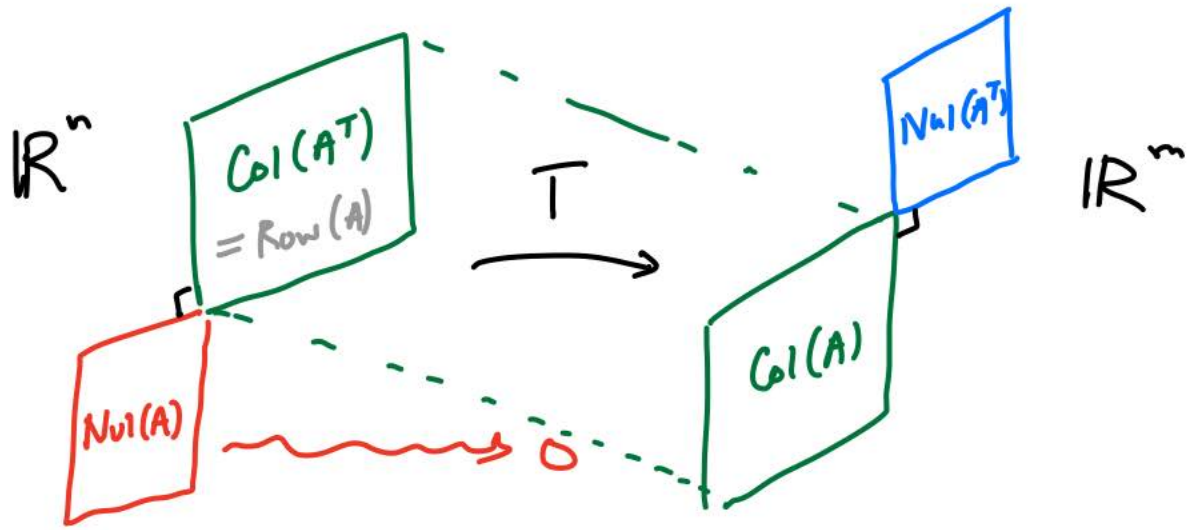
Lecture 38: The Singular Value Decomposition

Let A be an $m \times n$ matrix, $T: \mathbb{R}^n \to \mathbb{R}^m$ the linear transformation $\vec{x} \mapsto A\vec{x}$. Recall the picture



in which

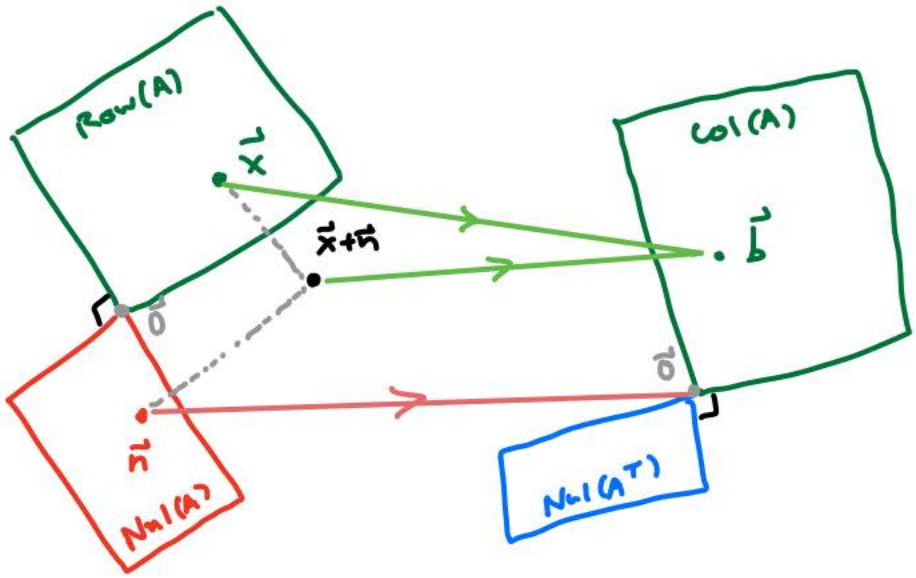
- T restricts to an isomorphism $\mathrm{Row}(A) \xrightarrow{\cong} \mathrm{Col}(A)$, and so they have the same dimension r
- $\mathrm{Col}(A) \perp \mathrm{Nul}(A^T)$ and $\underbrace{\mathrm{Col}(A^T) \perp \mathrm{Nul}(A)}$

  Null space of A is vectors $\perp$ to its rows, i.e. columns of A^T.

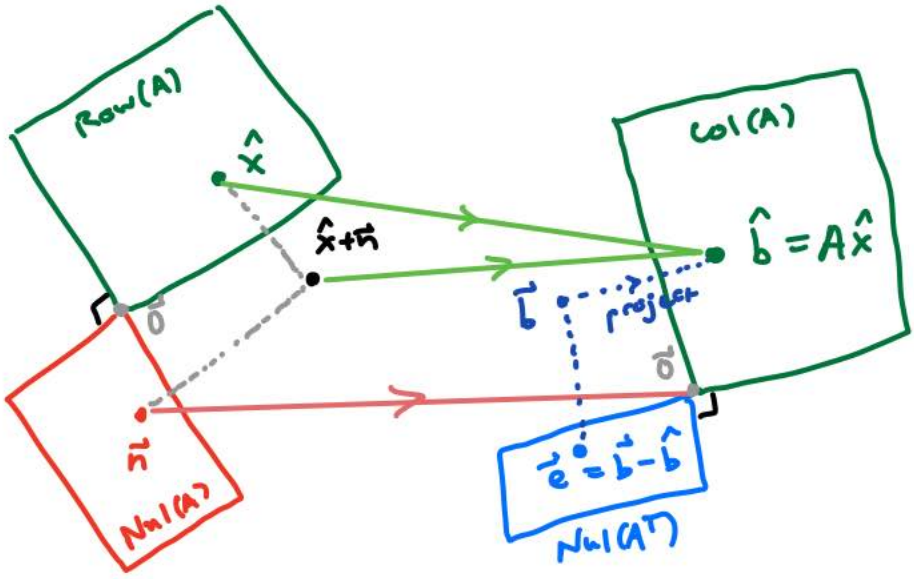
- $\left.\begin{array}{ll} \mathrm{Col}(A) \oplus \mathrm{Nul}(A^T) & \text{span } \mathbb{R}^m \\ \mathrm{Col}(A^T) \oplus \mathrm{Nul}(A) & \text{span } \mathbb{R}^n \end{array}\right\}$ recovers Rank + Nullity; $\dim \mathrm{Nul}(A^T) = m-r$, $\dim \mathrm{Nul}(A) = n-r$.

If $\vec{b} \in \mathrm{Col}(A)$, then there is a unique solution to $A\vec{x} = \vec{b}$ in $\mathrm{Col}(A^T)$, and many more if we add an arbitrary vector $\vec{n}$

in $\mathrm{Nul}(A)$
to $\vec{x}$:



If $\vec{b} \notin \mathrm{Col}(A)$, then $A\vec{x} = \vec{b}$ is inconsistent and the best we can do is minimize $\underbrace{A\vec{x} - \vec{b}}_{\text{error } \vec{e}}$, which is done by a least-squares solution $\hat{x}$:



To minimize $\vec{e}$, we take $\hat{b} = \mathrm{proj}_{\mathrm{Col}(A)}\vec{b}$ and solve

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

for $\hat{x}$; then $\vec{e} = \vec{b} - A\hat{x} = \vec{b} - \hat{b} \in \mathrm{Col}(A)^{\perp} = \mathrm{Nul}(A^T)$.

And so $A^T(\vec{b} - A\hat{x}) = 0$ recovers the normal equations

$$A^T A \hat{x} = A^T \vec{b}.$$

Now let's talk about bases. $\overset{n \times n}{A^T A}$ is symmetric, and also positive-semidefinite, since (for any $\vec{x} \in \mathbb{R}^n$) $\vec{x}^T A^T A \vec{x} = A\vec{x} \cdot A\vec{x} = \|A\vec{x}\|^2 \geq 0$. So $A^T A$ has an orthonormal eigenbasis $\mathcal{B} = \{\vec{v}_i\}_{i=1}^n$ with all $\lambda_i \geq 0$. The 0-eigenspace E_0 is $\mathrm{Nul}(A^T A) = \mathrm{Nul}(A)$ (since $A^T A \vec{x} = \vec{0} \Rightarrow \vec{x}^T A^T A \vec{x} = 0 \Rightarrow \|A\vec{x}\|^2 = 0 \Rightarrow A\vec{x} = \vec{0}$), and the other eigenspaces are orthogonal to E_0 hence contained in $\mathrm{Row}(A)$. This means (after reordering) we have

$$\left.\begin{array}{l} \{\vec{v}_1, \dots, \vec{v}_r\} \subset \mathrm{Row}(A) \\ \{\vec{v}_{r+1}, \dots, \vec{v}_n\} \subset \mathrm{Nul}(A) \end{array}\right. \quad \text{both (o.n.) bases,}$$

while $\lambda_1, \dots, \lambda_r > 0$ and $\lambda_{r+1} = \dots = \lambda_n = 0$.

<u>Definition</u>: The <u>singular values</u> of A are $\sigma_1, \dots, \sigma_r$, where $\sigma_i := \sqrt{\lambda_i}$. (We'll also write $\sigma_{r+1} = \dots = \sigma_n = 0$.)

So $A^T A \vec{v}_i = \sigma_i^2 \vec{v}_i$ $(1 \leq i \leq r) \Rightarrow \vec{v}_i^T A^T A \vec{v}_i = \sigma_i^2 v_i^T v_i \Rightarrow$ $\|Av_i\|^2 = \sigma_i^2 \|v_i\|^2 = \sigma_i^2$ (since $\vec{v}_i$ is normalized) $\Rightarrow$

$\vec{u}_i := \frac{1}{\sigma_i} A\vec{v}_i$ has unit length $\Rightarrow$

$$\{\vec{u}_1, \dots, \vec{u}_r\} \subset \mathrm{Col}(A) \quad \text{is an o.n. basis.}$$

Letting $\{\vec{u}_{r+1}, \dots, \vec{u}_m\} \subset \mathrm{Nul}(A^T)$ be an o.n. basis,

$\mathcal{C} := \{\vec{u}_1, \dots, \vec{u}_m\}$ is an o.n. basis of $\mathbb{R}^m$.

Now define matrices as follows:

- $U = \begin{pmatrix} \uparrow & & \uparrow \\ \vec{u}_1 & \cdots & \vec{u}_m \\ \downarrow & & \downarrow \end{pmatrix}$ $m \times m$ orthogonal matrix (why?)
- $V = \begin{pmatrix} \uparrow & & \uparrow \\ \vec{v}_1 & \cdots & \vec{v}_n \\ \downarrow & & \downarrow \end{pmatrix}$ $n \times n$ orthogonal matrix
- $\Sigma = \left(\begin{array}{c|c} \begin{matrix} \sigma_1 & & 0 \\ & \ddots & \\ 0 & & \sigma_r \end{matrix} & 0 \\ \hline 0 & 0 \end{array}\right)$ $m \times n$ matrix

We have then

$$U\Sigma = \begin{pmatrix} \uparrow & & \uparrow & \uparrow & & \uparrow \\ \sigma_1\vec{u}_1 & \cdots & \sigma_r\vec{u}_r & \vec{0} & \cdots & \vec{0} \\ \downarrow & & \downarrow & \downarrow & & \downarrow \end{pmatrix} = \begin{pmatrix} \uparrow & & \uparrow & \uparrow & & \uparrow \\ A\vec{v}_1 & \cdots & A\vec{v}_r & A\vec{v}_{r+1} & \cdots & A\vec{v}_n \\ \downarrow & & \downarrow & \downarrow & & \downarrow \end{pmatrix} = AV$$

and multiplying by V^T on the right gives $U\Sigma V^T = AVV^T$. But $VV^T = I_n$ since V is orthogonal, yielding the

$$\boxed{\underline{\text{SINGULAR VALUE DECOMPOSITION}}\ (\text{SVD}) \quad A = U\Sigma V^T}$$

When A is <u>itself</u> square and symmetric, we can take $U = V := P$ and $\Sigma := D$ — i.e. $A = PDP^T$ <u>is</u> the decomposition*. But the SVD <u>extends this to matrices which might not be symmetric, or diagonalizable, or even square</u> (since we didn't require $m = n$).

* And the σ_i are its eigenvalues

The SVD has been called the "fundamental theorem of matrix algebra". Its efficient computer implementation for large matrices has been the subject of countless articles in numerical analysis. While these algorithms are too complicated to describe here, they are more efficient (and accurate) than the orthogonal diagonalization of A^TA, which are implemented in things like MATLAB.

<u>Application #1</u>: Principal Component Analysis

The SVD reads

$$A = \begin{pmatrix} \uparrow & & \uparrow \\ \vec{u}_1 & \cdots & \vec{u}_m \\ \downarrow & & \downarrow \end{pmatrix} \left(\begin{array}{ccc|c} \sigma_1 & & 0 & \\ & \ddots & & 0 \\ 0 & & \sigma_r & \\ \hline & 0 & & 0 \end{array}\right) \begin{pmatrix} \leftarrow & \vec{v}_1^T & \rightarrow \\ & \vdots & \\ \leftarrow & \vec{v}_n^T & \rightarrow \end{pmatrix}$$

$$= \begin{pmatrix} \uparrow & & \uparrow & \uparrow & & \uparrow \\ \sigma_1\vec{u}_1 & \cdots & \sigma_r\vec{u}_r & \vec{0} & \cdots & \vec{0} \\ \downarrow & & \downarrow & \downarrow & & \downarrow \end{pmatrix} \begin{pmatrix} \leftarrow & \vec{v}_1^T & \rightarrow \\ & \vdots & \\ \leftarrow & \vec{v}_n^T & \rightarrow \end{pmatrix}$$

$$= \sigma_1 \underbrace{\vec{u}_1 \vec{v}_1^T}_{\substack{(m\times 1)(1\times n) \\ = m\times n}} + \sigma_2 \vec{u}_2 \vec{v}_2^T + \cdots + \sigma_r \vec{u}_r \vec{v}_r^T .$$

Let's say A is a big fat matrix of data. For instance, in a country of $n = 1{,}000{,}000$ internet users, you want to target ads (or fake news) based on online behavior, where the # m of possible clicks or purchases is also

very large. Assume however that only k (cultural, demographic, etc.) attributes of a person generally determine this behavior: that is, there exists an $n \times k$ "attribute matrix" and a $k \times m$ "behavior matrix" whose product is roughly A, the $n \times m$ matrix of raw user data. (In other words, we expect A to be well-approximated by a rank k matrix, with k much smaller than n or m.) Alternatively, A might be a matrix of numbers controlling brightness of pixels in an image. Either way, it will typically be the case that <u>most of the σ_i are very small compared to the first few.</u> So we get

$$A \approx \sigma_1 \vec{u}_1 \vec{v}_1^T + \cdots + \sigma_k \vec{u}_k \vec{v}_k^T,$$

the "principal components" of the SVD

where k is maybe 5. Then instead of $1{,}000{,}000^2$ numbers, you only need $2 \times 5 \times 1{,}000{,}000 + 5$ numbers. And this will typically be a very good approximation, <u>much</u> better than Fourier analysis would give.

<u>Application #2</u>: Least squares and the "pseudoinverse"

Define a transformation

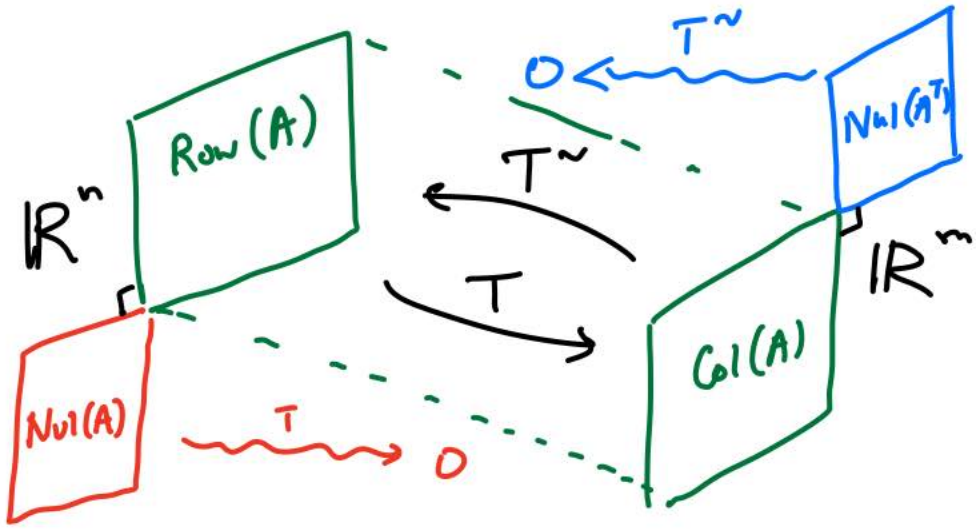
$$T^{\sim} : \mathbb{R}^m \to \mathbb{R}^n$$

by sending

$$\vec{u}_i \longmapsto \frac{1}{\sigma_i} \vec{v}_i \qquad (i = 1, \ldots, r)$$

$$\vec{u}_i \longmapsto \vec{0} \qquad (i = r+1, \ldots, m).$$

Its restriction to $\mathrm{Col}(A)$ inverts the restriction of T to $\mathrm{Row}(A)$, since T sends $\vec{v}_i \longmapsto \sigma_i \vec{u}_i$ $(i=1,\dots,r)$:



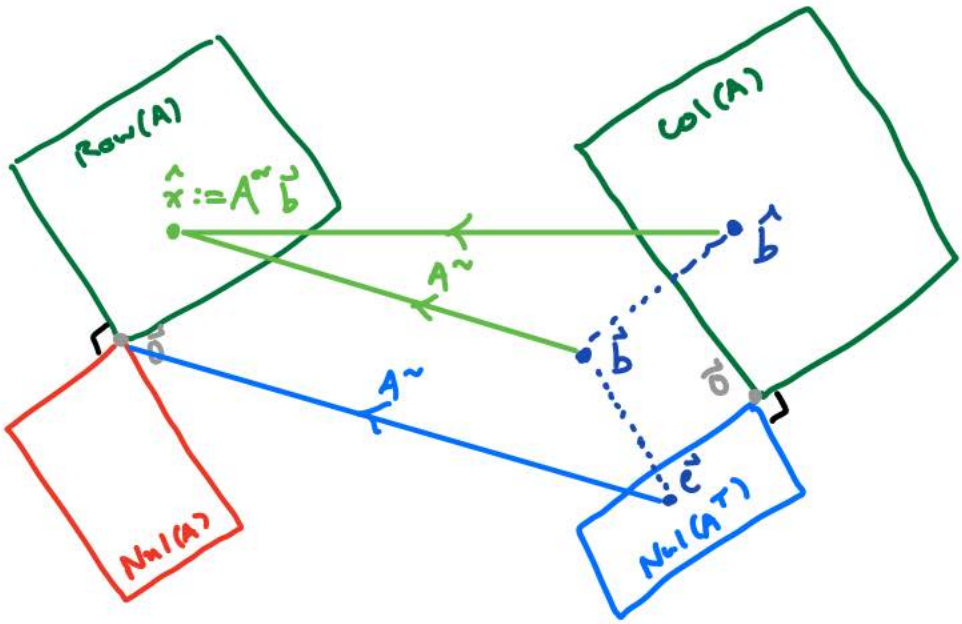
The matrix of this pseudoinverse is

$$A^\sim := V \tilde{\Sigma} U^T$$

where $\tilde{\Sigma} := \left(\begin{array}{ccc|c} \sigma_1^{-1} & & 0 & \\ & \ddots & & 0 \\ 0 & & \sigma_r^{-1} & \\ \hline & 0 & & 0 \end{array}\right)$ is $n \times m$. (Why?)

<u>Claim</u>: $\hat{x} := A^\sim \vec{b}$ gives the unique least-squares solution to $A\vec{x} = \vec{b}$ in $\mathrm{Row}(A)$ (called the "minimal least-squares solution").

Why does it work? In this case, a picture is worth a thousand math symbols:



That is, $(\hat{x} =) A^{\sim}\vec{b} = A^{\sim}\hat{b}$ and $A^{\sim}$ inverts A on Col(A).

So $A\hat{x} = \hat{b}$.

Let's finish with a numerical example:

$$A = \begin{pmatrix} 1 & -1 \\ 1 & -1 \\ 2 & -2 \end{pmatrix}$$

The SVD is

only "singular value" of A

$$A = \begin{pmatrix} 1/\sqrt{6} & 1/\sqrt{2} & 1/\sqrt{3} \\ 1/\sqrt{6} & -1/\sqrt{2} & 1/\sqrt{3} \\ 2/\sqrt{6} & 0 & -1/\sqrt{3} \end{pmatrix} \begin{pmatrix} \sqrt{12} & 0 \\ 0 & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} 1/\sqrt{2} & -1/\sqrt{2} \\ 1/\sqrt{2} & 1/\sqrt{2} \end{pmatrix}$$

and so the pseudo inverse is

$$A^{\sim} = \begin{pmatrix} 1/\sqrt{2} & 1/\sqrt{2} \\ -1/\sqrt{2} & 1/\sqrt{2} \end{pmatrix} \begin{pmatrix} 1/\sqrt{12} & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 1/\sqrt{6} & 1/\sqrt{6} & 2/\sqrt{6} \\ 1/\sqrt{2} & -1/\sqrt{2} & 0 \\ 1/\sqrt{3} & 1/\sqrt{3} & -1/\sqrt{3} \end{pmatrix}$$

$$= \frac{1}{12} \begin{pmatrix} 1 & 1 & 2 \\ -1 & -1 & -2 \end{pmatrix}$$

So the minimal least-squares solution to $A\vec{x} = \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix}$ is

$$\hat{x} = A^{\sim} \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 1/4 \\ -1/4 \end{pmatrix}.$$

Doesn't seem like the easiest way to do this example (The normal equations are), but it is numerically far superior for large matrices.

Friday we'll review for the exam.