Argonne NATIONAL LABORATORY

... for a brighter future

U.S. Department of Energy

THE UNIVERSITY OF CHICAGO

Office of Science

A U.S. Department of Energy laboratory managed by The University of Chicago

A Casual Introduction to Least-squares Fitting: A [mostly] descriptive approach

Brian H. Toby

Outline

- Linear Algebra: a cheap intro
- Least-Squares Minimization
  - Linear
  - Non-linear
- Least-square's weakness: Correlation
- Uncertainty estimation for fitted parameters
- Resistance: one bad point can do you in

Linear Algebra: for solution of simultaneous equations

- Linear Algebra provides a compact way to deal with simultaneous equations:

$A_{11}x_1 + A_{12}x_2 + A_{13}x_3 + \ldots A_{1m}x_m = b_1$
$A_{21}x_1 + A_{22}x_2 + A_{23}x_3 + \ldots A_{2m}x_m = b_2$
$A_{n1}x_1 + A_{n2}x_2 + A_{n3}x_3 + \ldots A_{nm}x_m = b_n$

or equivalently with n equations, $\Sigma_j A_{ij}x_j = b_i$, where we want to find the x_j values knowing A_{ij} and b_i

can be written as **A x** = **b** where
- **A** *is a (n by m) matrix;*
- **b** *is a column vector(or m by 1 matrix)*
- **x** *is a row vector(or 1 by n matrix)*

- Solving for **x**: $\mathbf{A}^{-1}\mathbf{A}\,\mathbf{x} = \mathbf{A}^{-1}\mathbf{b}$ *or* $\mathbf{x} = \mathbf{A}^{-1}\mathbf{b}$

Linear Algebra

- Matrix **A** with m rows and n columns is composed of $n\times m$ elements A_{ij}:

$$\mathbf{A} = \begin{pmatrix} A_{11} & A_{21} & \ldots & A_{m1} \\ A_{12} & A_{22} & \ldots & A_{m2} \\ \ldots & & & \\ A_{1n} & A_{2n} & \ldots & A_{mn} \end{pmatrix}$$

- Matrix multiplication: $\mathbf{C} = \mathbf{A}\,\mathbf{B}$, $C_{ij} = \Sigma_k A_{ik} B_{kj}$

  Note that in general, $\mathbf{A}\,\mathbf{B} \neq \mathbf{B}\,\mathbf{A}$,

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix}\begin{pmatrix} \alpha & \beta \\ \gamma & \delta \end{pmatrix} = \begin{pmatrix} a\alpha + b\gamma & a\beta + b\delta \\ c\alpha + d\gamma & c\beta + d\delta \end{pmatrix}$$

- Matrix transpose, $\mathbf{A}^T$
  - *if* $\mathbf{B} = \mathbf{A}^T$ *then* $\mathbf{B}_{ij} = \mathbf{A}_{ji}$

$$\mathbf{A} = \begin{pmatrix} A_{11} & A_{21} & A_{31} \\ A_{12} & A_{22} & A_{32} \end{pmatrix} \quad \mathbf{A}^T = \mathbf{B} = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \\ A_{31} & A_{32} \end{pmatrix}$$

Matrix Inversion

- Identity Matrix:
  - diagonal elements = 1
  - off-diagonal elements = 0

$$\mathbf{1} = \begin{pmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \dots & 1 \end{pmatrix}$$

- Inverse of Matrix: $\mathbf{A}^{-1}\,\mathbf{A} = \mathbf{1}$

$$\mathbf{A}^{-1}\mathbf{A} = \begin{pmatrix} A_{11}^{-1} & A_{12}^{-1} & A_{13}^{-1} \\ A_{21}^{-1} & A_{22}^{-1} & A_{23}^{-1} \\ A_{31}^{-1} & A_{32}^{-1} & A_{33}^{-1} \end{pmatrix} \begin{pmatrix} A_{11} & A_{21} & A_{31} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

- Inverse of 3x3 matrix
  (from wikopedia)

$$A^{-1} = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix}^{-1} = \frac{1}{|A|} \begin{bmatrix} ei - fh & ch - bi & bf - ce \\ fg - di & ai - cg & cd - af \\ dh - eg & bg - ah & ae - bd \end{bmatrix}$$

$$|A| = a(ei - fh) - b(di - fg) + c(dh - eg)$$

Summary: Part 1

- You have now had a very brief introduction to linear algebra and should understand the concept of a matrix

Singular Matrices

- If any column (or row) in a matrix is repeated, the matrix cannot be inverted. The same is true if a column (or row) is repeated multiplied by a constant
- A matrix that cannot be inverted is called **singular**

$$\mathbf{A} = \begin{pmatrix} a & b & na \\ d & e & nd \\ g & h & ng \end{pmatrix}^{-1} \qquad |\mathbf{A}| = \begin{vmatrix} a & b & na \\ d & e & nd \\ g & h & ng \end{vmatrix} = a(eng - ndh) - b(dng - ndg) + na(dh - eg) = 0$$

$$A^{-1} = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix}^{-1} = \frac{1}{|A|} \begin{bmatrix} ei - fh & ch - bi & bf - ce \\ fg - di & ai - cg & cd - af \\ dh - eg & bg - ah & ae - bd \end{bmatrix}$$

$$|A| = a(ei - fh) - b(di - fg) + c(dh - eg)$$

Nearly Singular Matrices

- When columns are nearly equivalent, we start subtracting numbers that are almost equal from each other.
  - This is a very bad thing in computer math as it causes round-off errors to be increased.

Round-off error example

∞-precision arithmetic: $64\left(\frac{1}{8}\right)\left(\frac{1}{8}\right) - 1 = 0$

Repeat with two significant-figures: $64(0.13)(0.13) - 1 = 64(0.017) - 1 = 1.1 - 1 = 0.1$

$$\mathbf{A} = \begin{pmatrix} a & b & na \\ d & e & nd \\ g & h & n(g+\delta) \end{pmatrix}^{-1} \qquad |\mathbf{A}| = a(en[g+\delta] - ndh) - b(dn[g+\delta] - ndg) + na(dh - eg)$$

Summary: Part 2

- You should now understand that a singular matrix is one that cannot be inverted
- A matrix that is nearly singular in theory can be inverted, but in practice inversion will be highly inaccurate due to round-off errors

Terminology of Least-Squares

Data: n observations, y_i, measured at independent variable setting x_i

Model: a function that predicts the observations: $Y(x_i,\mathbf{p})$

- Linear Model: $Y(x_i,\mathbf{p}) = p_1 f_1(x_i) + p_2 f_2(x_i) + \ldots$
- Non-linear Model: $Y(x_i,\mathbf{p}) = f(x_i, p_1, p_2, \ldots)$

Parameters: m terms $p_1, p_2, p_3 \ldots p_m$ that determine the values that are computed from the model

Refine: Find values for parameters, **p**, to yield the <u>best fit</u> between the model $Y(x_i,\mathbf{p})$ and observations y_i

Best fit: Means the finding the minimum for $\Sigma\, w_i[y_i - Y(x_i,\mathbf{p})]^2$ where $w_i = [1 / \sigma(y_i)]^2$ (Note: σ is standard uncertainty on y_i)

Linear Least-Squares

Linear Model: $Y(x_i,\mathbf{p}) = p_1 f_1(x_i) + p_2 f_2(x_i) + \ldots = \Sigma_k p_k f_k(x_i)$

- Goal: Find $p_1, p_2, p_3 \ldots p_m$ that minimize $\Sigma_i w_i[y_i - Y(x_i,\mathbf{p})]^2$

  set derivative w/r each parameter to zero: $\partial/\partial p_j\, \Sigma_i w_i[y_i - Y(x_i,\mathbf{p})]^2 = 0$
  Gives m coupled equations: $\Sigma_i w_i\, y_i\, \partial Y/\partial p_j = \Sigma_i w_i\, Y(x_i,\mathbf{p})\, \partial Y/\partial p_j$

  Note that $\partial Y/\partial p_j = f_j(x_i)$ so the m coupled equations become:
  $\Sigma_i w_i\, y_i f_j(x_i) = \Sigma_i w_i\, [\Sigma_k p_k f_k(x_i)]\, f_j(x_i) = \Sigma_k p_k\, \Sigma_i w_i f_k(x_i)\, f_j(x_i)$

  Define: $A_{ij} = f_j(x_i) / \sigma(y_i)$; $b_i = y_i / \sigma(y_i)$
  This gives m coupled equations: $\Sigma_i b_i A_{ij} = \Sigma_k p_k\, \Sigma_i A_{ij} A_{ik}$

  Recast using linear algebra: $\mathbf{A}^T\mathbf{b} = \mathbf{A}^T\mathbf{A}\mathbf{p}$ *or solving for* **p**:
  $(\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{b} = \mathbf{p}$ <u>*This allows the* **p** *values to be determined directly*</u>

Non-Linear Least-Squares (Gauss-Newton)

- With a non-linear model, $Y(x_i,\mathbf{p}) = f(x_i, p_1, p_2, \ldots)$, it is not possible to solve for **p**
- Remembering the Taylor expansion:
  $f(x_i, p+\delta) = f(x_i, p) + \delta(\partial f/\partial p) + \delta^2(\partial^2 f/\partial p^2)/2 + \ldots$
- Multi-parameter Taylor expansion around approximate values for **p**:
  $Y(x_i, p_1+\delta_1, p_2+\delta_2, \ldots) = Y(x_i, p_1, p_2, \ldots) + \Sigma_k\, \delta_k(\partial Y/\partial p_k) + \Sigma_k\, \delta_k^2(\partial^2 Y/\partial p_k^2)/2 + \ldots$
  - as before, set $\partial/\partial p_j\, \Sigma_i w_i[y_i - Y(x_i,\mathbf{p})]^2 = 0$; solve for δ_k

  m coupled equations: $\Sigma_i w_i\, [y_i - Y(x_i,\mathbf{p})]\, (\partial Y/\partial p_j) = \Sigma_k \delta_k\, \Sigma_i w_i\, (\partial Y/\partial p_k)\, (\partial Y/\partial p_j)$

  Define: $A_{ij} = (\partial Y(x_i,\mathbf{p}) / \partial p_j) / \sigma(y_i)$; $b_i = [y_i - Y(x_i,\mathbf{p})] / \sigma(y_i)$
  This gives m coupled equations: $\Sigma_i b_i A_{ij} = \Sigma_k \delta_k\, \Sigma_i A_{ij} A_{ik}$
  Recast using linear algebra: $\mathbf{A}^T\mathbf{b} = \mathbf{A}^T\mathbf{A}\,\boldsymbol{\delta}$ *or* $(\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{b} = \boldsymbol{\delta}$

<u>*Refinement is iterative process, starting from approximate* **p** *values*</u>

More on Least-squares

A *is called the* **Design Matrix**: $A_{ij} = \partial Y/\partial p_j \,/\, \sigma(y_i)$

- $\mathbf{H} = \mathbf{A}^T\mathbf{A}$ *is called the* **Hessian Matrix**
- *The inverse of the Hessian*, $\mathbf{H}^{-1} = (\mathbf{A}^T\mathbf{A})^{-1}$, *is called the* **Covariance Matrix** (*Einstein called it the* **Variance-Covariance Matrix)**
- The Hessian measures, evaluated for all data points, how the model responds to changes in parameters :
  - $H_{ij} = \Sigma_k\, (\partial Y/\partial p_i)\, (\partial Y/\partial p_j) \,/\, \sigma(y_k)$

Argonne NATIONAL LABORATORY

Summary: Part 3

You should now understand

- the difference between linear and non-linear least squares
- why non-linear least squares is iterative and requires starting with approximate values for parameters
- how LS refinement uses weights, differences and depends of the derivatives of w/r to parameters
- commonly used terms: covariance matrix, Hessian matrix

Argonne NATIONAL LABORATORY

Correlation:

- If two (or more) parameters have the same effect on the model, the derivatives are the same and the Hessian is singular
- If two (or more) parameters have very similar effects, the derivatives are nearly the same and the Hessian is nearly singular -- round-off dominates!

When parameters have similar effects on the fit they are said to be correlated

Argonne NATIONAL LABORATORY

Correlation: the Achilles' Heel of Least-Squares

- Least-squares works best with parameters that have very different effects on the model
- If parameters have exactly the same effect on the model (are completely correlated), the **p** values cannot be determined
- Least-squares performs poorly when p_i values have similar effects (are correlated.)

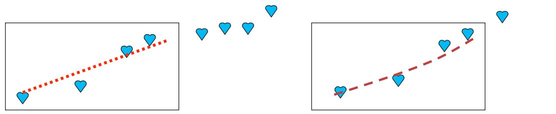
So why use Least-Squares?

- If $\sigma(y_i)$ accurately describes the estimated error (standard uncertainty) in y_i and the model produces an ideal fit to the data ($\chi^2 \cong 1$) then the diagonal elements of the covariance matrix give the standard uncertainty in the parameters: $(\mathbf{A}^T\mathbf{A})^{-1}_{jj} = \sigma(p_j)$
- Least-squares makes optimum use of data -- giving the result with the smallest possible statistically uncertainty.

Argonne NATIONAL LABORATORY

Correlation: Example 1

Fit $y = \underline{n}\sin(x) + \underline{m}x + \underline{b}$ (all $x \ll \pi/2$)



$y = \underline{m}x + \underline{b}$ $\qquad$ $y = \underline{n}\sin(x) + \underline{b}$

Note that $\underline{m}$ and $\underline{n}$ have almost exactly the same effect

The least-squares refinement will be prone to diverge!

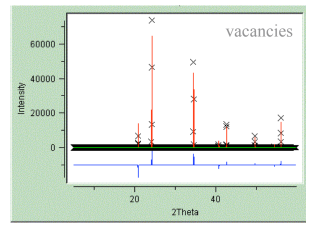
In order to fit both m and n well: data over a wide range in x or extremely precise data are needed

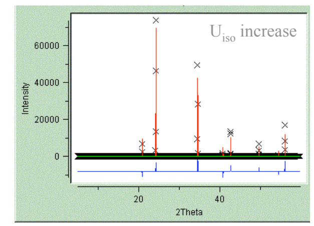
Correlation: Example 2

Occupancies and displacement parameters correlate.

Why? Decreasing occupancy has a similar effect to increasing U_{iso}.

- Example: Simulate change in NaCl X-ray diffraction due to 20% Na vacancies or due to a ×2.8 increase in Na Uiso.

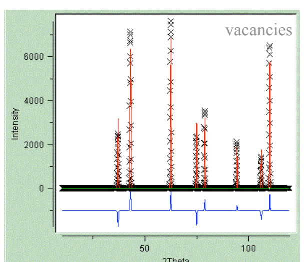


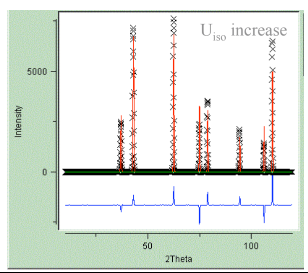


Note: with a smaller Q range, these changes would be even harder to discern

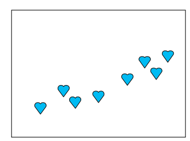
Correlation: Neutrons vs. X-rays

Same example: NaCl with 20% Na vacancies or ×2.8 increase in Na Uiso but now simulated with neutron diffraction (can discern much better)





Correlation: Example 3



Fit $y = \underline{m}x + \underline{b} + \underline{c}$

Note: $\underline{b}$ & $\underline{c}$ *are completely equivalent*

Since the effect of changing $\underline{b}$ & $\underline{c}$ is exactly the same, the Hessian matrix cannot be inverted (is singular).

GSAS treats a singular Hessian by ignoring one of the two identical variables.

Exact correlation: Crystallographic Examples

- When symmetry is lowered, there will be complete correlation between:
  - [Formerly] equivalent unit cell constants
  - Sets of [formerly] equivalent atoms
  - Either manually change the parameters to break the equivalence or vary only one of the set to start.
- Vacancies are equivalent to partial substitution by a “lighter” atom
- Refining all atom positions in space groups with only translational symmetry (arbitrary origin)
- Complete correlation occurs any time two (or more) parameters have exactly the same effect on the fit.

Summary: Part 4

You should now understand

- how exact correlation in parameters leads to a singular Hessian
- why highly correlated parameters leads to a very inaccurate inversion of the Hessian – possibly causing a refinement to fail.

Uncertainty estimation for derived parameters

- Statistical error estimates are computed using covariance matrix
  - If $\sigma(y_i)$ accurately describes the estimated error (standard uncertainty) in y_i and the model produces an ideal fit to the data ($\chi^2 \cong 1$) then the diagonal elements of the covariance matrix give the standard uncertainty in the parameters: $\mathbf{H}^{-1}_{jj} = [\sigma(p_j)]^2$
- For functions of fitted parameters, uncertainty also computed:

$$\text{If } s = \mathbf{f}\,\mathbf{p} \text{ then } \sigma(s) = (\mathbf{H}^{-1})^T\, \mathbf{f}\, \mathbf{H}^{-1}$$

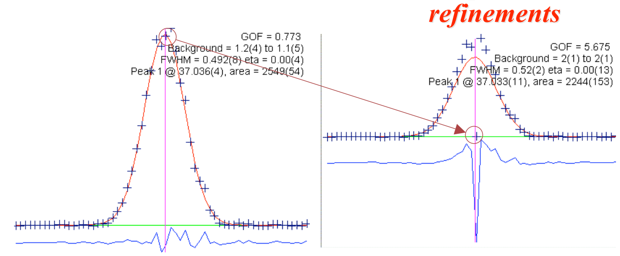
  - Used for bond distances & angles (DISAGL)
  - Can also be used for total composition (from refined occupancies, implemented in GEOMETRY)

Least-Squares is not Resistant

Least-squares weighting assumes:

- uncertainty estimates on data are accurate
- model is accurate (no systematic errors)

“Bad” points skew refinements



Robust-Resistant algorithms limit the maximum leverage a poorly fitting data point may have (for example, by changing weighting.)

Final Summary

You have now seen

- the strength of least-squares: error estimates for refined parameters
- that weights need to reflect the actual uncertainty on a observation or “bad data” can yield a bad fit.

In conclusion:

- Linear algebra simplifies least squares fitting
- Understand how least squares fitting works
- Understand the strengths and weaknesses of least squares

Argonne NATIONAL LABORATORY