Least-Squares Estimation of Nonlinear Parameters

Team 5

Saumya Shah - 20171193 & Anoushka Vyas - 20171057

International Insitute of Information Technology, Hyderabad, India

19 April 2021

Overview

1. Problem Statement & Definition
2. Previous Algorithms
3. Levenberg-Marquardt Method
4. Applications
5. Experimental Setup and Analysis
6. Conclusion

Problem Statement

Aim

The term paper is written as part of the course *Signal Detection and Estimation Theory* of *Team 5*. The survey describes various algorithms for least-squares estimation for nonlinear parameters as well as some experiments are done to validate the theoritical observations.

Problem Definition

The model to be fitted to the data is defined as

$$E(y) = f(x_1, x_2, ..., x_m; b_1, b_2, ..., b_k) = f(\mathrm{x}, \mathrm{b}), \tag{1}$$

where $x_1, x_2, ..., x_m$ are independent variables, $b_1, b_2, ..., b_k$ are the k parameters to estimate, $E(y)$ is the expected value of the dependent variable y.
To define the objective function, let the data points be denoted by

$$(Y_i, X_{1i}, X_{2i}, ..., X_{mi}), \quad i = 1, 2, ..., n. \tag{2}$$

The objective function to minimize to get the k parameters is given as

$$\Phi = \sum_{i=1}^{n} [Y_i - \hat{Y}_i]^2 = ||\mathrm{Y} - \hat{\mathrm{Y}}||^2, \tag{3}$$

where $\hat{Y}_i$ is the value of predicted y at the *ith* data point.

Gauss-Newton Method

The method is similar to expanding f in a Taylor series and in the vectorized form it can be written as

$$\langle \mathrm{Y} \rangle = \mathrm{f}_0 + \mathrm{P}\boldsymbol{\delta_t}. \tag{4}$$

The vector $\boldsymbol{\delta_t}$ is a small correction to b, which can be found by least-squares method of setting $\frac{\partial \langle \Phi \rangle}{\partial \delta_j} = 0$, for all j. This can be written as

$$A\boldsymbol{\delta_t} = \mathrm{g}, \tag{5}$$

where

$$A^{[k \times k]} = P^T P, \tag{6}$$

$$P^{[n \times k]} = (\frac{\partial f_i}{\partial b_j}), \quad i = 1, 2, ..., n; \quad j = 1, 2, ..., k, \tag{7}$$

$$g^{[k \times 1]} = (\sum_{i=1}^{n} (Y_i - f_i) \frac{\partial f_i}{\partial b_j}) = P^T(\mathrm{Y} - \mathrm{f}_0), \quad j = 1, 2, ..., k. \tag{8}$$

Gradient Descent Method

The algorithm is implemented by correcting b by only a small value $\boldsymbol{\delta_g}$ in the direction of the negative gradient of Φ. A step size $K\boldsymbol{\delta_g}$, $0 < K \leq 1$ is used to control the amount of change in b after $\boldsymbol{\delta_g}$ has specified the direction.

$$\boldsymbol{\delta_g} = -(\frac{\partial \Phi}{\partial b_1}, \frac{\partial \Phi}{\partial b_2}, ..., \frac{\partial \Phi}{\partial b_k})^T \tag{9}$$

Theoretical Analysis

To provide theoretical basis, following theorems were established. Let $\lambda \geq 0$ be arbitrary and let $\boldsymbol{\delta_l}$ satisfy the equation

$$(A^{(r)} + \lambda^{(r)} I)\boldsymbol{\delta_l}^{(r)} = \mathrm{g}^{(r)}. \tag{10}$$

Theorem

$\boldsymbol{\delta_l}$ minimizes Φ on the sphere whose radius $||\boldsymbol{\delta}||$ satisfies $||\boldsymbol{\delta}||^2 = ||\boldsymbol{\delta_l}||^2$.

Theorem

$||\boldsymbol{\delta_l}(\lambda)||$ decreases to zero monotonically as $\lambda \to \infty$.

Theorem

$\boldsymbol{\delta_l}$ rotates from $\boldsymbol{\delta_t}$ to $\boldsymbol{\delta_g}$ monotonically as $\lambda \to \infty$.

Scale of Measurement

We should scale the b-space in units of the standard deviations of the derivatives $\frac{\partial f_i}{\partial b_j}$, taken over the sample points $i = 1, 2, ..., n$.
We define a scaled matrix A^* and a scaled vector g^*:

$$A^* = a^*_{jj'} = \frac{a_{jj'}}{\sqrt{a_{jj}}\sqrt{a_{j'j'}}}, \tag{11}$$

and

$$g^* = (g^*_j) = (\frac{g_j}{\sqrt{a_{jj}}}), \tag{12}$$

and solve for the Taylor series correction using

$$A^* \boldsymbol{\delta_t}^* = \boldsymbol{g}^*. \tag{13}$$

Then

$$\delta_j = \frac{\delta^*_j}{\sqrt{a_{jj}}}. \tag{14}$$

Construction of the Algorithm

In order to minimize Φ locally, equation for r^{th} iteration can be constructed as

$$(A^{*(r)} + \lambda^{(r)} I)\boldsymbol{\delta_I}^{*(r)} = \boldsymbol{g}^{*(r)}. \tag{15}$$

Once above equation is solved for $\boldsymbol{\delta_I}^{*(r)}$, $\boldsymbol{\delta_I}^{(r)}$ can be obtained using 14. The new trial vector is given by

$$\mathrm{b}^{(r+1)} = \mathrm{b}^{(r)} + \boldsymbol{\delta_I}^{(r)}. \tag{16}$$

It is necessary to select $\lambda^{(r)}$ such that

$$\Phi^{(r+1)} < \Phi^{(r)}. \tag{17}$$

Iterative Algorithm

1. Let $v > 1$, and initialize $\lambda^{(0)} = 10^{-2}$.
2. Compute $\Phi(\lambda^{(r-1)}/v)$ and $\Phi(\lambda^{(r-1)})$.
   2.1 If $\Phi(\lambda^{(r-1)}/v) \leq \Phi^{(r)}$, let $\lambda^{(r)} = \lambda^{(r-1)}/v$.
   2.2 If $\Phi(\lambda^{(r-1)}/v) > \Phi^{(r)}$ and $\Phi(\lambda^{(r-1)}) \leq \Phi^{(r)}$, let $\lambda^{(r)} = \lambda^{(r-1)}$.
   2.3 If $\Phi(\lambda^{(r-1)}/v) > \Phi^{(r)}$ and $\Phi(\lambda^{(r-1)}) > \Phi^{(r)}$, increase λ by successive multiplication by v until for some smallest w, $\Phi(\lambda^{(r-1)}v^w) \leq \Phi^{(r)}$. Let $\lambda^{(r)} = \lambda^{(r-1)}v^w$.
3. Set $\mathrm{b}^{(r)} \leftarrow \mathrm{b}^{(r-1)} + \boldsymbol{\delta_I}$, where $\boldsymbol{\delta_I}$ is obtained using 15 with $\lambda = \lambda^{(r)}$.
4. Set $r \leftarrow r + 1$.
5. Repeat steps 2 through 4.

Curve and Surface Fitting

Problem Statement: Given n data points $p_1, ..., p_n$ in $\mathbb{R}^2$ or $\mathbb{R}^3$. We need to fit a surface over these points.
Consider the family of polynomial curves of form

$$f(x, a_0, ..., a_k) = 0, \tag{18}$$

where $x = (x, y)$ or (x, y, z) and $a_0, ..., a_k$ are the coefficients.

Let the distance from p_i to the surface 18 be d_i. The fitting problem can then be modeled in a least-squares fashion as

$$\min_{a_0, \ldots a_k} \sum_{i=0}^{k} d_i^2 \tag{19}$$

Curve and Surface Fitting

Let q_i be the closest point to p_i on the surface to be determined.

$$f(q_i; a_0, ..., a_k) \approx f(p_i, a_0, ..., a_k) + \nabla f(p_i; a_0, ..., a_k).(q_i - p_i). \tag{20}$$

As the left hand side of the above equation is zero, we have

$$\nabla f(p_i; a_0, ..., a_k).(q_i - p_i) \approx -f(p_i; a_0, ...a_k). \tag{21}$$

$$d_i = ||q_i - p_i|| \approx \frac{|f(p_i; a_0, ...a_k)|}{||\nabla f(p_i; a_0, ..., a_k)||}. \tag{22}$$

Substituting above approximation in 19, the fitting problem can be reformulated as

$$\min_{a_0,\ldots,a_k} \frac{f^2(p_i; a_0, ..., a_k)}{||\nabla f(p_i; a_0, ..., a_k)||^2}. \tag{23}$$

Surface Patch Reconstruction

A robotic hand can reconstruct an unknown surface patch by touch.
The idea is to track along three concurrent curves on the surface while collecting tactile data points (x_k, y_k, z_k), $1 \leq k \leq n$.
For estimating surface patch, we need to fit parabola to the data points along each curve and estimate its curvature and process it through some complex functions. This problem can be formulated as

$$z(x, y) = \frac{1}{2}(k_1 x^2 + k_2 y^2) + \sum_{3 \leq i+j \leq d} a_{ij} x^i y^j, \tag{24}$$

Surface Patch Reconstruction

LSE objective is given by

$$f(a) = \frac{1}{n} \sum_{k=1}^{n} (z(x_k, y_k) - z_k)^2. \tag{25}$$

The coefficients of the polynomial are determined in a least-squares sense as

$$\min_a f(a) \tag{26}$$

Experimental Setup

The experiments are performed by considering f as a gaussian function

$$f(x) = a * \frac{1}{\sqrt{2\pi\sigma^2}} e^{(-\frac{1}{2}(\frac{x-\mu}{\sigma})^2)}, \tag{27}$$

where $a = 10$ is the scaling factor, $\mu = 0$ is the mean and $\sigma = 20$ is the standard deviation. These three are the nonlinear parameters to be estimated.

The source code of our experiments and theoretical analysis can be found in the following GitHub repository [Link].

Performed Experiments

- **Baseline:** The experiment is performed for initilization of all one, learning rate of 0.01 and 100 observation points.
- **Experiment 1:** In this experiment the initialization of the parameters is changed to $30, 35, 40$.
- **Experiment 2:** The number of observation points are changed to 20.
- **Experiment 3:** A gaussian noise of $\mu = 0.1$ and $\sigma = 0.1$ is added to the observations.

Results

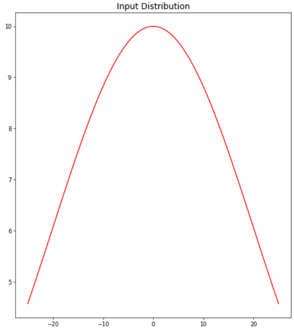
Algorithm	Baseline	Experiment 1	Experiment 2	Experiment 3
Gradient Descent	631	1985	2835	682
Gauss-Newton	500	1797	2212	548
Levenberg-Marquardt	506	1814	2244	524

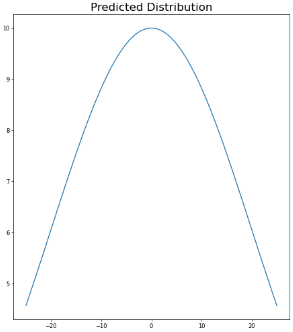
Table: Number of iteration required by the algorithms to converge to minimum

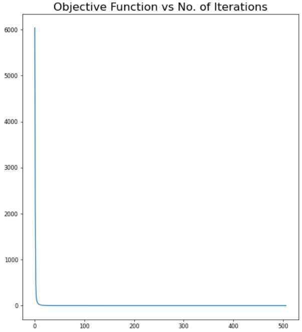
	Baseline			Experiment 1			Experiment 2			Experiment 3		
Algorithms	a	μ	σ	a	μ	σ	a	μ	σ	a	μ	σ
Gradient Descent	10.00	9.48e-14	20	10.00	-9.41e-14	20.00	10.00	3.17e-14	20.00	10.08	0.03	20.14
Gauss-Newton	10.00	7.76e-14	20	10.00	-7.75e-14	20.00	10.00	2.60e-14	20.00	10.08	0.03	20.14
Levenberg-Marquardt	10.00	7.79e-14	20	10.00	-7.81e-14	20.00	10.00	2.61e-14	20.00	10.08	0.03	20.14

Table: Estimated parameters of gaussian where a is the scaling factor, μ is the mean, and σ is the standard deviation

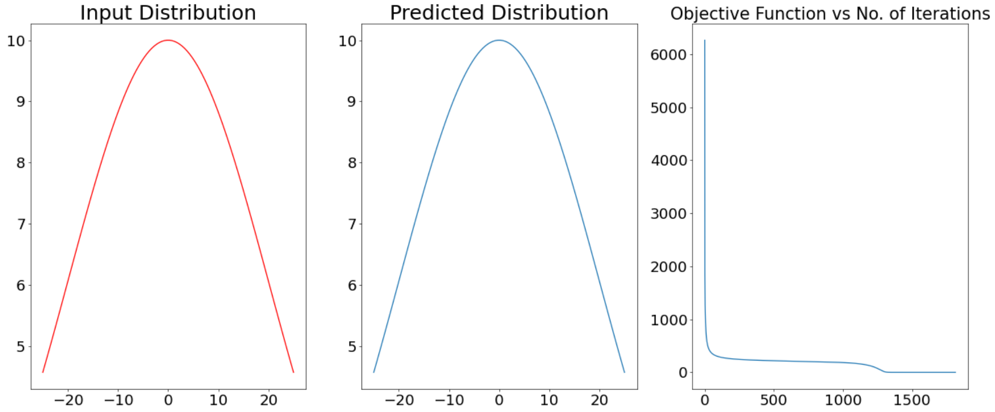
Baseline plots for the Levenberg-Marquardt Method



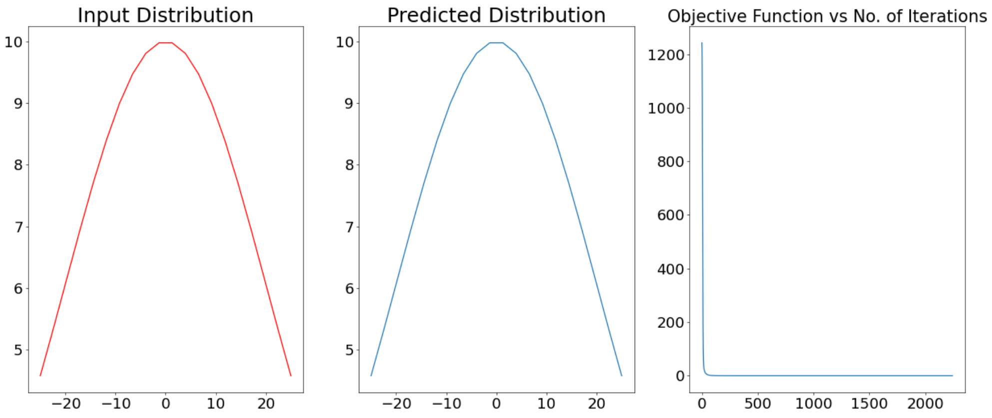




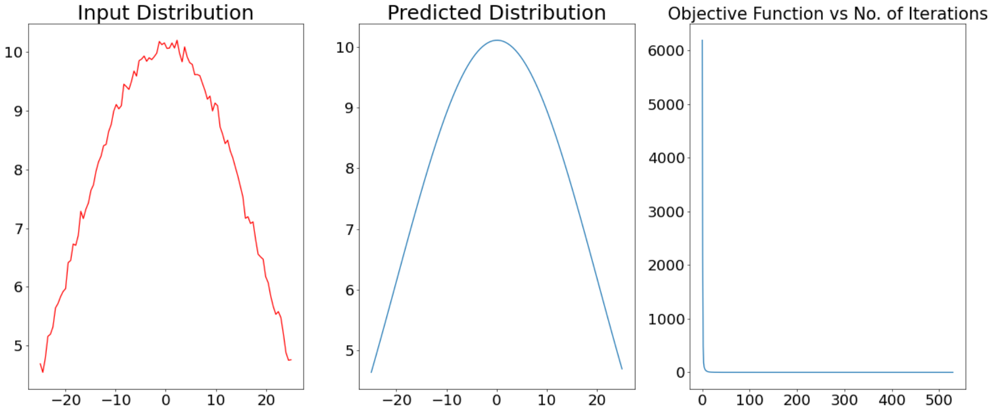
Experiment-1 plots for the Levenberg-Marquardt Method



Experiment-2 plots for the Levenberg-Marquardt Method



Experiment-3 plots for the Levenberg-Marquardt Method



Conclusion

The Levenberg-Marquardt method combines the property of Gradient Descent methods ability to converge from an initial guess and the property of Gauss-Newton method to converge to the solution faster after the neighbourhood values are reached but is able to eliminate the shortcomings of both algorithms which is slow convergence and divergence from the solution.

References

D. W. Marquardt (1963)
An algorithm for least-squares estimation of nonlinear parameters.
Journal of the Society for Industrial and Applied Mathematics 11(2), 431–441.

M. M. Blane, Z. Lei, H. Civi, D. B. Cooper (2000)
The 3L algorithm for fitting implicit polynomial curves and surfaces to data.
IEEE Transactions on Pattern Analysis and Machine Intelligence 22(3), 298–313.

Y.-B. Jia and J. Tian (2010)
Surface patch reconstruction from "one-dimensional" tactile data.
IEEE Transactions on Automation Science and Engineering 7(2), 400–407.

Thank You