*. Cambridge University Press
2. George Casella and Roger L. Berger (2002) *Statistical Inference*, University of Florida.
3. John A. Rice, (2007), *Mathematical Statistics Data Analysis* , University of California, Berkeley
4. Kyeongjun Lee and Jung-In Seo (2020). *Different Approaches to Estimation of the Gompertz*, *Journal of Probability and Statistics* Volume 2020, Article ID 3541946.
5. Mitchell, M. (1998). *An Introduction To Genetic Algorithms*. The United States: Cambridge University Press.
6. Sabah Manfi Rida and Yasser Ibrahim Saleh, (2015). *The use of the genetic algorithm*
7. *in solving the problem of multi-target allocation*, *Journal of the Iraqi Society for Information Technology* Issue 723.
8. Sanku Dey and Fernando Antonio Moala and Devendra Kumar (2018), *Statistical Properties and Different Methods of Estimation of Gompertz Distribution with Application*.