Robust kernel newton methods.

Optimization Problem

Optimization problem tries to find a set of best parameters to minimize the overall cost.

$$F = \sum_{i=0}^{n} \rho(f(x)) \tag{1}$$

- f is the objective function.
- x is the parameter vector.
- ρ is robust kernel.

Least-Squares Problem

When the objective function is presented by square form, we call this problem as least-squares problem.

$$f = \frac{r^T \Sigma r}{2} \tag{2}$$

Newton's method

Newton's method is an iterative method for solving nonlinear optimization problems. Δx is iterative step.

$$\Delta x = -H^{-1} g \tag{3}$$

here: g is the gradient vector; H is the hessian matrix.

No robust kernel.

$$g = \dot{f} \tag{4}$$

$$H = \ddot{f} \tag{5}$$

$\dot{f}$: Partial derivatives of objective function
$\ddot{f}$: Second-order partial derivatives of objective function

$$\dot{f} = r^T \Sigma \dot{r} \tag{6}$$

$$\ddot{f} = \dot{r}^T \Sigma \dot{r} + r^T \Sigma \ddot{r} \tag{7}$$

With robust kernel.

$$H = \ddot{\rho} \dot{f} \dot{f}^T + \dot{\rho} \ddot{f} \tag{8}$$

$$g = \dot{\rho} \dot{f} \tag{9}$$

- $\dot{\rho}$: Partial derivatives of robust kernel function
- $\ddot{\rho}$: Second-order partial derivatives of robust kernel function

Gaussian-Newton method

Unlike Newton's method, gauss-newton methods are not necessary to calculate second derivatves, which may difficult to compute in some cases.

No robust kernel.

$$g = \dot{f} = r^T \Sigma \dot{r} \tag{10}$$

$$H = \ddot{f} = \dot{r}^T \Sigma \dot{r} \tag{11}$$

With robust kernel.

$$g = \dot{\rho} \dot{f} \tag{12}$$

$$H = \dot{\rho} \dot{r}^T \Sigma \dot{r} \tag{13}$$

Jacobian and Hessian matrix

The jacobian matrix of r is a matrix, which contain the first-order partial derivatives of all parameters of r.
x denote the parameters of r.

- $x \in \mathbb{R}_n$
- $r \in \mathbb{R}_m$

Jacobian matrix of r is a $m \times n$ matrix.

$$\dot{r} = J_r = \begin{bmatrix} \frac{\partial r_1}{\partial x_1} & \cdots & \frac{\partial r_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial r_m}{\partial x_1} & \cdots & \frac{\partial r_m}{\partial x_n} \end{bmatrix} \tag{14}$$

The Hessian of r is a $m \times n \times n$ Tensor.

$$\ddot{r}_i = H_{r_i} = \begin{bmatrix} \frac{\partial^2 r_i}{\partial x_1 \partial x_1} & \frac{\partial^2 r_i}{\partial x_1 \partial x_2} & \cdots & \frac{\partial r_i}{\partial x_1 \partial x_n} \\ \frac{\partial^2 r_i}{\partial x_2 \partial x_1} & \frac{\partial^2 r_i}{\partial x_2 \partial x_2} & \cdots & \frac{\partial r_i}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 r_i}{\partial x_n \partial x_1} & \frac{\partial^2 r_i}{\partial x_n \partial x_2} & \cdots & \frac{\partial r_i}{\partial x_n \partial x_n} \end{bmatrix} \tag{15}$$