2) Data compression.

An image is made of pixels (say 2^9 by 2^{10} pixels). Each pixel has a grayscale value. You can keep track of this with a $2^9 \times 2^{10}$ matrix. This has 2^{19} entries. No big deal!

Until you start making movies with 30 images per second, or doing a CT scan which has tons of cross section images. Now there are lots of $2^9 \times 2^{10}$ matrices to keep track of.

Options: 1) Bigger pixels. Double pixel height + width and reduce the number of pixels by a factor of 4.

Downside: $\frac{1}{4}$ of 2^{19} (times many images per second) is still huge, and picture quality decrease

2) Use SVD! If A is your pixel matrix, approximate A using the largest singular values

$A \approx \sigma_1 \vec{u}_1 \vec{v}_1^T$ ($\vec{u}_1$ has $\underset{2^9}{m}$ entries, $\vec{v}_1$ has $\underset{2^{10}}{n}$ entries)

So this means we only need to keep track of $2^9 + 2^{10}$ numbers!

In reality, use the first few singular values (until the approx. is pretty good).

Say: $A \approx \sigma_1 \vec{u}_1 \vec{v}_1^T + \sigma_2 \vec{u}_2 \vec{v}_2^T + \sigma_3 \vec{u}_3 \vec{v}_3^T$

Keep track of $3(2^9 + 2^{10})$ numbers.
Way less than 2^{19}!

Class 23: 11/29
Collect HW.

Today: I. Two more uses for SVD
- Polar Decomposition
- Pseudo inverse

II. Complex number review

III. Complex matrices.

Recall: $A = U \Sigma V^T$ $\quad U, V$ orthogonal, Σ "diagonal"

We had $AV = U\Sigma$

$A\vec{v}_1 = \sigma_1 \vec{u}_1$ $\qquad A\vec{v}_{r+1} = 0$

$A\vec{v}_2 = \sigma_2 \vec{u}_2$ $\qquad \vdots$

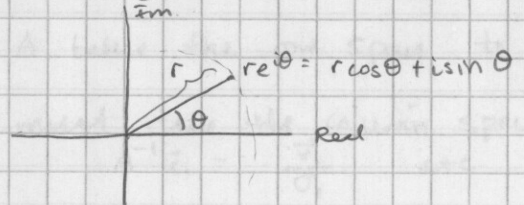
$\vdots$ $\qquad A\vec{v}_n = 0$

$A\vec{v}_r = \sigma_r \vec{u}_r$

Polar decomposition

This is analogous to writing a complex number as $re^{i\theta}$

$r \geq 0$; $e^{i\theta}$ ← tells you how much to rotate



Polar decomposition for matrices:

A a real square matrix. Write $A = QH$ where
Q is orthogonal
H is positive semidefinite and symmetric.

(If A is invertible, H is positive definite).

Orthogonal matrices are rotations (and reflections), so Q plays the role of $e^{i\theta}$

Symmetric positive semidefinite matrices stretch along the axes (eigenvector directions), generalizing r.

How do we compute it?

$$A = U\Sigma V^T = \underbrace{(UV^T)}_{= Q}\underbrace{(V\Sigma V^T)}_{H}$$

(U, V orthogonal $\Rightarrow UV^T$ is orthogonal.
H positive semidefinite because $H = V\Sigma V^T$ is $H = Q\Lambda Q^T$, so eigenvalues are the singular values)

Example: $A = \begin{bmatrix} 1 & 2 \\ 3 & 6 \end{bmatrix}$ (From HW due today).

$$A = \frac{1}{\sqrt{10}}\begin{bmatrix} 1 & -3 \\ 3 & 1 \end{bmatrix}\begin{bmatrix} \sqrt{50} & 0 \\ 0 & 0 \end{bmatrix}\frac{1}{\sqrt{5}}\begin{bmatrix} 1 & 2 \\ -2 & 1 \end{bmatrix}$$

So $Q = \frac{1}{5\sqrt{2}}\begin{bmatrix} 1 & -3 \\ 3 & 1 \end{bmatrix}\begin{bmatrix} 1 & 2 \\ -2 & 1 \end{bmatrix} = \frac{1}{5\sqrt{2}}\begin{bmatrix} 7 & -1 \\ 1 & 7 \end{bmatrix}$.

$$H = \frac{1}{5}\begin{bmatrix} 1 & -2 \\ 2 & 1 \end{bmatrix}\begin{bmatrix} \sqrt{50} & 0 \\ 0 & 0 \end{bmatrix}\begin{bmatrix} 1 & 2 \\ -2 & 1 \end{bmatrix} = \frac{1}{5}\begin{bmatrix} \sqrt{50} & 0 \\ 2\sqrt{50} & 0 \end{bmatrix}\begin{bmatrix} 1 & 2 \\ -2 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} \sqrt{2} & 2\sqrt{2} \\ 2\sqrt{2} & 4\sqrt{2} \end{bmatrix}$$

<u>Pseudo inverse</u>: Recall $A\vec{v}_1 = \sigma_1 \vec{u}_1$

$$A\vec{v}_r = \sigma_r \vec{u}_r$$

So A takes the row space to the column space.

A^{-1} must take the column space to the row space.

$$A^{-1}\vec{u}_1 = \frac{\vec{v}_1}{\sigma_1} \text{ etc.}$$

If A is not invertible, we can't find A^{-1} (not invertible could be not square, or square and singular).

Instead find A^+, the pseudoinverse.

A^+ takes column space to the row space.

If $A = U\Sigma V^T$, $A^+ = V\Sigma^+ U^T$,

where $\Sigma^+ = \left[\begin{array}{ccc|c} \sigma_1^{-1} & & & 0 \\ & \sigma_2^{-1} & & 0 \\ & & \ddots \; \sigma_r^{-1} & 0 \\ \hline 0 & & 0 \; 0 & 0 \end{array}\right]$ with r columns and $n-r$ columns; r rows and $m-r$ rows.

AA^+ is not the identity, but it is projection onto $C(A)$

$$AA^+ = U\Sigma\Sigma^+ U^T = U \left[\begin{array}{c|c} \underset{r}{I} & 0 \\ \hline 0 & 0 \end{array}\right] U^T = \left[\begin{array}{c|c} U_r U_r^T & 0 \\ \hline 0 & 0 \end{array}\right]$$

where $U_r = \begin{bmatrix} | & & | \\ u_1 & \cdots & u_r \\ | & & | \end{bmatrix}$

columns are a basis for $C(A)$.

So $AA^+A = A$

Similarly, A^+A is projection onto the row space of A and $A^+AA^+ = A^+$.

Why the pseudoinverse is good: Least squares.

Before, we solved $A^TA\hat{x} = A^Tb$ and A had indep. columns.

Now if A has dependent columns, there are many solutions $\hat{x}$.

But $\vec{x}^+ = A^+\vec{b}$ is the <u>shortest</u> solution.

$$AA^+\vec{b} = U\Sigma V^T V\Sigma^+ U^T b = U\Sigma\Sigma^+ U^T \vec{b} = \text{Proj. of } b \text{ onto } C(A)$$

Complex Numbers

$\sqrt{-1} = i$ A complex number is $a+bi$ where $a, b \in \mathbb{R}$

We can add, subtract, multiply and divide

$$(1+2i) + (3-4i) = 4-2i \quad \text{componentwise}$$
$$(1+2i)\cdot(3-4i) = 3-4i+6i-8i^2 = 11+2i \quad \text{FOIL}$$

$$\frac{1+2i}{3-4i} = \frac{(1+2i)}{(3-4i)}\frac{(3+4i)}{(3+4i)} = \frac{-5+10i}{9+16} = \frac{-1+2i}{5}$$

conjugate

Definition: If $z \in \mathbb{C}$, $\bar{z}$ is the complex conjugate. If $z = a+bi$, $\bar{z} = a-bi$

Definition: The norm of a complex number $z = a+bi$ is
$$|z| = \sqrt{z\bar{z}} = \sqrt{a^2+b^2}$$

Class 24: 12/4

Final exam 12/21 at 9am in Wilson 302.

Today: - Complex matrices
- Fourier matrix

Recall that the norm of a complex number is $|z| = \sqrt{a^2+b^2} = \sqrt{z\bar{z}}$

For length of a complex vector, we need to use the same complex conjugate trick.

$$\|\vec{z}\|^2 = \bar{\vec{z}}\vec{z} = \vec{z}^H\vec{z} = |z_1|^2 + |z_2|^2 + \dots + |z_n|^2$$

For a complex matrix, replace transpose with Hermitian (aka adjoint).

$$A^* = A^H = \bar{A}^T$$

Example If $A = \begin{bmatrix} 1+i & 2+3i \\ 5 & 2i \end{bmatrix}$, $A^H = \begin{bmatrix} 1-i & 5 \\ 2-3i & -2i \end{bmatrix}$

We use the Hermitian just like the transpose.

$\vec{v}^H\vec{u} = 0 \iff \vec{v}$ and $\vec{u}$ are perpendicular/orthogonal.

(Note $\vec{v}^H\vec{u} \neq \vec{u}^H\vec{v}$. Actually, $\vec{v}^H\vec{u} = \overline{\vec{u}^H\vec{v}}$).