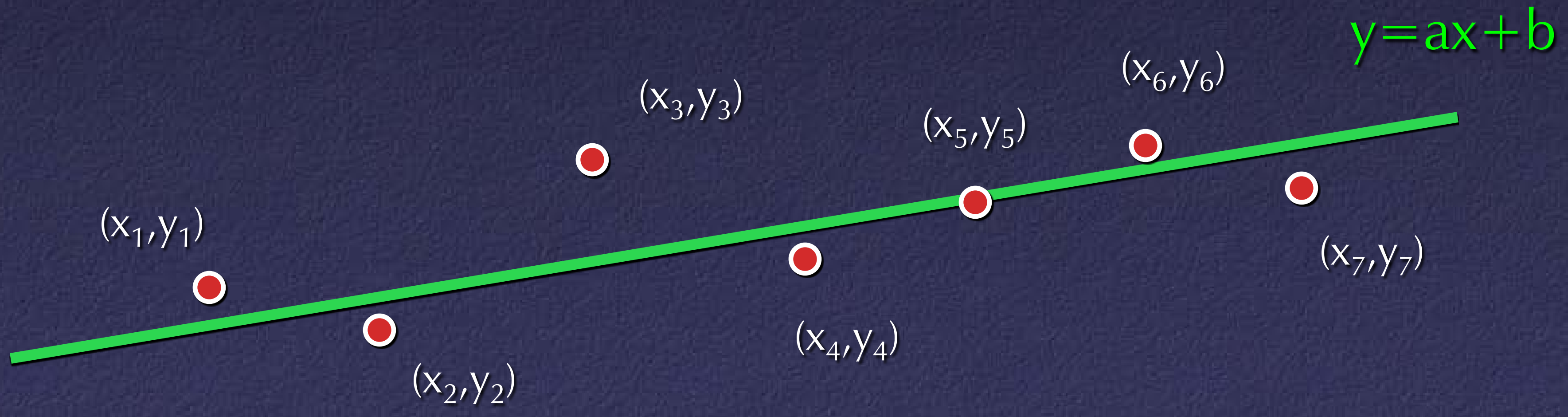
Data Modeling and Least Squares Fitting

COS 323

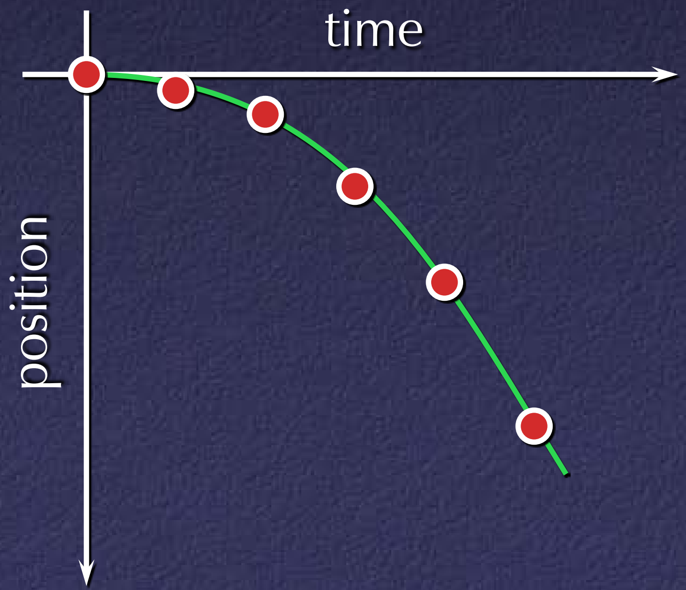
Data Modeling

- Given: data points, functional form, find constants in function
- Example: given (x_i, y_i), find line through them; i.e., find a and b in $y = ax+b$



Data Modeling

- You might do this because you actually care about those numbers…
  - Example: measure position of falling object, fit parabola

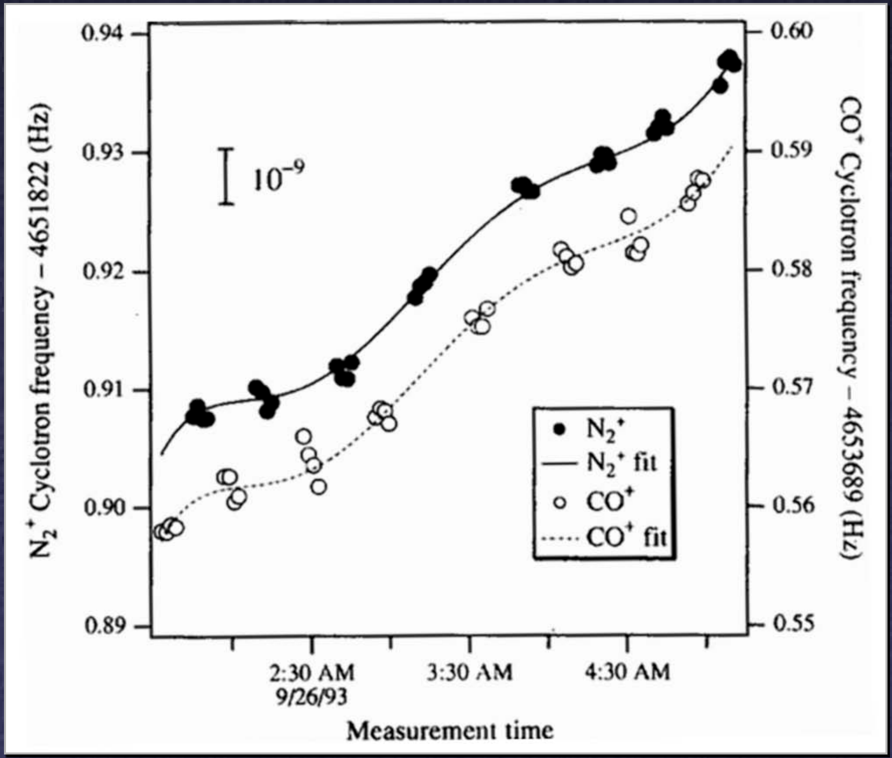


$p = -{}^1/_2\, gt^2$

$\Rightarrow$ Estimate g from fit

Data Modeling

- … or because some aspect of behavior is unknown and you want to ignore it
  - Example: measuring relative resonant frequency of two ions, want to ignore magnetic field drift



Least Squares

- Nearly universal formulation of fitting: minimize squares of differences between data and function
  - Example: for fitting a line, minimize

$$\chi^2 = \sum_i \left(y_i - (ax_i + b)\right)^2$$

    with respect to a and b
  - Most general solution technique: take derivatives w.r.t. unknown variables, set equal to zero

Least Squares

- Computational approaches:
  - General numerical algorithms for function minimization
  - Take partial derivatives; general numerical algorithms for root finding
  - Specialized numerical algorithms that take advantage of form of function
  - Important special case: linear least squares

Linear Least Squares

- General pattern:

$$y_i = a\, f(\vec{x}_i) + b\, g(\vec{x}_i) + c\, h(\vec{x}_i) + \cdots$$

$$\text{Given } (\vec{x}_i, y_i), \text{ solve for } a, b, c, \ldots$$

- Note that *dependence on unknowns* is linear, not necessarily function!

Solving Linear Least Squares Problem

- Take partial derivatives:

$$\chi^2 = \sum_i \left(y_i - a\, f(x_i) - b\, g(x_i) - \cdots\right)^2$$

$$\frac{\partial}{\partial a} = \sum_i -2 f(x_i)\left(y_i - a\, f(x_i) - b\, g(x_i) - \cdots\right) = 0$$

$$a \sum_i f(x_i) f(x_i) + b \sum_i f(x_i) g(x_i) + \cdots = \sum_i f(x_i)\, y_i$$

$$\frac{\partial}{\partial b} = \sum_i -2 g(x_i)\left(y_i - a\, f(x_i) - b\, g(x_i) - \cdots\right) = 0$$

$$a \sum_i g(x_i) f(x_i) + b \sum_i g(x_i) g(x_i) + \cdots = \sum_i g(x_i)\, y_i$$

Solving Linear Least Squares Problem

- For convenience, rewrite as matrix:

$$\begin{bmatrix} \sum_i f(x_i) f(x_i) & \sum_i f(x_i) g(x_i) & \cdots \\ \sum_i g(x_i) f(x_i) & \sum_i g(x_i) g(x_i) & \\ & \vdots & \end{bmatrix} \begin{bmatrix} a \\ b \\ \vdots \end{bmatrix} = \begin{bmatrix} \sum_i f(x_i) y_i \\ \sum_i g(x_i) y_i \\ \vdots \end{bmatrix}$$

- Factor:

$$\sum_i \begin{bmatrix} f(x_i) \\ g(x_i) \\ \vdots \end{bmatrix} \begin{bmatrix} f(x_i) \\ g(x_i) \\ \vdots \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} a \\ b \\ \vdots \end{bmatrix} = \sum_i y_i \begin{bmatrix} f(x_i) \\ g(x_i) \\ \vdots \end{bmatrix}$$

Linear Least Squares

- There's a different derivation of this: overconstrained linear system

$$\mathbf{A}x = b$$

$$\begin{pmatrix} \\ \\ \mathbf{A} \\ \\ \\ \end{pmatrix} \begin{pmatrix} \\ x \\ \\ \end{pmatrix} = \begin{pmatrix} \\ \\ b \\ \\ \\ \end{pmatrix}$$

- A has n rows and m<n columns: more equations than unknowns

Linear Least Squares

- Interpretation: find x that comes "closest" to satisfying Ax=b
  - i.e., minimize b–Ax
  - i.e., minimize || b–Ax ||
  - Equivalently, minimize $\| b-Ax \|^2$ or (b–Ax)·(b–Ax)

$$\min\ (b-\mathbf{A}x)^{\mathrm{T}}\ (b-\mathbf{A}x)$$

$$\nabla\left((b-\mathbf{A}x)^{\mathrm{T}}\ (b-\mathbf{A}x)\right)=-2\mathbf{A}^{\mathrm{T}}(b-\mathbf{A}x)=\vec{0}$$

$$\mathbf{A}^{\mathrm{T}}\mathbf{A}x=\mathbf{A}^{\mathrm{T}}b$$

Linear Least Squares

- If fitting data to linear function:
  - Rows of A are functions of x_i
  - Entries in b are y_i
  - Minimizing sum of squared differences!

$$\mathbf{A} = \begin{bmatrix} f(x_1) & g(x_1) & \cdots \\ f(x_2) & g(x_2) & \cdots \\ & \vdots & \end{bmatrix}, \quad b = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \end{bmatrix}$$

$$\mathbf{A}^{\mathrm{T}}\mathbf{A} = \begin{bmatrix} \sum_i f(x_i)f(x_i) & \sum_i f(x_i)g(x_i) & \cdots \\ \sum_i g(x_i)f(x_i) & \sum_i g(x_i)g(x_i) & \cdots \\ & \vdots & \end{bmatrix}, \quad \mathbf{A}^{\mathrm{T}}b = \begin{bmatrix} \sum_i y_i f(x_i) \\ \sum_i y_i g(x_i) \\ \vdots \end{bmatrix}$$

Linear Least Squares

- Compare two expressions we've derived – equal!

$$\begin{bmatrix} \sum_i f(x_i) f(x_i) & \sum_i f(x_i) g(x_i) & \cdots \\ \sum_i g(x_i) f(x_i) & \sum_i g(x_i) g(x_i) & \cdots \\ & \vdots & \end{bmatrix} \begin{bmatrix} a \\ b \\ \vdots \end{bmatrix} = \begin{bmatrix} \sum_i y_i f(x_i) \\ \sum_i y_i g(x_i) \\ \vdots \end{bmatrix}$$

$$\sum_i \begin{bmatrix} f(x_i) \\ g(x_i) \\ \vdots \end{bmatrix} \begin{bmatrix} f(x_i) \\ g(x_i) \\ \vdots \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} a \\ b \\ \vdots \end{bmatrix} = \sum_i y_i \begin{bmatrix} f(x_i) \\ g(x_i) \\ \vdots \end{bmatrix}$$

Ways of Solving Linear Least Squares

- Option 1:

  for each x_i, y_i

  compute $f(x_i)$, $g(x_i)$, etc.

  store in row i of A

  store y_i in b

  compute $(A^TA)^{-1} A^Tb$

- $(A^TA)^{-1} A^T$ is known as "pseudoinverse" of A

Ways of Solving Linear Least Squares

- Option 2:

  for each x_i, y_i

  compute $f(x_i)$, $g(x_i)$, etc.

  store in row i of A

  store y_i in b

  compute A^TA, A^Tb

  solve $A^TAx = A^Tb$

- These are known as the "normal equations" of the least squares problem

Ways of Solving Linear Least Squares

- These can be inefficient, since A typically much larger than A^TA and A^Tb
- Option 3:

  for each x_i, y_i

  compute $f(x_i)$, $g(x_i)$, etc.

  accumulate outer product in U

  accumulate product with y_i in v

  solve $Ux=v$

Normal Equations

- Solving linear least squares via normal equations can be inaccurate
  - Independent of solution method
  - $\text{cond}(A^T A) = [\text{cond}(A)]^2$
- Next week: SVD
  - More expensive, but more accurate
  - Also allows diagnosing insufficient data

Special Case: Constant

- Let's try to model a function of the form

$$y = a$$

- In this case, $f(x_i)=1$ and we are solving

$$\sum_i [1]\ [a] = \sum_i [y_i]$$

$$\therefore \quad a = \frac{\sum_i y_i}{n}$$

- Punchline: mean is least-squares estimator for best constant fit

Special Case: Line

- Fit to y=a+bx

$$\sum_i \begin{bmatrix} 1 \\ x_i \end{bmatrix} \begin{bmatrix} 1 & x_i \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \sum_i y_i \begin{bmatrix} 1 \\ x_i \end{bmatrix}$$

$$(\mathbf{A}^{\mathrm{T}}\mathbf{A})^{-1} = \begin{bmatrix} n & \Sigma x_i \\ \Sigma x_i & \Sigma x_i^2 \end{bmatrix}^{-1} = \frac{\begin{bmatrix} \Sigma x_i^{\ 2} & -\Sigma x_i \\ -\Sigma x_i & n \end{bmatrix}}{n\Sigma x_i^{\ 2} - (\Sigma x_i)^2}, \quad \mathbf{A}^{\mathrm{T}} b = \begin{bmatrix} \Sigma y_i \\ \Sigma x_i y_i \end{bmatrix}$$

$$a = \frac{\Sigma x_i^{\ 2}\, \Sigma y_i - \Sigma x_i\, \Sigma x_i y_i}{n\Sigma x_i^{\ 2} - (\Sigma x_i)^2}, \quad b = \frac{n\Sigma x_i y_i - \Sigma x_i\, \Sigma y_i}{n\Sigma x_i^{\ 2} - (\Sigma x_i)^2}$$

Weighted Least Squares

- Common case: the (x_i, y_i) have different uncertainties associated with them
- Want to give more weight to measurements of which you are more certain
- Weighted least squares minimization

$$\min \chi^2 = \sum_i w_i \left(y_i - f(x_i)\right)^2$$

- If uncertainty is σ, best to take $w_i = 1/\sigma_i^2$

Weighted Least Squares

- Define weight matrix W as

$$\mathbf{W} = \begin{pmatrix} w_1 & & & & 0 \\ & w_2 & & & \\ & & w_3 & & \\ & & & w_4 & \\ 0 & & & & \ddots \end{pmatrix}$$

- Then solve weighted least squares via

$$\mathbf{A}^{\mathrm{T}}\mathbf{W}\mathbf{A}\,x = \mathbf{A}^{\mathrm{T}}\mathbf{W}\,b$$

Error Estimates from Linear Least Squares

- For many applications, finding values is useless without estimate of their accuracy
- Residual is b – Ax
- Can compute $\chi^2 = (b - Ax) \cdot (b - Ax)$
- How do we tell whether answer is good?
  - Lots of measurements
  - χ^2 is small
  - χ^2 increases quickly with perturbations to x

Error Estimates from Linear Least Squares

- Let's look at increase in χ^2:

$$x \to x + \delta x$$

$$\left(b - \mathbf{A}(x + \delta x)\right)^{\mathrm{T}} \left(b - \mathbf{A}(x + \delta x)\right)$$

$$= \left((b - \mathbf{A}x) - \mathbf{A}\delta x\right)^{\mathrm{T}} \left((b - \mathbf{A}x) - \mathbf{A}\delta x\right)$$

$$= (b - \mathbf{A}x)^{\mathrm{T}}(b - \mathbf{A}x) - 2\delta x^{\mathrm{T}}\mathbf{A}^{\mathrm{T}}(b - \mathbf{A}x) + \delta x^{\mathrm{T}}\mathbf{A}^{\mathrm{T}}\mathbf{A}\delta x$$

$$= \chi^2 - 2\delta x^{\mathrm{T}}(\mathbf{A}^{\mathrm{T}}b - \mathbf{A}^{\mathrm{T}}\mathbf{A}x) + \delta x^{\mathrm{T}}\mathbf{A}^{\mathrm{T}}\mathbf{A}\delta x$$

$$\text{So, } \chi^2 \to \chi^2 + \delta x^{\mathrm{T}}\mathbf{A}^{\mathrm{T}}\mathbf{A}\delta x$$

- So, the *bigger* A^TA is, the *faster* error increases as we move away from current x

Error Estimates from Linear Least Squares

- $C=(A^TA)^{-1}$ is called *covariance* of the data
- The "standard variance" in our estimate of x is

$$\sigma^2 = \frac{\chi^2}{n-m}\mathbf{C}$$

- This is a matrix:
  - Diagonal entries give variance of estimates of components of x
  - Off-diagonal entries explain mutual dependence
- n–m is (# of samples) minus (# of degrees of freedom in the fit): consult a statistician…

Special Case: Constant

$$y = a$$

$$\sum_i [1] \; [a] = \sum_i [y_i]$$

$$\therefore \quad a = \frac{\sum_i y_i}{n}$$

$$\chi^2 = \sum_i (y_i - a)^2$$

$$\sigma^2 = \frac{\sum_i (y_i - a)^2}{n-1} \left[\tfrac{1}{n} \right]$$

$$\sigma_a = \sqrt{\frac{\sum_i (y_i - a)^2}{n-1}} \Big/ \sqrt{n}$$

"standard deviation of mean"

"standard deviation of samples"

Things to Keep in Mind

- In general, uncertainty in estimated parameters goes down slowly: like 1/sqrt(# samples)
- Formulas for special cases (like fitting a line) are messy: simpler to think of $A^TAx=A^Tb$ form
- All of these minimize "vertical" squared distance
  - Square not always appropriate
  - Vertical distance not always appropriate