

Overview

SVD



Solve Square Linear Systems

Can the SVD $A = U\Sigma V^T$ be used to solve *square* linear systems?
At what cost (once the SVD is known)?

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

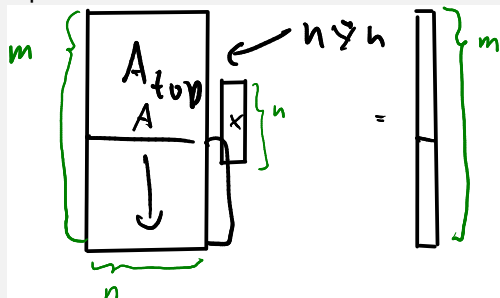
$$U \Sigma V^T \vec{x} = \vec{b}$$

$$\vec{x} = V \underbrace{\Sigma^{-1} U^T \vec{b}}_{\vec{h}} \quad \vec{h}^2$$

$$\left(\begin{array}{c|c|c} \sigma_1 & & \sigma_1 x_1 \\ & \ddots & \vdots \\ & & \sigma_n x_n \end{array} \right)$$

Tall and Skinny Systems

Consider a 'tall and skinny' linear system, i.e. one that has more equations than unknowns:



In the figure: $m > n$. How could we solve that?

"overdetermined"

$$\rightarrow A_{top} x = b_{top} \leftarrow$$

Instead of $Ax=b$, ask for

"Which \vec{x} makes $\|Ax-b\|_2$ as small as possible?"

$(\vec{r}) = A\vec{x}-b$ "residual"

$$\|Ax-b\|_2^2 = \|\vec{r}\|_2^2 = r_1^2 + \dots + r_m^2$$

\hookrightarrow "least squares"

Solving Least-Squares

How can I actually solve a least-squares problem $Ax \cong b$?

Suppose Q is orthogonal: $\|Q\vec{y}\|_2 = \|\vec{y}\|_2$

$$\min_x \|Ax - b\|_2^2 = \|U \Sigma V^T x - b\|_2^2$$

$$\stackrel{\textcircled{1}}{=} \underbrace{\|U^T (U \Sigma V^T x - b)\|_2^2}_{\text{I}} = \underbrace{\|\Sigma V^T x - U^T b\|_2^2}_{\vec{y}' = V^T x}$$

$$= \min_y \|\Sigma \vec{y} - U^T b\|_2^2 \quad \leftarrow$$

$$= \left\| \begin{pmatrix} \sigma_1 & & & \\ & \ddots & & \\ & & \sigma_n & \\ & & & 0 \end{pmatrix} \begin{matrix} y_1 \\ \vdots \\ y_n \end{matrix} - \underbrace{U^T z}_{z'} \right\|$$

$$= (\sigma_1 y_1 - z_1)^2 + \dots + (\sigma_n y_n - z_n)^2$$

$$|\sigma_1| \geq |\sigma_2| \dots \geq |\sigma_n| \geq 0$$

$$\sigma_1 \dots \sigma_k > 0 \quad \sigma_{k+1} \dots \sigma_L = 0$$

$$= \left(\sigma_1 y_1 - z_1 \right)^2 + \dots + \left(\sigma_k y_k - z_k \right)^2 + \left[\begin{array}{l} + \left(\cancel{\sigma_{k+1}} y_{k+1} - z_{k+1} \right)^2 \\ \vdots \\ + \left(\cancel{\sigma_L} y_L - z_L \right)^2 \end{array} \right] \begin{array}{l} z_{k+1} \\ \vdots \\ z_L \end{array}$$

$$y_i = \frac{z_i}{\sigma_i} \Rightarrow \left(\sigma_i y_i - z_i \right)^2 = \left(\cancel{\sigma_i} \frac{z_i}{\cancel{\sigma_i}} - z_i \right)^2 = 0$$

$$\min \| \varepsilon \vec{y} - U^T b \|$$

$$y_i = \begin{cases} (U^T b)_i / \sigma_i & \sigma_i \neq 0 \\ 0 & \sigma_i = 0 \end{cases}$$

doesn't matter, pick

$$\Sigma^{-1} = \begin{pmatrix} 1/\sigma_1 & & \\ & \ddots & \\ & & 1/\sigma_n \end{pmatrix}$$

↖ not allowed

$$\Sigma^+ = \begin{pmatrix} 1/\sigma_1 & & & \\ & \ddots & & \\ & & 1/\sigma_n & \\ & & & 0 \dots 0 \end{pmatrix}$$

↑
"p pseudo inverse of a diagonal"

with that:

$$\vec{y} = \Sigma^T U^T \vec{b}$$

Still wanted: $\|Ax - b\|_2^2$

$$\vec{y} = V^T \vec{x} \Leftrightarrow V \vec{y} = \vec{x}$$

full
& skinny

→

$$\vec{x} = V \Sigma^T U^T \vec{b}$$

square
& inv.

→

$$\vec{x} = V \Sigma^{-1} U^T \vec{b}$$

$$\underline{V} \Sigma^+ \underline{U}^T = A^+$$



pseudoinv. of A

$$\underline{x} = A^+ \underline{b}$$

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



solves $\min \|Ax - b\|_2^2$

$$A^T A x = A^T b$$

↪ normal equations

$$\text{cond}(A^T A) \approx \text{cond}(A^T) \text{cond}(A)$$

$$\approx \text{cond}(A)^2$$

$$(10^6)^2$$

In-class activity: SVD and Least Squares

The Pseudoinverse: A Shortcut for Least Squares

How could the solution process for $A\mathbf{x} \cong \mathbf{b}$ be with an SVDA = $U\Sigma V^T$ be 'packaged up'?

The Normal Equations

You may have learned the 'normal equations' $A^T A \mathbf{x} = A^T \mathbf{b}$ to solve $A \mathbf{x} \cong \mathbf{b}$.

Why not use those?

$h-1$

$$p(x) = \alpha_0 + \alpha_1 x + \dots + \alpha_n x^n$$

$$V \begin{pmatrix} \alpha_0 \\ \vdots \\ \alpha_n \end{pmatrix} = \begin{pmatrix} y_0 \\ \vdots \\ y_n \end{pmatrix}$$

$$p(x_0) = y_0$$

$$p(x_n) = y_n$$

Outline

Python, Numpy, and Matplotlib
Making Models with Polynomials
Making Models with Monte Carlo
Error, Accuracy and Convergence
Floating Point
Modeling the World with Arrays
 The World in a Vector
 What can Matrices Do?
 Graphs
 Sparsity
Norms and Errors
The 'Undo' Button for Linear Operations: LU
Repeating Linear Operations:
Eigenvalues and Steady States
Eigenvalues: Applications

Approximate Undo: SVD and Least Squares

SVD: Applications

Solving Funny-Shaped Linear Systems
Data Fitting
Norms and Condition Numbers
Low-Rank Approximation

Interpolation

Iteration and Convergence

Solving One Equation

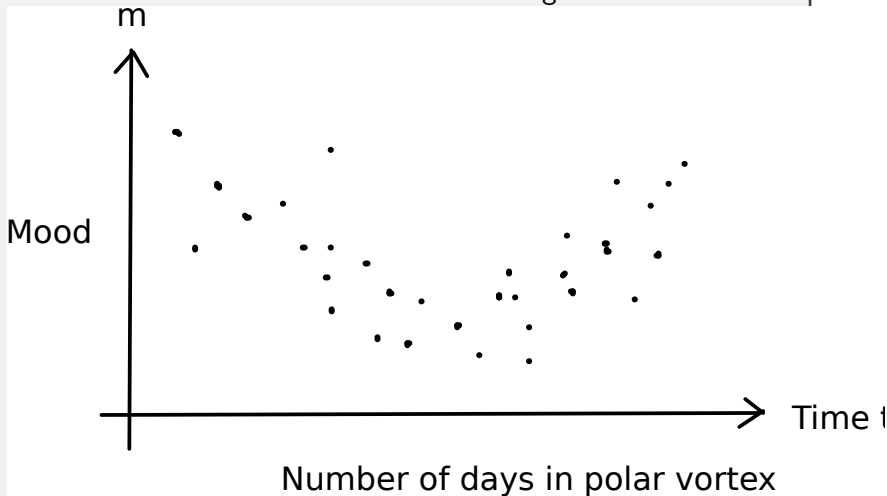
Solving Many Equations

Finding the Best: Optimization in 1D

Optimization in n Dimensions

Fitting a Model to Data

How can I fit a model to measurements? E.g.:



Demo: Data Fitting using Least Squares

Outline

Python, Numpy, and Matplotlib
Making Models with Polynomials
Making Models with Monte Carlo
Error, Accuracy and Convergence
Floating Point
Modeling the World with Arrays
 The World in a Vector
 What can Matrices Do?
 Graphs
 Sparsity
Norms and Errors
The 'Undo' Button for Linear Operations: LU
Repeating Linear Operations:
Eigenvalues and Steady States
Eigenvalues: Applications

Approximate Undo: SVD and Least Squares

SVD: Applications

Solving Funny-Shaped Linear Systems
Data Fitting
Norms and Condition Numbers
Low-Rank Approximation

Interpolation

Iteration and Convergence

Solving One Equation

Solving Many Equations

Finding the Best: Optimization in 1D

Optimization in n Dimensions