Numerical Optimization

Lecture Notes #23
Nonlinear Least Squares Problems — Algorithms

Peter Blomgren,
⟨blomgren.peter@gmail.com⟩

Department of Mathematics and Statistics
Dynamical Systems Group
Computational Sciences Research Center
San Diego State University
San Diego, CA 92182-7720

http://terminus.sdsu.edu/

Fall 2018

Outline

The Nonlinear Least Squares Problem

Problem: Nonlinear Least Squares

$$\bar{\mathbf{x}}^* = \arg\min_{\bar{\mathbf{x}}\in\mathbb{R}^n} [f(\bar{\mathbf{x}})] = \arg\min_{\bar{\mathbf{x}}\in\mathbb{R}^n} \left[\frac{1}{2} \sum_{j=1}^{m} r_j(\bar{\mathbf{x}})^2 \right], \quad m \geq n,$$

where the **residuals** $r_j(\bar{\mathbf{x}})$ are of the form $r_j(\bar{\mathbf{x}}) = y_j - \Phi(\bar{\mathbf{x}}; t_j)$. Here, y_j are the **measurements** taken at the **locations/times** t_j, and $\Phi(\bar{\mathbf{x}}; t_j)$ is our **model**.

$$\textbf{The Jacobian} \quad J(\bar{\mathbf{x}}) = \left[\frac{\partial r_j(\bar{\mathbf{x}})}{\partial x_i} \right]_{\substack{j=1,2,\ldots,m \\ i=1,2,\ldots,n}}$$

$$\textbf{The Gradient} \quad \nabla f(\bar{\mathbf{x}}) = \sum_{j=1}^{m} r_j(\bar{\mathbf{x}}) \nabla r_j(\bar{\mathbf{x}}) = J(\bar{\mathbf{x}})^T \bar{\mathbf{r}}(\bar{\mathbf{x}})$$

$$\textbf{The Hessian} \quad \nabla^2 f(\bar{\mathbf{x}}) = J(\bar{\mathbf{x}})^T J(\bar{\mathbf{x}}) + \sum_{j=1}^{m} r_j(\bar{\mathbf{x}}) \nabla^2 r_j(\bar{\mathbf{x}})$$

The Nonlinear Least Squares Problem: Algorithms

We now turn our attention to the solution of the nonlinear least squares problem.

Of course, we could just use our unconstrained nonlinear minimization methods as-is. However, as always we would like to implement the most efficient algorithms possible. In order to achieve this, we should take the **special structure** of the **gradient** and **Hessian** into consideration.

The first algorithm carries the name of two of the giants of mathematics — Gauss[1] and Newton[2]!

[1] — Johann Carl Friedrich Gauss (30 Apr 1777 – 23 Feb 1855).
[2] — Sir Isaac Newton (4 Jan 1643 – 31 Mar 1727).

The Gauss-Newton Method

The Gauss-Newton method can be viewed as a modification of Newton's method with line search.

Instead of generating the Newton search directions $\bar{\mathbf{p}}_k^{\mathrm{N}}$ as solutions of the linear systems

$$\left[\nabla^2 f(\bar{\mathbf{x}}_k)\right] \bar{\mathbf{p}}_k^{\mathrm{N}} = -\nabla f(\bar{\mathbf{x}}_k),$$

we use the particular form of the Hessian, and exclude the second order term from it. Thus we get the Gauss-Newton search directions $\bar{\mathbf{p}}_k^{\mathrm{GN}}$ as the solutions of the linear systems

$$\left[J(\bar{\mathbf{x}}_k)^T J(\bar{\mathbf{x}}_k)\right] \bar{\mathbf{p}}_k^{\mathrm{GN}} = -J(\bar{\mathbf{x}}_k)^T \bar{\mathbf{r}}(\bar{\mathbf{x}}_k).$$

This simple modification has many advantages over the standard Newton's method.

The Gauss-Newton Method: Advantages 1 of 2

1. The approximation

$$\nabla^2 f(\bar{\mathbf{x}}_k) \approx J(\bar{\mathbf{x}}_k)^T J(\bar{\mathbf{x}}_k)$$

saves the work of computing the m individual Hessians $\nabla^2 r_j(\bar{\mathbf{x}})$. If we compute the Jacobian $J(\bar{\mathbf{x}}_k)$ in the process if evaluating the gradient $\nabla f(\bar{\mathbf{x}}) = J(\bar{\mathbf{x}}_k)^T \bar{\mathbf{r}}(\bar{\mathbf{x}}_k)$, then this approximation is essentially "free."

2. In many situations the term $J(\bar{\mathbf{x}}_k)^T J(\bar{\mathbf{x}}_k)$ dominates $\sum_{j=1}^m r_j(\bar{\mathbf{x}}) \nabla^2 r_j(\bar{\mathbf{x}})$, so that the Gauss-Newton method gives performance similar to that of Newton's method, even when the latter term is neglected. This happens when $r_j(\bar{\mathbf{x}})$ are small (the **small residual case**), or when $r_j(\bar{\mathbf{x}})$ are nearly linear, so that $\|\nabla^2 r_j(\bar{\mathbf{x}})\|$ is small. **In practice**, many least-squares problems fall into the first category, and rapid local convergence of Gauss-Newton is observed.

The Gauss-Newton Method: Advantages 2 of 2

3. When $J(\bar{\mathbf{x}}_k)$ has full rank and $\nabla f(\bar{\mathbf{x}}_k) \neq \bar{\mathbf{0}}$, the Gauss-Newton direction $\bar{\mathbf{p}}_k^{\mathrm{GN}}$ is a **descent direction**, and works for line-search. We have

$$\begin{aligned} [\bar{\mathbf{p}}_k^{\mathrm{GN}}]^T \nabla f(\bar{\mathbf{x}}) &= [\bar{\mathbf{p}}_k^{\mathrm{GN}}]^T J(\bar{\mathbf{x}}_k)^T \bar{\mathbf{r}}_k = -[\bar{\mathbf{p}}_k^{\mathrm{GN}}]^T J(\bar{\mathbf{x}}_k)^T J(\bar{\mathbf{x}}_k) \bar{\mathbf{p}}_k^{\mathrm{GN}} \\ &= -\|J(\bar{\mathbf{x}}_k)\bar{\mathbf{p}}_k^{\mathrm{GN}}\|_2^2 \leq 0 \end{aligned}$$

where the final inequality is strict unless we are in a stationary point.

4. The Gauss-Newton equations are very similar to the **normal equations** for the linear least squares problem. $\bar{\mathbf{p}}_k^{\mathrm{GN}}$ is the solution of the linear least squares problem

$$\bar{\mathbf{p}}_k^{\mathrm{GN}} = \arg\min_{\bar{\mathbf{p}} \in \mathbb{R}^n} \|J(\bar{\mathbf{x}}_k)\bar{\mathbf{p}} + \bar{\mathbf{r}}(\bar{\mathbf{x}}_k)\|_2^2$$

Therefore, we can use our knowledge of solutions to the linear least squares problem to find the Gauss-Newton search direction.

The Gauss-Newton–Linear-Least-Squares Connection

The connection to the linear least squares problem shows another motivation for the Gauss-Newton step.

Instead of building a quadratic model of the objective $f(\bar{\mathbf{x}})$, we form a linear model of the vector function

$$\bar{\mathbf{r}}(\bar{\mathbf{x}} + \bar{\mathbf{p}}) \approx \bar{\mathbf{r}}(\bar{\mathbf{x}}) + J(\bar{\mathbf{x}})\bar{\mathbf{p}}$$

The step $\bar{\mathbf{p}}^{\mathrm{GN}}$ is obtained by using this model in the expression $f(\bar{\mathbf{x}}) = \frac{1}{2}\|\bar{\mathbf{r}}(\bar{\mathbf{x}})\|_2^2$, and minimizing over $\bar{\mathbf{p}}$.

Searching in the Gauss-Newton Direction — Convergence

Once we have the Gauss-Newton direction, we perform a line-search thus identifying a step α_k which satisfies the Wolfe / Strong Wolfe / Goldstein conditions.

The theory from our discussion on line-search can be applied to guarantee **global convergence** of the Gauss-Newton method. For the proof, we need the following

Assumption

The **singular values** of the Jacobians $J(\bar{\mathbf{x}})$ are bounded away from zero, *i.e.*

$$\|J(\bar{\mathbf{x}})\bar{\mathbf{z}}\|_2 \geq \gamma\|\bar{\mathbf{z}}\|_2, \quad \gamma > 0$$

for all $\bar{\mathbf{x}}$ in a neighborhood of the level set

$$\mathcal{L}(\bar{\mathbf{x}}_0) = \{\bar{\mathbf{x}} \in \mathbb{R}^n \,:\, f(\bar{\mathbf{x}}) \leq f(\bar{\mathbf{x}}_0)\}$$

where $\bar{\mathbf{x}}_0$ is the starting point.

Gauss-Newton: Convergence Theorem

Theorem

Suppose that each residual function $r_j(\bar{\mathbf{x}})$ is Lipschitz continuously differentiable in a neighborhood of $\mathcal{N}$ of the level set

$$\mathcal{L}(\bar{\mathbf{x}}_0) = \{\bar{\mathbf{x}} \in \mathbb{R}^n \,:\, f(\bar{\mathbf{x}}) \leq f(\bar{\mathbf{x}}_0)\},$$

and that the Jacobians satisfy the uniform full-rank condition

$$\|J(\bar{\mathbf{x}})\bar{\mathbf{z}}\|_2 \geq \gamma\|\bar{\mathbf{z}}\|_2, \quad \gamma > 0.$$

Then, if the iterates $\{\bar{\mathbf{x}}_k\}$ are generated by the Gauss-Newton method with step lengths α_k that satisfy the Wolfe conditions, we have

$$\lim_{k\to\infty} J(\bar{\mathbf{x}}_k)^T\bar{\mathbf{r}}_k = 0.$$

Gauss-Newton: Breaking the Convergence Theorem

If the Jacobian is **rank deficient**, *i.e.*

$$\|J(\bar{\mathbf{x}})\bar{\mathbf{z}}\|_2 \not\geq \gamma\|\bar{\mathbf{z}}\|_2, \quad \gamma > 0,$$

then the coefficient matrix $J(\bar{\mathbf{x}}_k)^T J(\bar{\mathbf{x}}_k)$ is singular. However, the system $J(\bar{\mathbf{x}}_k)^T J(\bar{\mathbf{x}}_k)\bar{\mathbf{p}}_k^{\text{GN}} = -J(\bar{\mathbf{x}}_k)^T\nabla f(\bar{\mathbf{x}}_k)$ still has a least-squares solution *since it is equivalent to the minimization problem*

$$\bar{\mathbf{p}}_k^{\text{GN}} = \arg\min_{\bar{\mathbf{p}}\in\mathbb{R}^n} \|J(\bar{\mathbf{x}}_k)\bar{\mathbf{p}} + \bar{\mathbf{r}}(\bar{\mathbf{x}}_k)\|_2^2.$$

In this case, there are infinitely many solutions of the form

$$\bar{\mathbf{x}}^* = \sum_{i\,:\,\sigma_i\neq 0} \frac{\bar{\mathbf{u}}_i^T\bar{\mathbf{r}}_0}{\sigma_i} + \sum_{i\,:\,\sigma_i=0} \tau_i\bar{\mathbf{v}}_i.$$

The convergence result falls since the search direction may become perpendicular to $\nabla f(\bar{\mathbf{x}}_k)$ (thus the Zoutendijk condition $\sum_{k=0}^{\infty}\cos^2\theta_k\|\nabla f(\bar{\mathbf{x}}_k)\|^2 < \infty$ does not show convergence).

Gauss-Newton: Local Convergence Rate — 1 of 2

The convergence rate of Gauss-Newton depends on how much the term $J(\bar{\mathbf{x}}_k)^T J(\bar{\mathbf{x}}_k)$ dominates the neglected term in the Hessian.

Near $\bar{\mathbf{x}}^*$, the step $\alpha_k = 1$ will be accepted, and we have

$$\begin{aligned}\underbrace{\bar{\mathbf{x}}_k + \bar{\mathbf{p}}_k^{\text{GN}}}_{\bar{\mathbf{x}}_{k+1}} - \bar{\mathbf{x}}^* &= \bar{\mathbf{x}}_k - \bar{\mathbf{x}}^* - \left[J(\bar{\mathbf{x}}_k)^T J(\bar{\mathbf{x}}_k)\right]^{-1}\nabla f(\bar{\mathbf{x}}_k) \\ &= \left[J(\bar{\mathbf{x}}_k)^T J(\bar{\mathbf{x}}_k)\right]^{-1}\left[J(\bar{\mathbf{x}}_k)^T J(\bar{\mathbf{x}}_k)(\bar{\mathbf{x}}_k - \bar{\mathbf{x}}^*) + \nabla f(\bar{\mathbf{x}}^*) - \nabla f(\bar{\mathbf{x}}_k)\right].\end{aligned}$$

If we let $H_r(\bar{\mathbf{x}}) = \sum_{j=1}^m r_j(\bar{\mathbf{x}})\nabla^2 r_j(\bar{\mathbf{x}})$ represent the neglected term in the Hessian, we can show (using Taylor's Theorem) that

$$\|\bar{\mathbf{x}}_{k+1} - \bar{\mathbf{x}}^*\| \leq \left\|\left[J(\bar{\mathbf{x}}^*)^T J(\bar{\mathbf{x}}^*)\right]^{-1} H_r(\bar{\mathbf{x}}^*)\right\| \cdot \|\bar{\mathbf{x}}_k - \bar{\mathbf{x}}^*\| + \mathcal{O}\left(\|\bar{\mathbf{x}}_k - \bar{\mathbf{x}}^*\|^2\right)$$

Gauss-Newton: Local Convergence Rate 2 of 2

We have the result

$$\|\bar{\mathbf{x}}_{k+1} - \bar{\mathbf{x}}^*\| \leq \underbrace{\left\| \left[J(\bar{\mathbf{x}}^*)^T J(\bar{\mathbf{x}}^*) \right]^{-1} H_r(\bar{\mathbf{x}}^*) \right\|}_{s(\bar{\mathbf{x}}^*)} \cdot \|\bar{\mathbf{x}}_k - \bar{\mathbf{x}}^*\| + \mathcal{O}\left(\|\bar{\mathbf{x}}_k - \bar{\mathbf{x}}^*\|^2 \right)$$

- Unless $s(\bar{\mathbf{x}}^*) < 1$, we cannot expect Gauss-Newton to converge at all.
- In the small-residual case, it is usually true that $s(\bar{\mathbf{x}}^*) \ll 1$, and we have very rapid convergence[1] of the Gauss-Newton method.
- When $H_r(\bar{\mathbf{x}}^*) = 0$, the rate of convergence is quadratic.

[1] Note that the convergence rate is actually **linear**, but we do not "feel" the linear term until $\|\bar{\mathbf{x}}_k - \bar{\mathbf{x}}^*\| < s(\bar{\mathbf{x}}^*)$. If $s(\bar{\mathbf{x}}^*)$ is small enough (smaller than the required tolerance on $\bar{\mathbf{x}}_N \approx \bar{\mathbf{x}}^*$), then this will not slow down the method.

The Levenberg-Marquardt Method: Introduction

Levenberg and Marquardt are slightly less famous than Gauss and Newton (at least to non-optimizers?!)

The Levenberg-Marquardt method is the **trust-region equivalent** of the Gauss-Newton method.

The Levenberg-Marquardt method avoids one of the weaknesses of the Gauss-Newton method, the (global convergence) behavior when the Jacobian $J(\bar{\mathbf{x}})$ is rank-deficient (or nearly rank-deficient) [see slide 11].

Since the second-order Hessian component is still ignored, the local convergence properties of the LM- and GN-methods are similar.

The original description of the LM-method (first published in 1944) did not make the connection with the trust-region concept.

The connection was made by Moré in 1978.

The Levenberg-Marquardt Method

For a spherical trust region, the subproblem to be solved at iteration $\#k$ is

$$\bar{\mathbf{p}}_k^{\text{LM}} = \arg\min_{\bar{\mathbf{p}} \in \mathbb{R}^n} \frac{1}{2} \|J(\bar{\mathbf{x}}_k)\bar{\mathbf{p}} + \bar{\mathbf{r}}_k\|_2^2, \quad \text{subject to } \|\bar{\mathbf{p}}\| \leq \Delta_k.$$

This corresponds to the model function

$$m_k(\bar{\mathbf{p}}) = \frac{1}{2}\|\bar{\mathbf{r}}_k\|^2 + \bar{\mathbf{p}}^T J(\bar{\mathbf{x}}_k)^T \bar{\mathbf{r}}_k + \frac{1}{2}\bar{\mathbf{p}}^T J(\bar{\mathbf{x}}_k)^T J(\bar{\mathbf{x}}_k)\bar{\mathbf{p}},$$

where we can identify the Hessian approximation as

$$B_k = J(\bar{\mathbf{x}}_k)^T J(\bar{\mathbf{x}}_k).$$

Levenberg-Marquardt vs. Gauss-Newton

If the Gauss-Newton step $\bar{\mathbf{p}}_k^{\text{GN}}$ lies inside the trust region, *i.e.* $\|\bar{\mathbf{p}}_k^{\text{GN}}\| \leq \Delta_k$, then $\bar{\mathbf{p}}_k^{\text{LM}} = \bar{\mathbf{p}}_k^{\text{GN}}$.

Otherwise, there is a $\lambda > 0$ such that $\|\bar{\mathbf{p}}_k^{\text{LM}}\| = \Delta_k$ and

$$\left[J(\bar{\mathbf{x}}_k)^T J(\bar{\mathbf{x}}_k) + \lambda I \right] \bar{\mathbf{p}}_k^{\text{LM}} = -J(\bar{\mathbf{x}}_k)^T \bar{\mathbf{r}}_k.$$

This can be **interpreted** as the normal equations for the linear least squares problem

$$\bar{\mathbf{p}}_k = \arg\min_{\bar{\mathbf{p}} \in \mathbb{R}^n} \frac{1}{2} \left\| \begin{bmatrix} J(\bar{\mathbf{x}}_k) \\ \sqrt{\lambda} I \end{bmatrix} \bar{\mathbf{p}} + \begin{bmatrix} \bar{\mathbf{r}}_k \\ 0 \end{bmatrix} \right\|^2.$$

As in the Gauss-Newton case, this equivalence yields a way of solving the subproblem without computing the matrix-matrix product $J(\bar{\mathbf{x}}_k)^T J(\bar{\mathbf{x}}_k)$ and its Cholesky factorization.

The Levenberg-Marquardt Method: Implementation 1/3

In order to find the value of λ which gives $\|\bar{\mathbf{p}}_k^{\text{LM}}\| = \Delta_k$, we can apply the root-finding algorithm from Lecture#9 **(Nearly Exact Solutions to the Subproblem)**.

However, due to the special structure $B_k = J(\bar{\mathbf{x}}_k)^T J(\bar{\mathbf{x}}_k)$, the approximate Hessian is always positive semi-definite, hence the Cholesky factorization cannot break down for any $\lambda^{(j)} > 0$.

The structure of B_k can further be exploited to get the Cholesky factorization

$$R_\lambda^T R_\lambda = J^T J + \lambda I,$$

by means of the **QR-factorization**

$$\begin{bmatrix} R_\lambda \\ 0 \end{bmatrix} = Q_\lambda^T \begin{bmatrix} J \\ \sqrt{\lambda} I \end{bmatrix}, \quad \begin{cases} Q_\lambda \text{ orthogonal} \\ R_\lambda \text{ upper triangular} \end{cases}$$

The Levenberg-Marquardt Method: Implementation 2/3

We defer the discussion of the most efficient way of implementing the QR-factorization to another forum (like `Math 543`).

Least-squares problems are ofter poorly scaled. The separation of scales between different parameters can easily be several orders of magnitude, *e.g.* $x_1 \sim 10^{14} x_n$.

This is a scenario where the use of scale factors, and ellipsoidal trust-regions $\|D\bar{\mathbf{p}}\| \le \Delta_k$, where the diagonal matrix $D = \text{diag}(d_1, d_2, \ldots, d_n)$ captures the scales of the various parameters [see Lecture#10] **(Trust-Region Methods: Global Convergence and Enhancements)**.

The Levenberg-Marquardt Method: Implementation 3/3

In the scaled case, the trust-region subproblem, and the two characterizations of the solution become:

$$\bar{\mathbf{p}}_k^{\text{LM}} = \arg\min_{\bar{\mathbf{p}} \in \mathbb{R}^n} \frac{1}{2} \|J(\bar{\mathbf{x}}_k)\bar{\mathbf{p}} + \bar{\mathbf{r}}_k\|_2^2, \quad \text{subject to } \|\mathbf{D_k}\bar{\mathbf{p}}\| \le \Delta_k,$$

$$\left[J(\bar{\mathbf{x}}_k)^T J(\bar{\mathbf{x}}_k) + \lambda \mathbf{D_k^2}\right] \bar{\mathbf{p}}_k^{\text{LM}} = -J(\bar{\mathbf{x}}_k)\bar{\mathbf{r}}_k,$$

$$\bar{\mathbf{p}}_k = \arg\min_{\bar{\mathbf{p}} \in \mathbb{R}^n} \frac{1}{2} \left\| \begin{bmatrix} J(\bar{\mathbf{x}}_k) \\ \sqrt{\lambda}\mathbf{D_k} \end{bmatrix} \bar{\mathbf{p}} + \begin{bmatrix} \bar{\mathbf{r}}_k \\ 0 \end{bmatrix} \right\|^2.$$

The convergence properties are preserved even though D_k is allowed to change from iteration-to-iteration, within certain bounds.

If $D_k^2 = \texttt{extract\_diagonal}(J_k^T J_k)$, then the algorithm is invariant under diagonal scaling of $\bar{\mathbf{x}}$.

Large-Residual Problems 1 of 3

In all the previous discussion we have assumed, explicitly or implicitly, that the approximation

$$\nabla^2 f(\bar{\mathbf{x}}) \approx J(\bar{\mathbf{x}})^T J(\bar{\mathbf{x}}),$$

works well. However, when the neglected second order part of the Hessian

$$\sum_{j=1}^{m} r_j(\bar{\mathbf{x}}) \nabla^2 r_j(\bar{\mathbf{x}}),$$

is large, the approximation does not produce good results.

The question looms large: **What should be done in this case?**

Large-Residual Problems 2 of 3

Solution #1: Head-in-the-sand

In statistical applications the importance of such problems are (sometimes) downplayed. The argument is that if the residuals are large at the solution, then the model is not good enough.

However, often large residuals are caused by *outliers* — caused by anomalous readings. Solving the least-squares problem may help us identify the outliers. These can then be treated in an appropriate way...

Both the Gauss-Newton and Levenberg-Marquardt methods perform poorly in the large-residual case. The local convergence is linear and thus slower than general-purpose algorithms for unconstrained optimization.

Large-Residual Problems 3 of 3

Solution #2: Hybrid Methods — Type #1

In a hybrid method, we start with Gauss-Newton or Levenberg-Marquardt (which have much lower cost/iteration compared to general-purpose methods), and if it turns out that the residuals at the solution are large we *switch* to a quasi-Newton or Newton method.

Solution #3: Hybrid Methods — Type #2

It is also possible to combine Gauss-Newton and quasi-Newton ideas to maintain approximations to the second order part of the Hessian. *I.e.* we maintain a sequence $S_k \approx \sum_{j=1}^{m} r_j(\bar{\mathbf{x}}_k)\nabla^2 r_j(\bar{\mathbf{x}}_k)$ (think BFGS-style), and then use the overall Hessian approximation $B_k = J_k^T J_k + S_k$ in a trust-region or line-search calculation.

This can get "somewhat" complex...

Index