姓名：林皓泓 学号：3170105156

An Application of Nonlinear Least Square Method on Respiratory Physiology

Abstract

Partial pressure of Oxygen (P_{O_2}) has become a crucial indicator to evaluate health condition for COVID-19 infections. However, it is very hard to measure the P_{O_2} directly since it requires catheterization, which is an invasive procedure and is extremely perilous for doctors and nurses.

In order to get the P_{O_2} in an indirect and safer way, we utilize the indicator of fractional O_2 saturation (%). It is our purpose to find an analytical relationship between oxygen saturation S_{O_2} and partial oxygen pressure (P_{O_2}).

In this work, we refer to a previous dataset of P_{O_2} and S_{O_2}, designing a series of unlinear least square algorithm and evaluate their performance on the target curve.

1 Problem Restatement

According to the reference [1], Richards family of growth curves has a dramatically outstanding performance to carry out the nonlinear fitting. Therefore, we adopt the following parameters x_1, x_2, x_3, x_4, x_5 to be determined. The relationship between $S_{O_2} = \hat{y}$ and $P_{O_2} = t$ is given approximately as:

$$\hat{y} = C_5 C_1^{C_2^{(1-C_3^t)^{C_4}}}$$

Where

$$C_1 = e^{-x_1}, C_2 = e^{-x_2}, C_3 = e^{-x_3}, C_4 = x_4, C_5 = x_5$$

We design an algorithm to solve the $x_i, i = 1, 2, ..., 5$ so that the difference between $\hat{y}$ and y is minimized.

2 Introduction to nonlinear least square problem.

2.1 Guass Newton Method

The objective function is given as:

$$\frac{1}{2}\sum_{j=1}^{m}[\phi(x;t_j) - y_j]^2$$

If we consider to use local gradient and Hessian matrix to design the algorithm, there is a tricky modification from linesearch Newton Method, usually known as Guass-Newton Method. Since we make the approximation [2]:

$$\nabla^2 f(x) = \sum_{j=1}^{m} \nabla r_j(x) \nabla r_j(x)^T + \sum_{j=1}^{m} r_j(x) \nabla^2 r_j(x) \approx J(x)^T J(x)$$

Thus we have a descendant direction p^{GN} that satisfies $J_k^T J_k p_k^{GN} = -J_k^T r_k$.

It is a linear LS problem for each iteration of GN Method:

$$\min_p \frac{1}{2}||J_k p + r_k||_2^2$$

Notice that whenever J_k has full column rank, p^{GN} is descendant. But things would go wrong when J_k does not have full column rank. Here we could utilize a Modification based on Cholesky Fractorization[6] to guarantee the positiveness of $J_k^T J_k$ (*This will be also discussed in LM Method, but here I introduce it to modify the original GN Method to improve its performance*).

Algorithm 1 Modified Gauss Newton Method

Require: $k \leftarrow 0, x \leftarrow x_0$

while $k < max\_iter$ **do**
  $J_k \leftarrow Jacobi(x)$
  $r_k \leftarrow \frac{1}{2}||\phi(x) - y||_2^2$
  $H \leftarrow J_k^T J_k,\ b \leftarrow J_k^T r_k$
  $H = CholeskyModify(H)$
  $p_k^{GN} = -H^{-1} b$
  $\alpha = linesearch(x_k)$
  $x_{k+1} = x_k + \alpha p_l^{GN}$
  if $||x_{k+1} - xk|| < bound$ **then**
    break
  end if
  $k \leftarrow k + 1$
end while

2.2 Levenberg-Marquardt Method

When GN Method become singular, the convergence of linesearch direction may be lost. Notice the dataset is far from high dimensional, we could save J_k and calculate $J^T J$ with little cost. Therefore, there is no necessity to utilize Givens and Householder Transform. We simply adopt Trust Region Method and Dogleg Method to finish the LM Algorithm.[3]

Notice that we are doing exactly the same tasks with GN iteration, except for the constraints of the range of $||P||$ [4]:

$$\min_p \frac{1}{2}||J_k p + r_k||^2, \quad s.t.\ ||P||_2 \leq \Delta_k$$

We further explain the dogleg method as follows:

We have Newton Point $p^N = -B_k^{-1} g_k$ and Cauchy Point $P^u = -\frac{g_k^T g_k}{g_k^T B_k g_k} g_k$. The dogleg curve is defined as:

$$p = \begin{cases} \tau p^u & \tau \in [0,1] \\ p^u + (\tau - 1)(p^N - p^u) & \tau \in (1,2] \end{cases}$$

Thus we could solve suitable τ precisely according to the current radius Δ of trust region. [5]

Algorithm 2 Trust Region with *Dogleg*

Require: $\hat{\Delta} \leftarrow 10, \Delta_0 \leftarrow 0.8\hat{\Delta}, \eta \leftarrow 1e-4.$

while $k < max\_iter$ **do**
  Update $f_k \leftarrow frac12||r_k||_2^2$, $g_k = J_k^T r_k$, $B_k = J_k^T J_k$
  $m_k(p) = f_k + g_k^T p + \frac{1}{2} p^T B_k p$
  $p_k \leftarrow Dogleg(x_k, \Delta_k)$
  $\rho_k = \frac{f(x_k) - f(x_k + p_k)}{m_k(0) - m_k(p_k)}$
  if $\rho_k < \frac{1}{4}$ **then**
    $\Delta_{k+1} = \frac{1}{4}\Delta_k$
  else
    if then$\rho_k > \frac{3}{4}$ and $||\rho_k|| = \Delta_k$
      $\Delta = min(2\Delta_k, \hat{\Delta})$
    end if
  end if
  if $\rho_k > \eta$ **then** $x_{k+1} = x_k + p_k$
  end if
end while

Algorithm 3 Dogleg Algorithm

Require: p^u, p^N, Δ

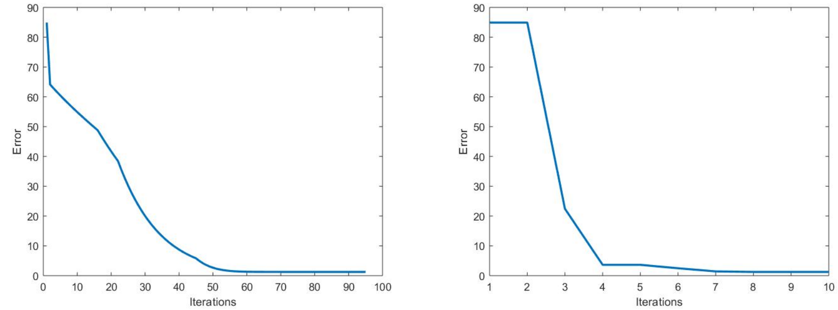
if $||p^N|| < \Delta$ **then**
  $p_k = p^N$
else
  if $||p^u|| \geq \Delta$ **then**
    $p_k = \frac{\Delta}{||p^u||} p^u$
  else
    Solve Quadratic Equation: $||p^u + (\tau - 1)(p^N - p^u)|| = \Delta, \tau \in (1, 2]$
  end if
end if

3 Numerical Result of Fitting

In this section, we utilize the GN and LM method respectively, and illustrate their results.

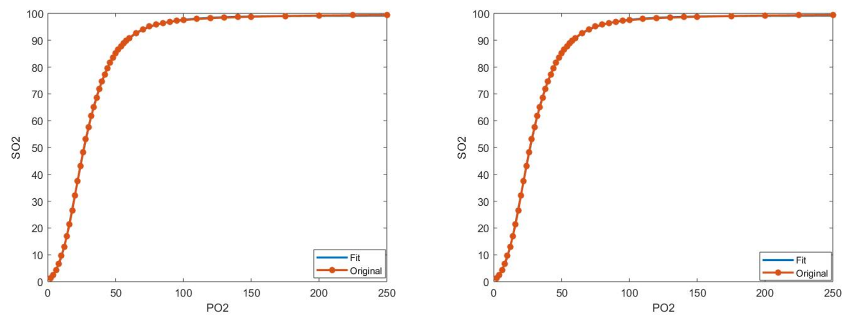
Here we start the iteration. We give them the same initial value $x_0 = [5; 8; 0.01; 1.2; 100]$, and get the following iteration results.

3.1 Comparison between GN and LM algorithm

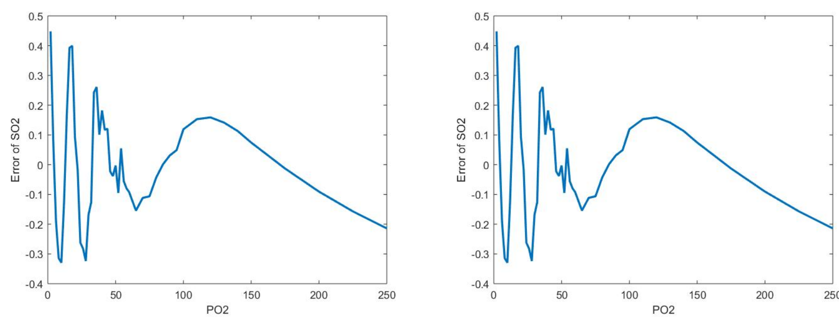


(a) Gauss-Newton (b) Levenberg-Marquardt

Figure 1: Convergence Ability of GN and LM. We notice that GN spends much more iterations than LM to achieve the same convergence effect.



(a) Gauss-Newton (b) Levenberg-Marquardt

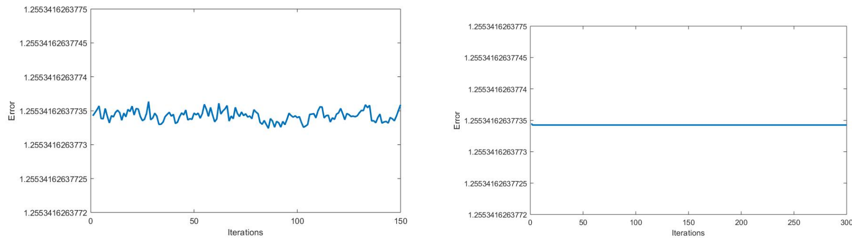


(c) Gauss-Newton (d) Levenberg-Marquardt

Figure 2: Fitting Curve and Final error, which seems to have very similar performance. The final error $||r_k|| = 1.2553$ for both mothods, which is the best numerical results I've ever achieved.

3.2 Best Numerical Result

My best numerical result is $||r_k|| = 1.2553$, and the residual mean square of the final error is 0.0335. The optimal value of X is $X = [4.5109, 7.5680, 0.0146, 1.2381, 99.4902]^T$. This result is close to the paper's result [1].

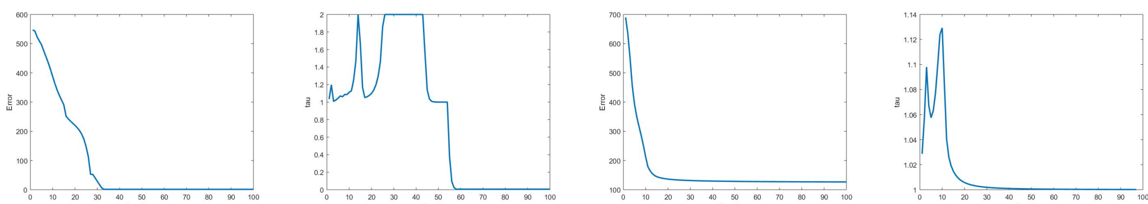


(a) Gauss-Newton (b) Levenberg-Marquardt

Figure 3: After achieve somewhere around the best results, neither GN nor LM method could reduce the loss anymore.

3.3 Discussion on Initial Value Dependency

If we add 2-10 times of the optimal value as initial value, we could still get an optimal result. For instance, we select $X_0 = [50; 50; 0.15; 5; 150]$. LM could achieve convergence after 33 iterations while GN could not converge. We make several other tests, showing that LM Method could converge on most of the situations while GN could achieve just part of them. Still, there are some situations where LM could hardly converge, like when $x_3^0 > 0.5$ or $x_4^0 > 10$. This might cause serious problems on the gradient explosion for x_3 and x_4, since they are at the higher level of exponential function and their gradients usually vanish/explode.



(a) Iteration error (b) Iteration τ (c) Failed: Iteration error (d) Failed: Iteration τ

Figure 4: Here we visualize the τ that indicate the step length decided by dogleg method. The second initial value has $x_4^0 = 10$ as initial value. It couldn't converge since the trust region shrink at rapid speed after several reasonable iterations. Then it couldn't reach any steps close to Newton direction while the Cauchy direction is not descendant enough.

4 Further Optimization

We could use:

- Ellipsodal trust region to replace the original trust region, and adjust the diagonal matrix in Cholesky Modification adaptively. We could find greater turbulance on the error at smaller value, where numerical instability usually occurs.
- Large residual problem: To achieve the descendant direction even on poor initial value, we could use hybrid strategy of Newton or Quasi-Newton direction to replace the Jacobian approximation.

They might achieve a better numerical result.

5 Reference

[1] The choice of an appropriate nonlinear model for the relationship between certain variables in respiratory physiology, S.H. C. du Toit et al., South African Statistics Journal (1984), 18, 161-176.

[2] Numerical Optimization, Jorge Nocedal et al., Second Edition, Springer, 254-255.

[3] Numerical Optimization, Jorge Nocedal et al., Second Edition, Springer, 258-259.

[4] Numerical Optimization, Jorge Nocedal et al., Second Edition, Springer, 69.

[5] Numerical Optimization, Jorge Nocedal et al., Second Edition, Springer, 73-75.

[6] Numerical Optimization, Jorge Nocedal et al., Second Edition, Springer, 53-55.