Overview

- Eigenvalues: app
- SVD
- SVD: least squares
Understanding Time Behavior

- Many important systems in nature are modeled by describing the time rate of change of something.
  
  - E.g. every bird will have 0.2 baby birds on average per year.
  
  - But there are also foxes that eat birds. Every fox present decreases the bird population by 1 birds a year. Meanwhile, each fox has 0.3 fox babies a year. And for each bird present, the population of foxes grows by 0.9 foxes for every bird present.

Set this up as equations and see if eigenvalues can help us understand what's going on.

\[ \frac{db}{dt} = 0.2 \cdot b - f \]
\[ \frac{df}{dt} = 0.9 \cdot b + 0.3 \cdot f \]
\[
\frac{d\hat{y}}{dt} = A\hat{y} \\
\text{plug in } \hat{y}(t) = e^{\lambda t}\hat{y}_0 \\
A e^{\lambda t}\hat{y}_0 = \lambda A\hat{y}_0 \\
\Rightarrow A\hat{y} = \lambda \hat{y}
\]
**Demo:** Understanding the birds and the foxes with eigenvalues

**In-class activity:** Eigenvalues 2
12 Approximate Undo: SVD and Least Squares
Singular Value Decomposition

What is the Singular Value Decomposition (‘SVD’)?

\[ \mathbf{A} = \mathbf{U} \Sigma \mathbf{V}^T \]

\( \mathbf{A} : m \times n \)

\( \mathbf{U} : m \times m \) orthogonal \( \leftarrow \) columns \( \leftarrow \) left singular vec.

\( \Sigma \) : \( m \times n \)

\( \sigma_1 \geq \sigma_2 \geq \sigma_3 \cdots \geq \sigma_n \geq 0 \)

\( \mathbf{V} : n \times n \) orthogonal \( \leftarrow \) columns \( \leftarrow \) right singular vec.
Computing the SVD

- How can I compute an SVD of a matrix $A$?

Find eigenvalues/vectors of $A^*A V = V D$

$A^*A$ symmetric, pos. def; $D = \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix}$ real-valued, non-orig. col. of $V$ (eigenvectors) are orth.

$\sigma_i = \sqrt{\lambda_i}$

$Z = \begin{pmatrix} z_1 & \cdots & z_n \end{pmatrix}$

$A = U C V^*$ $\Rightarrow$ $U = A V Z^{-1}$

$U^* U = (A V Z^{-1})^* A V Z^{-1} = Z^{-1} V^* A^* A V Z^{-1}$

$= Z^{-1} V^* V D Z^{-1} = Z^{-1} C Z^{-1} = I$
Demo: Computing the SVD
How Expensive is it to Compute the SVD?

Demo: Relative Cost of Matrix Factorizations
‘Reduced’ SVD

- Is there a ‘reduced’ factorization for non-square matrices?
13 SVD: Applications
13.1 Solving Funny-Shaped Linear Systems
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)?

$$Ax = b$$

Let $\hat{x} = V^T x$

$$V \hat{x} = \hat{b}$$

$$\hat{y} = V^T \hat{x}$$

Works, but more expensive than full/lu subr.
Tall and Skinny Systems

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

\[
\begin{bmatrix}
A \\
\end{bmatrix}
\begin{bmatrix}
x \\
\end{bmatrix} =
\begin{bmatrix}
b \\
\end{bmatrix}
\]

In the figure: \( m > n \). How could we solve that?

\[
\begin{align*}
\min \quad & \| A\hat{x} - b \|_2 \\
\text{equiv.} \quad & \min \quad \| A\hat{x} - b \|_2^2 = \| r \|_2^2 = r_1^2 + \ldots + r_m^2
\end{align*}
\]

This is called a least-squares problem.

Find \( \hat{x} \) so that \( \| A\hat{x} - b \|_2^2 \) is as small as possible.
Solving Least-Squares

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

\[
\begin{align*}
\text{minimize} & \quad \|Ax - b\|_2^2 \\
= & \quad \|U \Sigma V^T x - b\|_2^2 \\
= & \quad \|U^	op (U \Sigma V^T x - b)\|_2^2 \\
= & \quad \|U \Sigma V^T x - U \Sigma V^T b\|_2^2 \\
\end{align*}
\]

Given $A = U \Sigma V^T$

For any $y$:

\[
\|U y\|_2 = \|y\|_2
\]

\[
\|U y\|_2 \leq \|y\|_2
\]

\[
y_i = \begin{cases} (U^\top b)_i / \sigma_i & \text{if } 1 \leq i \leq k \\ 0 & \text{if } k + 1 \leq i \leq n \end{cases}
\]

\[
\tilde{y} \text{ figure out}
\]

\[
\Rightarrow \text{find } x = V \tilde{y}.
\]
In-class activity: SVD and Least Squares

Solve \( A\hat{x} = \hat{b} \)

1. Get \( A = U\Sigma V^T \)
2. Solve \( \Sigma \hat{y} = U^T \hat{b} \)
3. \( \hat{x} = V\hat{y} \)

\[ \Sigma^+ = \begin{pmatrix} \gamma & 0 & \cdots & 0 \\ 0 & \gamma & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \gamma \end{pmatrix} \]  \( \hat{x} = \binom{U^T \hat{y}}{\text{pseudo inverse}} \)
The Pseudoinverse: A Shortcut for Least Squares

- How could the solution process for $Ax \simeq b$ be with an $SVDA = U\Sigma V^T$ be ‘packaged up’?
The Normal Equations

- You may have learned the ‘normal equations’ $A^T A x = A^T b$ to solve $A x \approx b$. Why not use those?