Gauss-Newton / Levenberg-Marquardt Optimization

Ethan Eade

Updated March 20, 2013

1 Definitions

Let $\mathbf{x} \in X$ be the state parameters to be optimized, with n degrees of freedom. The goal of the optimization is to maximize the likelihood of a set of observations given the parameters, under a specified observation model.

1.1 Observations

For some problems, the observations are represented directly, either as vectors in $\mathbb{R}^m$ or as elements on a manifold with m degrees of freedom. For example, observations of points in an image are vectors in $\mathbb{R}^2$ ($m = 2$). Observations of coordinate transformations in 3D are elements of $\mathrm{SE}(3)$ ($m = 6$). In these cases, we refer to the collective observation as $\mathbf{z} \in \mathbf{Z}$. Often the collective observation is built by stacking up M independent observations $\{\mathbf{z}_i \in Z\}$:

$$\mathbf{Z} \equiv Z^M \tag{1}$$

$$\mathbf{z} \equiv \begin{pmatrix} \mathbf{z}_1 \\ \vdots \\ \mathbf{z}_M \end{pmatrix} \tag{2}$$

The observation model $\mathbf{h}(\mathbf{x})$ predicts the value of $\mathbf{z}$ given the state parameters:

$$\mathbf{h} : X \to \mathbf{Z} \tag{3}$$

The error vector $\mathbf{v}$ is then the difference (in a vector space) between the observations and the predictions, as a function of the parameters $\mathbf{x}$:

$$\mathbf{v}(\mathbf{x}) \equiv \mathbf{z} \ominus \mathbf{h}(\mathbf{x}) \tag{4}$$

When $\mathbf{z}$ is composed of independent observations $\{\mathbf{z}_i\}$, we can also refer to the corresponding pieces of $\mathbf{v}$:

$$\mathbf{v}_i(\mathbf{x}) \equiv \mathbf{z}_i \ominus \mathbf{h}_i(\mathbf{x}) \tag{5}$$

$$\mathbf{v}(\mathbf{x}) \equiv \begin{pmatrix} \mathbf{v}_1 \\ \vdots \\ \mathbf{v}_M \end{pmatrix} \tag{6}$$

The operator $\ominus$ yields a vector difference between two elements in Z:

$$\ominus : Z \times Z \to \mathbb{R}^m \tag{7}$$

Its definition depends on that space. For observations in a vector space (e.g. image points), $\ominus$ is just the plain vector difference. For a Lie group G, and two elements $a, b \in G$, we can define

$$a \ominus b \equiv \ln\left(a \cdot b^{-1}\right) \in \mathfrak{g} \tag{8}$$

where $\mathfrak{g}$ is the Lie algebra vector space corresponding to G.

For some problems, the error function is easier to express directly, rather than as a difference between observation and model prediction. For instance, when estimating epipolar geometry, the errors are the distances between points in an image and their epipolar lines. Each distance $\mathbf{v}_i$ is a scalar ($m = 1$), though the points themselves are 2-vectors and the predictions are lines. In such scenarios, we refer to the error vector $\mathbf{v}(\mathbf{x})$ without explicitly defining it in terms of $\mathbf{z}$ and $\mathbf{h}(\mathbf{x})$. Nonetheless, we still refer to the pieces $\mathbf{v}_i$ as observations.

We describe the uncertainty of $\mathbf{v}$ with a covariance matrix $\mathbf{R}$. In the case of Eq. 4, $\mathbf{R}$ is typically just the covariance over $\mathbf{z}$ itself. Otherwise, $\mathbf{R}$ is computed by projecting uncertainties of the measured quantities into the space of the error $\mathbf{v}$. Again using the epipolar geometry example, the covariance of each point measurement is propagated through the distance-to-line function to yield variance over the epipolar error.

When the errors for each observation are independent, as is common in many optimizations, the matrix $\mathbf{R}$ is block diagonal with M blocks, and we refer to the i^{th} block as $\mathbf{R}_i$:

$$\mathbf{R} = \begin{pmatrix} \mathbf{R}_1 & & \\ & \ddots & \\ & & \mathbf{R}_M \end{pmatrix} \tag{9}$$

1.2 Jacobians

We define $\mathbf{J}$ to be the negative Jacobian (differential) of the error $\mathbf{v}$ as a function of $\mathbf{x}$:

$$\mathbf{J} \equiv -\frac{\partial \mathbf{v}(\mathbf{x})}{\partial \mathbf{x}} \tag{10}$$

We use the negative Jacobian because when $\mathbf{v} \equiv \mathbf{z} \ominus \mathbf{h}(\mathbf{x})$, it is more natural to compute the Jacobian of $\mathbf{h}(\mathbf{x})$, and then

$$\mathbf{J} = \frac{\partial \mathbf{h}(\mathbf{x})}{\partial \mathbf{x}} \tag{11}$$

As with $\mathbf{v}$ and $\mathbf{R}$, for independent errors the whole Jacobian is just the stacked matrix of individual Jacobians:

$$\begin{aligned} \mathbf{J}_i &\equiv -\frac{\partial \mathbf{v}_i(\mathbf{x})}{\partial \mathbf{x}} \\ \mathbf{J} &= \begin{pmatrix} \mathbf{J}_1 \\ \vdots \\ \mathbf{J}_M \end{pmatrix} \end{aligned}$$

1.3 Parameter Perturbations

Above, the operator $\ominus$ was defined for computing differences in the observation space, in such a way that $\mathbf{z}_i \ominus \mathbf{h}_i(\mathbf{x})$ is a vector in $\mathbb{R}^m$ even when $\mathbf{z}_i$ and $\mathbf{h}_i(\mathbf{x})$ are not represented as vectors. Similarly, we define the operator $\oplus$ for "adding" a perturbation to our parameters. The parameter space X could be a vector space like $\mathbb{R}^n$, or instead some other manifold with n degrees of freedom. For pose estimation, $X = \mathrm{SE}(3)$ and $n = 6$.

Consider a parameter perturbation vector $\boldsymbol{\delta} \in \mathbb{R}^n$. Then for $\mathbf{x} \in X$, we have

$$\oplus \quad : \quad X \times \mathbb{R}^n \to X \tag{12}$$

When $X = \mathbb{R}^n$, this is just standard vector addition:

$$\mathbf{x} \oplus \boldsymbol{\delta} \equiv \mathbf{x} + \boldsymbol{\delta} \tag{13}$$

When $X = G$ for a Lie group G, the perturbation is expressed as left multiplication in the group:

$$\mathbf{x} \oplus \boldsymbol{\delta} \equiv \exp(\boldsymbol{\delta}) \cdot \mathbf{x} \tag{14}$$

Thus in Lie groups, using Eqs. 8 and 14, our intuition for $\oplus$ and $\ominus$ holds:

$$\begin{aligned} (\mathbf{x} \oplus \boldsymbol{\delta}) \ominus \mathbf{x} &= \ln\left(\exp(\boldsymbol{\delta}) \cdot \mathbf{x} \cdot \mathbf{x}^{-1}\right) && (15) \\ &= \ln(\exp(\boldsymbol{\delta})) && (16) \\ &= \boldsymbol{\delta} && (17) \end{aligned}$$

In manifolds without the exponential map, the perturbation can be computed as an update that might violate the manifold constraints, followed by a projection back onto the manifold.

1.4 Objective Function

The goal is to adjust $\mathbf{x}$ so that the likelihood of the observations is maximized:

$$p\left(\mathbf{z}|\mathbf{x}\right) \propto \exp\left(-\frac{1}{2}\mathbf{v}^T \cdot \mathbf{R}^{-1} \cdot \mathbf{v}\right) \tag{18}$$

As the logarithm is monotonic, this is equivalent to minimizing the negative log-likelihood objective function:

$$L = \mathbf{v}^T \cdot \mathbf{R}^{-1} \cdot \mathbf{v} \tag{19}$$

When the individual observations are independent, the covariance matrix $\mathbf{R}$ is block diagonal. Then the objective function reduces to a sum over the observations (again as a function of $\mathbf{x}$):

$$\begin{aligned} L_i &\equiv \mathbf{v}_i^T \cdot \mathbf{R}_i^{-1} \cdot \mathbf{v}_i & (20)\\ L &= \sum_i L_i & (21) \end{aligned}$$

The value of this objective function for some specific parameters $\mathbf{x}$ is often called the *residual*.

2 Gauss-Newton Method

We approximate $\mathbf{v}_i$ as a function of $\mathbf{x}$ by a first-order Taylor expansion:

$$\begin{aligned} \mathbf{v}_i\left(\mathbf{x} \oplus \boldsymbol{\delta}\right) &\approx \mathbf{v}_i(\mathbf{x}) + \frac{\partial \mathbf{v}_i(\mathbf{x})}{\partial \mathbf{x}} \cdot \boldsymbol{\delta} \\ &= \mathbf{v}_i(\mathbf{x}) - \mathbf{J}_i \cdot \boldsymbol{\delta} \end{aligned}$$

This approximation then extends trivially to the whole error vector:

$$\mathbf{v}\left(\mathbf{x} \oplus \boldsymbol{\delta}\right) \approx \mathbf{v}(\mathbf{x}) - \mathbf{J} \cdot \boldsymbol{\delta} \tag{22}$$

Substituting this approximation into 19 yields

$$L = \left(\mathbf{v} - \mathbf{J} \cdot \boldsymbol{\delta}\right)^T \cdot \mathbf{R}^{-1} \cdot \left(\mathbf{v} - \mathbf{J} \cdot \boldsymbol{\delta}\right) \tag{23}$$

To minimize this residual, we differentiate with respect to δ, set equal to zero, and solve for δ:

$$\begin{aligned}
\frac{\partial L}{\partial \delta} &= -2\left(\mathbf{v} - \mathbf{J}\cdot\delta\right)^T \cdot \mathbf{R}^{-1} \cdot \mathbf{J} && (24)\\
\mathbf{0} &= \mathbf{v}^T \cdot \mathbf{R}^{-1} \cdot \mathbf{J} - \delta^T \cdot \mathbf{J}^T \cdot \mathbf{R}^{-1} \cdot \mathbf{J} && (25)\\
\mathbf{v}^T \cdot \mathbf{R}^{-1} \cdot \mathbf{J} &= \delta^T \cdot \mathbf{J}^T \cdot \mathbf{R}^{-1} \cdot \mathbf{J} && (26)\\
\mathbf{J}^T \cdot \mathbf{R}^{-1} \cdot \mathbf{J} \cdot \delta &= \mathbf{J}^T \cdot \mathbf{R}^{-1} \cdot \mathbf{v} && (27)\\
\delta &= \left(\mathbf{J}^T \cdot \mathbf{R}^{-1} \cdot \mathbf{J}\right)^{-1} \cdot \mathbf{J}^T \cdot \mathbf{R}^{-1} \cdot \mathbf{v} && (28)
\end{aligned}$$

The Fisher information matrix $\left[\mathbf{J}^T \cdot \mathbf{R}^{-1} \cdot \mathbf{J}\right]$ is symmetric and positive definite, so the linear system can be efficiently solved with a Cholesky or LDL^T decomposition. Further, if the observations are independent, the information matrix and information vector are simply accumulated over the observations:

$$\begin{aligned}
\mathbf{J}^T \cdot \mathbf{R}^{-1} \cdot \mathbf{J} &= \sum_i \mathbf{J}_i^T \cdot \mathbf{R}_i^{-1} \cdot \mathbf{J}_i && (29)\\
\mathbf{J}^T \cdot \mathbf{R}^{-1} \cdot \mathbf{v} &= \sum_i \mathbf{J}_i^T \cdot \mathbf{R}_i^{-1} \cdot \mathbf{v}_i && (30)
\end{aligned}$$

The update from Eq. 31 is then applied by pertubring $\mathbf{x}$ by δ:

$$\mathbf{x} \leftarrow \mathbf{x} \oplus \delta \quad (31)$$

The whole process is iterated by evaluating $\mathbf{J}$ and $\mathbf{v}$ at the new parameters, recomputing δ (Eq. 28), and applying the update (Eq. 31). The iteration continues until some convergence criterion is met, or the iteration count reaches a bound.

Note that upon convergence to a minimum of the residual, $\left(\mathbf{J}^T \cdot \mathbf{R}^{-1} \cdot \mathbf{J}\right)^{-1}$ (the inverse of the information matrix) is the Cramer-Rao lower bound for the covariance of the parameters.

3 Assuring Convergence

Convergence of the Gauss-Newton method is not guaranteed, and it converges only to a local optimum that depends on the starting parameters. In practice, if the objective function $L(\mathbf{x})$ is locally well-approximated by a quadratic form, then convergence to a local minimum is quadratic. However, the curvature of the error surface of a nonlinear observation model can vary significantly over the parameter space. The Levenberg-Marquardt method is a refinement to the Gauss-Newton procedure that increases the chance of local convergence and prohibits divergence. Note that the results still depend on the starting point.

3.1 Levenberg Method

Define a modified information matrix, with a damping factor λ:

$$\mathbf{A} \equiv \mathbf{J}^T \cdot \mathbf{R}^{-1} \cdot \mathbf{J} + \lambda \cdot \mathbf{I} \tag{32}$$

As $\lambda \to 0$, $\mathbf{A}$ approaches the unmodified information matrix. For $\lambda \to \infty$, $\mathbf{A}$ is dominated by the identity matrix. As λ increases, the computed update $\boldsymbol{\delta}$ tends to the scaled gradient descent direction:

$$\boldsymbol{\delta} \to \frac{1}{\lambda} \left(\mathbf{J}^T \cdot \mathbf{R}^{-1} \cdot \mathbf{v} \right) \tag{33}$$

with decreasing step size. Thus if the residual L is not currently at a minimum, increasing λ and computing the update $\boldsymbol{\delta}$ will eventually lead to a decrease in L.

To control convergence behavior, we modify λ according to a simple schedule, controlled by two factors $1 < a < b$. Typical values are $a = 2$ and $b = 10$. Starting with parameters $\mathbf{x}$, residual $L(\mathbf{x})$, and damping value λ, an update $\boldsymbol{\delta}_\lambda$ is computed and applied:

$$\mathbf{x}' = \mathbf{x} \oplus \boldsymbol{\delta}_\lambda \tag{34}$$

Then the residual $L(\mathbf{x}')$ is computed under the new parameters. If the residual has decreased, such that $L(\mathbf{x}') < L(\mathbf{x})$, then the update is valid, and the damping factor is decreased by factor a:

$$\mathbf{x} \leftarrow \mathbf{x}' \tag{35}$$

$$\lambda \leftarrow \frac{1}{a} \cdot \lambda \tag{36}$$

If the residual has increased, or has not decreased by some threshold amount, the parameters are left unchanged and λ is increased:

$$\lambda \leftarrow b \cdot \lambda \tag{37}$$

Thus only parameter updates that decrease the residual are kept. The process is iterated similarly to the Gauss-Newton method, and can be terminated when λ reaches a large threshold value (which corresponds to a vanishingly small update). Note that in the case where the parameter update is rejected and λ increases, the information matrix and vector need not be recomputed. Instead only the matrix $\mathbf{A}$ needs to be updated using the new λ value, and the linear system solved to find a new $\boldsymbol{\delta}_\lambda$.

3.2 Levenberg-Marquardt Method

A refinement due to Marquardt changes how $\mathbf{A}$ is defined in terms of λ. Instead of damping all parameter dimensions equally (by adding a multiple of the identity matrix), a scaled version of of the diagonal of the information matrix itself can be added:

$$\mathbf{A} \equiv \mathbf{J}^T \cdot \mathbf{R}^{-1} \cdot \mathbf{J} + \lambda \cdot \mathbf{diag}\left(\mathbf{J}^T \cdot \mathbf{R}^{-1} \cdot \mathbf{J}\right) \tag{38}$$

As λ grows, $\boldsymbol{\delta}_\lambda$ again tends towards a gradient descent update, but with each dimension scaled according to the diagonal of the information matrix. This can lead to faster convergence than the Levenberg damping term when some dimensions of the error surface have much different curvature than others.

4 Robust Cost Functions

When dealing with non-Gaussian distributed errors, such as arise when some observations are outliers, a cost function other than simple quadratic error is appropriate. An alternative cost function $\rho : \mathbb{R}^+ \to \mathbb{R}^+$ can be injected into the objective function inside the sum of per-observation negative log-likelihoods by generalizing Eq. 20:

$$\begin{aligned} C_i &\equiv \rho\left(L_i\right) && (39)\\ &= \rho\left(\mathbf{v}_i^T \cdot \mathbf{R}_i^{-1} \cdot \mathbf{v}_i\right) && (40)\\ C &= \sum_i C_i && (41) \end{aligned}$$

The standard Gaussian least-squares objective function is thus the special case $\rho\left(L_i\right) = L_i$.

The optimization method presented here assumes the function ρ is continuously differentiable. M-estimators have nontrivial ρ, though often M-estimator cost functions are specified in the literature as functions of $\sqrt{L_i}$.

For example, the Huber cost function $\rho_{H[k]}$ with scale k can be defined in terms of L_i:

$$\rho_{H[k]}\left(L_i\right) \equiv \begin{cases} L_i & \text{if } L_i < k^2 \\ 2k \cdot \sqrt{L_i} - k^2 & \text{if } L_i \geq k^2 \end{cases} \tag{42}$$

We can differentiate the more general objective function of Eq. 41 around our parameters with the aid of the chain rule:

$$\frac{\partial C}{\partial \mathbf{x}} = \sum_i \left[\frac{\partial \rho\left(L_i\right)}{\partial L_i} \cdot \frac{\partial L_i}{\partial \mathbf{x}} \right] \tag{43}$$

$$= \sum_i \left[\frac{\partial \rho\left(L_i\right)}{\partial L_i} \cdot 2\mathbf{v}_i^T \cdot \mathbf{R}_i^{-1} \cdot \frac{\partial \mathbf{v}_i(\mathbf{x})}{\partial \mathbf{x}} \right] \tag{44}$$

$$= \sum_i \left[\frac{\partial \rho\left(L_i\right)}{\partial L_i} \cdot 2\mathbf{v}_i^T \cdot \mathbf{R}_i^{-1} \cdot -\mathbf{J}_i \right] \tag{45}$$

Linearizing the model $\mathbf{h}_i\left(\mathbf{x}+\boldsymbol{\delta}\right)$ and setting the differential to zero yields a linear system for the update vector $\boldsymbol{\delta}$ similar to the standard case:

$$\mathbf{0} = \sum_i \left[\frac{\partial \rho\left(L_i\right)}{\partial L_i} \cdot \left[\mathbf{z}_i - \mathbf{h}_i\left(\mathbf{x}+\boldsymbol{\delta}\right)\right]^T \cdot \mathbf{R}_i^{-1} \cdot \mathbf{J}_i \right] \tag{46}$$

$$\approx \sum_i \left[\frac{\partial \rho\left(L_i\right)}{\partial L_i} \cdot \left(\mathbf{v}_i - \mathbf{J}_i\boldsymbol{\delta}\right)^T \cdot \mathbf{R}_i^{-1} \cdot \mathbf{J}_i \right] \tag{47}$$

$$\sum_i \left[\frac{\partial \rho\left(L_i\right)}{\partial L_i} \cdot \boldsymbol{\delta}^T \cdot \mathbf{J}_i^T \cdot \mathbf{R}_i^{-1} \cdot \mathbf{J}_i \right] = \sum_i \left[\frac{\partial \rho\left(L_i\right)}{\partial L_i} \cdot \mathbf{v}_i^T \cdot \mathbf{R}_i^{-1} \cdot \mathbf{J}_i \right] \tag{48}$$

The differential of the cost function is typically called the weight function:

$$W_i \equiv \frac{\partial \rho\left(L_i\right)}{\partial L_i} \tag{49}$$

For the example of the Huber cost function, the weight function is then:

$$W_{H[k]}\left(L_i\right) \equiv \begin{cases} 1 & \text{if } L_i < k^2 \\ k/\sqrt{L_i} & \text{if } L_i \geq k^2 \end{cases} \tag{50}$$

Incorporating this abbreviation yields an update equation similar to Eq. 28:

$$\boldsymbol{\delta} = \left(\sum_i \left[W_i \cdot \mathbf{J}_i^T \cdot \mathbf{R}_i^{-1} \cdot \mathbf{J}_i \right] \right)^{-1} \cdot \left(\sum_i \left[W_i \cdot \mathbf{J}_i^T \cdot \mathbf{R}_i^{-1} \cdot \mathbf{v}_i \right] \right) \tag{51}$$

Note that the scalar weights W_i are evaluated at each iteration from the squared errors L_i computed from the current parameters $\mathbf{x}$. This method of optimizing robust cost functions is called iterated re-weighted least-squares.

Depending on the properties of ρ, when the parameter vector $\mathbf{x}$ is far from the optimum, the weights W_i might all tend to zero, in which case the iteration will not converge or will converge very slowly (i.e. not quadratically). Such failure

modes can sometimes be avoided by first optimizing a convex cost function (e.g. Huber) to find a reasonable estimate of the parameters before switching to a non-convex cost function (e.g. Tukey or Cauchy). The latter cost functions are able to more strongly reject outlier observations by assigning them very low or zero weights.

5 References

- K. Madsen, H.B. Nielsen, O. Tingleff. Methods for Non-linear Least Squares problems. Informatics and Mathematical Modelling, Technical University of Denmark. April 2004.