Data Reduction Notes

Tobias Marriage

Feb 2011

1 Errors

1.1 Errors on Raw Data

1.1.1 Counts

In some experiments, the data are counts. For instance in the classical Rutherford experiment, alpha particles were counted. When studying nuclear processes, product gamma rays may be counted. When studying galaxy populations, you may bin (count) galaxies in brightnesses intervals. In these cases, the data follow Poisson statistics: for a given count N, $\sigma_N^2 = N$.

2 Linear Modeling

Let $\vec{d}$ be an n-long vector of data that we model with a linear model $M\vec{p}$, where $vecp$ is an m-long vector of parameters and M is an $n \times m$ projection matrix which oper parameters p into the data space. Furthermore, let the noise covariance matrix N be an $n \times n$ diagonal matrix with the data's variance σ^2 on the diagonal:

$$N = \begin{bmatrix} \sigma_1^2 & & 0 \\ & \ddots & \\ 0 & & \sigma_n^2 \end{bmatrix}. \tag{1}$$

The goodness of fit for this model is evaluated using the chi-squared statistic:

$$\chi^2(\vec{p}) = (\vec{d} - M\vec{p})^T N^{-1} (\vec{d} - M\vec{p}) = \sum_i \frac{(d_i - [M\vec{p}]_i)^2}{\sigma_i^2} \tag{2}$$

The quantity is $\vec{d} - M\vec{p}$ is called the "residual". The chi-squared statistic is the square of this residual over the variance. As the sum in Eq 2 suggests, for properly estimated errors σ_i^2, a good fit minimizes chi-squared and has a value close to n, the so-called number of "degrees of freedom".

In order to minimize the chi-squared with respect to the linear model parameters, we simply set the derivative of Eq 2 with respect to these parameters to zero:

$$\begin{aligned} \frac{d\chi}{d\vec{p}} &= 2M^T N^{-1} (\vec{d} - M\vec{p}) = 0 \\ \vec{p} &= (M^T N^{-1} M)^{-1} M^T N^{-1} \vec{d} \end{aligned} \tag{3}$$

3 Non-linear Modeling

To interpret a dataset with measurements $\{d_i\}$, we have a model which, given n model parameters $\vec{p}$, gives corresponding values $\{m_i(\vec{p})\}$. The chi-squared for this model is

$$\chi^2(\vec{p}) = \sum_i \frac{[d_i - m_i(\vec{p})]^2}{\sigma_i^2}, \tag{4}$$

where σ_i^2 is the estimate for the variance on d_i.

The errors on d_i are assumed uncorrelated. For linear models, the chi-squared function was a quadratic in the parameters, so we can write

$$\chi^2(\vec{p}) = \chi^2(\vec{p_0}) + \frac{d\chi^2}{d\vec{p}}(\vec{p_0})(\vec{p} - \vec{p_0}) + \frac{1}{2}(\vec{p} - \vec{p_0})^T \frac{d^2\chi^2}{d^2\vec{p}}(\vec{p_0})(\vec{p} - \vec{p_0}), \tag{5}$$

where $\vec{p_0}$ is an arbitrary starting point in parameter space. The minimizing "jump" in parameter space, $\delta\vec{p} = (\vec{p_{min}} - \vec{p_0})$, is given by

$$\begin{aligned} 0 &= \frac{d\chi^2}{d\vec{p}}(\vec{p_0}) + \frac{d^2\chi^2}{d^2\vec{p}}(\vec{p_0})\delta\vec{p} \\ \delta\vec{p} &= -\left(\frac{d^2\chi^2}{d^2\vec{p}}(\vec{p_0})\right)^{-1} \frac{d\chi^2}{d\vec{p}}(\vec{p_0}) \end{aligned} \tag{6}$$

Note that in general the first derivative of chi-squared is a vector (the gradient), and the second derivative is a matrix (the hessian).

Many times, however, the chi-squared function is not purely quadratic. This is the case for non-linear models. In these cases, the above prescription for a purely quadratic function can fail. In these cases, we must follow our nose, potentially taking many steps around parameter space to find the minimum of the chi-squared. One easy prescription is to follow the gradient of chi-squared. The step in parameter space is then

$$\delta\vec{p} = -\vec{C}\frac{d\chi^2}{d\vec{p}}(\vec{p_0}), \tag{7}$$

where $\vec{C}$ is a constant vector of length n (same as $\vec{p}$). In general, we can use both step formulae (Eq 6 and Eq 7) to explore the parameter space in search of the best fit values.

3.1 Gradient and Hessian

The next step towards an algorithm for non-linear model fitting is constructing the gradient vector and hessian matrix of the chi-squared function (Eq 4).The gradient is straight forward. The j^{th} element of this vector is

$$\frac{\partial\chi^2}{\partial p_j}(\vec{p_0}) = -2\sum_i \frac{[d_i - m_i(\vec{p_0})]}{\sigma_i^2}\left(\frac{\partial m_i}{\partial p_j}(\vec{p_0})\right). \tag{8}$$

And the $(j,k)^{th}$ element of the hessian is

$$\begin{aligned} \frac{\partial^2\chi^2}{\partial p_j \partial p_k}(\vec{p_0}) &= -2\sum_i \sigma_i^{-2}\left(\frac{\partial m_i}{\partial p_j}(\vec{p_0})\right)\left(\frac{\partial m_i}{\partial p_k}(\vec{p_0})\right) + \frac{[d_i - m_i(\vec{p_0})]}{\sigma_i^2}\left(\frac{\partial^2 m_i}{\partial p_j \partial p_k}(\vec{p_0})\right) \\ &\approx -2\sum_i \sigma_i^{-2}\left(\frac{\partial m_i}{\partial p_j}(\vec{p_0})\right)\left(\frac{\partial m_i}{\partial p_k}(\vec{p_0})\right). \end{aligned} \tag{9}$$

The second term in the sum on the first line tends to cancel out for a good model near the chi-squared minimum: the quantity $d_i - m_i(\vec{p}_0)$ will be randomly positive and negative across d_i such that its sum over i should be small (Why isn't this the case for Eq 8? My hypothesis is that both of these terms are small near the minimum and it is the first term in the hessian which drives the final steps of the search.). The second term is also supposedly a numerical nuisance, so we drop it for the effective expression of the hessian.

3.2 Levenburg-Marquardt

The final step towards solving a non-linear least squares system is providing a prescription for stepping through parameter space and in particular choosing the constant in Eq 7. The Levenberg-Marquardt approach is the classic way to do this [1]. In this approach the constant for gradient decent is chosen to be proportional to the inverse of the second derivative with respect to the parameter in question. Following this prescription, Eq 6 and Eq 7 can be written

$$-\begin{bmatrix} \frac{\partial^2\chi^2}{\partial p_1 \partial p_1} & \cdots & \frac{\partial^2\chi^2}{\partial p_1 \partial p_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2\chi^2}{\partial p_n \partial p_1} & \cdots & \frac{\partial^2\chi^2}{\partial p_n \partial p_n} \end{bmatrix} \begin{bmatrix} \delta p_1 \\ \vdots \\ \delta p_n \end{bmatrix} = \begin{bmatrix} \frac{\partial\chi^2}{\partial p_1} \\ \vdots \\ \frac{\partial\chi^2}{\partial p_n} \end{bmatrix}, \tag{10}$$

$$\lambda \begin{bmatrix} \frac{\partial^2\chi^2}{\partial p_1 \partial p_1} & & 0 \\ & \ddots & \\ 0 & & \frac{\partial^2\chi^2}{\partial p_1 \partial p_1} \end{bmatrix} \begin{bmatrix} \delta p_1 \\ \vdots \\ \delta p_n \end{bmatrix} = \begin{bmatrix} \frac{\partial\chi^2}{\partial p_1} \\ \vdots \\ \frac{\partial\chi^2}{\partial p_n} \end{bmatrix}, \tag{11}$$

where λ is a constant. Eq 10 and Eq 11 specify, respectively, the quadratic (Eq 6) and gradient (Eq 7) contributions to $\delta\vec{p}$. These equations can be rewritten as a single operation on the gradient

$$\begin{bmatrix} \delta p_1 \\ \vdots \\ \delta p_n \end{bmatrix} = \left(\begin{bmatrix} \frac{\partial^2\chi^2}{\partial p_1 \partial p_1}(-1+\lambda) & \cdots & \frac{\partial^2\chi^2}{\partial p_1 \partial p_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2\chi^2}{\partial p_n \partial p_1} & \cdots & \frac{\partial^2\chi^2}{\partial p_n \partial p_n}(-1+\lambda) \end{bmatrix} \right)^{-1} \begin{bmatrix} \frac{\partial\chi^2}{\partial p_1} \\ \vdots \\ \frac{\partial\chi^2}{\partial p_n} \end{bmatrix}, \tag{12}$$

Setting the matrix on the right hand side to $\boldsymbol{\alpha}(\vec{p}, \lambda)$, the expression simplifies to

$$\delta\vec{p} = \boldsymbol{\alpha}(\vec{p}, \lambda)^{-1} \bigtriangledown_p \chi^2. \tag{13}$$

As a prescription for choosing λ and stepping around parameter space, here is a suggestion from the ever useful Numerical Recipes books [2]

1. Solve Eq 13 for $\delta\vec{p}$.
2. If $\chi^2(\vec{p}+\delta\vec{p}) > \chi^2(\vec{p})$, do *not* set $\vec{p} = \vec{p}+\delta\vec{p}$ (reject the step) and increase λ by a factor of 10. Return to start.
3. If $\chi^2(\vec{p}+\delta\vec{p}) < \chi^2(\vec{p})$, set $\vec{p} = \vec{p}+\delta\vec{p}$ and decrease λ by a factor of 10. Return to start.

You stop iterating this sequence when the quantity $\chi^2(\vec{p}+\delta\vec{p}) - \chi^2(\vec{p})$ is significantly less than one.

3.3 Error Propagation

Let the variance on a dataset $\{d_i\}$ (n long) be given by $\{\sigma_i^2\}$. The variance of a function of the data $p(\{d_i\})$ is given by

$$\sigma_p^2 = \sum_i \left|\frac{\partial p}{\partial d_i}\right|^2 \sigma_i^2 \tag{14}$$

A corollary of this is that, for functions linear in $\{d_i\}$, the fractional variance (σ_p^2/p^2) of the function is the quadrature sum of the fractional variance of the data. This is corollary does not hold for non-linear functions.

Eq 14 can be written in vector form as

$$\sigma_p^2 = \vec{\nabla}_d p \cdot N \cdot \vec{\nabla}_d p, \tag{15}$$

where N is given by Eq 1. More generally we can define the covariance matrix of a set of m dependent variables (e.g., model parameters) as

$$C = \vec{\nabla}_d \vec{p} \cdot N \cdot \vec{\nabla}_d \vec{p} \tag{16}$$

where $\vec{\nabla}_d \vec{p}$ has dimensions of $(n \times m)$. Referring to Eq 3 for an expression for $\vec{p}$, the covariance matrix takes the form

$$C = (M^T N^{-1} M)^{-1}. \tag{17}$$

This expression can be directly related to the Hessian of the chi-squared in Eq 2:

$$\frac{d\chi^2}{d^2\vec{p}} = 2M^T N^{-1} M \equiv 2\alpha. \tag{18}$$

Thus half the Hessian (defined here as α) is the inverse of the covariance matrix. For a non-linear model which is well approximated by linear terms near the minimum of the chi-squared function, the inverse of the α matrix is the formal covariance. This quantity can be calculated at the end of a non-linear model fit in order to give errors – it is often returned by algorithms based on the Levenburg-Marquardt chi-squared minimization.

Generally near the minimum, the chi-squared function can approximated by a second order function:

$$\Delta\chi^2 = \delta\vec{p} \cdot \alpha \cdot \delta\vec{p} \tag{19}$$

where we have used the definition of the matrix α from Eq 18, $\Delta\chi^2 = \chi^2 - \chi^2_{min}$ and $\delta\vec{p} = \vec{p} - \vec{p}_{min}$. In order to find the uncertainty in a parameter δp_1, we allow this parameter to take an arbitrary value and minimize over the other parameters. This latter condition implies that the gradient of χ^2 will be zero along directions besides that corresponding to δp_1.

$$\frac{d\Delta\chi^2}{d\vec{p}} = \alpha \cdot \delta\vec{p} = \begin{bmatrix} c \\ \vdots \\ 0 \end{bmatrix} \tag{20}$$

Furthermore, for a single degree of freedom (δp_1), the change in chi-square (Eq 19) corresponding to the 68% confidence interval is $\Delta\chi^2 = \delta\vec{p} \cdot \alpha \cdot \delta\vec{p} = 1$. This condition constrains the value of c in Eq 20 such that

$$\delta\vec{p}(68\%) = C \begin{bmatrix} 1/\sqrt{C_{11}} \\ \vdots \\ 0 \end{bmatrix}, \tag{21}$$

Table 1: Chi-squared values corresponding to confidence levels (CL) for various numbers of degrees of freedom.

C.L.	1 dof	2 dof	3 dof
68%	1.00	2.30	3.53
95.4%	4.00	6.17	8.02

where we have used the identity $C = \alpha^{-1}$. It then follows that the standard deviation of parameter p_1 is $\sqrt{C_{11}}$.

More generally, one can plot the confidence contours of multiple parameters. Given the constraints of the printed page, two parameter error ellipses are most frequently used and provide insight to degeneracy among parameters. For ν parameters, one constructs a $\nu \times \nu$ sub-covariance matrix by extracting the intersections of rows and columns associated with the parameters of interested. Then one plots chi-squared contours corresponding to the appropriate confidence intervals. See Table 1.

References

[1] Kenneth Levenberg. “A Method for the Solution of Certain Non-Linear Problems in Least Squares”. The Quarterly of Applied Mathematics 2: 164-168 (1944).

[2] William Press et al. *Numerical Recipes in C.* Cambridge: Cambridge University Press, 1992.