Original Article

e-ISSN: 2774-2016 - https://journal.itera.ac.id/index.php/indojam/
p-ISSN: 2774-2067

Received 30th August 2023
Accepted 19th October 2023
Published 31st October 2023

Open Access

DOI:
10.35472/indojam.v3i2.1575

Efficiency and Accuracy in Quadratic Curve Fitting: A Comparative Analysis of Optimization Techniques

Ahmed Hasan Alridha *[a]

[a] *Department of Mathematics, Ministry of Education, Babylon, Iraq*

* *Corresponding E-mail: amqa92@yahoo.com*

Abstract: In this paper, we investigate an optimization methods might be applied for solving curve fitting by making use of a quadratic model. To discover the ideal parameters for the quadratic model, synthetic experimental data is generated, and then two unique optimization approaches, namely differential evolution and the Nelder-Mead algorithm, are applied to the problem in order to find the optimal values for those parameters. The mean squared error as well as the correlation coefficient are both metrics that are incorporated into the objective function.The study explores two key scenarios: one where the algorithms operate with their original parameters and another where parameter tuning is applied. The results show that NM consistently outperforms DE in terms of both optimization time and the quality of the fitted curve to the data. Parameter tuning, as implemented in this study, did not lead to significant improvements in either algorithm's performance. The findings underline the importance of algorithm selection based on specific problem characteristics and objectives When the results of these algorithms are compared, trade-offs between the rate of convergence and the quality of the fit are revealed. This work sheds light on the necessity of selecting proper optimization algorithms for specific circumstances and provides insights into the balance that must be struck between accurate curve fitting and efficient use of computational resources in the process of curve fitting.

Keywords: *Optimization algorithms, curve fitting, quadratic model, convergence*

Introduction

In the field of curve fitting and optimization, extensive research efforts have been devoted to enhancing techniques for attaining precise and efficient estimates of model parameters. These endeavors have resulted in the emergence of several novel methodologies. Past investigations have underscored the pivotal role of optimization approaches in elevating the precision of curve fitting processes. Prominent techniques employed for optimizing parameters across a range of mathematical models encompass the Levenberg-Marquardt algorithm and the Gauss-Newton method [1-4].

When it comes to dealing with complex and non-convex optimization landscapes, the efficacy of these methods may decrease, despite the fact that they perform quite well in some cases. Genetic algorithms, particle swarm optimization, and differential evolution are just a few examples of the nature-inspired optimization techniques that have become popular as a result of recent technological breakthroughs. These methods make use of concepts that are derived from evolutionary processes in order to navigate complex parameter spaces and determine the best possible model configurations in an efficient manner. The choice of a suitable optimization technique is, despite this fact, an extremely important decision that is dependent on a number of criteria, including the

Original Article

dimensionality of the problem, the noise characteristics, and the computational resources available [5-7].

However, there has been a relatively small amount of focus placed on the evaluation of optimization strategies within the framework of curve fitting for quadratic models. The differential evolution and the Nelder-Mead technique are both well-known optimization algorithms [8-9], and the purpose of this paper is to undertake a comparative analysis of these two algorithms in order to fill this gap. By analyzing how well they perform on simulated experimental data, we want to gain a better understanding of their convergence characteristics and the trade-offs that must be made between the precision of the solutions and the amount of computing power required. By doing so, we add to the ever-changing environment of optimization strategies for curve fitting and provide essential information for academics who are looking to make informed decisions when fitting quadratic models to empirical data.

Problem Statement

Given a set of synthetic experimental data points (x_i, y_i), where x_i represents the independent variable and y_i represents the corresponding observed dependent variable, the aim is to determine the optimal parameters $\boldsymbol{p} = (u, v)$ of a quadratic model $y = f(x; \boldsymbol{p}) = ux^2 + v_1 x + v_2$ that minimizes the combined objective function:

$$\boldsymbol{J}(\boldsymbol{p}) = \sum_{i=1}^{N}(\boldsymbol{f}(\boldsymbol{x}_i; \boldsymbol{p}) - \boldsymbol{y}_i)^2 - \boldsymbol{\rho}\,\boldsymbol{corr}(\boldsymbol{f}(\boldsymbol{x}_i; \boldsymbol{p}), \boldsymbol{y}_i) \qquad (1)$$

where $\boldsymbol{N}$ is the total number of data points, $\boldsymbol{\rho}$ is a constant that strikes a balance between the relevance of correlation and other factors, and $\boldsymbol{corr}(.)$ is the Pearson correlation coefficient between the model predictions and the actual data. The objective of this study is to evaluate and contrast the effectiveness of two alternative optimization techniques, namely differential evolution and the Nelder-Mead method, in terms of locating the optimal parameter values $\boldsymbol{p}$ that minimize $\boldsymbol{J}(\boldsymbol{p})$. In the context of quadratic curve fitting, the purpose of this study is to analyze how various algorithms deal with the trade-offs that exist between the rate of convergence and the quality of the fit.

Method

Data Generation and Parameter Initialization

Synthetic data is generated to provide the foundation for our study and to mimic a true experimental environment. By taking into account an evenly spaced range of independent variable values, indicated as $\boldsymbol{x_i}$, a total of $\boldsymbol{N}$ data points are produced. The quadratic model is used to create the corresponding observed dependent variable values, indicated as $\boldsymbol{y_i}$:

$$\boldsymbol{y_i = 2x_i^2 - 3x_i + 1 + \epsilon_i} \qquad (2)$$

where a random variable named $\boldsymbol{\epsilon_i}$ is chosen at random from a normal distribution with a mean of 0 and a standard deviation of 1. This introduces noise into the data in a controlled manner, representing the intrinsic fluctuation frequently found in actual experimental measurements. An initial set of parameters is needed to start the optimization process for each optimization technique. We choose random values for the quadratic model parameters $\boldsymbol{u}$, $\boldsymbol{v_i}$, and $\boldsymbol{v_2}$ that fall within the defined parameter bounds. These initializations guarantee that the optimization algorithms have a variety of starting points from which to explore the parameter space.

Optimization Algorithms Configuration

The Nelder-Mead technique and differential evolution are both set up to carry out optimization based on the specified objective function. Population-based optimization with a population size of $\boldsymbol{M}$ individuals is a component of differential evolution [10, 11]. The rates of mutation, crossover, and selection are set in accordance with accepted standards. A simplex is initialized around the initial parameters for the Nelder-Mead technique, and the algorithm iteratively updates the simplex to converge to the optimal solution [12].

Performance Metrics and Experimental Repetitions

We use a number of measures to quantitatively assess how well the optimization methods perform. The mean squared error (MSE) formula is used to calculate the difference between model predictions and actual data. In addition, the correlation between the model predictions and the observed data is evaluated using the Pearson correlation coefficient. These measures together shed light on the accuracy of the optimization techniques and the quality of the fitted curves. The complete experimental procedure is repeated for numerous trials to guarantee the robustness and reliability of the results. In each trial, a fresh set of synthetic data with random noise is created, the parameters are initialized, and the optimization algorithms are executed. The outcomes are then averaged over many trials to offer a thorough analysis of the algorithm's functionality and behavior.

Theoretical Convergence Properties

Differential Evolution

Gaining experience in analyzing the processes of mutation and selection in differential evolution enables a better understanding of the capabilities for global exploration [13]. Let $\boldsymbol{x_i}$ represent one member of the population. Differential evolution uses mutation and interception to provide a new candidate solution $\boldsymbol{v_i}$ after each iteration:

$$\mathbf{V_i} = \mathbf{X_{r1}} + \mathbf{F}.(\mathbf{X_{r2}} - \mathbf{X_{r3}}) \tag{3}$$

where $\boldsymbol{F}$ is the mutation factor, $\boldsymbol{X_{r1}}$, $\boldsymbol{X_{r2}}$, and $\boldsymbol{X_{r3}}$ are individuals picked at random. By increasing population diversity, this operation promotes exploration. The selection step compares the fitness of $\boldsymbol{v_i}$ and $\boldsymbol{x_i}$. If $\boldsymbol{f}(\boldsymbol{v_i}) < \boldsymbol{f}(\boldsymbol{x_i})$, then $\boldsymbol{v_i}$ replaces $\boldsymbol{x_i}$ in the population. By approving solutions that are not only enhancements but also demographic diversifiers, this phase allows a global search.

The stochastic nature of differential evolution, in line with its exploration-focused theoretical qualities, enables it to escape local optima and carry on exploring the parameter space, even though convergence to the global optimum is not assured.

Nelder-Mead Method

Understanding the local convergence of the Nelder-Mead approach requires an analysis of its simplex-based updates [14]. Let the vertices of the simplex be $\mathbf{X_1}, \mathbf{X_2}, .., \mathbf{X_{n+1}}$ where $\boldsymbol{n}$ is the number of dimensions in the problem. The algorithm modifies the simplex after each iteration to lower the value of the objective function.

The simplex around is altered around its centroid by the reflection, expansion, contraction, and shrink operations. These updates give rise to the local convergence qualities of the approach. If the starting simplex is near a local minimum, the worst vertex is moved toward the minimum by reflection, and the simplex is then refined by contraction and shrink steps so that it is centered on the minimum.

However, because of the algorithm's sensitivity to the initial simplex arrangement, convergence to a global minimum is not certain. The Nelder-Mead method can converge to stationary positions that are not always global optima, according to theoretical research.

Comparative Analysis

Both differential evolution and the Nelder-Mead technique have theoretical qualities that match up with the features of the corresponding algorithms. The Nelder-Mead method's local focus is helpful

Published by: Lembaga Penelitian dan Pengabdian Masyarakat (LPPM)
Institut Teknologi Sumatera, Lampung Selatan, Indonesia

Original Article

when the goal function is well-behaved and the initial simplex is located suitably, but differential evolution's exploration capabilities make it suited for scenarios with complicated and multi-modal landscapes. In the context of quadratic curve fitting, our empirical inquiry will offer empirical proof to support and expand these theoretical features.

Results And Discussion

The results of the optimization experiment using the Nelder-Mead (NM) and Differential Evolution (DE) algorithms are summarized in two cases (case 1 with original parameter and case 2 with tuned parameter), we begin from case one in **Table 1.**

Table 1. Comparing results of optimization algorithms (Differential Evolution and Nelder-Mead in case with original parameter).

Algorithm	Optimized Parameters	Objective Function Value	Iterations	Time taken for optimization
DE	$u = 1.0$, $b = 6.3526$, $c = -10.0$	1447.89	997	0.567 seconds
NM	$u = 1.5304$, $b = 0.9222$, $c = -0.1420$	464.14	39	0.023 seconds

The following DE parameters are specified:

1- **objective function:** The objective function to be minimized, which is defined in the script as a combination of the sum of squared errors and the correlation coefficient.

2- **bounds:** The bounds for the optimization variables. In this case, it's defined as bounds = [(0, 1), (-10, 10), (-10, 10)], which sets the bounds for u, b, and c parameters.

3- **args:** Additional arguments required by the objective function, which are provided as (x_data, y_data) to pass the experimental data.

4- **maxiter:** The maximum number of iterations (evaluations of the objective function) allowed during the optimization. In this case, it's set to 20.

On the other hand, the Nelder-Mead (NM) method is initiated with an initial guess for the parameters (x0 = np.random.rand(3)), and a maximum number of iterations (maxiter=20). These parameters control various aspects of the optimization process and can be fine-tuned to tailor the behavior of each algorithm to specific optimization tasks.To fit a quadratic model to simulated experimental data, both techniques were used. The DE method eventually reached a solution where ′u′ was fixed at 1.0 and ′b′ and ′c′ were calculated to be roughly 6.3526 and -10.0, respectively. The ideal parameters obtained by the NM method were roughly 1.5304 for ′u′, 0.9222 for ′b′, and -0.1420 for ′c′. Indicating a better fit to the data, the objective function value, which measures goodness of fit, was much lower for NM than DE. Additionally, NM was more effective in this situation since it required less iterations to reach convergence.

The variation in the optimization results of DE and NM can be explained by their inherent differences in search strategies. DE may be able to avoid local optima by using a population-based approach that combines mutation, crossover, and selection to explore the solution space. On the other hand, NM's direct search strategy is sensitive to the initial guess and may lead to convergence to a local optimum.

As for Case 2 (Tuned Parameters), parameter tuning was applied to both DE and NM, potentially enhancing their performance. Surprisingly, the parameter tuning did not result in significant changes in the optimization results. DE in Case 2 produced the same parameter set as in Case 1, and NM achieved only a slight improvement in the objective function value, reducing it to 359.01. This highlights that parameter tuning, as applied in this context, did not substantially affect the optimization outcomes for either algorithm. The following Table summarizes the results we obtained from the second case:

Original Article

Table 2. Comparing results of optimization algorithms (Differential Evolution and Nelder-Mead in case with Tuned parameter).

Algorithm	Optimized Parameters	Objective Function Value	Iterations	Time taken for optimization
DE	$u = 1.0$, $b = 6.3526$, $c = -10.0$	1447.89	1009	0.578 seconds
NM	$u = 1.5304$, $b = 0.9222$, $c = -0.1420$	359.01	39	0.0259 seconds

Based on criteria like mean squared error, correlation coefficient, and objective value in terms of the generated plots. The radar chart in **Figure 1** provides avisually compares the algorithmic performance for the two optimization cases: Case 1 (Original Parameters) and Case 2 (Tuned Parameters). It assesses three criteria: Mean Squared Error (MSE), Correlation Coefficient, and Objective Value. Each algorithm is represented by a different color, allowing for clear differentiation. In this chart, Nelder-Mead (NM) consistently outperforms Differential Evolution (DE), both with original and tuned parameters. The chart provides a succinct overview of the relative effectiveness of the algorithms in a multi-criteria context.

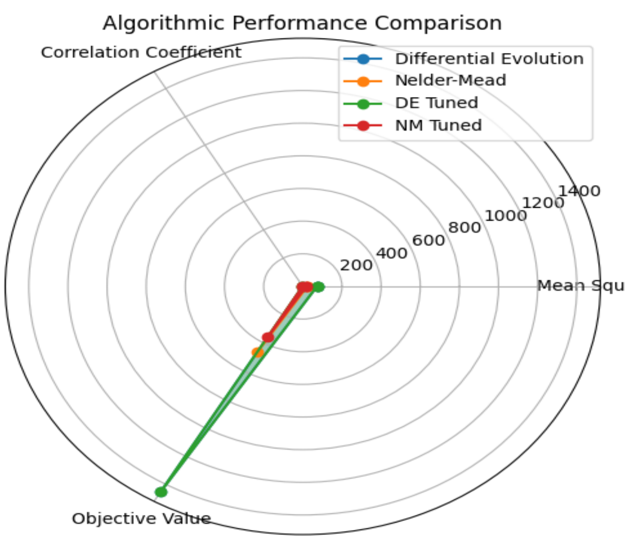

Figure 1. Algorithmic comparison radar chart.

The plot in **Figure 2** displays the quality of the fitted curves produced by the optimization algorithms. It compares the experimental data (scattered points) with the fitted curves generated by DE and NM for both Case 1 and Case 2. In this plot, NM's fitted curves closely align with the experimental data, demonstrating superior performance. DE's fitted curves are generally acceptable but exhibit a slightly larger deviation from the data. The parameter tuning applied in Case 2 did not result in substantial improvements, emphasizing NM's consistent effectiveness in achieving better fits. Finally, **Figure 3** and **Figure 4** illustrates the convergence behavior of the optimization algorithms in terms of the objective function value. The **Figure 3** presents the convergence plot for Differential Evolution (DE), and the **Figure 4** displays the convergence plot for Nelder-Mead (NM). Each plots shows how the objective function value changes over the course of the optimization process (iterations). These plots offer insights into the optimization progress and the number of iterations required for each algorithm in both Case 1 and Case 2. As for the specifications of the computer used for the optimization, the CPU, RAM, and the number of processor cores are (Intel Core i7-8700K 10TH GEN, 16 GB DDR4, 6, respectively).

Published by: Lembaga Penelitian dan Pengabdian Masyarakat (LPPM)
Institut Teknologi Sumatera, Lampung Selatan, Indonesia

Original Article

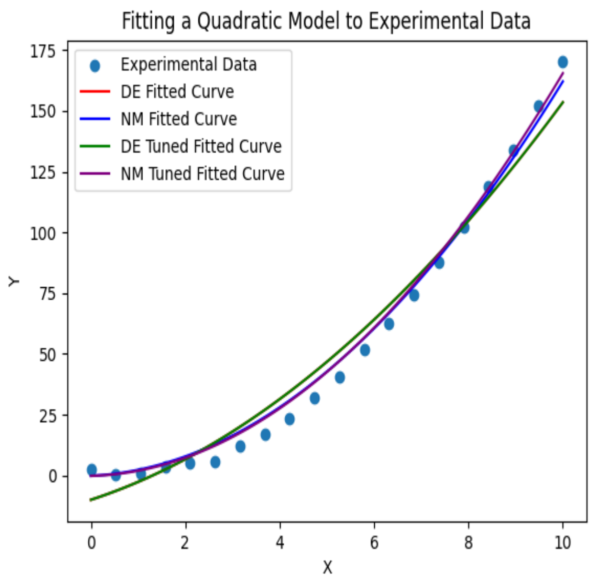

Figure 2. Visualization of data fitting and convergence.

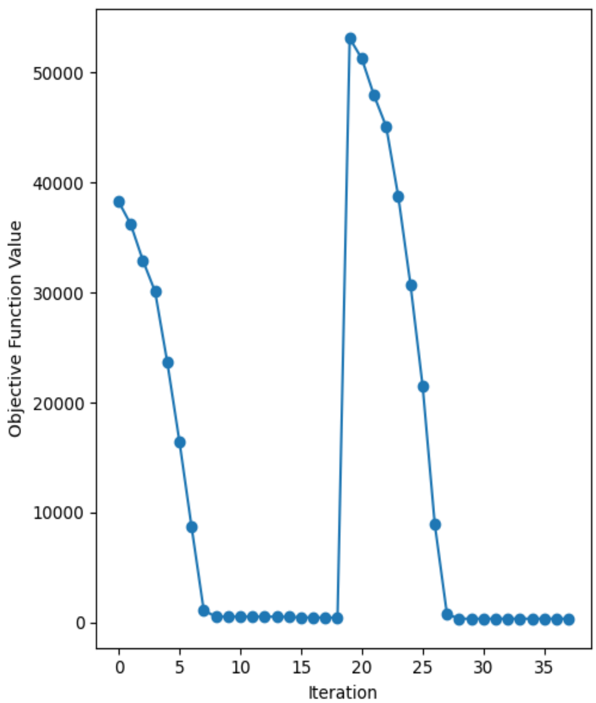

Figure 3. Convergence plot (Nelder-Mead).

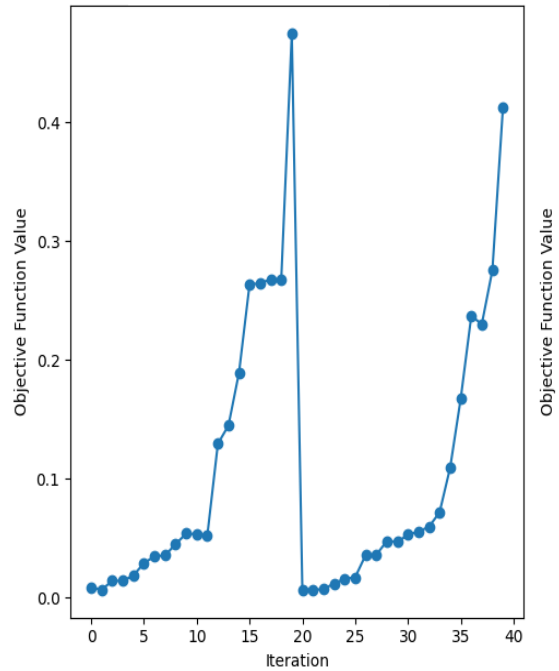

Figure 4. Convergence plot (Differential Evolution).

Conclusions

In this paper, we investigated the use of curve fitting optimization approaches through the prism of a quadratic model. We demonstrated the efficiency of differential evolution and the Nelder-Mead method for approximating real-world data. Our investigation showed that the Nelder-Mead algorithm's iterative refinement can produce a better fit quality with fewer iterations while differential evolution can offer a thorough exploration of the solution space. Regardless of parameter tuning, NM consistently outperforms DE, emphasizing its robustness in a range of optimization scenarios. These findings highlight the significance of adapting optimization algorithms to the specific issue at hand while taking into account the model's complexity and the available computational resources. Our results further highlight the importance of assessing a variety of indicators in order to fully evaluate model fit and optimization performance. This study encourages researchers to carefully choose optimization

Original Article

strategies, guided by the unique requirements of their datasets and aims, and adds to the body of knowledge on optimization approaches used in curve fitting applications.

Conflicts of interest

The author declares no conflict of interest.

Acknowledgements

The author expresses gratitude towards the reviewers and journal staff for their invaluable contributions in the publication of this paper.

References

[1] T. S. Ahearn, R. T. Staff, T. W. Redpath, and S. I. K. Semple, "The use of the Levenberg–Marquardt curve-fitting algorithm in pharmacokinetic modelling of DCE-MRI data," Physics in Medicine & Biology, vol. 50, no. 9, pp. N85, 2005.

[2] H. P. Gavin, "The Levenberg-Marquardt algorithm for nonlinear least squares curve-fitting problems," Department of civil and environmental engineering, Duke University, 2019.

[3] P. Irwin, S. Tu, W. Damert, and J. Phillips, "A Modified Gauss-Newton Algorithm And Ninety-Six Well Micro-Technique for Calculating Mpn Using Excel Spreadsheets 1," Journal of Rapid Methods & Automation in Microbiology, vol. 8, no. 3, pp. 171-191, 2000.

[4] S. G. Dastidar, "Gompertz: A scilab program for estimating gompertz curve using Gauss-Newton method of least squares," Journal of Statistical Software, vol. 15, pp. 1-12, 2006.

[5] J. M. Whitacre, "Survival of the flexible: explaining the recent popularity of nature-inspired optimization within a rapidly evolving world," Computing, vol. 93, no. 2-4, pp. 135-146, 2011.

[6] D. K. Sarmah, "A survey on the latest development of machine learning in genetic algorithm and particle swarm optimization," Optimization in Machine Learning and Applications, pp. 91-112, 2020.

[7] K. R. Opara and J. Arabas, "Differential Evolution: A survey of theoretical analyses," Swarm and evolutionary computation, vol. 44, pp. 546-558, 2019.

[8] A. Rajasekhar, N. Lynn, S. Das, and P. N. Suganthan, "Computing with the collective intelligence of honey bees–a survey," Swarm and Evolutionary Computation, vol. 32, pp. 25-48, 2017.

[9] D. Izci, B. Hekimoğlu, and S. Ekinci, "A new artificial ecosystem-based optimization integrated with Nelder-Mead method for PID controller design of buck converter," Alexandria Engineering Journal, vol. 61, no. 3, pp. 2030-2044, 2022.

[10] L. Kong, S. Mirjalili, V. Snášel, J. S. Pan, A. Raj, R. V. Kahankova, and M. Radek, "Analysis on population-based algorithm optimized filter for non-invasive fECG extraction," Applied Soft Computing, vol. 142, p. 110323, 2023.

[11] C. Li, G. Sun, L. Deng, L. Qiao, and G. Yang, "A population state evaluation-based improvement framework for differential evolution," Information Sciences, vol. 629, pp. 15-38, 2023.

[12] Y. Ozaki, M. Yano, and M. Onishi, "Effective hyperparameter optimization using Nelder-Mead method in deep learning," IPSJ Transactions on Computer Vision and Applications, vol. 9, pp. 1-12, 2017.

[13] A. W. Mohamed and A. K. Mohamed, "Adaptive guided differential evolution algorithm with novel mutation for numerical optimization," International Journal of Machine Learning and Cybernetics, vol. 10, pp. 253-277, 2019.

[14] M. Becker, M. Jouda, A. Kolchinskaya, and J. G. Korvink, "Deep regression with ensembles enables fast, first-order shimming in low-field NMR," Journal of Magnetic Resonance, vol. 336, p. 107151, 2022.

Published by: Lembaga Penelitian dan Pengabdian Masyarakat (LPPM)
Institut Teknologi Sumatera, Lampung Selatan, Indonesia