Data Modeling and Least Squares Fitting

COS 323

Last time: Linear Systems Part 2

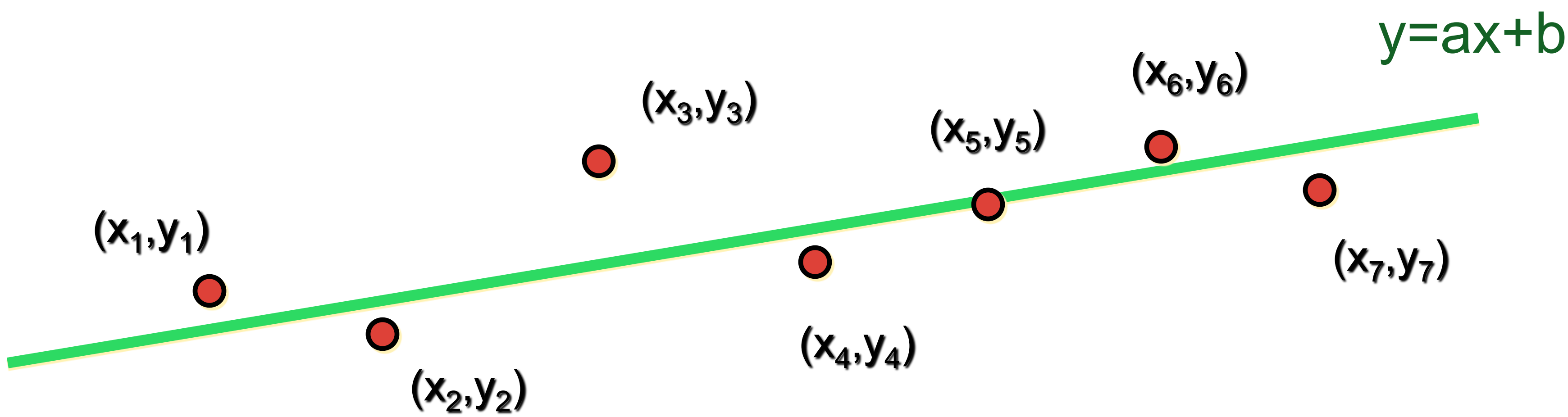
- Iterative refinement
- Fixed-point stationary methods
  - Formulations for root-finding and linear systems
  - Iterative refinement as a stationary method
  - Iterative methods for large systems:
    - Jacobi, Gauss-Seidel, Successive Over-Relaxation
- Sherman-Morrison: Rank-1 updating
- Conjugate gradient (formulating a linear system as an optimization problem)
- Representing sparse systems

Outline

- What is data modeling and why do it?
- Why choose a model that minimizes sum of squared error?
- How to formulate and compute the optimal least-squares linear model?
- Illustrating least-squares with special cases: constant, line
- Weighted least squares
- How to judge the quality of a model?

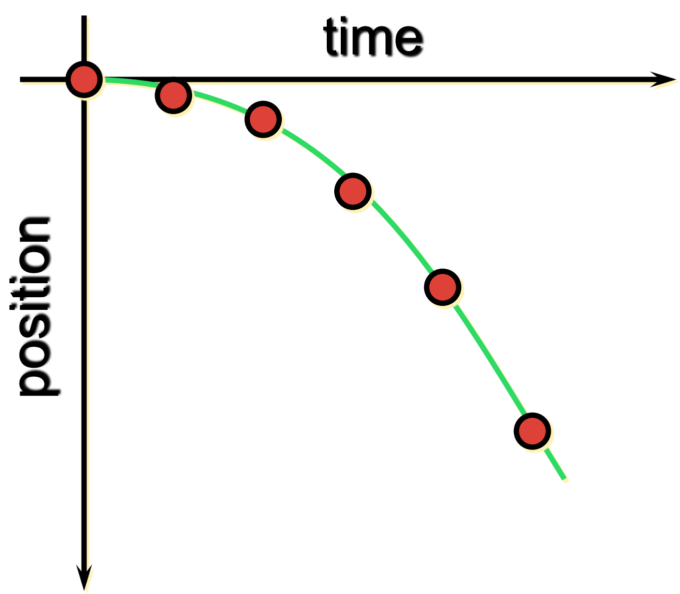
Data Modeling

- 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

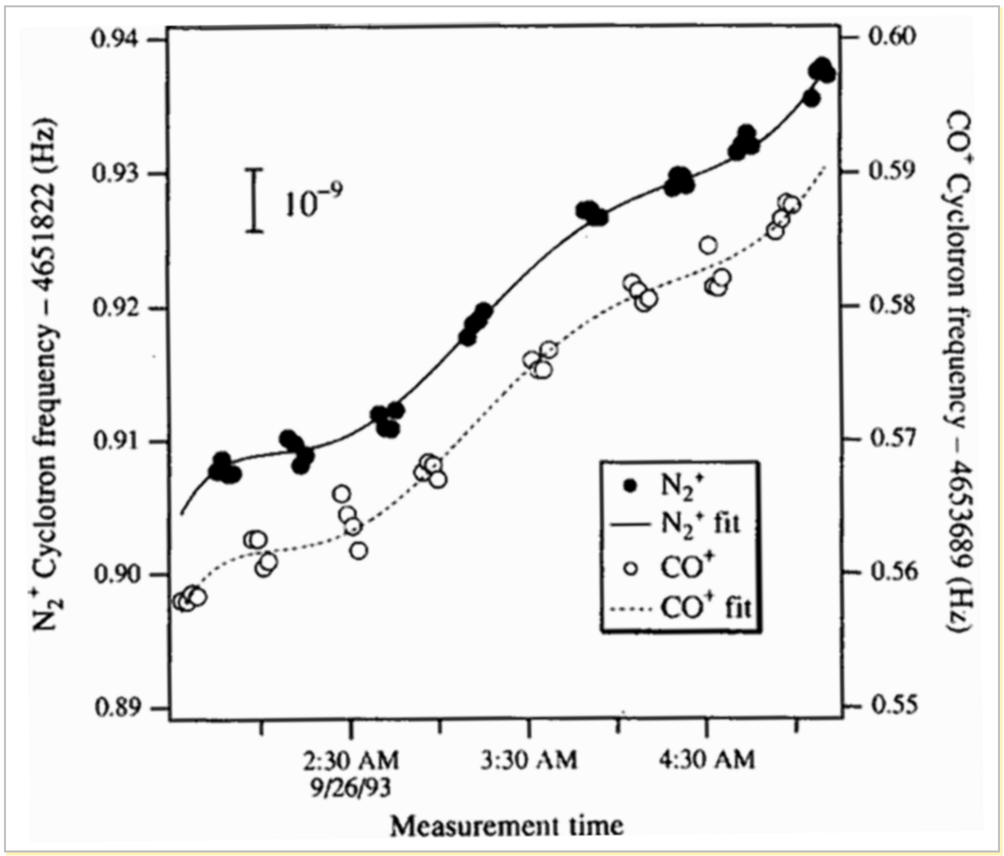


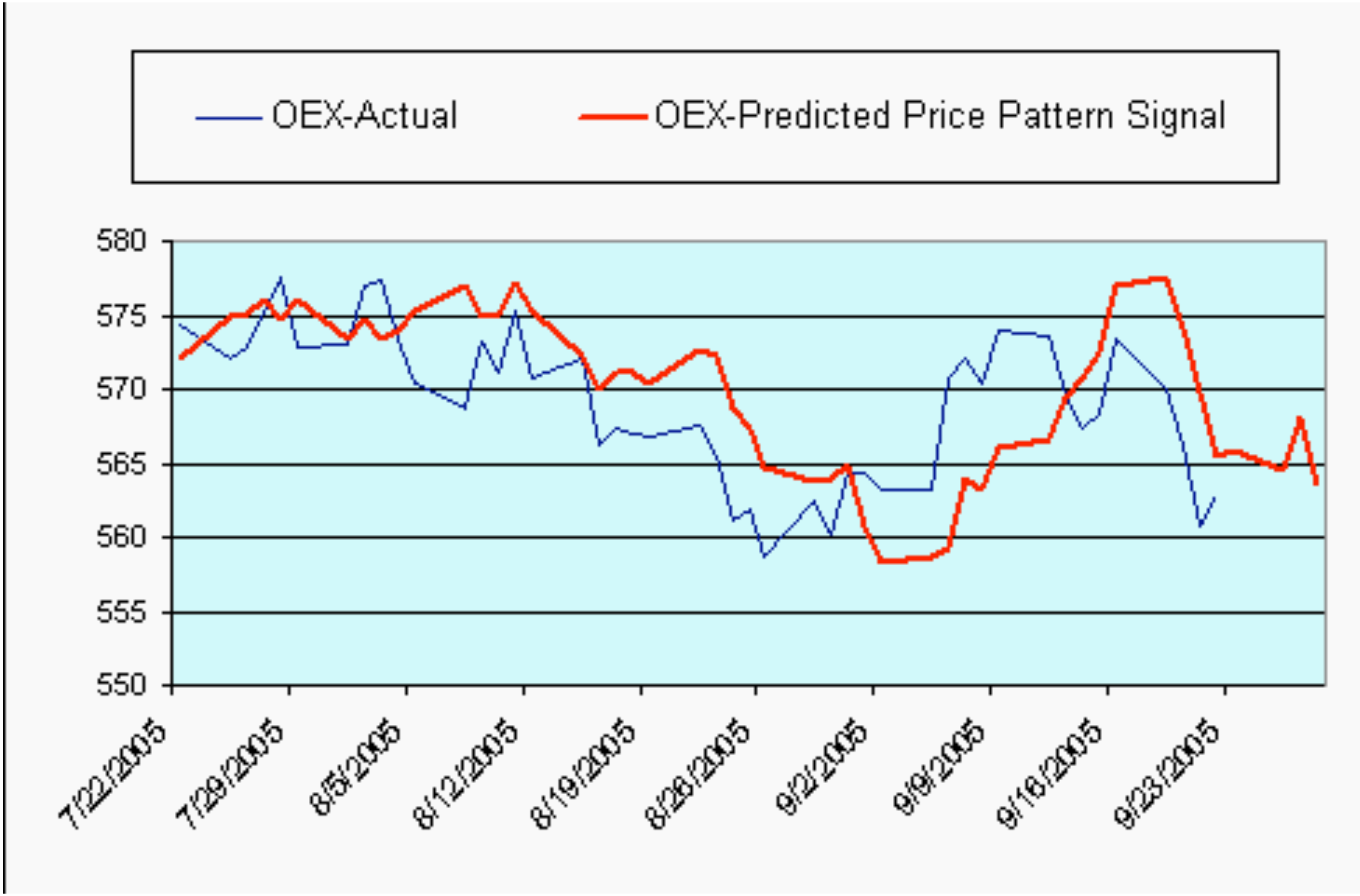
$p = -{}^{1}/_{2}\, gt^2$

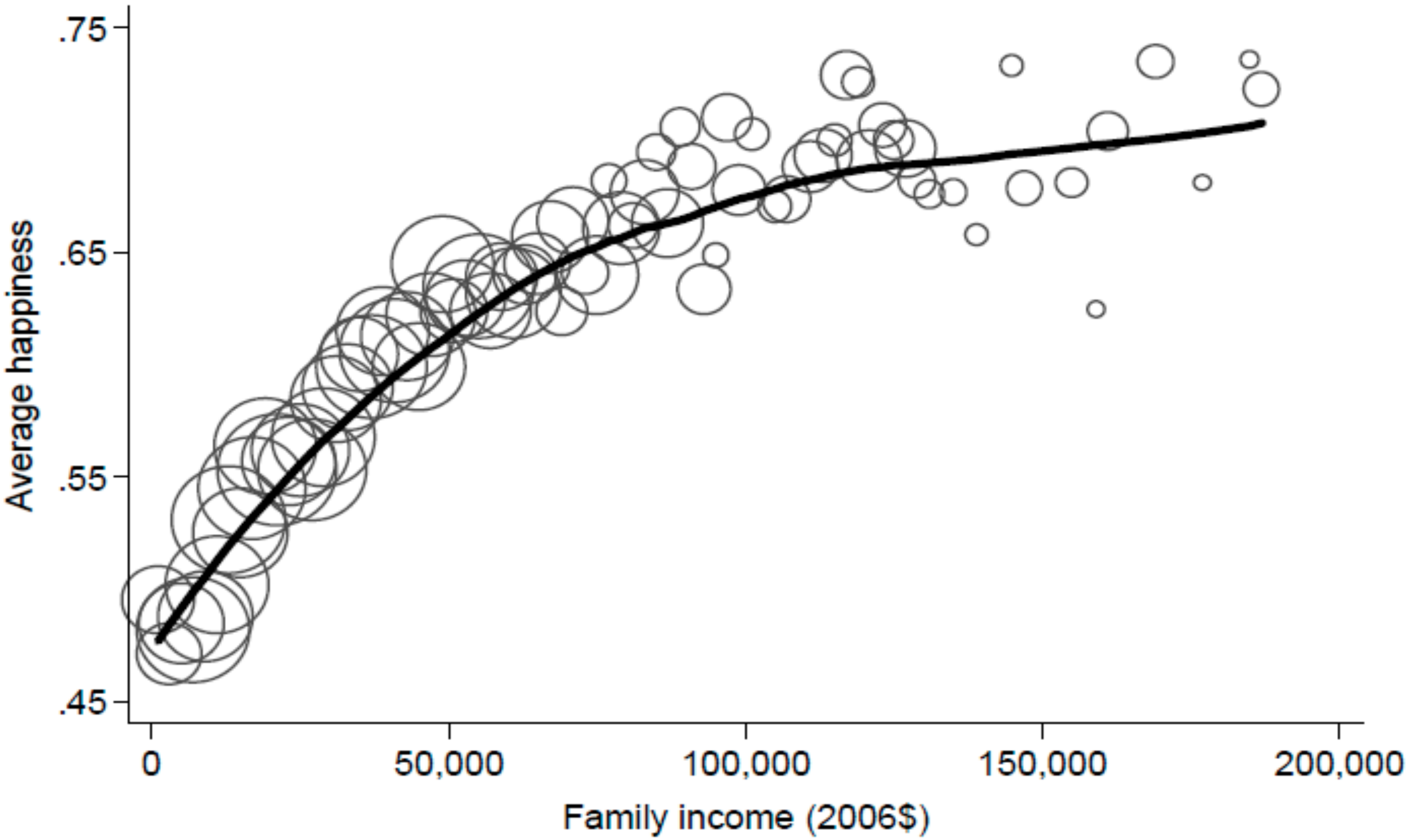
$\Rightarrow$ Estimate g from fit

Data Modeling

- … or because some aspect of behavior is unknown and you want to ignore it




OEX-Actual
OEX-Predicted Price Pattern Signal
580
575
570
565
560
555
550
7/22/2005
7/29/2005
8/5/2005
8/12/2005
8/19/2005
8/26/2005
9/2/2005
9/9/2005
9/16/2005
9/23/2005



Note: 1972 to 2006. Sample size: 41,795. Each circle represents an income range of $2,000 (e.g., $10,001 to $12,000), in 2006$. Its diameter is proportional to the number of people in that range.

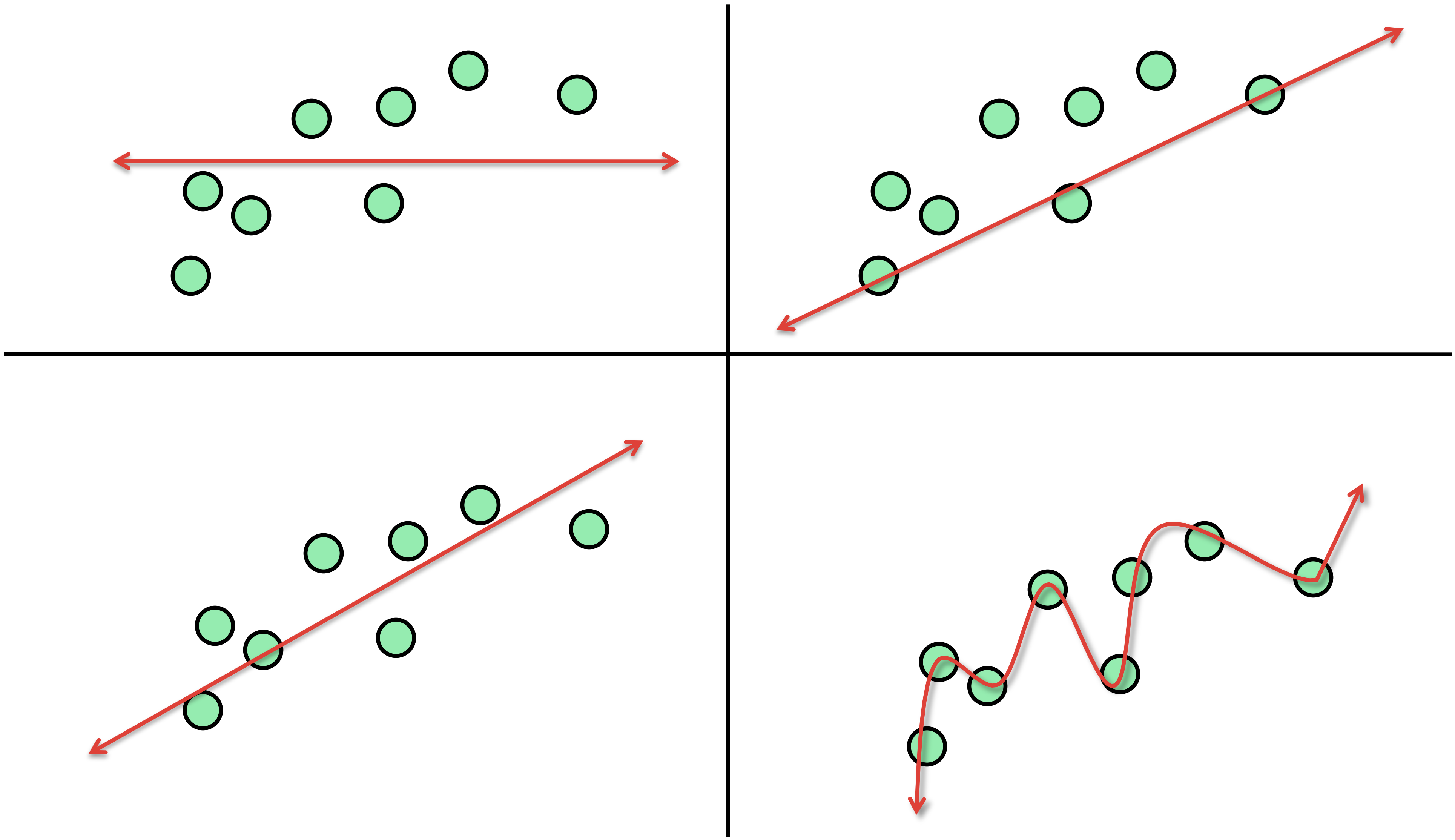
Source: My calculations from General Social Survey data.

Historical context

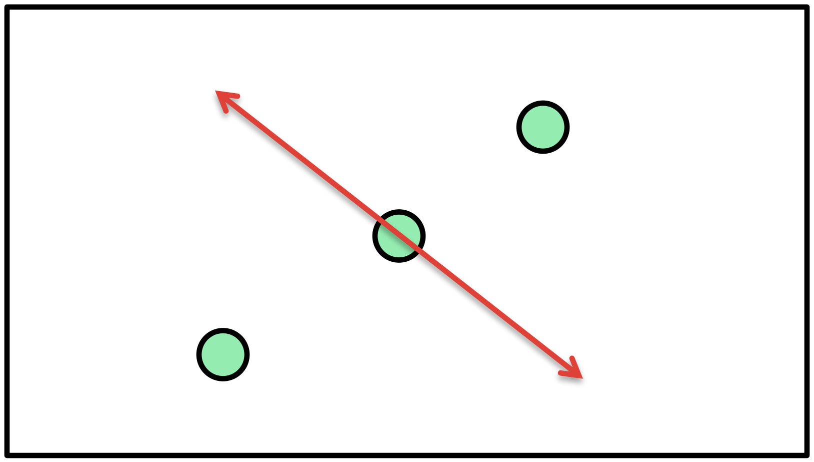
Legendre

Gauss

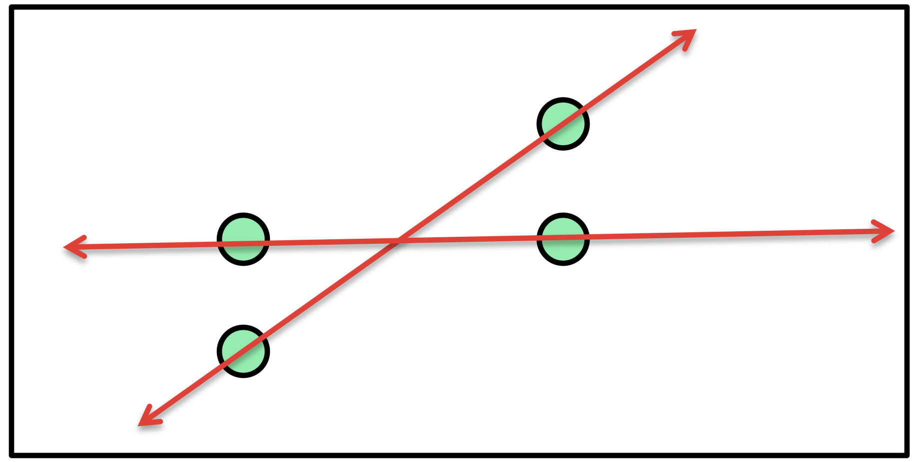
Which model is best?



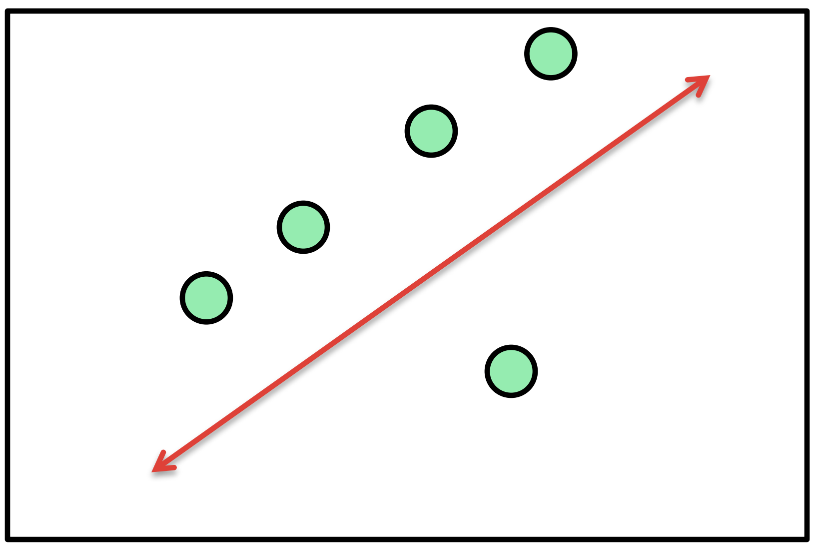
Best-fit lines under different metrics



Sum of residuals



Sum of absolute values of residuals



Maximum error of
any point

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

  - Finds one unique best-fit model for a dataset

Linear Least Squares

- (Also called “Ordinary least squares”)
- General pattern:

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

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

- *Dependence on unknowns (a, b, c…)* is linear, but f, g, etc. might not be!

Linear Least Squares Examples

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

  Given $(\vec{x}_i, y_i)$, solve for $a, b, c, \ldots$
- **Linear** regression:

  $y_i = a * x_i + b$
- **Multiple** linear regression: **x** has many dimensions

  $y_i = a * x_{1i} + b * x_{2i} + c$
- **Polynomial** regression:

  $y_i = a * x_i^2 + b * x_i + c$

How do we compute the model parameters?

Solving Linear Least Squares Problem (one simple approach)

- 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:

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

Alternative perspective: [Approximate] linear system

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

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

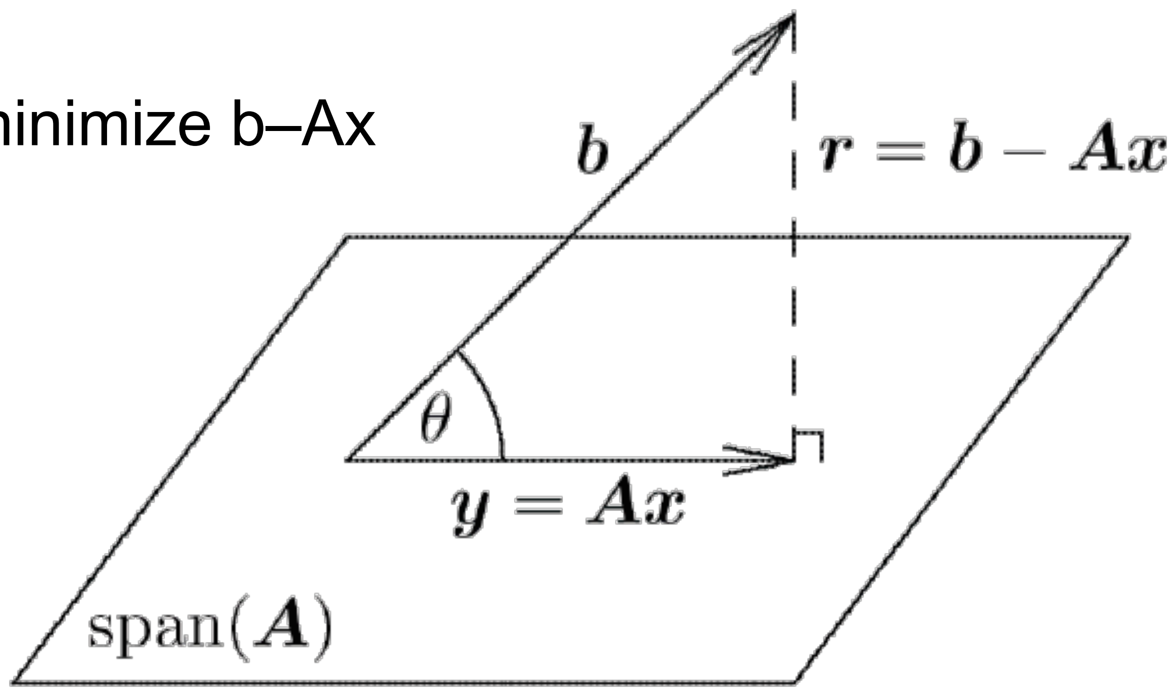
$$\begin{pmatrix} \\ \\ & \mathbf{A} & \\ \\ \\ \end{pmatrix} \begin{pmatrix} \\ x \\ \\ \end{pmatrix} = \begin{pmatrix} \\ \\ b \\ \\ \\ \end{pmatrix}$$

Notation change:
•***A*** is now basis functions computed on observations ($f(x_i)$, $g(x_i)$, …)
•*x* is now model parameters (a, b, c…)
•*b* is now "y"

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

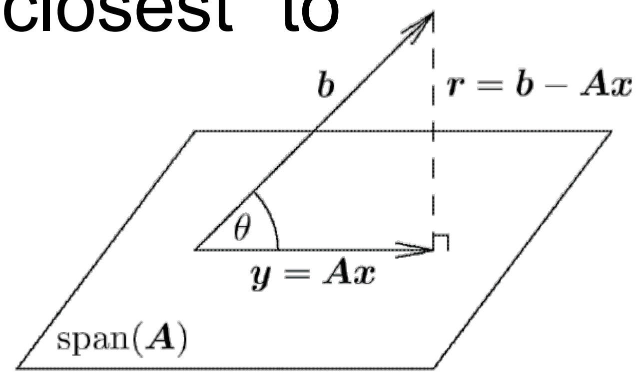
Geometric Interpretation for Over-determined System

- Find the x that comes "closest" to satisfying Ax=b
  - i.e., minimize b–Ax



Geometric Interpretation

- Interpretation: find x that comes "closest" to satisfying Ax=b
  - i.e., minimize b–Ax
  - i.e., minimize || b–Ax ||
  - Equivalently, find x such that r is orthogonal to span(A)



$$0 = \mathbf{A}^T\mathbf{r} = \mathbf{A}^T(\mathbf{b} - \mathbf{Ax})$$

$$\mathbf{A}^T\mathbf{Ax} = \mathbf{A}^T\mathbf{b}$$

Forming the equation

- What are A and b?
  - Row i of A is basis functions computed on x_i
  - Row i of b is y_i

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

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

Minimizing Sum of Squares = Finding Closest Ax in span(A)

- Compare two expressions we've derived: They're equal!

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

Starting from goal of minimizing sum of squares

Starting from goal of finding Ax in span(A) closest to b outside span(A)

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

Great, but how do we solve it?

1: Normal Equations

- Pseudocode:

  for each x_i, y_i

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

  store in column i of A

  store y_i in b

  compute A^TA, A^Tb

  solve $A^TAx=A^Tb$

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

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

2: More efficient normal equations

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

$$\underbrace{\begin{bmatrix} \sum_i f(x_i)f(x_i) & \sum_i f(x_i)g(x_i) & \cdots \\ \sum_i g(x_i)f(x_i) & \sum_i g(x_i)g(x_i) & \cdots \\ & \vdots & \end{bmatrix}}_{\text{U}} \underbrace{\begin{bmatrix} a \\ b \\ \vdots \end{bmatrix}}_{\text{x}} = \underbrace{\begin{bmatrix} \sum_i y_i f(x_i) \\ \sum_i y_i g(x_i) \\ \vdots \end{bmatrix}}_{\text{v}}$$

3: Using the pseudoinverse

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

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

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

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 x = $\mathbf{(A^T A)^{-1} A^T b}$

- $(A^T A)^{-1} A^T$ is known as “pseudoinverse” of A

The Problem with Normal Equations

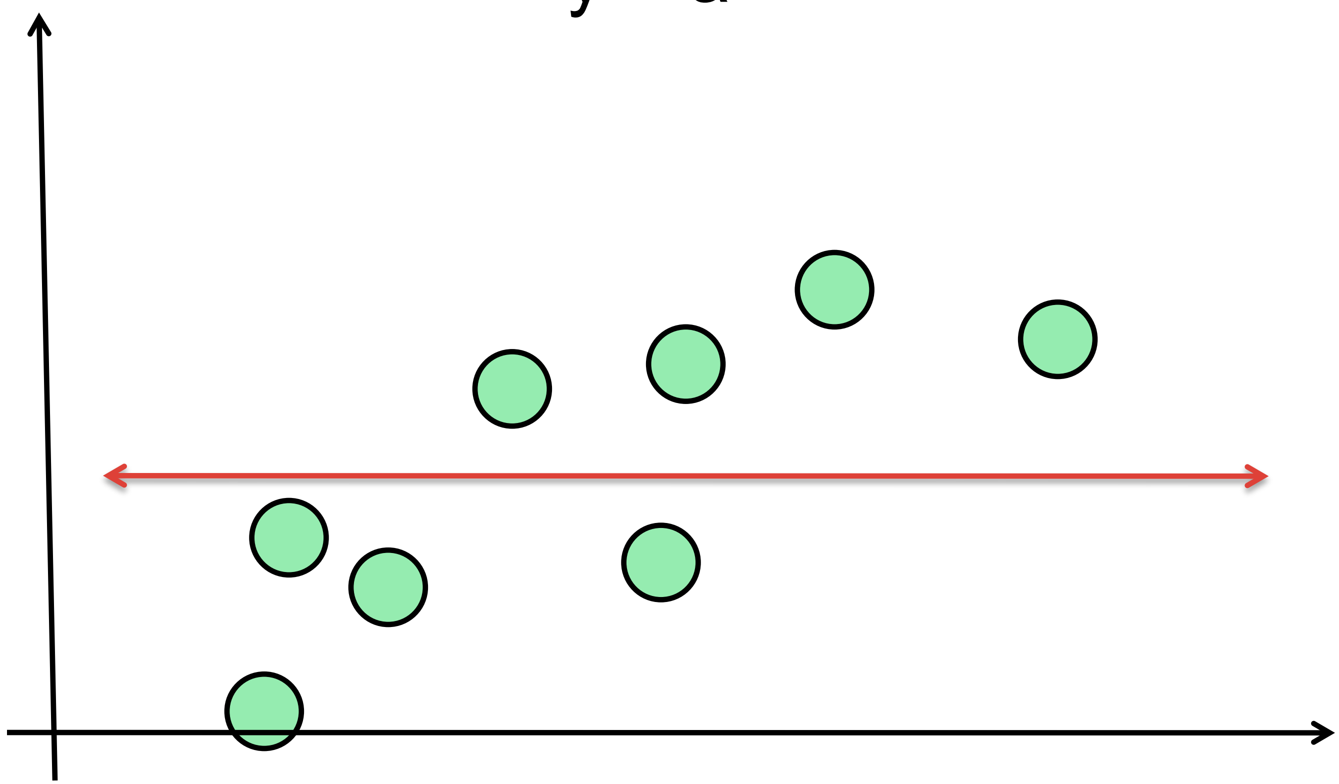
- Involves solving $\mathbf{A^TAx=A^Tb}$
- This can be inaccurate
  - Independent of solution method
  - Remember: $$\frac{\|\Delta x\|}{\|x\|} \leq cond(A)\frac{\|\Delta A\|}{\|A\|}$$
  - $\mathrm{cond}(A^TA) = [\mathrm{cond}(A)]^2$
- Next week: computing pseudoinverse
  - More expensive, but more accurate
  - Also allows diagnosing insufficient data

Special Cases

Special Case: Constant

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

$$y = a$$



Special Case: Constant

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

$$y = a$$

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

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

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

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

Special Case: Line

- Fit to y=a+bx
- $f(x_i)=1$, $g(x_i)=x$. So, solve:

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

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

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

Variant: Weighted Least Squares

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" (stdev) 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

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

Understanding Error and Uncertainty

Error Estimates from Linear Least Squares

- For many applications, finding model is useless without estimate of its 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 (→ **standard variance** of estimate is small)
  - R^2 (“**coefficient of determination**”) is near 1

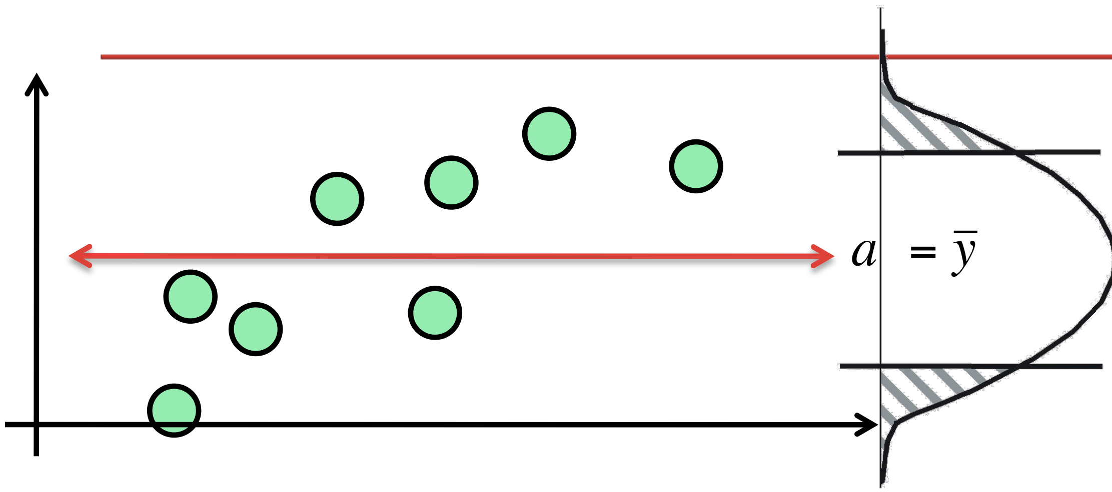
Error Estimates from Linear Least Squares

- $C=(A^TA)^{-1}$ 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: e.g., var(a_0)
  - Off-diagonal entries explain mutual dependence: e.g., cov(a_0, a_1)
- n–m is (# of samples) minus (# of degrees of freedom in the fit): consult a statistician…

Special Case: Error in Constant Model



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

$$\mathbf{C} = \left[\frac{1}{n}\right]$$

$$\text{standard deviation of data}: \quad \sigma = \sqrt{\frac{\sum_i (y_i - a)^2}{n-1}}$$

$$\text{standard error of a}: \quad \sigma_{\mathrm{a}} = \underbrace{\sqrt{\frac{\sum_i (y_i - a)^2}{n-1}} \Bigg/ \sqrt{n}}_{\text{“standard deviation of mean”}}$$

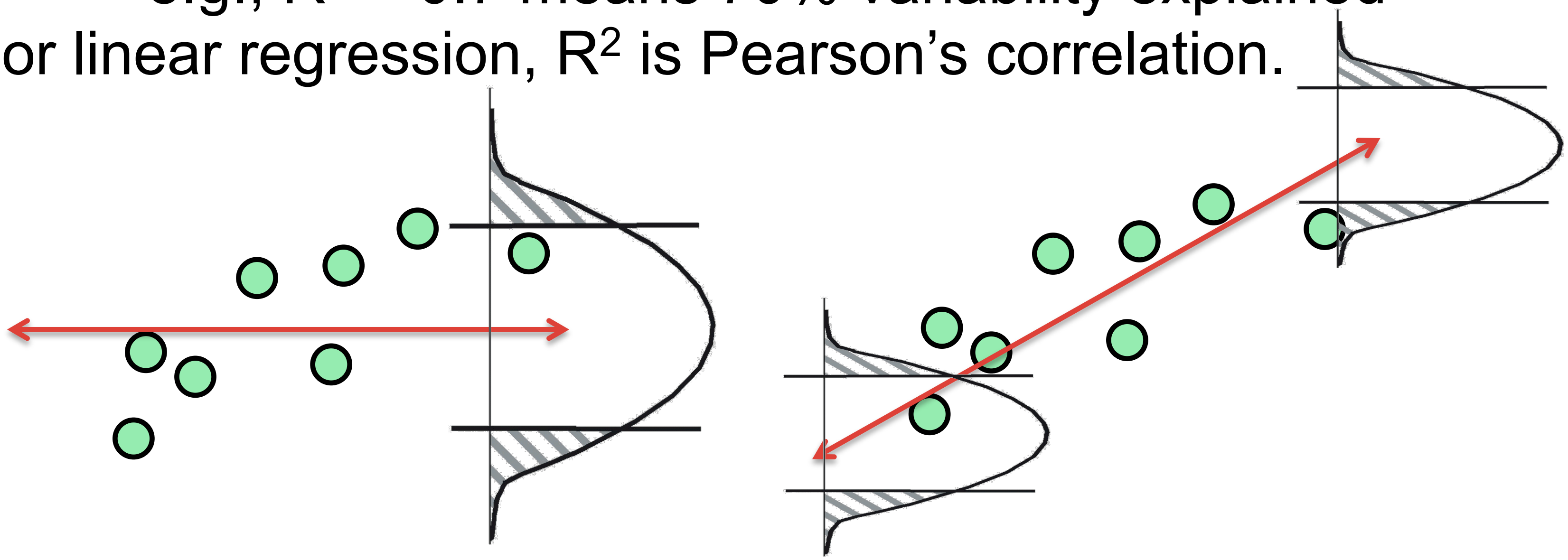
Coefficient of Determination

$$R^2 \equiv 1 - \frac{\chi^2}{\sum_i y_i - \bar{y}}$$

R^2 : Proportion of observed variability that is explained by the model

e.g., $R^2 = 0.7$ means 70% variability explained

For linear regression, R^2 is Pearson's correlation.



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
- Normal equations method often not numerically stable: orthogonal decomposition methods used instead
- Linear least squares is not always the most appropriate modeling technique…

Next time

- Non-linear models
  - Including logistic regression
- Dealing with outliers and bad data
- Practical considerations
  - Is least squares an appropriate method for my data?
- Examples with Excel and Matlab