Fitting coefficients in a linear equation

Mark Jones

General type of problem

- Measured a set of N data points with values A_1 … A_N with errors σ_1 … σ_N
- Have a model to characterize data: B(p_1 … p_k) where p_k are coefficients that one wants to determine.
- Goodness of the fit determined by which set of p_1 … p_k minimize chi-squared

$$\chi^2 = \frac{1}{N-1}\sum_{j=1}^{N}\left(\frac{A_j - B_j}{\sigma_j}\right)^2$$

- Best solution has derivative of chi-squared with respect to each model parameter equal to zero

$$\frac{\partial \chi^2}{\partial p_1} = 0 \qquad \frac{\partial \chi^2}{\partial p_2} = 0 \qquad \ldots \qquad \frac{\partial \chi^2}{\partial p_k} = 0$$

Example Hall C optimization needs

- The spectrometer optics matrix

$$Y_{tar} = \sum_{k,l,m,n=1}^{nord} a_{klmn} y_{fp}^{k} x_{fp}^{l} yp_{fp}^{m} xp_{fp}^{n}$$

- Calorimeter gain constants

$$E_{cluster} = a_0 * \mathrm{Eblock}_0 + a_1 * \mathrm{Eblock}_1 + \ldots + a_{nmax} * \mathrm{Eblock}_{nmax}$$

Example Hall C optimization needs

- The spectrometer optics matrix

$$Y_{tar} = \sum_{k,l,m,n=1}^{nord} a_{klmn} y_{fp}^{k} x_{fp}^{l} yp_{fp}^{m} xp_{fp}^{n}$$

- Calorimeter gain constants

$$E_{cluster} = a_0 * \mathrm{Eblock}_0 + a_1 * \mathrm{Eblock}_1 + \ldots + a_{nmax} * \mathrm{Eblock}_{nmax}$$

Need to be linear in the coefficients !

Need Minuit to fit : $T(x) = a_o + x^{a_1}$

Example of polynomial fitting

- General polymonial $F(x) = a_0 + a_1 x + a_2 x^2 + \ldots + a_m x^m$

$$\chi^2 = \frac{1}{N - m - 1} \sum_{j=1}^{N} \left(\frac{y_j - F(x_j)}{\sigma_j} \right)^2$$

$$\frac{\partial \chi^2}{\partial a_0} = 0 \qquad \frac{\partial \chi^2}{\partial a_1} = 0 \ldots \frac{\partial \chi^2}{\partial a_m} = 0$$

Derivative of chi-squared with respect to a_0

- General polymonial $F(x) = a_0 + a_1 x + a_2 x^2 + \ldots + a_m x^m$

$$\frac{\partial \chi^2}{\partial a_0} = \frac{2}{N-m-1} \sum_{j=1}^{N} \left(\frac{y_j - F(x_j)}{\sigma_j^2} \right) \left(\frac{\partial F(x_j)}{\partial a_0} \right) = 0 \qquad \frac{\partial F(x_j)}{\partial a_0} = 1$$

$$\sum_{j=1}^{N} \frac{y_j}{\sigma_j^2} = \sum_{j=1}^{N} \frac{a_0 + a_1 x + a_2 x^2 + \ldots + a_m x^m}{\sigma_j^2}$$

Rewrite as: $$\sum_{j=1}^{N} \frac{y_j}{\sigma_j^2} = a_0 \sum_{j=1}^{N} \frac{1}{\sigma_j^2} + a_1 \sum_{j=1}^{N} \frac{x}{\sigma_j^2} + \ldots + a_m \sum_{j=1}^{N} \frac{x^m}{\sigma_j^2}$$

Derivative of chi-squared with respect to a$_M$

- General polymonial $F(x) = a_0 + a_1 x + a_2 x^2 + \ldots + a_m x^m$

$$\frac{\partial \chi^2}{\partial a_m} = \frac{2}{N-m-1} \sum_{j=1}^{N} \left(\frac{y_j - F(x_j)}{\sigma_j^2} \right) \left(\frac{\partial F(x_j)}{\partial a_m} \right) \qquad \frac{\partial F(x_j)}{\partial a_m} = x^m$$

$$\sum_{j=1}^{N} \frac{y_j x^m}{\sigma_j^2} = \sum_{j=1}^{N} \frac{x^m (a_0 + a_1 x + a_2 x^2 + \ldots + a_m x^m)}{\sigma_j^2}$$

Rewrite as:

$$\sum_{j=1}^{N} \frac{y_j x^m}{\sigma_j^2} = a_0 \sum_{j=1}^{N} \frac{x^m}{\sigma_j^2} + a_1 \sum_{j=1}^{N} \frac{x^{(m+1)}}{\sigma_j^2} + \ldots + a_m \sum_{j=1}^{N} \frac{x^{(m+m)}}{\sigma_j^2}$$

Set of linear equations

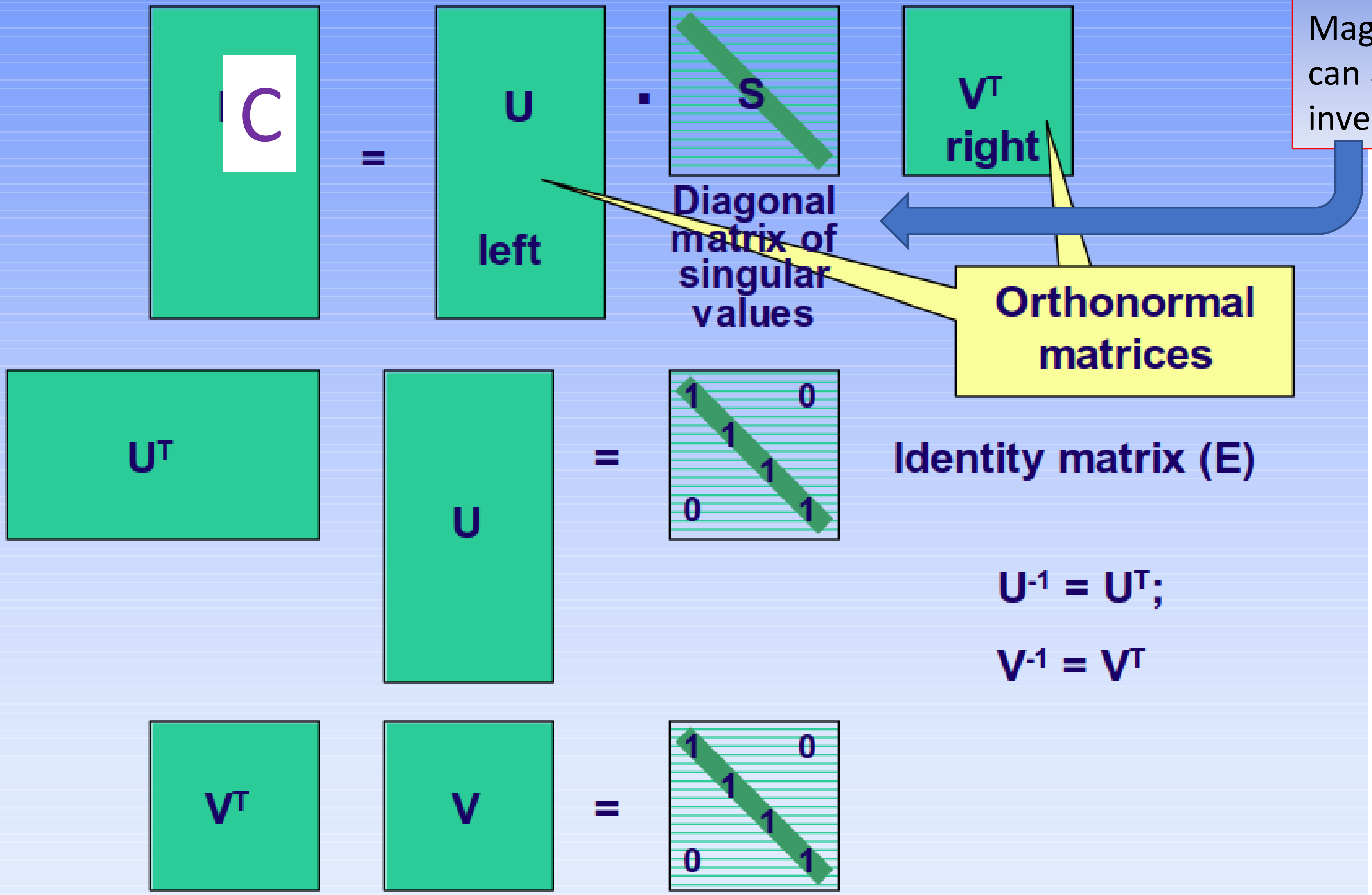
An equation for each $\frac{\partial \chi^2}{\partial a_m} = 0$

$$\boxed{\sum_{j=1}^{N}\frac{y_j}{\sigma_j^2}} = a_0\boxed{\sum_{j=1}^{N}\frac{1}{\sigma_j^2}} + a_1\boxed{\sum_{j=1}^{N}\frac{x}{\sigma_j^2}} + \ldots + a_m\boxed{\sum_{j=1}^{N}\frac{x^m}{\sigma_j^2}}$$

$$\boxed{\sum_{j=1}^{N}\frac{y_j x^m}{\sigma_j^2}} = a_0\boxed{\sum_{j=1}^{N}\frac{x^m}{\sigma_j^2}} + a_1\boxed{\sum_{j=1}^{N}\frac{x^{(m+1)}}{\sigma_j^2}} + \ldots + a_m\boxed{\sum_{j=1}^{N}\frac{x^{(m+m)}}{\sigma_j^2}}$$

- Matrix B = CA with solution A = $C^{-1}B$
- Need a robust way to invert matrix C .
- In general may not be able to invert C.

Singular value decomposition (SVD)



Magic is that S can always be inverted

Orthonormal matrices

Identity matrix (E)

$U^{-1} = U^T$;

$V^{-1} = V^T$

Solve linear equation using Singular Value Decomposition

$CA = B$ $C=USV^T$ therefore $USV^TA = B$

- $U^TUSV^TA = U^TB$ $U^TU = 1$ therefore $SV^TA = U^TB$
- $S^{-1}SV^TA = S^{-1}U^TB$ $S^{-1}S = 1$ therefore $V^TA = S^{-1}U^TB$
- $VV^TA = VS^{-1}U^TB$ $VV^T = 1$ therefore $A = VS^{-1}U^TB$

- ROOT provides class TDecompSVD with methods to calculate V, S and U from C and then solve for A