Derivative-Free Optimization Methods for the Least Absolute Deviation Objective

Egor Sementul
Department of Advanced Computing Sciences
Faculty of Science and Engineering
Maastricht University
Maastricht, The Netherlands

***Abstract*—This thesis evaluates the effectiveness of derivative-free optimization methods for the Least Absolute Deviations (LAD) objective, which is known for being non-smooth and non-differentiable at certain points. The study compares traditional methods like *Nelder-Mead*, *Trust Region*, *Subgradient Line Search*, and *Smoothing with Gradient Descent* with state-of-the-art techniques such as *IRLS*, *Wesolowsky's Direct Descent*, and *Li-Arce's Maximum Likelihood Approach*. Results show that *Nelder-Mead* and *Trust Region* perform well for both linear and nonlinear LAD problems, while state-of-the-art methods excel in linear cases. *Smoothing with Gradient Descent* was effective for linear problems but faced challenges in nonlinear landscapes. The *Subgradient Line Search* method struggled with convergence and accuracy, highlighting the need for further refinement.**

***Index Terms*—nonsmooth optimization, derivative-free optimization, least absolute deviation, robust model fitting.**

I. Introduction

Least Absolute Deviations (LAD) [1] is a method used to fit models by minimizing the sum of absolute deviations from the predicted values. Unlike the Ordinary Least Squares (OLS) [2] approach that minimizes the sum of squared residuals and is highly sensitive to outliers, LAD provides a more robust alternative; In OLS, large residuals exert a disproportionately high influence due to the squaring process, but in LAD, this influence is mitigated, making LAD more suitable in the presence of outliers or non-normal error distributions.

Fitting a model using the LAD objective involves finding the parameters that minimize the absolute differences between observed and predicted values. In this context, the function $f(x_i; a)$ represents the model's prediction for the input x_i, where $\mathbf{a}$ is a vector of parameters. The goal is to adjust the parameters $\mathbf{a}$ of f such that the sum of the absolute deviations between the observed values y_i and the predicted values $f(x_i; a)$ is minimized. The LAD objective function is defined as:

$$\min_{a} \sum_{i=1}^{n} |y_i - f(x_i; a)|$$

Here, f can be any predictive model, ranging from a simple linear function to a more complex non-linear function, depending on the nature of the data and the specific problem being addressed.

The challenge in LAD optimization stems from the non-differentiability of the absolute value function at points where $y_i = f(x_i)$. This non-differentiability makes traditional gradient-based optimization methods unsuitable, as they rely on the smoothness and differentiability of the objective function to guide the search for the optimal parameters. The lack of smoothness in the LAD objective function can lead to convergence issues and suboptimal performance when using standard optimization techniques. This necessitates the use of derivative-free optimization methods, which do not rely on gradient information and can handle non-differentiable objective functions effectively.

The current literature on LAD and derivative-free optimization methods is extensive, highlighting the significance of addressing non-differentiable objective functions. Key contributions in the field include the book ”Introduction to Derivative-Free Optimization” by Conn, Scheinberg, and Vicente [3] which provides an in-depth discussion on the theoretical foundations and practical implementations of derivative-free optimization methods. This book is a useful resource for understanding the various approaches to optimization when derivative information is unavailable or unreliable. The authors categorize these methods into several classes, such as direct search methods, model-based methods, and techniques for handling constraints, and provide insights into their convergence properties and practical applications. In addition to this book, a comprehensive review by Larson, Menickelly, and Wild [4] categorizes derivative-free optimization methods based on objective function properties and optimization features. These methods are particularly relevant for LAD problems, where traditional gradient-based techniques are ineffective.

Several state-of-the-art techniques have been developed for LAD, including:

- **Barrodale-Roberts Algorithm** [5]: This algorithm modifies the simplex method for linear programming to minimize the sum of absolute deviations, efficiently handling both linear and through linearization techniques nonlinear data fitting.
- **Iteratively Re-weighted Least Squares (IRLS)** [6]: IRLS minimizes absolute deviations by solving a se-

This thesis was prepared in partial fulfilment of the requirements for the Degree of Bachelor of Science in Data Science and Artificial Intelligence, Maastricht University. Supervisor(s): [Dr. Pieter Collins, Dr. Barbara Franci]

quence of weighted least squares problems, effectively reducing the influence of outliers. Primarily used for robust regression and sparse signal recovery.

- **Wesolowsky's Direct Descent Method** [7]: This method iteratively solves LAD linear regression by following the steepest descent path, using a weighted median operation to update parameters and reduce outlier influence.
- **Li-Arce's Maximum Likelihood Approach** [8]: This iterative method applies a coordinate transformation and weighted median operation, simplifying the computational process for both simple and multivariate linear models.

These methods have shown effectiveness in various scenarios but are tailored specifically to linear LAD problems and may not generalize well to other forms of model fitting, like the nonlinear models or higher-dimensional spaces. Moreover, the focus of existing literature has been primarily on problem-specific adaptations rather than on the application of general-purpose optimization techniques.

Problem Statement: This thesis aims to assess the viability and effectiveness of traditional derivative-free optimization techniques for the Least Absolute Deviations objective.

Research Questions:

1) Which derivative-free optimization techniques demonstrate the fastest convergence in LAD problems?
2) How do specific derivative-free optimization techniques perform in terms of accuracy and robustness compared to traditional methods for LAD objectives?
3) Does the performance of derivative-free techniques depend on the smoothness at optimum points?
4) What are the limitations of applying these methods to practical, real-world datasets?

II. Methods

Several traditional derivative-free optimization algorithms (DFO) were implemented. Including two direct search methods, one model based method and a smoothing technique followed by a vanilla gradient descent. Three state-of-the-art algorithms were implemented, which are the Wesolowsky's Direct Descent, IRLS and Li-Arce's Maximum Likelihood Approach. In this section we will to discus the chosen DFO algorithms in detail.

A. Direct Search Methods

Direct search methods are a class of optimization algorithms used for the minimization of functions $f : \mathbb{R}^n \to \mathbb{R}$ that do not require gradient information to find the minimum of a function. These methods are particularly useful for problems where the objective function is non-differentiable, discontinuous, or noisy. In this section, we describe the Nelder-Mead algorithm [9] and the Subgradient Line Search method [10], both of which are used to optimize the LAD objective.

1) Nelder-Mead Algorithm: The Nelder-Mead algorithm is a popular derivative-free optimization method that is well-suited for optimizing functions with non-differentiable components. The algorithm operates by maintaining a simplex of $n + 1$ points (where n is the number of dimensions) and iteratively refines this simplex to approach the minimum of the objective function.

a) Hyperparameters of the Nelder-Mead Algorithm:

- **Reflection Coefficient** (α): Determines the distance to reflect the worst vertex across the centroid.
- **Expansion Coefficient** (γ): Used to expand the simplex if the reflection yields a better solution.
- **Contraction Coefficient** (ρ): Determines how much to contract the simplex towards the centroid when the reflection does not yield a better solution.
- **Shrinkage Coefficient** (σ): Used to shrink the simplex towards the best vertex when contraction fails.
- **Step Size** (h): Initial step size used to construct the initial simplex.
- **No Improvement Threshold** (no_imp_thr): Threshold for considering an improvement in the objective function value.
- **No Improvement Break** (max_no_imp): Number of iterations to continue without improvement before terminating the algorithm.
- **Maximum Iterations** (N): Maximum number of iterations to perform before stopping the algorithm.

b) Key Steps of the Nelder-Mead Algorithm:

1) **Initialization**: Construct an initial simplex consisting of $n+1$ vertices. Each vertex represents a set of parameters, and the initial simplex is constructed by taking steps (h) from the initial guess along each dimension. Evaluate the objective function at each vertex to get the initial set of function values.
2) **Sorting and Centroid Calculation**: Sort the vertices of the simplex according to their function values. Calculate the centroid ($\mathbf{x}_o$) of the best n vertices.
3) **Reflection**: Reflect the worst vertex ($\mathbf{x}_w$) through the centroid. This is done to explore the opposite side of the simplex and is computed as:

$$\mathbf{x}_r = \mathbf{x}_o + \alpha(\mathbf{x}_o - \mathbf{x}_w)$$

where $\mathbf{x}_r$ is the reflected point and α is the reflection coefficient. Evaluate the objective function at the reflected point.

4) **Expansion**: If the reflected point is better than the best point, expand the simplex in this direction to potentially find a better solution:

$$\mathbf{x}_e = \mathbf{x}_o + \gamma(\mathbf{x}_r - \mathbf{x}_o)$$

where $\mathbf{x}_e$ is the expanded point and γ is the expansion coefficient. Evaluate the objective function at the expanded point and replace the worst vertex with the better of the reflected or expanded point.

5) **Contraction**: If the reflected point is not better than the second worst point, contract the simplex towards the best point to reduce the search area:

$$\mathbf{x}_c = \mathbf{x}_o + \rho(\mathbf{x}_w - \mathbf{x}_o)$$

where $\mathbf{x}_c$ is the contracted point and ρ is the contraction coefficient. Evaluate the objective function at the contracted point. If the contracted point is better than the worst point, replace the worst vertex with the contracted point.

6) **Shrinkage**: If contraction fails to improve the simplex, shrink the entire simplex towards the best point:

$$\mathbf{x}_i = \mathbf{x}_1 + \sigma(\mathbf{x}_i - \mathbf{x}_1)$$

for $i = 2, \ldots, n+1$, where $\mathbf{x}_1$ is the best vertex and σ is the shrinkage coefficient. Evaluate the objective function at each new point.

7) **Termination**: Check for convergence based on (N) or (max_no_imp).

The Nelder-Mead method does not require the computation of gradients, which is particularly advantageous for functions with absolute values that are not differentiable at all points. The algorithm is inherently robust to noise and discontinuities in the function landscape, making it well-suited to data that may not behave smoothly.

2) Subgradient Line Search: Subgradient methods are a generalization of gradient methods that can be applied to non-differentiable functions. The subgradient of a function at a given point is a vector that generalizes the concept of a gradient for non-differentiable functions. Unlike the gradient, which is unique at differentiable points, a subgradient can have multiple values at non-differentiable points.

A vector $\mathbf{g} \in \mathbb{R}^n$ is a subgradient of $f : \mathbb{R}^n \to \mathbb{R}$ at $\mathbf{x} \in \mathrm{dom}\, f$ if for all $\mathbf{z} \in \mathrm{dom}\, f$,

$$f(\mathbf{z}) \geq f(\mathbf{x}) + \mathbf{g}^\top(\mathbf{z} - \mathbf{x}).$$

If f is convex and differentiable, then its gradient at $\mathbf{x}$ is a subgradient, and is unique. However, a subgradient can exist even when f is not differentiable at $\mathbf{x}$. At this point $\mathbf{x}$, f can have many subgradients, and the set of these subgradients is called the subdifferential of f at $\mathbf{x}$, and is denoted $\partial f(\mathbf{x})$.

In the context of the LAD objective, the subgradient can be understood through the absolute value function. For the function $f(x) = |x|$, the subgradient at any point $x \neq 0$ is $\mathrm{sgn}(x)$, and at $x = 0$, any value in the interval $[-1, 1]$ is a subgradient.

a) Hyperparameters of the Subgradient Line Search Method:

- **Initial Step Size** (η): Controls the initial magnitude of steps taken in the direction of the subgradient.
- **Step Reduction Factor** (β): Factor by which the step size is reduced during backtracking line search.
- **Tolerance** (τ): Threshold for the change in coefficients below which the algorithm will terminate, indicating convergence.
- **Maximum Iterations** (N): Maximum number of iterations to perform before stopping the algorithm.

b) Steps of the Subgradient Line Search Method:

1) **Initialization**: Choose an initial point $\mathbf{x}$ and set η, N, τ, and β.

2) **Subgradient Computation**: Compute the residuals as the difference between observed and predicted values. Compute the subgradient $\mathbf{g}$ based on the sign of the residual and how changes in each parameter affect the residuals, which corresponds to the partial derivatives of the prediction function with respect to each parameter.

3) **Backtracking Line Search**: Update the current point using the subgradient and step size:

$$\mathbf{x}_{\text{new}} = \mathbf{x} - \eta\mathbf{g}.$$

Evaluate the new objective function value. If the new objective function value is smaller, accept the new point. Otherwise, reduce the step size η by the step reduction factor β and repeat.

4) **Convergence Check**: Check if the change in coefficients is below the tolerance τ or if the maximum number of iterations N is reached.

The subgradient line search method can handle the non-differentiability of the LAD objective by using subgradients instead of gradients. The inclusion of a backtracking line search helps ensure that each step made during the optimization process leads to a decrease in the objective function, thereby enhancing the convergence properties of the method.

B. Model Based Methods

Model-based methods use surrogate models to approximate the objective function within a region around the current iterate. In this section, we discuss a trust region framework.

1) Trust Region Method: Trust region methods are a class of iterative optimization algorithms that maintain a model of the objective function within a region around the current estimate. The size of this region, called the trust region, is adjusted dynamically based on the performance of the model in predicting the objective function's behavior.

The trust region method discussed in this thesis follows the approach proposed by Liuzzi, Lucidi, Rinaldi, and Vicente [11] for the derivative-free optimization of non-smooth black-box functions. This method builds a trust-region model that is the sum of a max-linear term with a quadratic one so that the function nonsmoothness can be properly captured, but at the same time the curvature of the function in smooth subdomains is not neglected.

a) Hyperparameters:

- **Initial trust region radius** (Δ_0): The starting size of the trust region.
- **Lower bound for considering an iteration successful** (η_0): The minimum value of ρ_k for an iteration to be considered successful.
- **Upper and lower bounds for adjusting the trust region** (η_1, η_2): Thresholds for shrinking or expanding the trust region.
- **Non-smooth model parameters** $(\eta_0^n, \eta_1^n, \eta_2^n)$: Thresholds specific to the non-smooth model.
- **Trust region adjustment factors** (γ_1, γ_2): Factors by which the trust region is contracted or expanded.

- **Tolerance for the trust region radius** (ϵ): The threshold for stopping based on the trust region radius.
- **Maximum number of iterations** (N): The upper limit on the number of iterations to perform.
- **Exponent parameter for the non-smooth model** (p): Used in the calculation of the non-smooth model.
- **Weighting parameter for the quadratic term** (ω): Used in both the smooth and non-smooth models.

b) Steps of the Advanced DFO-TRNS Algorithm:

1) **Initialization**: Select an initial point x_0, initial trust region radius Δ_0, and set the parameters $\eta_0, \eta_1, \eta_2, \eta_0^n, \eta_1^n, \eta_2^n, \gamma_1, \gamma_2, \epsilon, p, \omega$.
2) **Model Construction**: At each iteration k, construct a model of the objective function within the trust region as follows:
   - Generate the sample point set Y around x_k using coordinate perturbations. Evaluate the objective function at each sample point.
   - Construct the quadratic model by minimizing the Frobenius norm of the difference between the actual function values at points in Y and the values predicted by the quadratic model to estimate the Hessian H and gradient g.
   - Construct the max-linear model:
     - Generate random vectors g_i on the unit sphere. These vectors replace the information given by the subgradients with the one obtained for a set of randomly generated normalized directions.
     - Calculate displacement terms β_k^{ij} for each random vector and sample point:
       $$\beta_k^{ij} = \max \begin{Bmatrix} 0, f(x_k) - f(y_k^j) + (g_i)^\top (y_k^j - x_k) \\ + \delta \|y_k^j - x_k\|^2 \end{Bmatrix}$$
       where, δ = 10^{-5}
     - Aggregate these displacements to form the max-linear term:
       $$\bar{\beta}_k^i = \max_j \{\beta_k^{ij}\}.$$
   - Combine the max-linear and quadratic terms to form the complete nonsmooth approximating mode:
     $$m_k(s) = \max_i \{f(x_k) + g_i^T s - \bar{\beta}_k^i\} + \frac{\omega}{2} s^T H s$$
3) **Trust Region Subproblem**: Solve for s by minimizing the model within the trust region radius:
   $$s_k = \arg\min_s m_k(s) \quad \text{subject to} \quad \|s\| \leq \Delta_k$$
   Any suitable nonlinear constrained optimization algorithm can be used to solve this subproblem.
4) **Update**: Evaluate the objective function at the new point $x_k + s_k$. Calculate the ratio ρ_k of actual to predicted reduction:
   $$\rho_k = \frac{f(x_k) - f(x_k + s_k)}{m_k(0) - m_k(s_k)}$$
   Update Δ_k and x_{k+1} based on ρ_k:
   - If $\rho_k \geq \eta_1$: Successful step, update:
     $$x_{k+1} = x_k + s_k, \quad \Delta_{k+1} = \gamma_2 \Delta_k$$
   - If $\rho_k < \eta_1$: Unsuccessful step, update:
     $$x_{k+1} = x_k, \quad \Delta_{k+1} = \gamma_1 \Delta_k$$
5) **Iteration and Convergence**: Repeat the process until convergence criteria are met, such as the trust region radius being smaller than a threshold, maximum iterations or function evaluations reached.

The Advanced DFO-TRNS method is designed to handle the non-differentiability and complexity of black-box functions. By dynamically adjusting the trust region and utilizing both smooth and non-smooth models, the method ensures robust and efficient optimization even in challenging problem landscapes.

C. Smoothing with Subsequent Gradient-Based Optimization

The chosen approach for smoothing the LAD objective is based on the method proposed by Jimenez-Fernandez [12]. This method transforms the canonical piecewise-linear model into a smooth-piecewise representation.

The smoothing transformation involves approximating the absolute value function using natural logarithmic and exponential functions, resulting in a smooth and differentiable model. The smooth approximation for the absolute value function is given by:

$$abs_\alpha(x) \approx \frac{2}{\alpha} \ln \left(e^{\frac{\alpha x}{2}} + e^{-\frac{\alpha x}{2}} \right)$$

where α is a parameter that controls the smoothness of the approximation. Higher values of α result in a more accurate approximation.

By substituting the absolute value terms in the LAD objective with this smooth approximation, the objective function becomes:

$$\min_a \sum_{i=1}^{n} abs_\alpha(y_i - f(x_i; a))$$

a) Key steps of the algorithm:

1) **Initialization**: Choose an initial point x_0 and parameters such as the smoothing parameter α, learning rate, tolerance level, and maximum number of iterations.
2) **Smoothing Transformation**: Apply the smoothing transformation to the LAD objective function as shown above.
3) **Gradient-Based Optimization**: Use gradient-based optimization techniques, such as gradient descent, on the smoothed objective function. Compute gradients using the smoothed model and update the solution iteratively:
   $$x_{k+1} = x_k - \eta \nabla f(x_k)$$
   where η is the learning rate and $\nabla f(x_k)$ is the gradient of the smoothed objective function at iteration k.
4) **Convergence Check**: Check if the change in the objective function value or the norm of the gradient is below

a specified tolerance level. If the convergence criteria are met, terminate the optimization process.

The smoothing transformation method is suitable for LAD model fitting as it addresses the non-differentiability issue inherent in the LAD objective. By converting the piecewise-linear model into a smooth and differentiable form, it enables the application of gradient-based optimization techniques, which can be more efficient in certain scenarios.

III. Experiments

This section details the experimental setup for evaluating various optimization techniques in the context of LAD model fitting. We analyze two population dynamics models along with their objective functions and noise characteristics. Then, we analyze the smoothness and landscape of these objective functions to understand the challenges they present for optimization. Finally, we establish the evaluation metrics and hyperparameters used for comparing the different optimization algorithms.

A. Problem Setup and Objective Study

Two types of population dynamics models are considered: a linear population dynamics model and a nonlinear logistic growth model [13]. These models are chosen due to their relevance in various scientific and engineering applications.

1) Linear Population Dynamics Model: The linear population dynamics model aims to optimize two variables: the growth rate (r) and a constant term (K). The data used in this model contains additive noise, which can be of two types: Gaussian noise and Laplace noise.

The population dynamics model can be represented as:

$$P_t = P_{t-1} \cdot r + K + \text{noise}$$

where P_t is the population at time t, r is the growth rate, and K is a constant term. The noise is proportional to $\sqrt{P_{t-1}}$. Specifically, the noise term can be described as:

$$\text{noise} \sim \begin{cases} \mathcal{N}(0, b \cdot \sqrt{P_{t-1}}) & \text{(Gaussian noise)} \\ \text{Laplace}(0, b \cdot \sqrt{P_{t-1}}) & \text{(Laplace noise)} \end{cases}$$

where b are standard deviations for the Gaussian and Laplace noise, respectively.

a) Data Generation: To generate the data for this model, simple time series with the following parameters was used: r=0.15, K=10, Number of time steps=100, Gaussian noise with b=1.

b) Objective Landscape and Smoothness Analysis: The objective landscape is characterized by its piecewise linear nature. The analysis of derivatives reveals that while the function is convex, the lack of smoothness can lead to convergence issues with traditional optimization techniques.

Figures (13 in appendix) and 1 illustrate the objective function landscape and its contour for the linear population dynamics model. The global minimum is highlighted in red.

To further analyze the smoothness, we computed the derivatives of the objective function with respect to the growth rate r. The derivatives are plotted for different values of K, both with Gaussian noise and without noise, as shown in Figure 2.

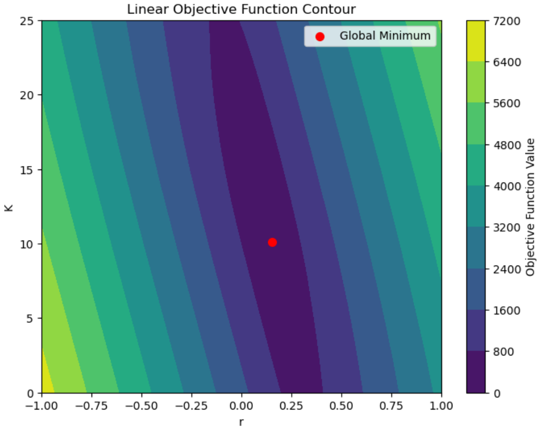

Fig. 1. Linear Objective Function Contour

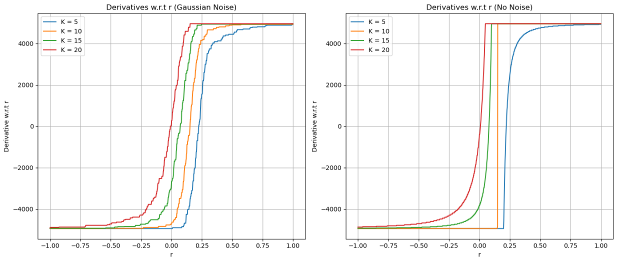

Fig. 2. Derivatives w.r.t r for Gaussian Noise (left) and No Noise (right)

The plots demonstrate the impact of noise on the derivatives of the objective function. The non-differentiability and the effect of noise are more evident in the zoomed-in derivative plot for $K = 10$, as shown in Figure (14 in appendix).

The presence of Gaussian noise causes the derivative to fluctuate widely. These fluctuations are due to the random variability introduced by the noise, which affects the residuals and consequently the derivative of the objective function. The noise-free case shows a piecewise linear behavior with an abrupt change, reflecting the inherent non-smoothness of the least absolute deviations objective. The sharp transitions and discontinuities in the derivative indicate points where the residuals change sign.

2) Nonlinear Logistic Growth Model: The logistic growth model is a nonlinear model commonly used to describe population dynamics where growth rate decreases as the population reaches its carrying capacity. This model introduces additional complexity compared to the linear model.

The logistic growth model can be represented as:

$$P_t = P_{t-1} + rP_{t-1}\left(1 - \frac{P_{t-1}}{K}\right) \cdot \text{noise}$$

where P_t is the population at time t, r is the intrinsic growth rate, and K is the carrying capacity. The noise factor can be either Gaussian or Laplace, scaled by $\sqrt{P_{t-1}}$, similar to the linear model.

a) Data Generation: To generate the data for this model, simple time series with the following parameters was used: r=0.1, K=500, Number of time steps=100, Initial population=10, Laplace noise with b=0.05.

b) Objective Landscape and Smoothness Analysis: The objective landscape is characterized by its nonlinear nature, which poses additional challenges for optimization methods due to the presence of multiple local minima and non-linear interactions between parameters.

Figures (15 in appendix) and 3 illustrate the objective function landscape and its contour for the logistic growth model. The global minimum is highlighted in red.

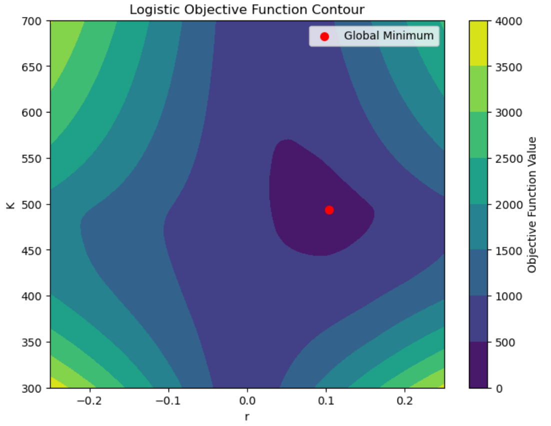

Fig. 3. Logistic Objective Function Contour

To further analyze the smoothness, we computed the derivatives of the objective function with respect to the intrinsic growth rate r and the carrying capacity K. The derivatives are plotted for different values, as shown in Figure 4.

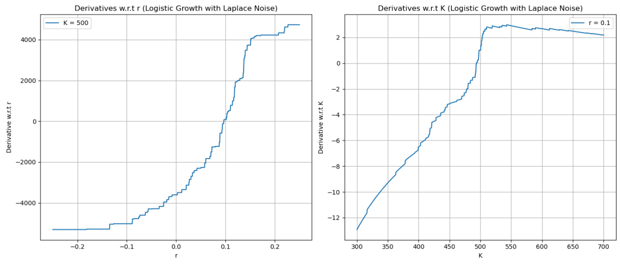

Fig. 4. Derivatives w.r.t r (left) and K (right) for Logistic Growth with Laplace Noise

The derivative plots for the logistic growth model reveal a non-linear landscape with multiple local minima and significant fluctuations due to Laplace noise. The sharp transitions and irregular steps in the derivatives underscore the non-smooth nature of the objective function, similar to the linear case.

B. Testing Methodologies and Scope

All optimization methods were tested on the linear population dynamics model to evaluate their performance in a simpler, more controlled setting. Traditional DFO methods, as well as state-of-the-art algorithms, were compared on this model.

For the nonlinear logistic growth model, classical DFO methods were tested to assess their ability to handle the increased complexity and non-linearity. While the state-of-the-art algorithms are primarily designed for linear cases, they can potentially be adapted for nonlinear models.

C. Evaluation Metrics

The evaluation metrics include Mean Absolute Error (MAE), which measures the average magnitude of errors between predicted and observed values, and Mean Squared Error (MSE), which quantifies the average squared differences, emphasizing larger errors. Median Absolute Deviation (MedAD) is also used for its robustness to outliers, measuring the median of absolute deviations from the data's median. The Breakdown Point (BP) indicates the smallest proportion of incorrect observations that an estimator can handle before it starts to give arbitrarily large results.

Additionally, the Number of Iterations to Converge and the Number of Function/Gradient Evaluations are tracked to judge the efficiency and computational cost of the algorithms. The Final Objective Value, representing the objective function value at the last iteration, helps compare the effectiveness of different algorithms in minimizing the objective function.

D. Hyperparameters Used

The same initial guess was provided for all algorithms: $r = -0.8$, $K = 20$ for the linear case, and $r = -0.2$, $K = 650$ for the nonlinear case.

a) Traditional DFO Algorithms: **Nelder-Mead Algorithm:** Linear and Nonlinear: h $= 0.1$, no_improve_thr $= 1e - 5$, max_no_imp $= 10$, $N = 500$, $\alpha = 1$, $\gamma = 2$, $\rho = 0.5$, $\sigma = 0.5$.

Subgradient Line Search:

- Linear: $\eta = 0.5$, $\beta = 0.2$, $N = 200$, $\tau = 1e - 5$.
- Nonlinear: $\eta = 0.5$, $\beta = 0.5$, $N = 200$, $\tau = 1e - 5$.

Smoothing with Subsequent Gradient Descent: Linear and Nonlinear: $\alpha = 5$, lr $= 0.00001$, $N = 100000$, tol $= 1e - 5$.

Trust Region Method:

- Common Parameters: $\epsilon = 1e - 8$, $\eta_0 = 0.001$, $\eta_1 = 0.25$, $\eta_2 = 0.75$, $\eta_0^n = 1e - 8$, $\eta_1^n = 0.001$, $\eta_2^n = 0.1$, $\theta = 0.001$, p $= 0.1$, $\gamma_1 = 0.1$, $\gamma_2 = 10/9$, $\epsilon = 1e - 6$, $N = 1000$.
- Linear: $\omega = 0.5$, max_f_eval $= 10000$, $\Delta_0 = 1$.
- Nonlinear: $\omega = 1.0$, max_f_eval $= 10000$, $\Delta_0 = 100$.

b) State-of-the-Art Algorithms: IRLS, Wesolowsky's Direct Descent Method and Li-Arce's Maximum Likelihood Approach were used on the linear case and initialized with the same hyperparameters: max_iter $= 100$, tol $= 1e - 5$.

IV. Results

In this section, we present the results of applying derivative-free optimization techniques to the LAD objective. The performance of each method is evaluated on both linear and nonlinear data.

Both objectives were evaluated on a two-dimensional (r and K) grid of 25000 x 25000 points.

For the linear data, the true global minimum was found at $r = 0.1448$, $K = 10.5741$ with an objective value of **326.9738**. The smoothed global minimum for the linear case was at $r = 0.1448$, $K = 10.5741$ with an objective value of 658.0823 ($\alpha = 5.0$).

For the nonlinear data, the true global minimum was found at $r = 0.1017$, $K = 493.0663$ with an objective value of **377.7065**. The smoothed global minimum for the nonlinear case was at $r = 0.0985$, $K = 493.9394$ with an objective value of 763.7800 ($\alpha = 5.0$).

The first question we address is whether the original objectives are differentiable and smooth at and around the global minima. For the linear case, the gradients at $r = 0.1448$, $K = 10.5741$ are $[98.0, 2.0]$, indicating significant step changes rather than smooth transitions (Figure 24 in appendix). For the nonlinear case, the gradients at $r = 0.1017$, $K = 493.0663$ are $[-8.1967, 0.0449]$, similarly showing step changes (Figure 23 in appendix). Both the linear and nonlinear LAD objective functions are not smoothly differentiable at the global minima and around it, as indicated by significant step changes shown in the plots.

Table I summarizes the results obtained.

For the linear data: The **state-of-the-art** techniques (IRLS, Wesolowsky, Li-Arce) performed exceptionally well, achieving objective values around 326.9738 with very few function evaluations and iterations. **Nelder-Mead**: Achieved an objective value of 326.9738, competitive with state-of-the-art techniques but with more function evaluations and iterations. **Trust-region**: Slightly higher objective value (329.7401) with significantly more function evaluations and iterations. **Subgradient line search**: Higher error metrics and an objective value of 482.6902, indicating less precision and a failure to find the global minima. **Smoothing with GD**: Successfully found the global minima but highly inefficient (57800 gradient evaluations, 57799 iterations).

For the nonlinear data: **Trust-region and Nelder-Mead**: Both performed well, with objective values close to the true minimum (377.8158 and 377.7082, respectively) and reasonable function evaluations and iterations. **Subgradient line search**: Struggled significantly, with high error metrics and numerous function evaluations, failed to find the global minima. **Smoothing with GD**: Failed to converge to the global minima, showing very high error metrics and function evaluations.

Detailed Analysis

In this section, we will provide a detailed analysis of each optimization method, discussing its behavior and performance on both linear and nonlinear data models.

A. Nelder-Mead Algorithm

a) Linear Case Analysis: The path taken by the Nelder-Mead algorithm on the linear data is depicted in Figure 5. The algorithm begins at the initial guess and explores the parameter space by reflecting, expanding, and contracting the simplex. The objective function's piecewise linear nature introduces abrupt changes, which the algorithm handles by adjusting the simplex to navigate these discontinuities effectively.

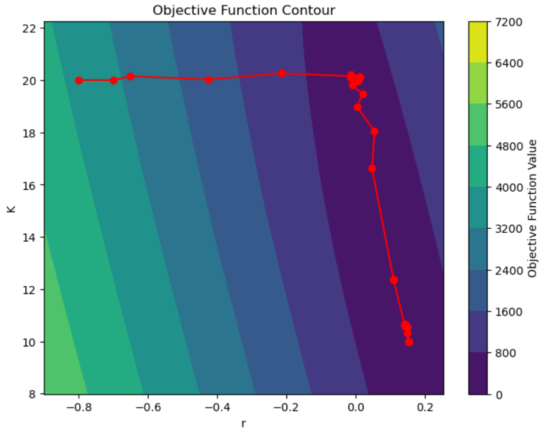

Fig. 5. Path taken by Nelder-Mead on Linear Data

b) Nonlinear Case Analysis: In the nonlinear case, shown in Figure 6, the Nelder-Mead algorithm demonstrates a similar approach. Around the point (0.02, 650), the algorithm appears to spend more time adjusting the parameters, indicating a region where the function continues to change, albeit more gradually. We plotted the plateau regions on the objective contour, to examine weather it is indeed a region of slower change. A plateau region in this case is considered a region where the magnitude of the gradient is smaller then 100. The resulting plot can be found in Figure 21, and indeed this is a region of slower change.

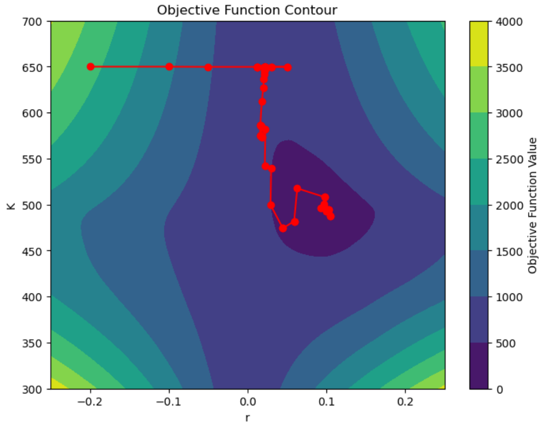

Fig. 6. Path taken by Nelder-Mead on Nonlinear Data

In summary, the Nelder-Mead algorithm's ability to handle non-differentiable and discontinuous objective functions

TABLE I
PERFORMANCE OF OPTIMIZATION METHODS ON LAD OBJECTIVE WITH LINEAR AND NONLINEAR DATA

Method (linear/nonlinear)	**MAE**	**MSE**	**MedAD**	**BP**	**Obj. Val**	**F. Eval**	**(Sub)Grad. Eval**	**Iter**
Trust-region (l)	3.30	17.01	2.64	0.76	329.7401	4087	–	418
Subgradient line search (l)	4.83	33.86	4.48	0.90	482.6902	372	45	45
Nelder-Mead (l)	3.27	16.78	2.80	0.81	**326.9738**	229	–	82
IRLS (l)	3.27	16.78	2.80	0.81	326.9778	10	–	10
Wesolowsky (l)	3.27	16.78	2.80	0.81	**326.9738**	5	–	5
Li-Arce (l)	3.27	16.78	2.80	0.81	**326.9738**	3	–	3
Smoothing with GD (l)	3.27	16.77	2.76	0.80	657.9460	–	57800	57799
Trust-region (n)	8.44	146.24	3.67	0.59	377.8158	805	–	84
Subgradient line search (n)	265.38	98110.47	320.75	0.00	543.8874	945	34	34
Nelder-Mead (n)	8.39	139.02	4.29	0.63	**377.7082**	227	–	80
Smoothing with GD (n)	265.85	98482.27	321.07	0.00	1088.8997	–	100001	–

makes it a good choice for LAD optimization.

B. Subgradient Line Search

a) Linear Case Analysis: The path taken by the Subgradient Line Search method on the linear data is depicted in Figure 7. The algorithm starts at the initial guess and initially moves towards the steepest descent direction. However, the trajectory quickly becomes zigzagged. This behavior is primarily due to the nature of subgradients in handling the absolute value function, which is highly sensitive to small changes in the data.

After examining the subgradients at each iteration, starting with the second iteration we see the same pattern. At one iteration, the subgradient is pointing in the direction [26. -258.] with a step size of 3.2e-04, in the next one, it changes to [32. 2.] with a step size of 0.4e-03. These big changes in subgradients and the relatively small changes in step sizes indicate that the method is responding to local variations in the objective caused by the noise. The zigzag behavior is also influenced by the backtracking line search, which adjusts the step size. If the step size is too large, the method overshoots, and if too small, it makes minimal progress, leading to oscillations and premature convergence.

The zig-zag behaviour of subgradient algorithms is discussed in more detail in [14], to avert the zig-zagging phenomenon and speed up the convergence behavior, the authors introduce a conditional subgradient method.

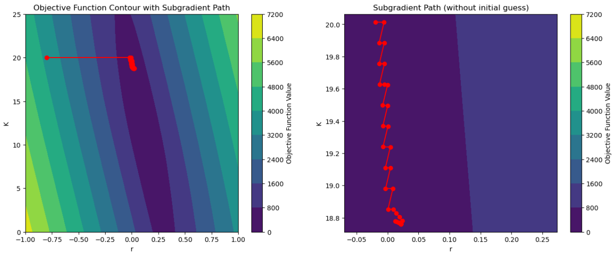

Fig. 7. Path taken by Subgradient Line Search on Linear Data

b) Nonlinear Case Analysis: In the nonlinear case, shown in Figure 8, the Subgradient Line Search method exhibits a similar zig-zag pattern.

In this case we observed a similar pattern in subgradients over iterations. After the first 4 iterations, the algorithm gets trapped in a plateau region. In one iteration the subgradient is [171.1573 -0.3099] with the step size 1.5258e-05, in the next one the subgradient is [-68.4664 -0.2512] with the step size 3.0517e-05. The lack of clear directional guidance due to relatively flat region and a complex landscape as well as backtracking line search mechanism make the method oscillate between two points.

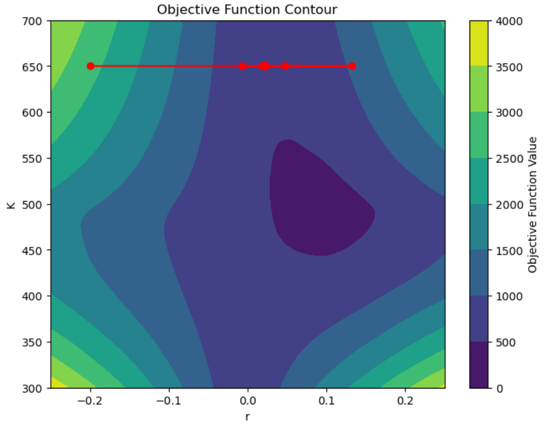

Fig. 8. Path taken by Subgradient Line Search on Nonlinear Data

In summary, in both linear and nonlinear cases, the subgradients lead to frequent directional changes due to local variations in the objective caused by the noise. Lack of clear directional guidance in regions of slower change and backtracking line search further complicates the search by overshooting or taking small inefficient steps. This results in zigzag paths and premature convergence.

C. Trust Region Algorithm

a) Linear Case Analysis: The path taken by the trust-region method on the linear data is depicted in Figure 9. The algorithm starts at the initial guess and iteratively refines the trust region to explore the parameter space. Initially, the algorithm makes larger steps, as shown by the wide movements. As it approaches regions with lower objective values, the steps become smaller, indicating a more refined search. This behavior is consistent with the trust-region approach, where the size of the trust region is dynamically adjusted based on the local landscape of the objective function.

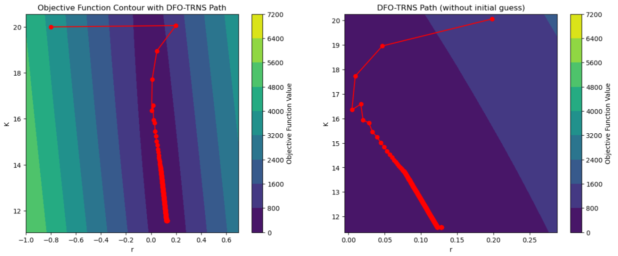

Fig. 9. Path taken by Trust Region Method on Linear Data

b) Nonlinear Case Analysis: In the nonlinear case, illustrated in Figure 10, the trust-region method shows a similar adaptive approach. The initial steps are larger to cover more ground quickly. As the algorithm detects more promising regions, the steps become progressively smaller. The path shows how the algorithm avoids getting trapped in local minima by dynamically adjusting the trust region, allowing it to navigate the complex landscape of the nonlinear objective function effectively.

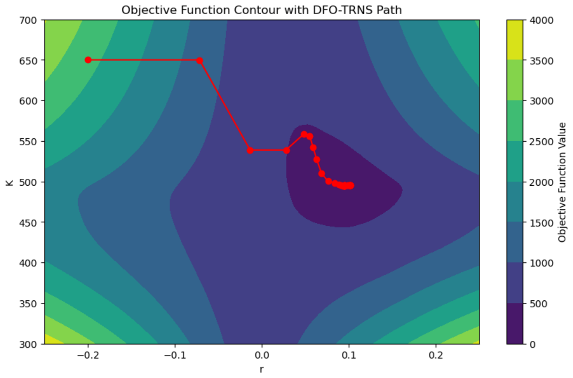

Fig. 10. Path taken by Trust Region Method on Nonlinear Data

In summary, the trust-region method's ability to adjust the search region dynamically makes it a robust choice for optimizing LAD objectives.

D. Smoothing with Gradient Descent

a) Smoothing Effect: The smoothing technique applied to the objective function aims to reduce the noise and abrupt changes in the function's landscape, facilitating a more stable gradient descent. This is evident in the derivative plots for both the linear and nonlinear cases.

Figure (16 in appendix) displays the derivatives of the objective function with respect to r for different values of K, comparing the original and smoothed objectives on linear data.

Figure (17 in appendix) displays the derivatives with respect to r and K of the smoothed objective function on logistic growth data. The smoothed derivatives show a more continuous and predictable pattern, which is crucial for the effectiveness of gradient-based optimization methods.

b) Linear Case Analysis: The path taken by the Gradient Descent algorithm on the smoothed linear data is depicted in Figure 11. The smoothing effect results in a more straightforward gradient path, allowing the GD algorithm to efficiently traverse the parameter space and converge towards the global minimum.

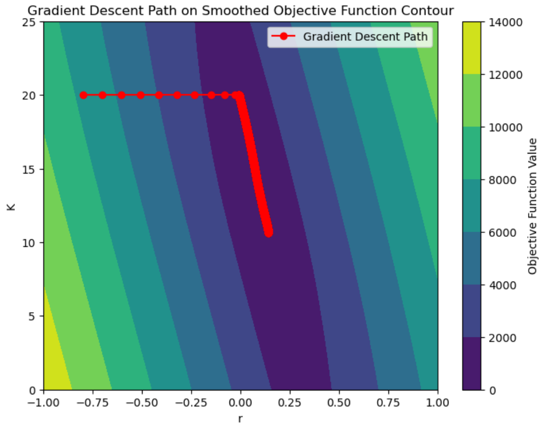

Fig. 11. Path taken by Gradient Descent on Smoothed Linear Obj.

c) Nonlinear Case Analysis: In the nonlinear case, shown in Figure 12, the GD algorithm initially follows a promising path but eventually gets stuck in a plateau region. The plot with plateau regions can be found in Figure 22. After examining the gradients at last iterations, we indeed see values close to 0, e.g [-1.1812e-04 5.4896e-01] w.r.t r and K. These values highlight that the objective is changing very slow around the estimated point, indicating a local minima.

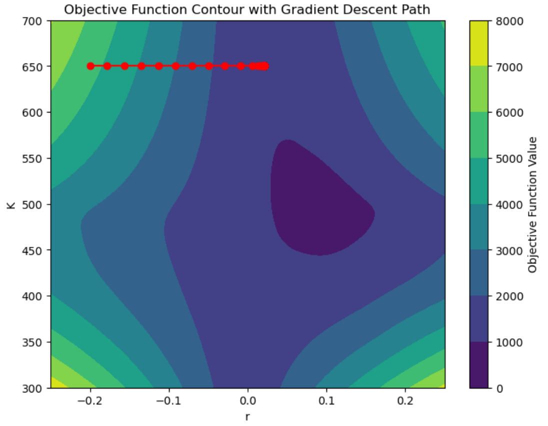

Fig. 12. Path taken by Gradient Descent on Smoothed Non-Linear Obj.

In summary, smoothing transformation effectively mitigates the non-differentiability of the LAD objective function, allowing gradient descent to perform well on linear cases. However, in nonlinear cases, gradient descent can still get trapped in local optima.

E. State-of-the-Art Algorithms

The state-of-the-art algorithms: IRLS, Wesolowsky's Direct Descent, and Li-Arce's Maximum Likelihood Approach—exhibit exceptional performance in optimizing the simple linear LAD objective. These methods efficiently handle the non-differentiability and outliers, leading to rapid convergence with minimal iterations and function evaluations (see appendix figures 18, 19, 20).

V. Discussion

In this section, we address the research questions posed at the beginning of the thesis and suggest potential avenues for future work.

RQ1: Among traditional derivative-free optimization techniques, the *Nelder-Mead algorithm* demonstrated the fastest convergence for both linear and nonlinear LAD problems. The *Trust Region method* also performed similarly to Nelder-Mead in the nonlinear case, effectively navigating complex landscapes. The *state-of-the-art methods* exhibited even faster convergence in the linear case, outperforming traditional methods in terms of function evaluations and iterations due to their tailored approaches to handling non-differentiability in the LAD objective.

RQ2: State-of-the-art techniques showed superior performance in terms of accuracy and robustness for linear LAD problem, reaching the true global minimum with minimal iterations and function evaluations. Among traditional methods, *Nelder-Mead* was the most accurate. For nonlinear LAD problem, *Nelder-Mead* and *Trust-region* performed well, achieving values close to the global minimum. The *Subgradient Line Search* struggled with both linear and nonlinear problems, failing to converge to the global minimum. The *Smoothing with Gradient Descent* approach was effective for linear problem but struggled with the complexity of nonlinear problem.

RQ3: The performance of derivative-free techniques is influenced by the smoothness at optimum points. *Trust-region* and *Nelder-Mead* effectively handled non-smooth objective functions by dynamically adjusting their search strategies. The *Subgradient Line Search* method struggled with non-differentiability and local variations in the objective caused by the noise, leading to zigzagging paths and inefficient convergence. *Smoothing with Gradient Descent* transformed the LAD objective into a smooth function, allowing for effective optimization in linear case, but faced challenges with nonlinear problem due to multiple local minima and complex landscape.

RQ4: Applying derivative-free optimization methods to practical datasets presents several challenges. Computational cost increases significantly with dimensionality, making *Nelder-Mead* and *Trust-region* less practical for high-dimensional problems due to the need in numerous function evaluations. Real-world data often contains noise, adversely affecting convergence and accuracy. The *Subgradient Line Search* struggled with these imperfections, leading to inefficient convergence. Multiple local minima in nonlinear and high-dimensional objective functions further complicate optimization.

Future work should focus on enhancing the performance of derivative-free optimization methods through careful hyperparameter tuning, which can significantly improve convergence rates and accuracy. Adapting state-of-the-art techniques to better handle nonlinear LAD problems, will broaden their applicability. Incorporating scale-invariant (sub)gradient methods could potentially improve the performance in the case where estimates are of different scale, as for example is the case in population dynamics models. These improvements could greatly advance the effectiveness of optimization methods in complex scenarios.

VI. Conclusion

This thesis evaluated derivative-free optimization techniques for the Least Absolute Deviations objective. *Nelder-Mead* and *Trust Region* methods demonstrated effective convergence for both linear and nonlinear problems, with state-of-the-art methods excelling in the linear case. While *Smoothing with Gradient Descent* was effective for linear problems, it struggled with nonlinear landscapes. The *Subgradient Line Search* method faced challenges with convergence and accuracy due to noise and highly non-smooth landscape, highlighting the need for further refinement in handling complex objective functions.

References

[1] *Least Absolute Deviation Regression*, pp. 299–302. New York, NY: Springer New York, 2008.

[2] F. Galton, "Regression towards mediocrity in hereditary stature.," *The Journal of the Anthropological Institute of Great Britain and Ireland*, vol. 15, pp. 246–263, 1886.

[3] A. R. Conn, K. Scheinberg, and L. N. Vicente, *Introduction to Derivative-Free Optimization*. Society for Industrial and Applied Mathematics, 2009.

[4] J. Larson, M. Menickelly, and S. M. Wild, "Derivative-free optimization methods," *Acta Numerica*, vol. 28, p. 287–404, May 2019.

[5] I. Barrodale and F. D. K. Roberts, "An improved algorithm for discrete l1 linear approximation," *SIAM Journal on Numerical Analysis*, vol. 10, no. 5, pp. 839–848, 1973.

[6] S. Burrus, "Iterative reweighted least squares * c .," 2018.

[7] G. O. Wesolowsky, "A new descent algorithm for the least absolute value regression problem," *Communications in Statistics - Simulation and Computation*, vol. 10, no. 5, pp. 479–491, 1981.

[8] G. Arce and L. Yinbo, "A maximum likelihood approach to least absolute deviation regression," *EURASIP Journal on Advances in Signal Processing*, vol. 2004, 09 2004.

[9] J. A. Nelder and R. Mead, "A simplex method for function minimization," *Comput. J.*, vol. 7, pp. 308–313, 1965.

[10] S. P. Boyd and J. Park, "Subgradient methods," 2007.

[11] G. Liuzzi, S. Lucidi, F. Rinaldi, and L. Vicente, "Trust-region methods for the derivative-free optimization of nonsmooth black-box functions," *SIAM Journal on Optimization*, vol. 29, pp. 3012–3035, 12 2019.

[12] V. Jimenez-Fernandez, M. Jimenez, H. Vazquez-Leal, E. Muñoz-Aguirre, H. Cerecedo-Núñez, U. Filobello-Niño, and F. Castro, "Transforming the canonical piecewise-linear model into a smooth-piecewise representation," *SpringerPlus*, vol. 5, p. 1612, 12 2016.

[13] A. Tsoularis and J. Wallace, "Analysis of logistic growth models," *Mathematical biosciences*, vol. 179, pp. 21–55, 07 2002.

[14] Y. Hu, C. Yu, C. Li, and X. Yang, "Conditional subgradient methods for constrained quasi-convex optimization problems," *Journal of nonlinear and convex analysis*, vol. 17, pp. 2143–2158, 10 2016.

Appendix A
Additional Figures

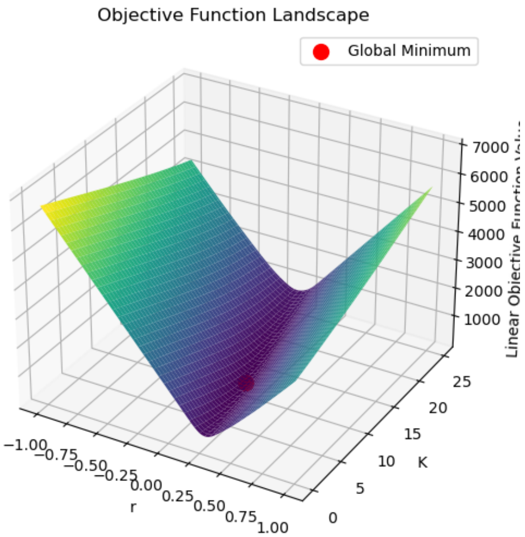

Fig. 13. Linear Objective Function Landscape

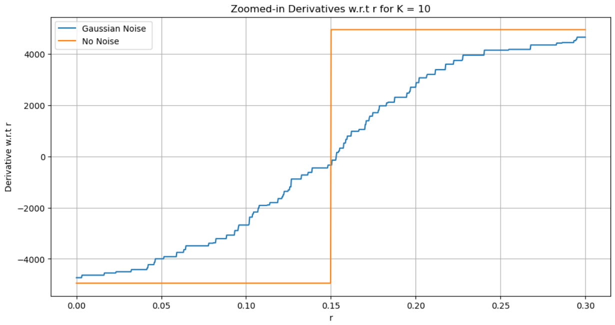

Fig. 14. Zoomed-in Derivatives w.r.t r for $K = 10$

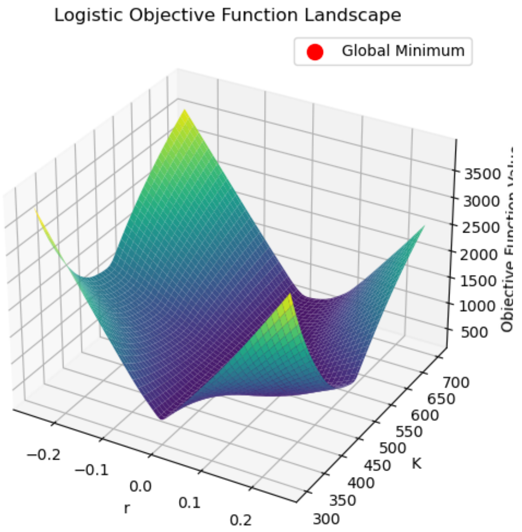

Fig. 15. Logistic Objective Function Landscape

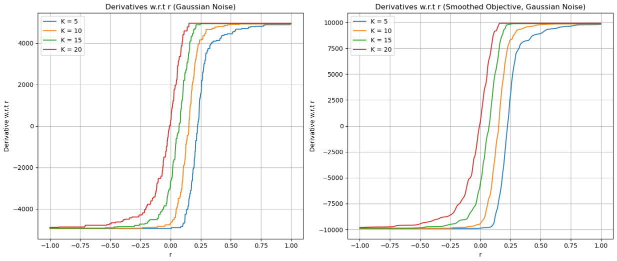

Fig. 16. Derivatives w.r.t r for Original Obj. (left) and Smoothed Obj. (right)

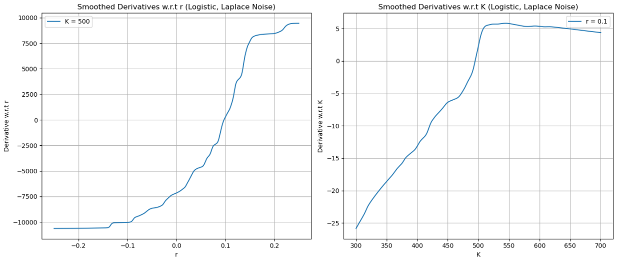

Fig. 17. Derivatives w.r.t r (left) and K (right) for Smoothed Obj

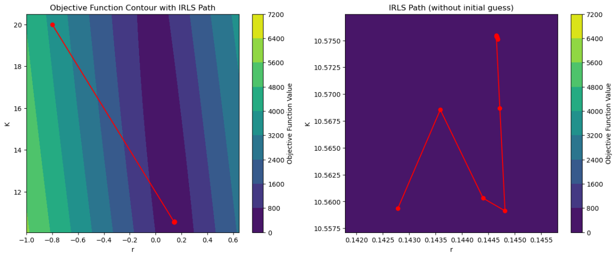

Fig. 18. Path taken by IRLS on Linear Data

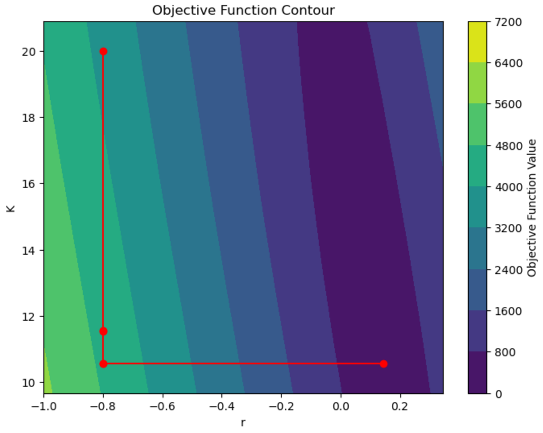

Fig. 19. Path taken by Wesolowsky's Direct Descent on Linear Data

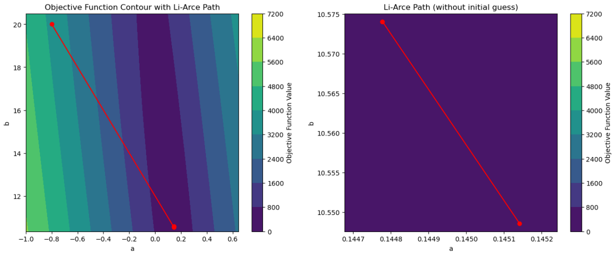

Fig. 20. Path taken by Li-Arce's Maximum Likelihood Approach on Linear Data

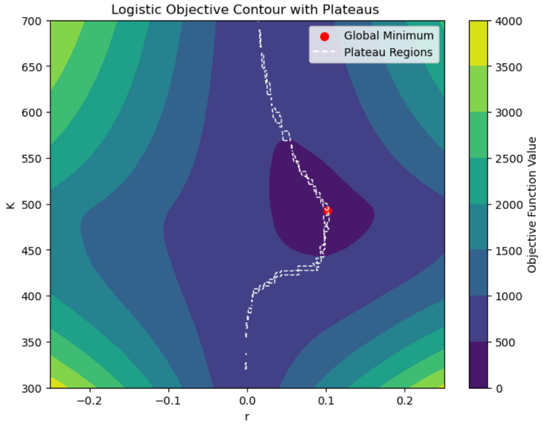

Fig. 21. Logistic Objective Contour with Plateau

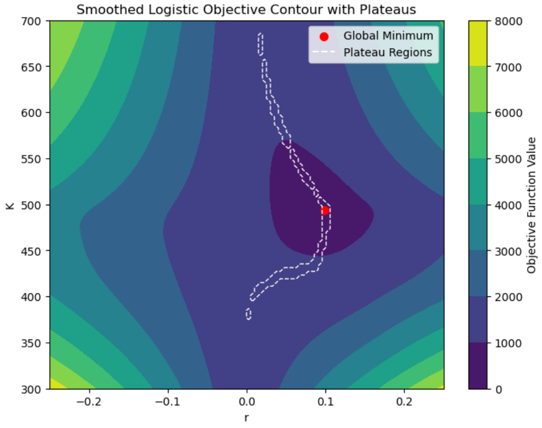

Fig. 22. Smoothed Logistic Objective Contour with Plateau

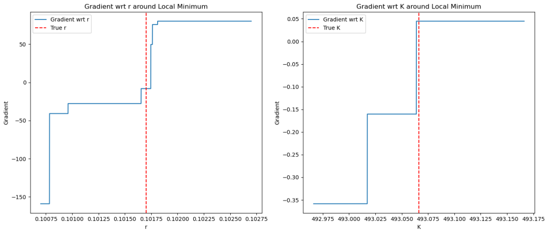

Fig. 23. Gradients with respect to r and K around the local minimum for the nonlinear case.

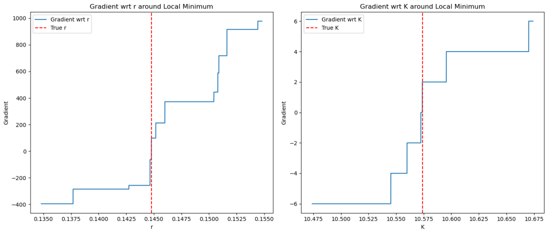

Fig. 24. Gradients with respect to r and K around the local minimum for the linear case.