

Nonlinear Least Squares

Ying Xiong

School of Engineering and Applied Sciences

Harvard University

yxiong@seas.harvard.edu

January 19th, 2014

1 Problem Statement

The least squares problem is to find a (local) minimizer for cost function

$$F(\mathbf{x}) = \sum_{i=1}^m (f_i(\mathbf{x}))^2 = \|\mathbf{f}(\mathbf{x})\|^2 = \mathbf{f}(\mathbf{x})^\top \mathbf{f}(\mathbf{x}), \quad (1)$$

where $f_i : \mathbb{R}^n \mapsto \mathbb{R}$, $i = 1, \dots, m$ are given nonlinear functions.

A least squares problem is a special variant of the more general nonlinear programming problem, and the special form provides useful structure that we can exploit. Define

$$(\mathbf{J}(\mathbf{x}))_{i,j} = \frac{\partial f_i}{\partial x_j}(\mathbf{x}) \quad (2)$$

the Jacobian matrix of $\mathbf{f}(\mathbf{x})$, then we have

$$\mathbf{F}'(\mathbf{x}) = 2\mathbf{J}(\mathbf{x})^\top \mathbf{f}(\mathbf{x}), \quad (3)$$

$$\mathbf{F}''(\mathbf{x}) = 2\mathbf{J}(\mathbf{x})^\top \mathbf{J}(\mathbf{x}) + 2 \sum_{i=1}^m f_i(\mathbf{x}) \mathbf{f}_i''(\mathbf{x}), \quad (4)$$

which means even when we do not have second-order information of $\mathbf{f}(\mathbf{x})$, we still know *something* about $\mathbf{F}''(\mathbf{x})$ from $\mathbf{J}(\mathbf{x})$ alone.

2 Algorithms

We make a linear approximation on $\mathbf{f}(\mathbf{x})$ near a given \mathbf{x} as

$$\mathbf{f}(\mathbf{x} + \mathbf{h}) \approx \ell(\mathbf{h}) = \mathbf{f}(\mathbf{x}) + \mathbf{J}(\mathbf{x}) \mathbf{h}, \quad (5)$$

which yields

$$F(\mathbf{x} + \mathbf{h}) \approx L(\mathbf{h}) = \ell(\mathbf{h})^\top \ell(\mathbf{h}) \quad (6)$$

$$= \mathbf{f}^\top \mathbf{f} + 2\mathbf{h}^\top \mathbf{J}^\top \mathbf{f} + \mathbf{h}^\top \mathbf{J}^\top \mathbf{J} \mathbf{h} \quad (7)$$

$$= F(\mathbf{x}) + 2\mathbf{h}^\top \mathbf{J}^\top \mathbf{f} + \mathbf{h}^\top \mathbf{J}^\top \mathbf{J} \mathbf{h}. \quad (8)$$

Note that this is equivalent to perform a second order Taylor expansion on $F(\mathbf{x})$ and approximate \mathbf{F}'' as $2\mathbf{J}^\top \mathbf{J}$.

2.1 Gauss-Newton algorithm

The Gauss-Newton algorithm minimize (8) directly, with

$$\mathbf{h}_{\text{gn}} = - \left(\mathbf{J}^\top \mathbf{J} \right)^{-1} \mathbf{J}^\top \mathbf{f}. \quad (9)$$

The algorithm has at least two short-comings: (1) $\left(\mathbf{J}^\top \mathbf{J} \right)$ might be singular and (2) \mathbf{h}_{gn} might not be a descending direction.

2.2 Levenberg-Marquardt algorithm [1, 2, 3]

Levenberg-Marquardt algorithm is a damped Gaussian-Newton method

$$\mathbf{h}_{\text{lm},1} = - \left(\mathbf{J}^\top \mathbf{J} + \mu \mathbf{I} \right)^{-1} \mathbf{J}^\top \mathbf{f}, \quad (10)$$

or, as suggested by Marquardt

$$\mathbf{h}_{\text{lm},2} = - \left(\mathbf{J}^\top \mathbf{J} + \mu \text{diag} \left(\mathbf{J}^\top \mathbf{J} \right) \right)^{-1} \mathbf{J}^\top \mathbf{f}. \quad (11)$$

We write the two forms together as

$$\mathbf{h}_{\text{lm}}^\top = - \left(\mathbf{J}^\top \mathbf{J} + \mu \mathbf{D} \right)^{-1} \mathbf{J}^\top \mathbf{f}, \quad (12)$$

where the “damping matrix” \mathbf{D} can either be \mathbf{I} or $\text{diag} \left(\mathbf{J}^\top \mathbf{J} \right)$.

2.2.1 Choice of damping factor [4]

Define a *gain ratio*

$$\varrho = \frac{F(\mathbf{x}) - F(\mathbf{x} + \mathbf{h}_{\text{lm}})}{L(\mathbf{0}) - L(\mathbf{h}_{\text{lm}})}, \quad (13)$$

where $L(\mathbf{h})$ is defined in (6), and the denominator can be calculated as

$$L(\mathbf{0}) - L(\mathbf{h}_{\text{lm}}) = \mathbf{h}_{\text{lm}}^\top \left(\mu \mathbf{D} \mathbf{h}_{\text{lm}} - \mathbf{J}^\top \mathbf{f} \right) \quad (14)$$

The update rule for μ will be

$$\mu_{k+1} = \begin{cases} \mu \cdot \max \left\{ \frac{1}{3}, 1 - (2\varrho - 1)^3 \right\}; & \nu = 2 & \text{if } \varrho > 0, \\ \mu \cdot \nu; & \nu = 2 \cdot \nu & \text{otherwise.} \end{cases} \quad (15)$$

The initial μ is usually set as $\tau \cdot \max_i \left\{ \left(\mathbf{J}^\top \mathbf{J} \right)_{i,i} \right\}$, where τ is a user specified parameter, which should be a small value, *e.g.* $\tau = 10^{-6}$ if \mathbf{x}_0 is a good approximation to the final local minimum, and 10^{-3} or even 1 otherwise.

2.2.2 Algorithm description

Algorithm 1: Levenberg-Marquardt method

Input : $f(\mathbf{x})$, $J(\mathbf{x})$: Input function and its Jacobian matrix.

Input : \mathbf{x}_0 : Initial guess.

Input : τ : A parameter specifying initial damping factor, default 10^{-3} .

Input : A stopping criterion.

Output: \mathbf{x} : A local minimum.

```
1  $\mathbf{x} = \mathbf{x}_0$ ,  $\mu = \tau \cdot \max_i \left\{ \left( J^\top J \right)_{i,i} \right\}$ ,  $\nu = 2$ .
2 while the stopping criterion is not met do
3   Calculate  $\mathbf{h}_{lm}$  according to (12).
4   Calculate  $\varrho$  according to (13).
5   if  $\varrho > 0$  then
6      $\mathbf{x} = \mathbf{x} + \mathbf{h}_{lm}$ ,  $\mu = \mu \cdot \max \left\{ \frac{1}{3}, 1 - (2\varrho - 1)^3 \right\}$ ,  $\nu = 2$ .
7   else
8      $\mu = \mu \cdot \nu$ ,  $\nu = 2\nu$ .
9   end
10 end
```

References

- [1] K. Levenberg, “A method for the solution of certain problems in least squares,” *Quarterly of applied mathematics*, vol. 2, pp. 164–168, 1944.
- [2] D. W. Marquardt, “An algorithm for least-squares estimation of nonlinear parameters,” *Journal of the Society for Industrial & Applied Mathematics*, vol. 11, no. 2, pp. 431–441, 1963.
- [3] Wikipedia, “Plagiarism — Wikipedia, the free encyclopedia,” 2014, [Online; accessed 20-January-2014]. [Online]. Available: http://en.wikipedia.org/wiki/Levenberg-Marquardt_algorithm
- [4] K. Madsen, H. B. Nielsen, and O. Tingleff, *Methods for non-linear least squares problems*, 2nd ed., 2004.