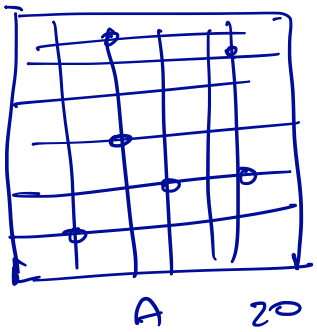


Nonlinear least squares

Ankush Aggarwal

November 9, 2020

Notation: repeated indices imply summation, normal font with a subscript indicates an element of vector, and a bold font indicates the whole vector. A comma in the subscript indicates differentiation.

A general least square problem is to find

$$\boldsymbol{\theta}^* = \underset{\boldsymbol{\theta}\in\mathbb{R}^M}{\arg\min}\, \mathcal{F}, \text{ where } \mathcal{F} = \frac{1}{2} r_i(\boldsymbol{\theta}) r_i(\boldsymbol{\theta}) = \frac{1}{2}\boldsymbol{r}^\top \boldsymbol{r}, \tag{1}$$

where $\boldsymbol{\theta} = [\theta_1, \theta_2, \ldots, \theta_M]$ and $M < n$. The residual functions r_i for $i = 1, \ldots, n$ are nonlinear in $\boldsymbol{\theta}$. The condition of minima is

$$\frac{\partial}{\partial \theta_J}\left[\frac{1}{2} r_i(\boldsymbol{\theta}) r_i(\boldsymbol{\theta})\right] = 0 \;\text{ for } J = 1, \ldots, M. \tag{2}$$

$$\Rightarrow r_i(\boldsymbol{\theta}^*)\left.\frac{\partial r_i}{\partial \theta_J}\right|_{\boldsymbol{\theta}^*} = 0 \;\text{ for } J = 1, \ldots, M. \tag{3}$$

We rewrite the above using a short-hand: find $\boldsymbol{\theta}^*$ such that

$$r_i(\boldsymbol{\theta}^*) r_{i,J}(\boldsymbol{\theta}^*) = 0 \;\text{ for } J = 1, \ldots, M. \tag{4}$$

Standard formulation

In order to solve the above system of equations, we linearize them around an initial guess $\boldsymbol{\theta}^0$:

$$r_i(\boldsymbol{\theta}^0) r_{i,J}(\boldsymbol{\theta}^0) + \frac{\partial}{\partial \theta_K}\left[r_i(\boldsymbol{\theta}) r_{i,J}(\boldsymbol{\theta})\right]_{\boldsymbol{\theta}^0}\left(\theta_K^* - \theta_K^0\right) + \text{higher order terms} = 0 \tag{5}$$

$$\left[r_{i,K}(\boldsymbol{\theta}^0) r_{i,J}(\boldsymbol{\theta}^0) + r_i(\boldsymbol{\theta}^0) r_{i,JK}(\boldsymbol{\theta}^0)\right]\left(\theta_K^* - \theta_K^0\right) + \text{higher order terms} = -r_i(\boldsymbol{\theta}^0) r_{i,J}(\boldsymbol{\theta}^0), \tag{6}$$

for $J = 1, \ldots, M$. Ignoring the higher order terms the above gives the full Newton's iteration to find the update in our guess $\Delta\theta_K$

$$\left[r_{i,K}(\boldsymbol{\theta}^0) r_{i,J}(\boldsymbol{\theta}^0) + r_i(\boldsymbol{\theta}^0) r_{i,JK}(\boldsymbol{\theta}^0)\right]\Delta\theta_K = -r_i(\boldsymbol{\theta}^0) r_{i,J}(\boldsymbol{\theta}^0). \tag{7}$$

Furthermore, ignoring the second derivative $r_{i,JK}$ gives us the Gauss-Newton iteration to find the update in our guess $\Delta\theta_K$

$$r_{i,K}(\boldsymbol{\theta}^0) r_{i,J}(\boldsymbol{\theta}^0)\Delta\theta_K = -r_i(\boldsymbol{\theta}^0) r_{i,J}(\boldsymbol{\theta}^0). \tag{8}$$

The above can also be written as

$$[\mathbf{J}]_{iK}[\mathbf{J}]_{iJ}\Delta\theta_K = -r_i(\boldsymbol{\theta}^0)[\mathbf{J}]_{iJ} \;\text{ for } J = 1, \ldots, M, \tag{9}$$

where $[\mathbf{J}]_{iK} = r_{i,K}(\boldsymbol{\theta}^0)$.

Another way to arrive at Gauss-Newton iteration is to linearize the residuals instead of (4). That is, if we write

$$r_i(\boldsymbol{\theta}) = r_i(\boldsymbol{\theta}^0) + r_{i,K}(\boldsymbol{\theta}^0)\left(\theta_K - \theta_K^0\right) + \text{higher order terms}, \tag{10}$$

and ignore the higher order terms

$$r_i(\boldsymbol{\theta}) \approx r_i(\boldsymbol{\theta}^0) + r_{i,K}(\boldsymbol{\theta}^0)\left(\theta_K - \theta_K^0\right). \tag{11}$$

Therefore,

$$r_{i,J}(\boldsymbol{\theta}) \approx r_{i,J}(\boldsymbol{\theta}^0). \tag{12}$$

Substituting the above into (4), we get

$$\left[r_i(\boldsymbol{\theta}^0) + r_{i,K}(\boldsymbol{\theta}^0)\left(\theta_K^* - \theta_K^0\right)\right] r_{i,J}(\boldsymbol{\theta}^0) = 0 \;\text{ for } J = 1, \ldots, M. \tag{13}$$

After rearranging we get the Gauss-Newton iteration (8).

Halley's method inspired formulation

Inspired by the Halley's method, we propose a two-step algorithm. The first step is the Gauss-Newton

$$r_{i,K}(\boldsymbol{\theta}^0) r_{i,J}(\boldsymbol{\theta}^0) \Delta\hat{\theta}_K = -r_i(\boldsymbol{\theta}^0) r_{i,J}(\boldsymbol{\theta}^0), \tag{14}$$

where $\Delta\hat{\theta}$ denotes the step that satisfies the above equation. This step is corrected by using the second derivative. In order to do this, we write the Taylor series expansion of the residual up to second order as:

$$\begin{aligned}
r_i(\boldsymbol{\theta}) &\approx r_i(\boldsymbol{\theta}^0) + r_{i,K}(\boldsymbol{\theta}^0)\Delta\theta_K + \frac{1}{2} r_{i,KL}(\boldsymbol{\theta}^0)\Delta\theta_K \Delta\theta_L && (15)\\
&= r_i(\boldsymbol{\theta}^0) + \left[r_{i,K}(\boldsymbol{\theta}^0) + \frac{1}{2} r_{i,KL}(\boldsymbol{\theta}^0)\Delta\theta_L\right]\Delta\theta_K && (16)\\
&\approx r_i(\boldsymbol{\theta}^0) + \left[r_{i,K}(\boldsymbol{\theta}^0) + \frac{1}{2} r_{i,KL}(\boldsymbol{\theta}^0)\Delta\hat{\theta}_L\right]\Delta\theta_K && (17)\\
&= r_i(\boldsymbol{\theta}^0) + [\hat{\mathbf{J}}]_{iK}\Delta\theta_K. && (18)
\end{aligned}$$

where, we used a new definition

$$[\hat{\mathbf{J}}]_{iK} = r_{i,K}(\boldsymbol{\theta}^0) + \frac{1}{2} r_{i,KL}(\boldsymbol{\theta}^0)\Delta\hat{\theta}_L = [\mathbf{J}]_{iK} + \frac{1}{2} r_{i,KL}(\boldsymbol{\theta}^0)\Delta\hat{\theta}_L. \tag{19}$$

Using this updated linear approximation of the residual we get the second (correction) step

$$[\hat{\mathbf{J}}]_{iK}[\hat{\mathbf{J}}]_{iJ}\Delta\bar{\theta}_K = -r_i(\boldsymbol{\theta}^0)[\hat{\mathbf{J}}]_{iJ} \;\text{ for } J = 1, \ldots, M, \tag{20}$$

where $\Delta\bar{\theta}_K$ is the corrected step.

Analysis of the step direction

Whether the target function $\mathcal{F}$ increases or decreases along the calculated step is an important measure of any optimization algorithm. Since the gradient of $\mathcal{F}$ at any point $\boldsymbol{\theta}$ is $\mathcal{F}_{,J} = r_i(\boldsymbol{\theta}) r_{i,J}(\boldsymbol{\theta})$. The rate of change of $\mathcal{F}$ along a direction $\Delta\boldsymbol{\theta} = \boldsymbol{d}$ is therefore $= \mathcal{F}_{,J} d_J = r_i(\boldsymbol{\theta}) r_{i,J}(\boldsymbol{\theta}) d_J$.

For Gauss-Newton algorithm, the gradient along the step can be calculated by using (9) and taking an inner product with $\Delta\theta_J$ on both sides:

$$\Delta\theta_J [\mathbf{J}]_{iK}[\mathbf{J}]_{iJ}\Delta\theta_K = -\Delta\theta_J r_i(\boldsymbol{\theta}^0)[\mathbf{J}]_{iJ}. \tag{21}$$

If $[\mathbf{J}]_{iK}$ is full rank, then the matrix $[\mathbf{J}]_{iK}[\mathbf{J}]_{iJ}$ is positive definite, and, therefore, the left hand side of the above equation is always positive

$$\Delta\theta_J [\mathbf{J}]_{iK}[\mathbf{J}]_{iJ}\Delta\theta_K = \Delta\theta_J \left([\mathbf{J}]_{iK}[\mathbf{J}]_{iJ}\right)\Delta\theta_K \geq 0. \tag{22}$$

Thus, the right hand side is always negative, which is the gradient along the step direction:

$$\Delta\theta_J \left(r_i(\boldsymbol{\theta}^0)[\mathbf{J}]_{iJ}\right) \leq 0. \tag{23}$$

This is an attractive property of Gauss-Newton algorithm, which leads to its global convergence as long as an appropriate line search algorithm is used.

On the other hand, the corrected step of the proposed algorithm is not always gradient descent. To see this, we take the inner product of (20) with $\Delta\bar{\theta}_J$ and get

$$\Delta\bar{\theta}_J \left([\hat{\mathbf{J}}]_{iK}[\hat{\mathbf{J}}]_{iJ}\right) \Delta\bar{\theta}_K = -\Delta\bar{\theta}_J \left(r_i(\boldsymbol{\theta}^0)[\hat{\mathbf{J}}]_{iJ}\right). \tag{24}$$

Again, if $[\hat{\mathbf{J}}]_{iK}$ is full-rank, the left hand side is ≥ 0. This implies:

$$\Rightarrow \Delta\bar{\theta}_J \left(r_i(\boldsymbol{\theta}^0)[\hat{\mathbf{J}}]_{iJ}\right) \leq 0 \tag{25}$$

$$\Rightarrow \Delta\bar{\theta}_J \left(r_i(\boldsymbol{\theta}^0)\left\{[\mathbf{J}]_{iJ} + \frac{1}{2} r_{i,JL}(\boldsymbol{\theta}^0)\Delta\hat{\theta}_L\right\}\right) \leq 0 \tag{26}$$

$$\Rightarrow \Delta\bar{\theta}_J \left(r_i(\boldsymbol{\theta}^0)[\mathbf{J}]_{iJ}\right) + \Delta\bar{\theta}_J \left(r_i(\boldsymbol{\theta}^0)\left\{\frac{1}{2} r_{i,JL}(\boldsymbol{\theta}^0)\Delta\hat{\theta}_L\right\}\right) \leq 0. \tag{27}$$

The first term is the gradient along the corrected step. However, because the sign of the second term is not clear, the corrected step can be in a descent or ascent direction.

scalar

$\min_{\underset{\sim}{\theta}} f(\underset{\sim}{\theta})$

Gradient $\frac{\partial f}{\partial \theta_J} = 0$ — J equations

$\Downarrow$ Newton method (second deriv.) on J equations.

Quasi-newton methods

e.g. BFGS.

$f = \frac{1}{2} r_i r_i$

$r_i(\underset{\sim}{\theta})$ $\quad i = 1$ to n

$r_i(\underset{\sim}{\theta}) = y_i - m(x_i, \underset{\sim}{\theta})$

→ Standard Newton (second deriv.)

→ Gauss-Newton (first deriv. only)

→ Halley's inspired (Corrected Gauss Newton) (second deriv.)

Thus, the right hand side is always negative, which is the gradient along the step direction:

$$\Delta\theta_J \left(r_i(\boldsymbol{\theta}^0)[\mathbf{J}]_{iJ}\right) \leq 0. \tag{23}$$

This is an attractive property of Gauss-Newton algorithm, which leads to its global convergence as long as an appropriate line search algorithm is used.

On the other hand, the corrected step of the proposed algorithm is not always gradient descent. To see this, we take the inner product of (20) with $\Delta\bar{\theta}_J$ and get

$$\Delta\bar{\theta}_J \left([\hat{\mathbf{J}}]_{iK}[\hat{\mathbf{J}}]_{iJ}\right) \Delta\bar{\theta}_K = -\Delta\bar{\theta}_J \left(r_i(\boldsymbol{\theta}^0)[\hat{\mathbf{J}}]_{iJ}\right). \tag{24}$$

Again, if $[\hat{\mathbf{J}}]_{iK}$ is full-rank, the left hand side is ≥ 0. This implies:

$$\Rightarrow \Delta\bar{\theta}_J \left(r_i(\boldsymbol{\theta}^0)[\hat{\mathbf{J}}]_{iJ}\right) \leq 0 \tag{25}$$

$$\Rightarrow \Delta\bar{\theta}_J \left(r_i(\boldsymbol{\theta}^0)\left\{[\mathbf{J}]_{iJ} + \frac{1}{2} r_{i,JL}(\boldsymbol{\theta}^0)\Delta\hat{\theta}_L\right\}\right) \leq 0 \tag{26}$$

$$\Rightarrow \Delta\bar{\theta}_J \left(r_i(\boldsymbol{\theta}^0)[\mathbf{J}]_{iJ}\right) + \Delta\bar{\theta}_J \left(r_i(\boldsymbol{\theta}^0)\left\{\frac{1}{2} r_{i,JL}(\boldsymbol{\theta}^0)\Delta\hat{\theta}_L\right\}\right) \leq 0. \tag{27}$$

The first term is the gradient along the corrected step. However, because the sign of the second term is not clear, the corrected step can be in a descent or ascent direction.

N
GN
CGN
$\}$ $\underset{\sim}{\theta}^n \rightarrow \Delta\underset{\sim}{\theta}$

line search

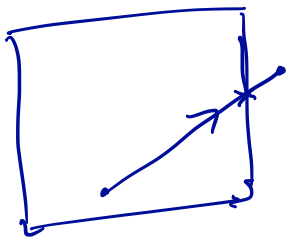
$\alpha \Delta\underset{\sim}{\theta}$

α: scalar

Find α to have best result.

Constraints

(later to consider)



Thus, the right hand side is always negative, which is the gradient along the step direction:

$$\Delta\theta_J \left(r_i(\boldsymbol{\theta}^0)[\mathbf{J}]_{iJ}\right) \leq 0. \tag{23}$$

This is an attractive property of Gauss-Newton algorithm, which leads to its global convergence as long as an appropriate line search algorithm is used.

On the other hand, the corrected step of the proposed algorithm is not always gradient descent. To see this, we take the inner product of (20) with $\Delta\bar{\theta}_J$ and get

$$\Delta\bar{\theta}_J \left([\hat{\mathbf{J}}]_{iK}[\hat{\mathbf{J}}]_{iJ}\right) \Delta\bar{\theta}_K = -\Delta\bar{\theta}_J \left(r_i(\boldsymbol{\theta}^0)[\hat{\mathbf{J}}]_{iJ}\right). \tag{24}$$

Again, if $[\hat{\mathbf{J}}]_{iK}$ is full-rank, the left hand side is ≥ 0. This implies:

$$\Rightarrow \Delta\bar{\theta}_J \left(r_i(\boldsymbol{\theta}^0)[\hat{\mathbf{J}}]_{iJ}\right) \leq 0 \tag{25}$$

$$\Rightarrow \Delta\bar{\theta}_J \left(r_i(\boldsymbol{\theta}^0)\left\{[\mathbf{J}]_{iJ} + \frac{1}{2} r_{i,JL}(\boldsymbol{\theta}^0)\Delta\hat{\theta}_L\right\}\right) \leq 0 \tag{26}$$

$$\Rightarrow \Delta\bar{\theta}_J \left(r_i(\boldsymbol{\theta}^0)[\mathbf{J}]_{iJ}\right) + \Delta\bar{\theta}_J \left(r_i(\boldsymbol{\theta}^0)\left\{\frac{1}{2} r_{i,JL}(\boldsymbol{\theta}^0)\Delta\hat{\theta}_L\right\}\right) \leq 0. \tag{27}$$

The first term is the gradient along the corrected step. However, because the sign of the second term is not clear, the corrected step can be in a descent or ascent direction.