

# GTSAM Robust Noise Model

Fan Jiang<sup>†</sup>, Yetong Zhang<sup>†</sup>

February 2020

## 1 Introduction

In gtsam, we are mostly concerned with the least square problems:

$$\min_x \sum_i \frac{1}{2} \|h_i(x_i) - z_i\|_{\Sigma_i}^2$$

However, when outliers exist in the measurements, they may have a large influence on our result. To resolve this issue, robust loss functions  $\rho$  were proposed such that turns the problem into

$$\min_x \sum_i \rho(\|h_i(x_i) - z_i\|_{\Sigma_i})$$

For simplicity, we use  $m_i$  to represent the Mahalanobis distance of the measurement error vector:  $m_i \doteq \|h_i(x_i) - z_i\|_{\Sigma_i}$ , such that each error term becomes  $\rho(m_i)$ .

## 2 Linear Reweighted Least Squares

In [1], linear robust estimation problems were solved with reweighted least squares. In linear cases, the objective is formulated as

$$\sum_i \rho(\|A_i x_i - b_i\|)$$

The objective is minimized by iteratively solving the reweighted least squares problem:

$$\sum_i w(m_i) \|A_i x_i - b_i\|^2$$

The weight is calculated based on the error's Mahalanobis distance  $m$  of the previous iteration

$$w(m) = \frac{\rho'(m)}{m}$$

We can see that in each iteration, a new linear least square problem is created, and solving the problem will generate the weights  $w$  for next iteration.

For the linear case, the current gtsam implementation strictly follows this algorithm (with Gauss-Newton optimizer):

---

**Algorithm 1** Linear function with robust noise model

---

set initial value for  $x$

**while** *not converge* **do**

    calculate weight for each factor  $w(m_i) = \frac{\rho'(m_i(x_i))}{m_i(x_i)}$

    solve the weighted linear LS problem  $\min_x \sum_i \frac{1}{2} w(m_i) \|H_i x_i - z_i\|_{\Sigma_i}^2$  with Cholesky or QR

    factorization

    update  $x$  with the optimization result

**end**

---

### 3 Nonlinear Reweighted Least Squares

When we turn the robust noise problem from linear to nonlinear, our objective function becomes

$$\sum_i \rho(\|h_i(x_i) - z_i\|_{\Sigma_i}) \quad (1)$$

One way to solve the problem is: Every iteration, we change the objective into weighted nonlinear least square form, and call the nonlinear solver (LM, Dogleg, etc) to solve the problem.

In the current implementation of GTSAM, a faster approach is used. In every iteration, we perform both reweighting and linearization, as in Algorithm 2

---

**Algorithm 2** trust-region method for nonlinear-robust noise problem

---

set initial value for  $x$

**while** *not converge* **do**

    calculate Mahalanobis distance of the error for each factor  $m_i = \|h_i(x_i) - z_i\|_{\Sigma_i}$

    calculate weight for each factor  $w_i = \frac{\rho'(m_i)}{m_i}$

    create a weighted nonlinear LS problem  $\sum_i \frac{1}{2} w_i \|h_i(x_i) - z_i\|_{\Sigma_i}^2$

    linearize the problem to  $\sum_i \frac{1}{2} w(m_i) \|A_i \Delta_i - b_i\|^2$

    solve the linearized LS problem

    maintain trust region

    update  $x$  with LM/Dogleg rule

**end**

---

Note that in every iteration, we only needs to solve a linear least square problem by minimizing

$$\sum_i \frac{1}{2} \|\sqrt{w(m_i)} A_i \Delta_i - \sqrt{w(m_i)} b_i\|^2 \quad (2)$$

An interpretation for Algorithm 2 is: we use the weighted linearized least square function (2) as an approximation to our original objective function (1). In the next section, we'll inspect how well the approximation is.

## 4 Properties of the Weighted Linearized Function

### 4.1 Function Value

At the linearization point  $x_0$ , the value for the original objective function is

$$\rho(m(x_0))$$

while the value for the reweighted linearized function is

$$\frac{1}{2}w(m)m(x_0)^2 = \frac{1}{2}\rho'(m(x_0))m(x_0)$$

Note: they are not necessarily the same as shown in the example in Section 5.

## 4.2 Jacobian Value

Interestingly, the Jacobian of the reweighted function agree with the Jacobian of the original function.

Let us define  $A_u, b_u$  to be the linearization result of  $h(x) - z$  at  $x_0$  by Taylor expansion. e.g.

$$h(x) - z = h(x_0 + \Delta) - z \approx h(x_0) - z + A_u \Delta = A_u \Delta - b_u$$

and we define  $A_w, b_w$  to be the whitened  $A_u, b_u$

$$\begin{aligned} A_w &= \Sigma^{-\frac{1}{2}} A_u \\ b_w &= \Sigma^{-\frac{1}{2}} b_u \end{aligned}$$

At the linearization point, the Jacobian for the original function is

$$\begin{aligned} \frac{\partial \rho}{\partial x} &= \frac{\partial \rho}{\partial r} \frac{\partial r}{\partial x} \\ &= \frac{\partial \rho}{\partial r} \frac{\partial r}{\partial(r^2)} \frac{\partial r^2}{\partial x} \\ &= \frac{\rho'(m)}{r} \frac{\partial r^2}{\partial x} \\ &= w(m) A_w^T b_w \end{aligned}$$

whereas the Jacobian of the *reweighted* linearized function is

$$w(m) A_w^T b_w$$

Thus, if we add a constant offset term to the reweighted linearized function to make it align with the original objective function, it is a local approximation, with the accuracy of first order derivative.

## 5 Example

We use a simple linear 1d function as our example, in this case the robust error function is huber loss with  $k = 2$ :

$$\rho(m) = \begin{cases} m^2/2 & \text{if } |x| \leq k \\ k(|m| - k/2) & \text{if } |x| > k \end{cases}$$

The measurement error function is defined as

$$h(x) - z = x - 2$$

Then, our objective function becomes

$$\rho(x - 2)$$

When linearizing the objective function at point  $x_0 = 5$ .

The corresponding error plot is shown in Figure 1. Note that the reweighted least square function (blue curve) is not aligned with the original objective function (black curve) at the linearization point  $x = 5$ . After we add the offset to the reweighted least square function, we get a good local approximation at the linearization point (orange curve).

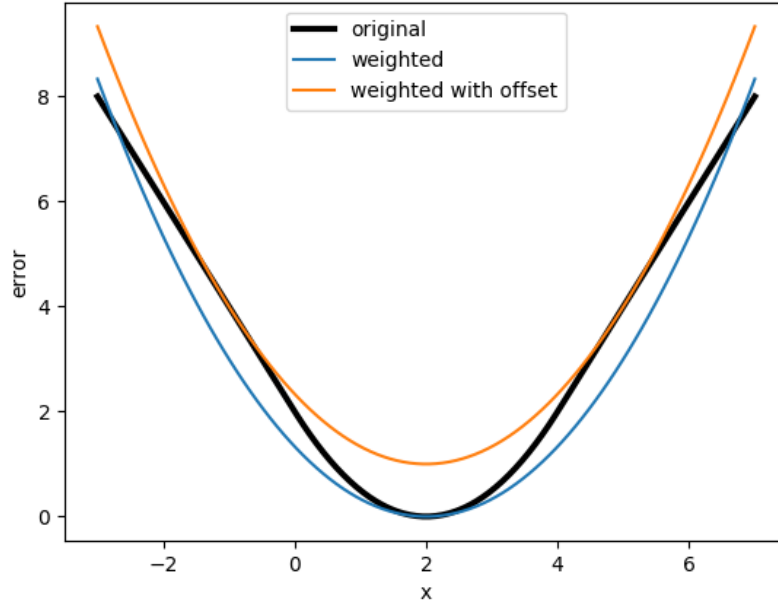


Figure 1: 1D Linear weighted least squares

## References

- [1] Paul W. Holland and Roy E. Welsch. Robust regression using iteratively reweighted least-squares. *Communications in Statistics - Theory and Methods*, 6(9):813–827, 1977.