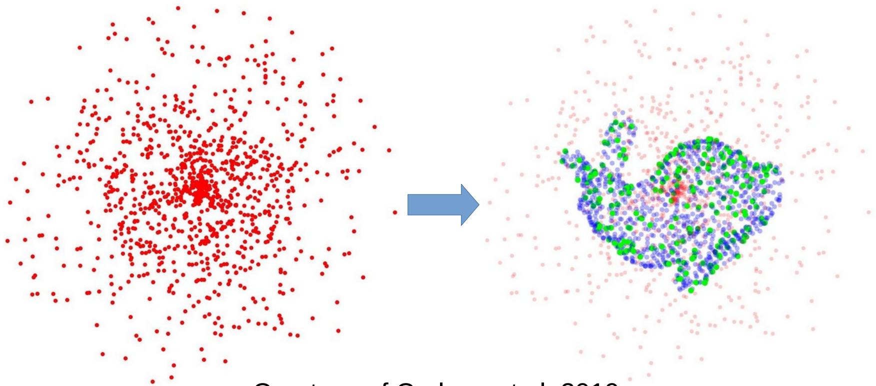
Probabilistic Robotics Course

Robust Estimators

Giorgio Grisetti
grisetti@diag.uniroma1.it

Department of Computer, Control and Management Engineering
Sapienza University of Rome



Courtesy of Carlone et al. 2019

Gauss-Newton

We have seen in the previous episodes that GN is an iterative method to find the minimum of the following function

$$F(\mathbf{x}) = \sum_i e^{[i]}(\mathbf{x})$$

$$e^{[i]}(\mathbf{x}) = \mathbf{e}^{[i]}(\mathbf{x})^T \mathbf{\Omega}^{[i]} \mathbf{e}^{[i]}(\mathbf{x})$$

$$= \|\mathbf{e}^{[i]}(\mathbf{x})\|^2_{\mathbf{\Omega}^{[i]}}$$

The latter expression is called the Omega L2 norm, and represents the squared norm of the a vector, modulated by a weight matrix Omega

$$\|\mathbf{v}\|^2_{\mathbf{\Omega}} = \mathbf{v}^T \mathbf{\Omega} \mathbf{v}$$

Omega Norm

We can then say that GN find the x that minimizes the sum of the squared L2 Omega norms of the errors.

- The L2 norm assigns to each term a cost that is **quadratic** in the magnitude of the error vector.
- There is a linear transformation of a vector that maps an omega norm as a plain squared norm

$$\mathbf{\Omega} = \mathbf{L}\mathbf{L}^T$$

$$\begin{aligned}\|\mathbf{v}\|^2_{\mathbf{\Omega}} &= \mathbf{v}^T\mathbf{\Omega}\mathbf{v} \\ &= \mathbf{v}^T\mathbf{L}\underbrace{\mathbf{L}^T\mathbf{v}}_{\mathbf{v}'} \\ &= \|\mathbf{v}'\|^2\end{aligned}$$

L1 Norm

In contrast with the L2 norm the L1 norm is just the magnitude of the vector, and grows linearly with the length of a vector.

It can be obtained by taking the square root of the L2 norm.

$$\|\mathbf{v}\|_{\boldsymbol{\Omega}} = \sqrt{\mathbf{v}^T \boldsymbol{\Omega} \mathbf{v}}$$

Using the L1 norm in minimization

- Might present some benefits, since it does not quadratically overweight few potential “wrong” measurements, that would hinder bias our estimation
- Has the shortcoming of invalidating all the nice Gaussian machinery where we live in

Outliers

Typical scenarios like ICP, or in general registration problems are characterized by heuristics to compute the association.

- In general, some associations **will** not be correct.
- We cannot tell **a priori** which associations are wrong.

There are two strategies to deal with the issue

- Consensus (like RANSAC, soon on these screens),

  Find the larger set of measurements that simultaneously agree with a solution

- M Estimators (in this episode)

  Assume the initial guess is good enough and consider “less” the large error terms

M estimators

Minimize

$$F(\mathbf{x}) = \sum_i \rho(\underbrace{\|\mathbf{e}^{[i]}(\mathbf{x})\|_{\mathbf{\Omega}^{[i]}}}_{u^{[i]}(\mathbf{x})})$$

- Here rho(u) is a scalar monotonically increasing function that maps the L1 Omega norm of the error to a scalar
- Note that with $\rho(u) = u^2$

  we have the plain GN algorithm

M estimators

Common cost functions here rho=f, and u=r.

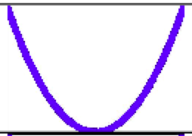
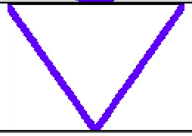
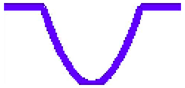
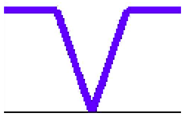
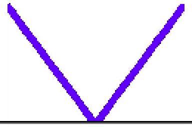
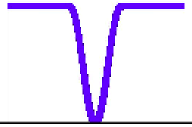
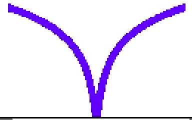
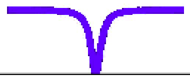
	$f(r)$	$f(r)$ graphical	$\omega(r)$
L^2	$\frac{1}{2}r^2$		1
L^1	$\|r\|$		$\frac{1}{\|r\|}$
L^2_{trunc}	$\begin{cases} \frac{1}{2}r^2 & \text{if } \|r\| \leq k \\ \frac{k^2}{2} & \text{otherwise} \end{cases}$		$\begin{cases} 1 & \text{if } \|r\| \leq k \\ 0 & \text{otherwise} \end{cases}$
L^1_{trunc}	$\begin{cases} \|r\| & \text{if } \|r\| \leq k \\ \text{k} & \text{otherwise} \end{cases}$		$\begin{cases} \frac{1}{\|r\|} & \text{if } \|r\| \leq k \\ 0 & \text{otherwise} \end{cases}$
$Huber$	$\begin{cases} \frac{r^2}{2} & \text{if } x \leq k \\ k(\|r - k/2\|) & \text{otherwise} \end{cases}$		$\begin{cases} 1 & \text{if } x \leq k \\ \frac{k}{\|r\|} & \text{otherwise} \end{cases}$
$Tukey$	$\begin{cases} \frac{k^2}{6}\left(1 - \left(1 - \left(\frac{r}{k}\right)^2\right)^3\right) & \text{if } x \leq k \\ \frac{k^2}{6} & \text{otherwise} \end{cases}$		$\begin{cases} \left(1 - \left(\frac{r}{k}\right)^2\right)^2 & \text{if } r \leq k \\ 0 & \text{otherwise} \end{cases}$
$Cauchy$	$\frac{k^2}{2} log\left(1 + \left(\frac{r}{k}\right)^2\right)$		$\frac{1}{1+\left(\frac{r}{k}\right)^2}$
$Geman-McClure$	$\frac{r^2/2}{1+r^2}$		$\frac{1}{(1+r^2)^2}$

TABLE II: Summary of the robust functions evaluated in this paper

- (courtesy of Civera and Concha)

M estimators in GN

We want to modify the GN algorithm to incorporate arbitrary robust cost functions.

- To minimize a function we need to set its derivative to zero.

Derivative of the L2 norm (omitting indices)

$$\frac{\partial \|\mathbf{e}(\mathbf{x})\|^2_{\mathbf{\Omega}}}{\partial \mathbf{x}} = 2\mathbf{e}^T(\mathbf{x})\mathbf{\Omega}\frac{\partial \mathbf{e}(\mathbf{x})}{\partial \mathbf{x}}$$

M estimators in GN

We want to modify the GN algorithm to incorporate arbitrary robust estimators.

- Derivative of the L1 norm (omitting indices)

$$\frac{\partial\sqrt{\|\mathbf{e}(\mathbf{x})\|^2_{\mathbf{\Omega}}}}{\partial\mathbf{x}} = \frac{1}{2\|\mathbf{e}(\mathbf{x})\|_{\mathbf{\Omega}}} 2\mathbf{e}^T(\mathbf{x})\mathbf{\Omega}\frac{\partial\mathbf{e}(\mathbf{x})}{\partial\mathbf{x}}$$

$$= \frac{1}{u(\mathbf{x})}\frac{\partial\|\mathbf{e}(\mathbf{x})\|^2_{\mathbf{\Omega}}}{\partial\mathbf{x}}$$

  is the product of

  - a scalar (the inverse of the L1 norm) and
  - the derivative of the L2 norm.

Robustifiers

Using the chain rule we can express the derivative of the error weighed by the cost function (robustifed error)

$$\frac{\partial \rho(u(\mathbf{x}))}{\partial \mathbf{x}} = \left.\frac{\partial \rho(u)}{\partial u}\right|_{u=u(\mathbf{x})} \frac{1}{u(\mathbf{x})} \frac{\partial \|\mathbf{e}(\mathbf{x})\|^2_{\mathbf{\Omega}}}{\partial \mathbf{x}}$$

$$= \left.\frac{\partial \rho(u)}{\partial u}\right|_{u=u(\mathbf{x})} \frac{1}{u(\mathbf{x})} \mathbf{e}^T(\mathbf{x}) \mathbf{\Omega} \frac{\partial \mathbf{e}(\mathbf{x})}{\partial \mathbf{x}}$$

$$\gamma(\mathbf{x}) = \left.\frac{\partial \rho(u)}{\partial u}\right|_{u=u(\mathbf{x})} \frac{1}{u(\mathbf{x})}$$

Robustifiers

Put side by side the derivatives of the L2 norm and of the robustified norm differ just by a scalar term gamma(x)

$$\frac{\partial \rho(u(\mathbf{x}))}{\partial \mathbf{x}} = \gamma(\mathbf{x})\mathbf{e}^T(\mathbf{x})\boldsymbol{\Omega}\frac{\partial \mathbf{e}(\mathbf{x})}{\partial \mathbf{x}}$$

$$\frac{\partial \|\mathbf{e}(\mathbf{x})\|^2_{\boldsymbol{\Omega}}}{\partial \mathbf{x}} = 2\mathbf{e}^T(\mathbf{x})\boldsymbol{\Omega}\frac{\partial \mathbf{e}(\mathbf{x})}{\partial \mathbf{x}}$$

Such a term can be absorbed as a scaling factor in the Omega matrix, that will be changed at each iteration, depending on *u*.

This scheme relies on the fact that the algorithm is iterative

Robust GN (one Iteration)

Clear **H** and **b** $\mathbf{H} \leftarrow 0 \quad \mathbf{b} \leftarrow 0$

For each measurement

- Compute error and jacobian

$$\mathbf{e}^{[i]} \leftarrow \mathbf{h}^{[i]}(\mathbf{x}^*) - \mathbf{z}^{[i]}$$

$$\mathbf{J}^{[i]} \leftarrow \left. \frac{\partial \mathbf{e}^{[i]}(\mathbf{x})}{\partial \mathbf{x}} \right|_{\mathbf{x}=\mathbf{x}^*}$$

- Update the scaling based on the robustifier

$$u^{[i]} \leftarrow \sqrt{\mathbf{e}^{[i]T} \mathbf{e}^{[i]}}$$

$$\gamma^{[i]} \leftarrow \frac{1}{u^{[i]}} \left. \frac{\partial \rho(u)}{\partial u} \right|_{u=u^{[i]}}$$

- Update the quadratic form

$$\mathbf{H} \leftarrow \mathbf{H} + \mathbf{J}^{[i]T} \gamma^{[i]} \mathbf{\Omega}^{[i]} \mathbf{J}^{[i]}$$

$$\mathbf{b} \leftarrow \mathbf{b} + \mathbf{J}^{[i]T} \gamma^{[i]} \mathbf{\Omega}^{[i]} \mathbf{e}^{[i]}$$

Solve the linear system

$$\mathbf{\Delta x} \leftarrow \mathrm{solve}(\mathbf{H} \mathbf{\Delta x} = -\mathbf{b})$$

Apply the increments

$$\mathbf{x}^* \leftarrow \mathbf{x}^* + \mathbf{\Delta x}$$