Least-Squares Problems

Oscar Dalmau
dalmau@cimat.mx

Centro de Investigación en Matemáticas
CIMAT A.C. Mexico

March 2016

Outline

1. Least Square Problems

2. Gauss Newton Method

Least Square

Least-square Problem

$$\boldsymbol{x}^* = \arg\min_{\boldsymbol{x}\in\mathbb{R}^n} f(\boldsymbol{x})$$
$$f(\boldsymbol{x}) = \frac{1}{2}\sum_{j=1}^{m} r_j(\boldsymbol{x})^2$$

where $r_j(\boldsymbol{x}) : \mathbb{R}^n \to \mathbb{R}$, $j = 1, 2, \cdots, m$ are smooth functions.

- $r_j(\boldsymbol{x})$, $j = 1, 2, \cdots, m$ are referred as *residuals*, ie $r_j(\boldsymbol{x}) = y_j - \phi(\boldsymbol{x}; t_j)$; $\phi(\boldsymbol{x}; t_j)$ is a model
- It is assumed that $m \geq n$

Least-square Problem: Example

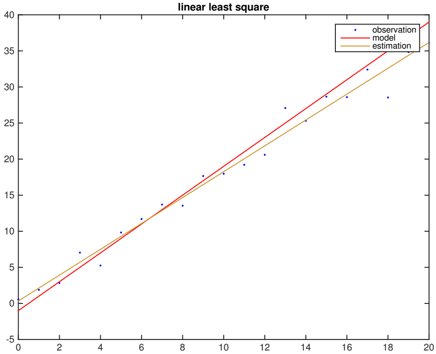
Linear Least-square

- $r_j(\boldsymbol{\beta}) = y_j - \phi(\boldsymbol{\beta}; t_j)$; $j = 1, 2, \cdots, m$
- $\phi(\boldsymbol{\beta}; t) = \beta_0 + \beta_1 t$; $\boldsymbol{\beta} = [\beta_0, \beta_1]^T$.
- $f(\boldsymbol{\beta}) = \frac{1}{2} \sum_{j=1}^{m} r_j(\boldsymbol{\beta})^2 = \frac{1}{2} \sum_{j=1}^{m} (y_j - \beta_0 - \beta_1 t_j)^2$

Least-square Problem: Example

Linear Least-square

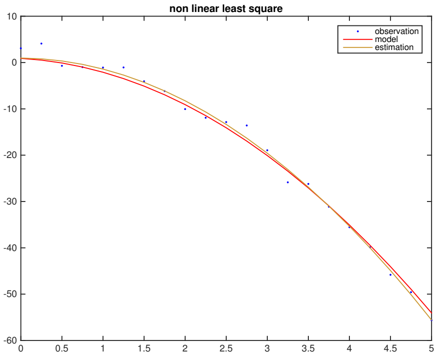
- $\phi(\boldsymbol{\beta}; t) = \beta_0 + \beta_1 t$; $\boldsymbol{\beta} = [\beta_0, \beta_1]^T$.
- $y = 2 * t - 1 + \eta$; $\eta \sim \mathcal{N}(0, 2)$, with $t = 0, 1, \cdots, 20$



Non Least-square Problem: Example

Non Linear Least-square

- $\phi(\boldsymbol{\beta}; t) = \beta_0 + \beta_1 t + \beta_2 t^2 + \beta_3 e^{-\beta_4 t}$
- $r_j(\boldsymbol{\beta}) = t_j - \phi(\boldsymbol{\beta}; t_j)$; $j = 1, 2, \cdots, m$; $\boldsymbol{\beta} = [\beta_0, \beta_1, \cdots, \beta_4]^T$.



Least Square

Least-square Problem

Defining $\boldsymbol{r}(\boldsymbol{x}) = [r_1(\boldsymbol{x}), r_2(\boldsymbol{x}), \cdots, r_m(\boldsymbol{x})]^T$

$$f(\boldsymbol{x}) = \frac{1}{2}\sum_{j=1}^{m} r_j(\boldsymbol{x})^2 = \frac{1}{2}\|\boldsymbol{r}(\boldsymbol{x})\|_2^2 = \frac{1}{2}\boldsymbol{r}(\boldsymbol{x})^T\boldsymbol{r}(\boldsymbol{x})$$

Least Square: Gradient

Then

$$
\begin{aligned}
Df(\boldsymbol{x}) &= \frac{1}{2}\left(\boldsymbol{r}(\boldsymbol{x})^T D\boldsymbol{r}(\boldsymbol{x}) + \boldsymbol{r}(\boldsymbol{x})^T D\boldsymbol{r}(\boldsymbol{x})\right) = \boldsymbol{r}(\boldsymbol{x})^T D\boldsymbol{r}(\boldsymbol{x}) \\
\nabla f(\boldsymbol{x}) &= D\boldsymbol{r}(\boldsymbol{x})^T \boldsymbol{r}(\boldsymbol{x}) = \mathbf{J}(\boldsymbol{x})^T \boldsymbol{r}(\boldsymbol{x})
\end{aligned}
$$

where $\mathbf{J}$ is the Jacobian of $\boldsymbol{r} : \mathbb{R}^n \to \mathbb{R}^m$ and

$$
\begin{aligned}
\mathbf{J}(\boldsymbol{x}) &= [J_{ij}]_{\substack{i=1,..,m \\ j=1,..,n}} \\
&= [\frac{\partial r_i(\boldsymbol{x})}{\partial x_j}]_{\substack{i=1,..,m \\ j=1,..,n}}
\end{aligned}
$$

Least Square: Jacobian

$$
\begin{aligned}
\mathbf{J}(\boldsymbol{x}) &= [J_{ij}]_{\substack{i=1,..,m \\ j=1,..,n}} \\
&= [\frac{\partial r_i(\boldsymbol{x})}{\partial x_j}]_{\substack{i=1,..,m \\ j=1,..,n}} \\
&= \begin{bmatrix} \nabla r_1(x)^T \\ \nabla r_2(x)^T \\ \vdots \\ \nabla r_m(x)^T \end{bmatrix} \\
\mathbf{J}(\boldsymbol{x})^T &= [\nabla r_1(x), \nabla r_2(x), \cdots, \nabla r_m(x)]
\end{aligned}
$$

Least Square: Gradient

$$
\begin{aligned}
\nabla f(\boldsymbol{x}) &= \mathbf{J}(\boldsymbol{x})^T \boldsymbol{r}(\boldsymbol{x}) \\
&= [\nabla r_1(x), \nabla r_2(x), \cdots, \nabla r_m(x)] \begin{bmatrix} r_1(x) \\ r_2(x) \\ \vdots \\ r_m(x) \end{bmatrix} \\
&= \sum_{i=1}^{m} r_i(x) \nabla r_i(x)
\end{aligned}
$$

Least Square: Hessian

Gradient

$$\nabla f(\boldsymbol{x}) = \sum_{i=1}^{m} r_i(x)\nabla r_i(x)$$

Hessian

$$\begin{aligned}
\nabla^2 f(\boldsymbol{x}) &= \sum_{i=1}^{m} \nabla r_i(x)\nabla r_i(x)^T + r_i(x)\nabla^2 r_i(x) \\
&= \mathbf{J}(\boldsymbol{x})^T\mathbf{J}(\boldsymbol{x}) + \sum_{i=1}^{m} r_i(x)\nabla^2 r_i(x) \\
&= \mathbf{J}(\boldsymbol{x})^T\mathbf{J}(\boldsymbol{x}) + S(\boldsymbol{x})
\end{aligned}$$

Linear Least Square

Least-square Problem

$$
\begin{aligned}
\boldsymbol{r}(\boldsymbol{x}) &= \mathbf{J}\boldsymbol{x} - \boldsymbol{b} \\
f(\boldsymbol{x}) &= \frac{1}{2}\|\mathbf{J}\boldsymbol{x} - \boldsymbol{b}\|_2^2 \\
\mathbf{J}(\boldsymbol{x}) &= D\boldsymbol{r}(\boldsymbol{x}) = \mathbf{J} \\
\nabla f(\boldsymbol{x}) &= \mathbf{J}(\boldsymbol{x})^T \boldsymbol{r}(\boldsymbol{x}) = \mathbf{J}^T(\mathbf{J}\boldsymbol{x} - \boldsymbol{b}) = \mathbf{J}^T\mathbf{J}\boldsymbol{x} - \mathbf{J}^T\boldsymbol{b} \\
\nabla^2 f(\boldsymbol{x}) &= \mathbf{J}^T\mathbf{J} = \mathbf{J}(\boldsymbol{x})^T\mathbf{J}(\boldsymbol{x}) + \mathbf{0}
\end{aligned}
$$

Note:

- $\mathbf{J}(\boldsymbol{x}) = \mathbf{J}$ is a constant matrix
- $\sum_{k=1}^{m} r_k(\boldsymbol{x})\nabla^2 r_k(\boldsymbol{x}) = \mathbf{0}$ due to $\nabla^2 r_k(\boldsymbol{x}) = \mathbf{0}$, ie, $r_k(\boldsymbol{x})$ is affine.

Linear Least Square

Least-square Problem

As

$$\nabla f(\boldsymbol{x}) \quad = \quad \mathbf{J}^T \mathbf{J} \boldsymbol{x} - \mathbf{J}^T \boldsymbol{b}$$

the optimum $\boldsymbol{x}^*$ satisfies

$$\mathbf{J}^T \mathbf{J} \boldsymbol{x} \quad = \quad \mathbf{J}^T \boldsymbol{b}$$

known as *normal equations*.

The Gauss Newton Method is used to solve the problem

$$
\begin{aligned}
\boldsymbol{x}^* &= \arg\min_{\boldsymbol{x}\in\mathbb{R}^n} f(\boldsymbol{x}) \\
f(\boldsymbol{x}) &= \frac{1}{2}\sum_{j=1}^{m} r_j(\boldsymbol{x})^2
\end{aligned}
$$

It exploits the structure of the Hessian $\nabla^2 f(\boldsymbol{x})$

Instead of the standard direction

$$\nabla^2 f(\boldsymbol{x}_k)\boldsymbol{d}_k^N \quad = \quad -\nabla f(\boldsymbol{x}_k)$$

one solves the following system of equation with respect to $\boldsymbol{d}_k^{GN}$

$$\mathbf{J}(\boldsymbol{x}_k)^T\mathbf{J}(\boldsymbol{x}_k)\boldsymbol{d}_k^{GN} \quad = \quad -\mathbf{J}(\boldsymbol{x}_k)^T\boldsymbol{r}(\boldsymbol{x}_k)$$

Gauss Newton Method

1. If $r_k(\boldsymbol{x}) \approx 0$ or $\nabla^2 r_k(\boldsymbol{x}) \approx 0$, $\forall k$ then

$$\nabla^2 f(\boldsymbol{x}) \;\; \approx \;\; \mathbf{J}(\boldsymbol{x})^T \mathbf{J}(\boldsymbol{x})$$

   then, we do not require to compute the individual residual Hessians $\nabla^2 r_k(\boldsymbol{x})$.

2. There are many situation where $\mathbf{J}(\boldsymbol{x})^T \mathbf{J}(\boldsymbol{x})$ dominates the second term. Therefore, $\mathbf{J}(\boldsymbol{x}_k)^T \mathbf{J}(\boldsymbol{x}_k)$ is a close approximation to $\nabla^2 f(\boldsymbol{x}_k)$ and the convergence rate of Gauss-Newton is similar to that of Newton's method.

Gauss Newton Method

1. If $\mathbf{J}_k$ has full rank and the gradient ∇f_k is nonzero, the direction $\boldsymbol{d}^{GN}$ is a descent direction, and therefore a suitable direction for a line search.

$$\begin{aligned} \boldsymbol{d}^{GN^T} \nabla f(\boldsymbol{x}) &= \boldsymbol{d}^{GN^T} \mathbf{J}(\boldsymbol{x}_k)^T \boldsymbol{r}(\boldsymbol{x}_k) \\ &= -\boldsymbol{d}^{GN^T} \mathbf{J}(\boldsymbol{x}_k)^T \mathbf{J}(\boldsymbol{x}_k) \boldsymbol{d}_k^{GN} \\ &= -\|\mathbf{J}(\boldsymbol{x}_k) \boldsymbol{d}_k^{GN}\|_2^2 \leq 0 \end{aligned}$$

What happens when $\mathbf{J}(\boldsymbol{x}_k)\boldsymbol{d}_k^{GN} = 0$?

Gauss Newton Method

1. The final inequality is strict unless $\mathbf{J}(\boldsymbol{x}_k)\boldsymbol{d}_k^{GN} = \mathbf{0}$, in which case we have by the full rank of $\mathbf{J}_k$

$$\begin{aligned}
\mathbf{J}(\boldsymbol{x}_k)^T\mathbf{J}(\boldsymbol{x}_k)\boldsymbol{d}_k^{GN} &= -\mathbf{J}(\boldsymbol{x}_k)^T\boldsymbol{r}(\boldsymbol{x}_k) \\
\mathbf{J}(\boldsymbol{x}_k)^T\mathbf{0} &= -\nabla f(\boldsymbol{x}_k) \\
\nabla f(\boldsymbol{x}_k) &= \mathbf{0}
\end{aligned}$$

then $\boldsymbol{x}_k$ is a stationary point.

Gauss Newton Method

1. The Gauss-Newton arises from the similarity between the equations

$$\mathbf{J}(\boldsymbol{x}_k)^T\mathbf{J}(\boldsymbol{x}_k)\boldsymbol{d}_k^{GN} \quad = \quad -\mathbf{J}(\boldsymbol{x}_k)^T\boldsymbol{r}(\boldsymbol{x}_k)$$

   and the *normal equations* for the linear least-squares problem.

2. The previous connection tells us that $\boldsymbol{d}_k^{GN}$ is in fact the solution of the linear least-squares problem

$$\arg\min_{\boldsymbol{d}} \|\mathbf{J}(\boldsymbol{x}_k)\boldsymbol{d} + \boldsymbol{r}(\boldsymbol{x}_k)\|^2$$

Gauss Newton Method

1. If the QR (with column pivoting) or SVD-based algorithms are used to solve the corresponding linear system

$$\mathbf{J}(\boldsymbol{x}_k)^T\mathbf{J}(\boldsymbol{x}_k)\boldsymbol{d}_k^{GN} \quad = \quad -\mathbf{J}(\boldsymbol{x}_k)^T\boldsymbol{r}(\boldsymbol{x}_k)$$

there is no need to calculate the Hessian approximation $\mathbf{J}(\boldsymbol{x}_k)^T\mathbf{J}(\boldsymbol{x}_k)$ explicitly; we can work directly with the Jacobian $\mathbf{J}(\boldsymbol{x}_k)$.

Gauss Newton Method

1. The linear least-squares problem

$$\arg\min_{\boldsymbol{d}} \|\mathbf{J}(\boldsymbol{x}_k)\boldsymbol{d} + \boldsymbol{r}(\boldsymbol{x}_k)\|^2$$

can be viewed as the linear model for the the vector function $\boldsymbol{r}(\boldsymbol{x}_k + \boldsymbol{d}) \approx \boldsymbol{r}(\boldsymbol{x}_k) + \mathbf{J}(\boldsymbol{x}_k)\boldsymbol{d}$ therefore

$$f(\boldsymbol{x}_k + \boldsymbol{d}) = \frac{1}{2}\|\boldsymbol{r}(\boldsymbol{x}_k + \boldsymbol{d})\|^2 \approx \frac{1}{2}\|\mathbf{J}(\boldsymbol{x}_k)\boldsymbol{d} + \boldsymbol{r}(\boldsymbol{x}_k)\|^2$$

2. Implementations of the Gauss-Newton method usually perform a line search in the direction $\boldsymbol{d}^{GN}$.

Theorem 2.1

Suppose each residual function r_j is Lipschitz continuously differentiable in a neighborhood $\mathcal{N}$ of the bounded level set

$$\mathcal{L} = \{\boldsymbol{x} | f(\boldsymbol{x}) \leq f(\boldsymbol{x}_0)\}$$

where $\boldsymbol{x}_0$ is the starting point for the algorithm, and that the Jacobians $\mathbf{J}(\boldsymbol{x})$ satisfy (the uniform full-rank condition) that there is a constant $\gamma > 0$ such that

$$\|\mathbf{J}(\boldsymbol{x})\boldsymbol{z}\| \geq \gamma \|\boldsymbol{z}\|$$

for all $\boldsymbol{x}$ in a neighborhood $\mathcal{N}$ of the level set $\mathcal{L}$. Then if the iterates $\boldsymbol{x}_k$ are generated by the Gauss-Newton method with step lengths α_k that satisfy the Wolfe conditions, we have

$$\lim_{k \to \infty} \mathbf{J}_k^T \boldsymbol{r}_k = 0.$$

Theorem 2.2

Let $\boldsymbol{r} : \mathbb{R}^n \to \mathbb{R}^m$*, and let* $f(\boldsymbol{x}) = \frac{1}{2}\|\boldsymbol{r}(\boldsymbol{x})\|^2$ *be twice continuously differentiate in an open convex set* Ω*. Assume that* $\mathbf{J}(\boldsymbol{x}) \in Lip_\gamma(\Omega)$ *with* $\|\mathbf{J}(\boldsymbol{x})\| \geq \alpha$ *for all* $\boldsymbol{x} \in \Omega$ *and there exists* $\boldsymbol{x}^* \in \Omega$ *and* $\lambda, \sigma \geq 0$ *such that* $\mathbf{J}(\boldsymbol{x}^*)^T \boldsymbol{r}(\boldsymbol{x}^*) = 0$*,* λ *is the smallest eigenvalue of* $\mathbf{J}(\boldsymbol{x}^*)^T \mathbf{J}(\boldsymbol{x}^*)$*, and*

$$\|(\mathbf{J}(\boldsymbol{x}) - \mathbf{J}(\boldsymbol{x}^*))^T \boldsymbol{r}(\boldsymbol{x}^*)\| \leq \sigma \|\boldsymbol{x} - \boldsymbol{x}^*\|$$

for all $\boldsymbol{x} \in \Omega$*. If* $\sigma < \lambda$ *for any* $c \in (1, \lambda/\sigma)$ *there exists* $\epsilon > 0$ *such that for all* $\boldsymbol{x}_0 \in \mathcal{N}(\boldsymbol{x}^*, \epsilon)$ *the sequence generated by the Gauss-Newton method is well defined, converges to* $\boldsymbol{x}^*$*, and obeys*

$$\|\boldsymbol{x}_{k+1} - \boldsymbol{x}^*\| \leq \frac{c\sigma}{\lambda}\|\boldsymbol{x}_k - \boldsymbol{x}^*\| + \frac{c\alpha\gamma}{2\lambda}\|\boldsymbol{x}_k - \boldsymbol{x}^*\|^2$$

and $\|\boldsymbol{x}_{k+1} - \boldsymbol{x}^*\| \leq \frac{c\sigma+\lambda}{2\lambda}\|\boldsymbol{x}_k - \boldsymbol{x}^*\| < \|\boldsymbol{x}_k - \boldsymbol{x}^*\|$

Corollary

Let the assumptions of the previous theorem be satisfied. If $\boldsymbol{r}(\boldsymbol{x}^*) = 0$, then there exists $\epsilon > 0$ such that for all $\boldsymbol{x}_0 \in \mathcal{N}(\boldsymbol{x}^*, \epsilon)$, the sequence $\{\boldsymbol{x}_k\}$ generated by the Gauss-Newton method is well defined and converges quadratically to $\boldsymbol{x}^*$.

Gauss Newton Method:Advantages

1. Locally quadratically convergent on zero-residual problems.
2. Quickly locally q-linearly convergent on problems that aren't too nonlinear and have reasonably small residuals.
3. Solves linear least-squares problems in one iteration.

Gauss Newton Method: Disadvantages

1. Slowly locally linearly convergent on problems that are sufficiently nonlinear or have reasonably large residuals.
2. Not locally convergent on problems that are very nonlinear or have very large residuals.
3. Not well defined if $\mathbf{J}(\boldsymbol{x}_k)$ doesn't have full column rank.
4. Not necessarily globally convergent.

Implementation using QR

Let us use QR Factorization with Column Pivoting, ie

$$\mathbf{JP} = \mathbf{QR} = \left[\begin{array}{cc} \mathbf{Q}_1 & \mathbf{Q}_2 \end{array} \right] \left[\begin{array}{c} \mathbf{R}_1 \\ 0 \end{array} \right]$$

where $\mathbf{J} \in \mathbb{R}^{m \times n}$, $\mathbf{P} \in \mathbb{R}^{n \times n}$ is a permutation matrix, $\mathbf{Q} \in \mathbb{R}^{m \times m}$ and $\mathbf{R} \in \mathbb{R}^{m \times n}$, $\mathbf{Q}_1 \in \mathbb{R}^{m \times n}$ $\mathbf{Q}_2 \in \mathbb{R}^{m \times m-n}$ with

$$\mathbf{Q}\mathbf{Q}^T = \mathbf{Q}^T\mathbf{Q} = \mathbf{I}$$

and $\mathbf{R}_1 \in \mathbb{R}^{n \times n}$ is an upper triangular matrix with elements of the diagonal satisfying

$$|r_{11}| \geq |r_{22}| \geq \cdots \geq |r_{nn}|$$

Implementation using QR

Considering $\|\mathbf{Q}\boldsymbol{x}\|_2 = \|\boldsymbol{x}\|_2$ if $\mathbf{Q}$ is orthogonal, then

$$
\begin{aligned}
\|\mathbf{J}_k \boldsymbol{d}_k + \boldsymbol{r}_k\|^2 &= \|\mathbf{J}_k \mathbf{P}\mathbf{P}^T \boldsymbol{d}_k + \boldsymbol{r}_k\|^2 \\
&= \|\mathbf{Q}\mathbf{R}\mathbf{P}^T \boldsymbol{d}_k + \boldsymbol{r}_k\|^2 \\
&= \|\mathbf{R}\mathbf{P}^T \boldsymbol{d}_k + \mathbf{Q}^T \boldsymbol{r}_k\|^2 \\
&= \left\| \begin{bmatrix} \mathbf{R}_1 \\ 0 \end{bmatrix} \mathbf{P}^T \boldsymbol{d}_k + \begin{bmatrix} \mathbf{Q}_1^T \\ \mathbf{Q}_2^T \end{bmatrix} \boldsymbol{r}_k \right\|^2 \\
&= \|\mathbf{R}_1 \mathbf{P}^T \boldsymbol{d}_k + \mathbf{Q}_1^T \boldsymbol{r}_k\|^2 + \|\mathbf{Q}_2^T \boldsymbol{r}_k\|^2
\end{aligned}
$$

Implementation using QR

From the last equation, ie,

$$\|\mathbf{J}_k \boldsymbol{d}_k + \boldsymbol{r}_k\|^2 \quad = \quad \|\mathbf{R}_1 \mathbf{P}^T \boldsymbol{d}_k + \mathbf{Q}_1^T \boldsymbol{r}_k\|^2 + \|\mathbf{Q}_2^T \boldsymbol{r}_k\|^2$$

Computing the gradient w.r.t $\boldsymbol{d}_k$ and due to $\mathbf{P}, \mathbf{R}_1$ have inverse,.. then

$$\begin{aligned} \mathbf{P}\mathbf{R}_1^T (\mathbf{R}_1 \mathbf{P}^T \boldsymbol{d}_k + \mathbf{Q}_1^T \boldsymbol{r}_k) &= 0 \\ \mathbf{R}_1 \mathbf{P}^T \boldsymbol{d}_k &= -\mathbf{Q}_1^T \boldsymbol{r}_k \end{aligned}$$

the previous system can be solved in two steps

Implementation using QR

Defining

$$\begin{aligned} \boldsymbol{b} &= -\mathbf{Q}_1^T \boldsymbol{r}_k \\ \boldsymbol{z} &= \mathbf{P}^T \boldsymbol{d}_k \end{aligned}$$

ie, $\boldsymbol{d}_k = \mathbf{P}\boldsymbol{z}$. From

$$\mathbf{R}_1 \mathbf{P}^T \boldsymbol{d}_k = -\mathbf{Q}_1^T \boldsymbol{r}_k$$

we solve the following systems, first for $\boldsymbol{z}$ and then for $\boldsymbol{d}_k$

$$\begin{aligned} \mathbf{R}_1 \boldsymbol{z} &= \boldsymbol{b} \\ \boldsymbol{d}_k &= \mathbf{P}\boldsymbol{z} \end{aligned}$$