Some Informal Background on the Gauss-Newton-Levenberg-Marquardt Algorithm

Mike Fienen

October 2, 2023

Linear Least Squares Regression

Starting with a linear relationship between measured and modeled values:

$$\mathbf{y} = \mathbf{X}\beta$$

where $\mathbf{y}$ are the observations, β are the parameters, and $\mathbf{X}$ is the Jacobian matrix of sensitivities, with

$$X_{ij} = \frac{\partial y_i}{\partial b_j}$$

In essence, $\mathbf{X}$ is a codification of the relationship between observations and parameters in the model. In other words, for any set of parameters, one could multiply them by $\mathbf{X}$ and get an estimate of $\mathbf{y}$. In practice, this quantity must be approximated. Traditionally, this has often been accomplished through a finite difference approximation as

$$X_{ij} = \frac{\partial y_i}{\partial b_j} \approx \frac{f\left(b_j + \Delta b_j\right)_i - f\left(b_j\right)_i}{\Delta b_j}$$

where

$$y_i = f\left(b_j\right)_i$$

In ensemble methods, this is further approximated using the vector of differences in y and b available through evaluating the model over an ensemble of parameter values.

Once we have this relationship, we can flip this in the inverse and could estimate parameters from a set of observations as

$$\hat{\beta} = \mathbf{X}^{-1}\mathbf{y}$$

However, this relationship is seldom perfectly one-to-one. Typically there is noise which we can assume is normally distributed and with zero mean. If we represent the error by ϵ we have

$$\mathbf{y} = \mathbf{X}\beta + \epsilon$$

Residuals are defined then as

$$\epsilon = \mathbf{y} - \mathbf{X}\beta$$

The sum of squared errors is

$$\epsilon^T\epsilon = \left(\mathbf{y}-\mathbf{X}\beta\right)^T\left(\mathbf{y}-\mathbf{X}\beta\right)$$

So, like with any function, if we want to find the parameters that minimize it, we take the derivative with respect to the parameters β and set it to zero

$$\frac{\partial \epsilon^T \epsilon}{\partial \beta^T} = (\mathbf{y} - \mathbf{X}\beta)^T (\mathbf{y} - \mathbf{X}\beta) = 0$$

multiplying out the terms

$$(\mathbf{y} - \mathbf{X}\beta)^T (\mathbf{y} - \mathbf{X}\beta)$$

$$\mathbf{y}^T\mathbf{y} - \beta^T\mathbf{X}^T\mathbf{y} - \mathbf{y}^T\mathbf{X}\beta + \beta^T\mathbf{X}^T\mathbf{X}\beta$$

note that $\beta^T\mathbf{X}^T\mathbf{y}$ and $\mathbf{y}^T\mathbf{X}\beta$ both collapse to a scalar and are, in fact, equivalent. So, we can simplify a bit

$$\mathbf{y}^T\mathbf{y} - 2\beta^T\mathbf{X}^T\mathbf{y} + \beta^T\mathbf{X}^T\mathbf{X}\beta$$

Evaluating the derivative

$$\frac{\partial \left(\cancelto{0}{\mathbf{y}^T\mathbf{y}} - 2\beta^T\mathbf{X}^T\mathbf{y} + (2)\beta^T\mathbf{X}^T\mathbf{X}\beta \right)}{\partial \beta^T} = 0$$

$$-2\mathbf{X}^T\mathbf{y} + 2\mathbf{X}^T\mathbf{X}\beta = 0$$

$$\cancel{2}\mathbf{X}^T\mathbf{X}\beta = \cancel{2}\mathbf{X}^T\mathbf{y}$$

$$\boxed{\hat{\beta} = \left(\mathbf{X}^T\mathbf{X}\right)^{-1}\mathbf{X}^T\mathbf{y}}$$

This can be solved in closed form if the problem is linear and is known as the **Newton solution**.

Adding Weights

Returning to the $\mathbf{X}$ matrix, we can think of the $\hat{\beta}$ solution as a mapping from observations $\mathbf{y}$ to estimated parameters $\hat{\beta}$, similar to in the noiseless case above $\left(\hat{\beta} = \mathbf{X}^{-1}\mathbf{y}\right)$. The mapping is provided by $\left(\mathbf{X}^T\mathbf{X}\right)^{-1}\mathbf{X}^T$. The rows of $\mathbf{X}$ each contain the sensitivity of of a single observation to all the parameters. So, in a sense, each parameter in $\hat{\beta}$ comprises a weighted product of all the observations. This is powerful in that it formally shows that the estimates of $\hat{\beta}$ are directly the result of both the observations used and the modeled sensitivity between parameters and the observations.

However, this also puts signficant faith in each observation used. In reality, observations have errors and are not equally informative. So, we can provide a matrix of weights, often but not necessarily diagonal, defined as $\mathbf{Q}$. This matrix is square, with dimensions of $NOBS \times NOBS$. This gets incorporated into the sum of squared errors formulation, replacing

$$\epsilon^T\epsilon = (\mathbf{y} - \mathbf{X}\beta)^T (\mathbf{y} - \mathbf{X}\beta)$$

with

$$\mathbf{\Phi} = (\mathbf{y} - \mathbf{X}\beta)^T \mathbf{Q} (\mathbf{y} - \mathbf{X}\beta)$$

Multiplying this out:

$$\mathbf{\Phi} = (\mathbf{y} - X\beta)^T \mathbf{Q}^{1/2}\mathbf{Q}^{1/2} (\mathbf{y} - X\beta)$$

$$\mathbf{y}^T\mathbf{Q}\mathbf{y} - \beta^T\mathbf{X}^T\mathbf{Q}\mathbf{y} - \mathbf{y}^T\mathbf{Q}\mathbf{X}\beta + \beta^T\mathbf{X}^T\mathbf{Q}\mathbf{X}\beta$$

Again we can simplify since $\beta^T\mathbf{X}^T\mathbf{Q}\mathbf{y}$ still multiplies out to a scalar

$$\mathbf{y}^T\mathbf{Q}\mathbf{y} - 2\beta^T\mathbf{X}^T\mathbf{Q}\mathbf{y} + \beta^T\mathbf{X}^T\mathbf{Q}\mathbf{X}\beta$$

Evaluate the derivative

$$\frac{\partial\left(\cancel{\mathbf{y}^T\mathbf{Q}\mathbf{y}}^{0} - 2\cancel{\beta^T}\mathbf{X}^T\mathbf{Q}\mathbf{y} + (2)\cancel{\beta^T}\mathbf{X}^T\mathbf{Q}\mathbf{X}\beta\right)}{\partial\beta^T} = 0$$

$$-2\mathbf{X}^T\mathbf{Q}\mathbf{y} + 2\mathbf{X}^T\mathbf{Q}\mathbf{X}\beta = 0$$

$$\cancel{2}^{1}\mathbf{X}^T\mathbf{Q}\mathbf{X}\beta = \cancel{2}^{1}\mathbf{X}^T\mathbf{Q}\mathbf{y}$$

$$\boxed{\hat{\beta} = \left(\mathbf{X}^T\mathbf{Q}\mathbf{X}\right)^{-1}\mathbf{X}^T\mathbf{Q}\mathbf{y}}$$

Nonlinear Least-Squares—the Gauss-Newton Algorithm

If the relationship, continuing in the weighted case $\mathbf{y} = \mathbf{X}\mathbf{Q}\beta + \epsilon$ is not linear, it gets expressed as

$$\mathbf{y} = f\left(\mathbf{X}\mathbf{Q}, \beta\right)$$

This can get expanded in a Taylor series to linearize it and then iterate using similar equations

$$\boxed{\hat{\beta}_{i+1} = \hat{\beta}_i + \left(\mathbf{X}^T\mathbf{Q}\mathbf{X}\right)^{-1}\mathbf{X}^T\mathbf{Q}\left(\mathbf{y} - \mathbf{X}\hat{\beta}_\mathbf{i}\right)}$$

This keeps iterating from some initial estimate until the difference between subsequent estimates of $\hat{\beta}$ decreases to within a tolerance or when $\mathbf{X}$, which is calculated as a function of $\hat{\beta}_i$, changes less than a tolerance from one iteration to another.

The Levenberg-Marquardt Adjustment

Given the relationship

$$\boxed{\hat{\beta}_{i+1} = \hat{\beta}_i + \left(\mathbf{X}^T\mathbf{Q}\mathbf{X}\right)^{-1}\mathbf{X}^T\mathbf{Q}\left(\mathbf{y} - \mathbf{X}\hat{\beta}_\mathbf{i}\right)}$$

Levenberg and Marquardt, separately, found an improvement that can help the algorithm progress. As it happents, the Newton direction is not always optimal as it progresses through parameter space "downhill" toward an optimum. An option is to form a trust region around the current parameters and assume the problem is linear in that trust region. At the extreme, if the trust region is really large, then the steepest-gradient direction is used. If the trust region is small, the Newton correction to the steepest-descent gets used.

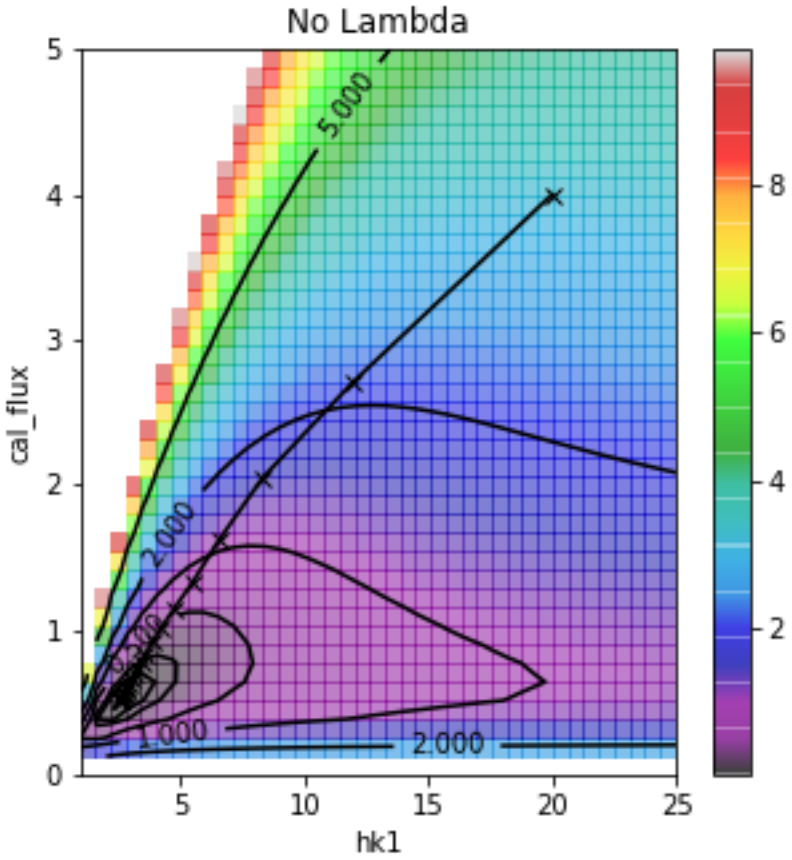
The Levenberg approach implements this by adding a weighted diagonal (identity) matrix to the normal equations like

$$\boxed{\hat{\beta}_{i+1} = \hat{\beta}_i + \left(\mathbf{X}^T\mathbf{Q}\mathbf{X} + \lambda\mathbf{I}\right)^{-1}\mathbf{X}^T\mathbf{Q}\left(\mathbf{y} - \mathbf{X}\hat{\beta}_\mathbf{i}\right)}$$

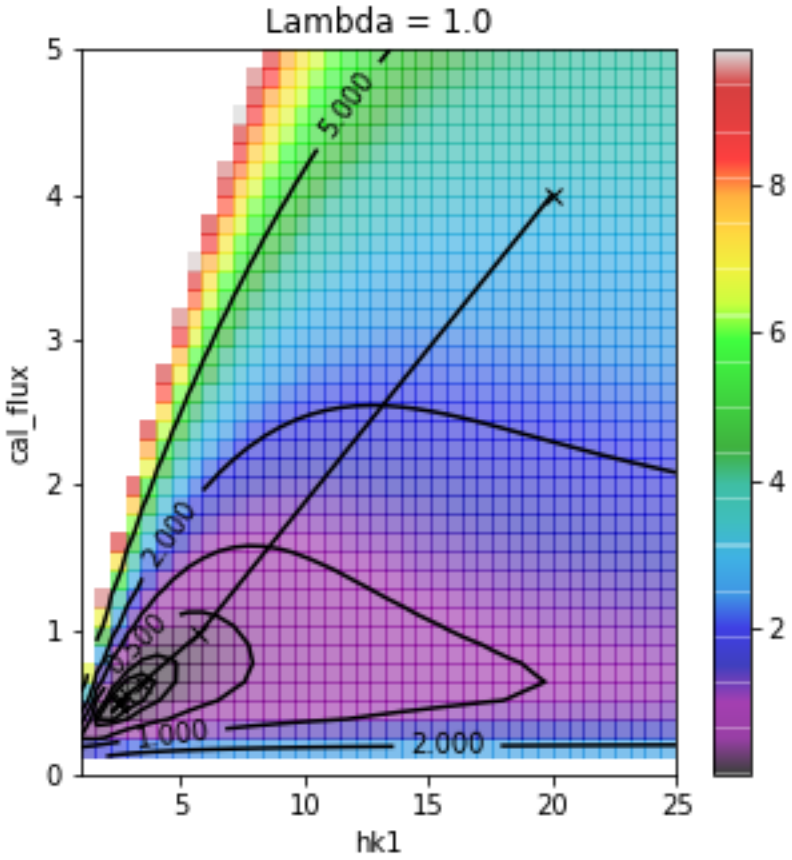
The Marquardt adaptation to Levenberg's method replaces $\mathbf{I}$ with the diagonal of $\mathbf{X}^T\mathbf{X}$ as

$$\boxed{\hat{\beta}_{i+1} = \hat{\beta}_i + \left(\mathbf{X}^T\mathbf{Q}\mathbf{X} + \lambda diag\left(\mathbf{X}^T\mathbf{X}\right)\right)^{-1}\mathbf{X}^T\mathbf{Q}\left(\mathbf{y} - \mathbf{X}\hat{\beta}_\mathbf{i}\right)}$$

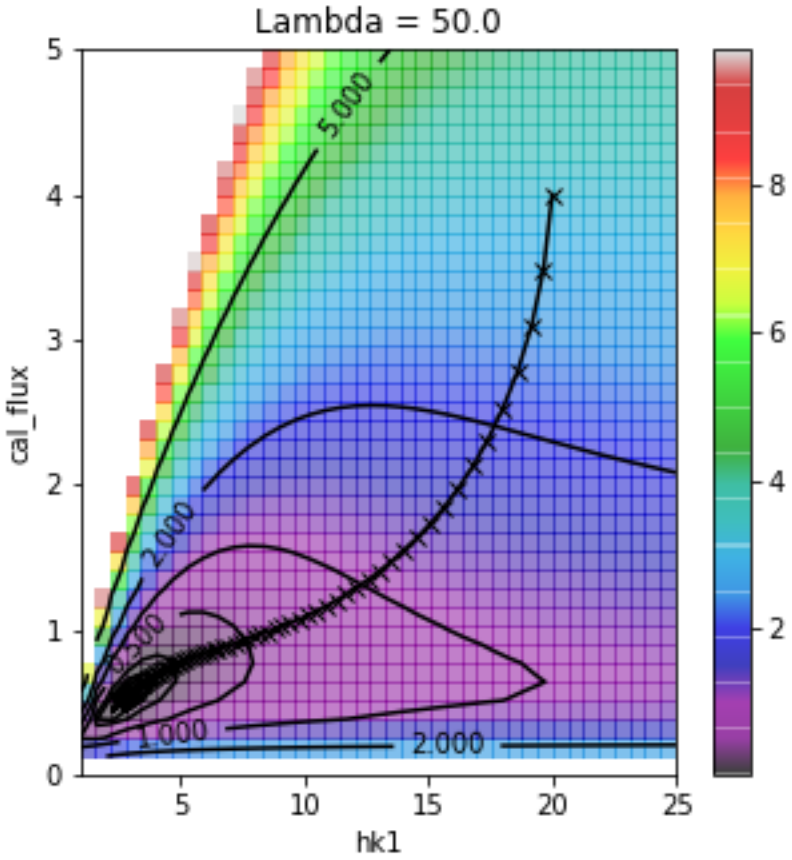
If λ is small, very little adjustment from the Newton direction takes place.



If λ is moderate, the descent direction moves toward steepest descent somewhat.



However, if λ is large, then $\left(\mathbf{X}^T\mathbf{Q}\mathbf{X} + \lambda diag\left(\mathbf{X}^T\mathbf{X}\right)\right)^{-1} \rightarrow \left(\cancel{\mathbf{X}^T\mathbf{Q}\mathbf{X}} + \lambda diag\left(\mathbf{X}^T\mathbf{X}\right)\right)^{-1}$.



A second impact of this is scaling the steepest descent direction $\mathbf{X}^T$. If λ is small, again, the step size implied by $\left(\mathbf{X}^T\mathbf{Q}\mathbf{X} + \lambda diag\left(\mathbf{X}^T\mathbf{X}\right)\right)^{-1}\mathbf{X}^T\mathbf{Q}$ doesn't change much, but if λ is large, $\left(\lambda diag\left(\mathbf{X}^T\mathbf{X}\right)\right)^{-1}\mathbf{X}^T\mathbf{Q}$ behaves like dividing the gradient by λ. This decreases the step size. There is an advantage to this over unaltered steepest descent in that overshooting (hemstitching) is decreased.

In practice, λ is helpful, but it's difficult to know what value is appropriate. PEST and PEST++ try multiple values at each iteration and, in each case, choose the value of λ that reduces $\mathbf{\Phi}$ the most.