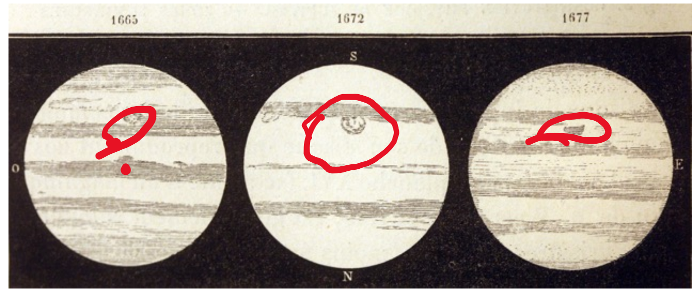
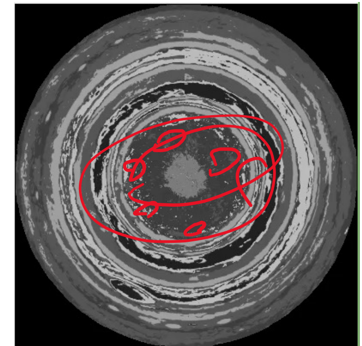
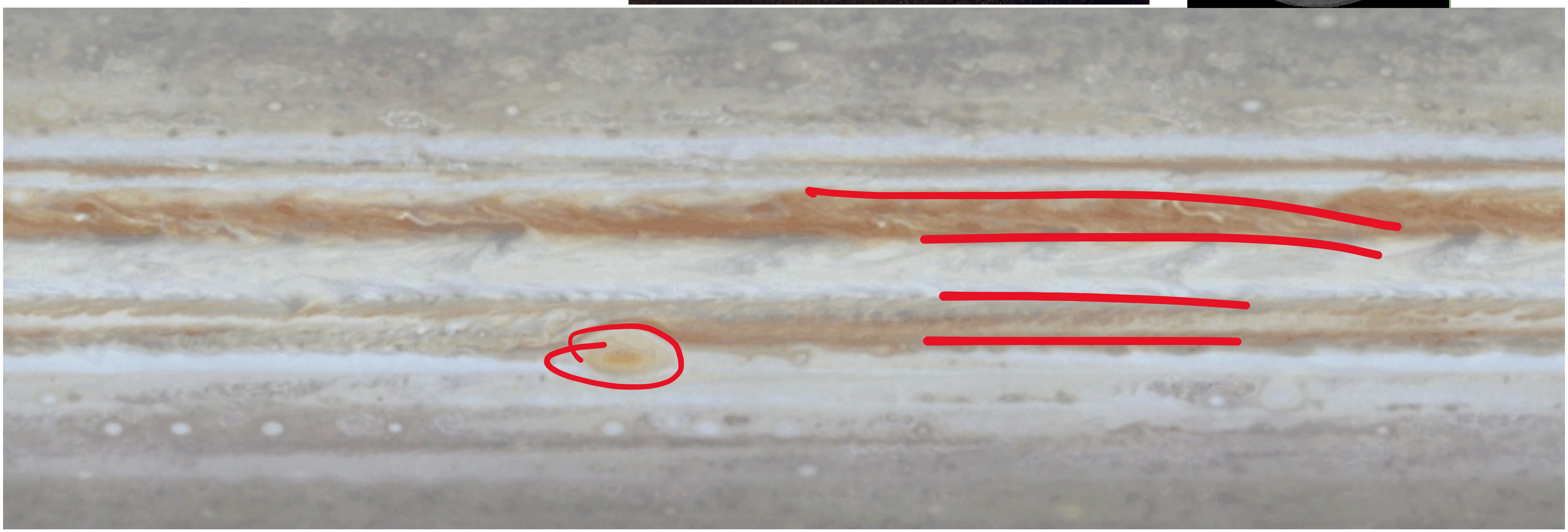
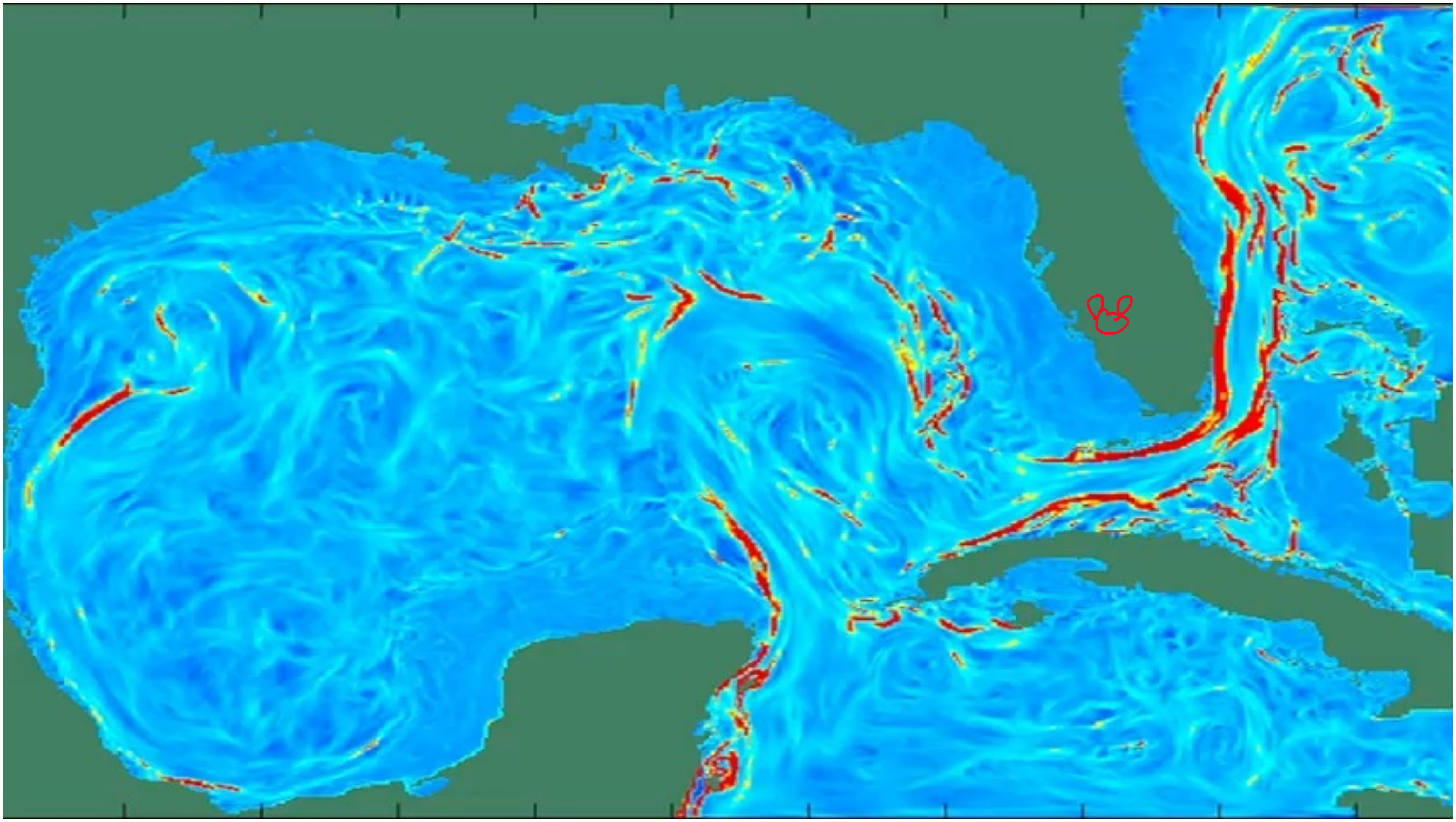
EE520 Data Driven Analysis of Complex Systems

Erik Bollt







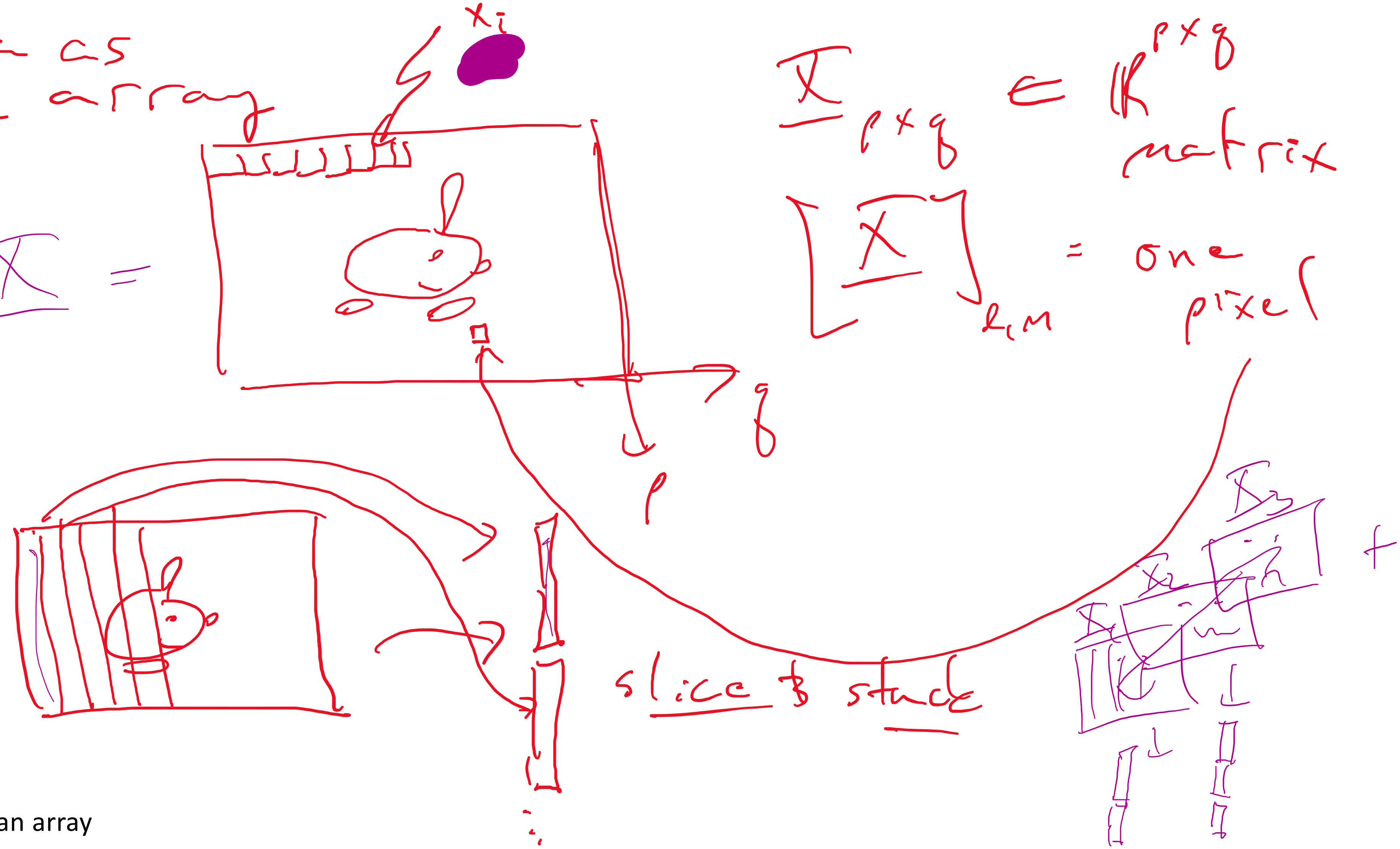


Data as an array

$\underline{X} =$

$\underline{X}_{p \times q} \in \mathbb{R}^{p \times q}$ matrix

$[\underline{X}]_{l,m}$ = one pixel



Data as an array

On Matrix Multiplication

$$A = \begin{pmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,n} \\ \vdots & & \ddots & \vdots \\ a_{m,1} & a_{m,2} & \cdots & a_{m,n} \end{pmatrix}, \text{ and each } a_{i,j} \in \mathbb{C}$$

$$\begin{aligned} L(\mathbf{z}) : \mathbb{R}^n &\to \mathbb{R}^m \\ \mathbf{z} &\mapsto \mathbf{z}' = A\mathbf{z}, \end{aligned}$$

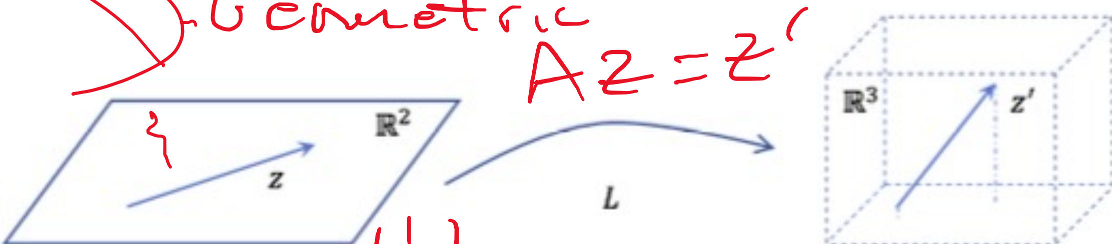
in terms of the usual matrix times vector multiplication,

$$[\mathbf{z}]'_i = \sum_{j=1}^{n} A_{i,j}[\mathbf{z}]_j, \text{ for each } i = 1, \cdots, m,$$

Geometric

• a vector has length & direction.

$Az = z'$



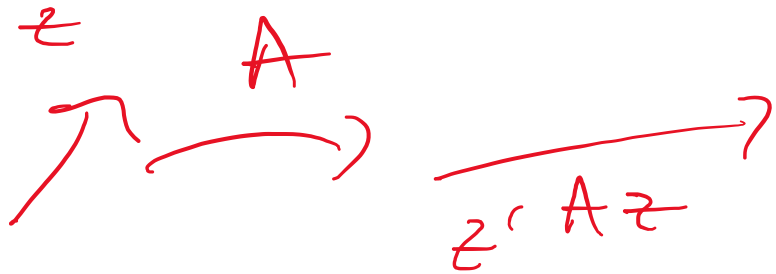
$n = 2$ $\quad z = \begin{pmatrix} 1 \\ 2 \end{pmatrix}$ $\quad m = 3$

But as linear algebra i.e. Matrices × vectors

$$A_{2\times 2} = \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}$$

$$A\begin{pmatrix} 3 \\ 4 \end{pmatrix} = \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}\begin{pmatrix} 3 \\ 4 \end{pmatrix} = \begin{pmatrix} 1\cdot 3 + 2\cdot 4 \\ 3\cdot 3 + 4\cdot 4 \end{pmatrix} = \begin{pmatrix} 11 \\ 25 \end{pmatrix}$$

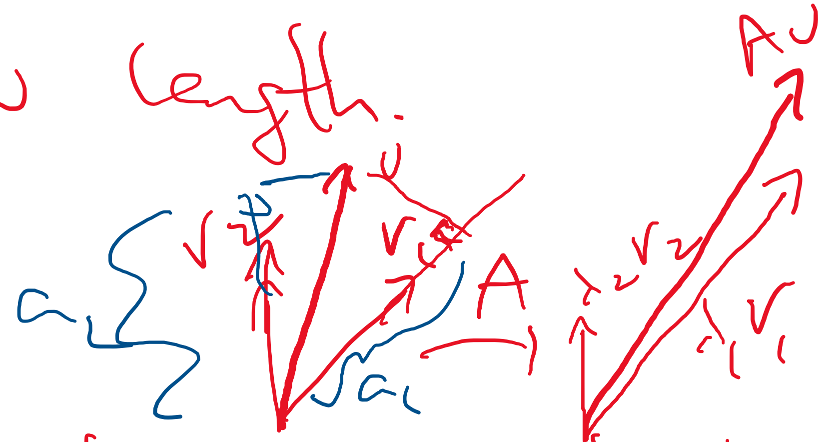
$$z' = Az$$



new direction, new length.

Eig for square.

∘ $\underset{2\times 2}{A} v = \lambda v$



Characterize matrices by knowing just these

∘ eig. special directions

Linear

∘ $Au = A(a_1 v_1 + a_2 v_2) = a_1 A v_1 + a_2 A v_2$

$= a_1 \lambda_1 v_1 + a_2 \lambda_2 v_2$

$\det(A - \lambda I) = 0$

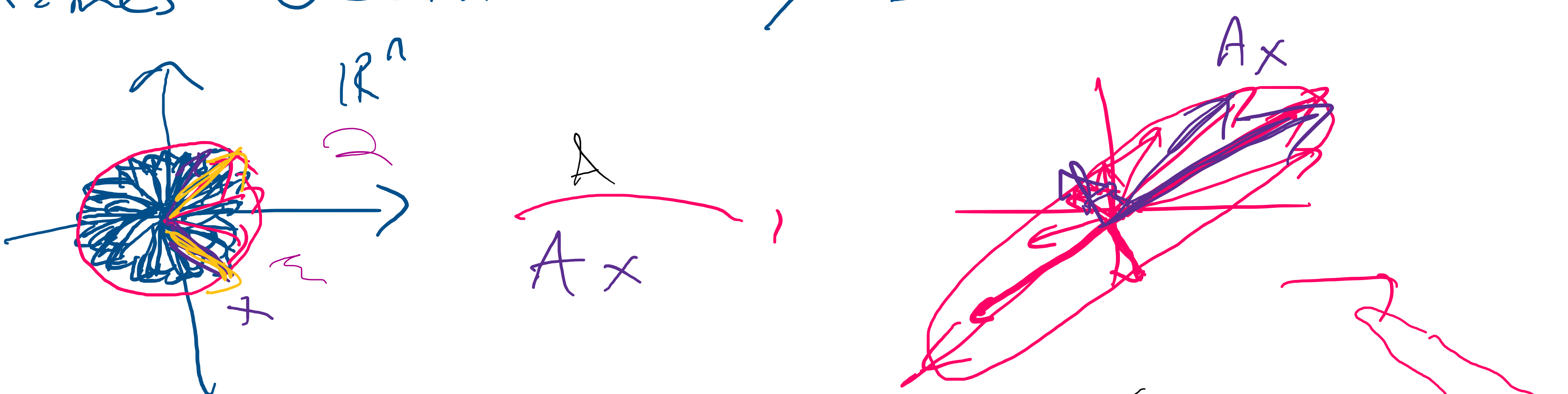
$(A - \lambda I)v = 0$

Matrix time circle =

all vectors of length 1.

? Matrix x circle ?! But matrix times vector.

$\nearrow = \nearrow$



$S = \{x \mid \|x\|_2 = 1,\ x \in E \cong \mathbb{R}^2\}$; $A \cdot S = \{y : y = Ax,\ x \in S\}$

Theorem 2.1.1 — Singular Value Decomposition. Let A be an $m \times n$ matrix whose entries come from the field $\mathcal{K}$, which is either the field of real numbers or the field of complex numbers. Then the singular value decomposition of A exists, and it takes the form of a product of matrices:

$$A_{m\times n} = U_{m\times m}\Sigma_{m\times n}V^*_{n\times n}, \tag{2.5}$$

where

- U is an $m \times m$ unitary matrix.
- Σ is a diagonal $m \times n$ matrix with non-negative real numbers on the diagonal.
- V is an $n \times n$ unitary matrix, and V^* is the conjugate transpose of V.

The singular values are the nonegative values: $\sigma_i \geq 0, i = 1, \cdots, n$,

The left singular vectors: u_i are the columns of $U = [u_1, u_2, ..., u_m]$.

The right singular vectors: v_i are the columns of $V = [v_1, v_2, ..., v_n]$.

$$\Sigma := diag(\sigma_1, \sigma_2, \cdots, \sigma_p), p = min(m, n),$$
$$\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_p \geq 0.$$

$\Sigma = \begin{pmatrix} \sigma_1 & & 0 \\ & \sigma_2 & \\ 0 & & \ddots \end{pmatrix}$

$A = U \Sigma V^T$

$u_1^T u_1 = u_1 \cdot u_1 = 1$

$\begin{pmatrix} u_1^T u_1 = 1 & u_1^T u_2 = 0 & u_1^T u_3 = 0 & \cdots \\ u_2^T u_1 & u_2^T u_2 & \cdots \end{pmatrix}$

$u_i^T u_j = \delta_{ij} = 1$ or 0

Definition 2.1.1 — Singular values and singular vectors. The singular values of A are the scalar values, σ_i, and the columns of U and V have columns that are the corresponding i^{th} left and right singular vectors, u_i and v_i:

The singular values are the nonegative values: $\sigma_i \geq 0, i = 1, \cdots, n$,

The left singular vectors: u_i are the columns of $U = [u_1, u_2, ..., u_m]$.

The right singular vectors: v_i are the columns of $V = [v_1, v_2, ..., v_n]$.

$\|u\|^2 \cos\theta = \vec{u_1} \cdot \vec{u_1} = I \qquad u_2^T u_1 = 0$

$\cdot\ U$ unitary $\Leftrightarrow U^* U = U U^* = I$

$U^T U = (u_1\ u_2 \cdots u_m)^T (u_1\ u_2 \cdots u_m) = I$

Since V is orthogonal, then right multiplying Eq. (2.5) by V,

$$AV = U\Sigma V^*V = U\Sigma, \tag{2.8}$$

$$Ax = b$$

$$\begin{pmatrix} 1 & 2 & 3 \\ 2 & 4 & 6 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix}$$

if

A is square

$$AV = U\Sigma$$

$$[A]\,[v_1 v_2 \cdots v_n] = [u_1 u_2 \cdots u_n]\, diag(\sigma_1, \sigma_2, \cdots, \sigma_n).$$

■ **Example 2.1** Let $A = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 4 & 6 \end{pmatrix}_{2\times 3}$. By SVD of the matrix A we have:

$$\begin{aligned} A &= U\Sigma V^T \\ &= \begin{pmatrix} \frac{1}{\sqrt{5}} & \frac{2}{\sqrt{5}} \\ \frac{-2}{\sqrt{5}} & \frac{1}{\sqrt{5}} \end{pmatrix} \begin{pmatrix} \sqrt{70} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} \frac{1}{\sqrt{14}} & \sqrt{\frac{2}{7}} & \frac{3}{\sqrt{14}} \\ \frac{-3}{\sqrt{10}} & 0 & \frac{1}{\sqrt{10}} \\ \frac{-1}{\sqrt{35}} & \sqrt{\frac{5}{7}} & \frac{-3}{\sqrt{35}} \end{pmatrix}. \end{aligned} \tag{2.28}$$

We see that the second singular value, $\sigma_2 = 2$, meaning that number of non-zero singular values $r < \min\{m, n\}$. Such matrix is called rank deficient matrix. If we take the economy version (with $r = 1$) of the SVD we will have:

$$\begin{aligned} u_1\sigma_1 v_1^T &= \begin{pmatrix} \frac{1}{\sqrt{5}} \\ \frac{-2}{\sqrt{5}} \end{pmatrix} \left(\sqrt{70}\right) \begin{pmatrix} \frac{1}{\sqrt{14}} & \sqrt{\frac{2}{7}} & \frac{3}{\sqrt{14}} \end{pmatrix} \\ &\approx \begin{pmatrix} 1 & 2 & 3 \\ 2 & 4 & 6 \end{pmatrix} \end{aligned} \tag{2.29}$$

■

$$A^TA = V\Sigma^T\Sigma V^T \cdot V$$

$$A^TAV = \Sigma^T\Sigma V$$

$$V = (v_1\ v_2 \cdots v_n)$$

$$[A]\,[v_1 v_2 \cdots v_n] = [u_1 u_2 \cdots u_n]\, diag(\sigma_1, \sigma_2, \cdots, \sigma_n).$$

$$AA^T = U\Sigma \underbrace{V^T (V}\Sigma^T U^T)$$

$$= U\Sigma\Sigma^T U^T$$

$$(AA^T)\,\underline{U} = U(\Sigma\Sigma^T) = (\Sigma\Sigma^T)\,\underline{U}$$

$$\underline{U} = \begin{pmatrix} | & | & & | \\ u_1 & u_2 & \cdots & u_m \\ | & | & & | \end{pmatrix}$$

Full ☆

The Economy SVD, and Reduced Rank SVD

The general SVD, Eq. (2.5) may be written in terms of submatrices.

Definition 2.1.3 — The Economy SVD. For any matrix $A \in \mathbb{R}^{m\times n}$, the general SVD Eq. (2.5) can be written in terms of smaller matrices,

$$A_{m\times n} = \hat{U}_{m\times n}\hat{\Sigma}_{n\times n}V^*_{n\times n}, \tag{2.21}$$

and $U = [\hat{U}_{m\times n}|\hat{U}_{(n-m)\times n}]$, written in terms of an orthogonal "buffer" matrix

$\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_r > \sigma_{r+1} = 0$

$= 0 \dots$

Definition 2.1.4 — Rank Deficient SVD. For a matrix $A \in \mathbb{R}^{m\times n}$ such that the SVD results in singular values

$$\sigma_r > \sigma_{r+1} = 0, \text{ for some } r < n. \tag{2.22}$$

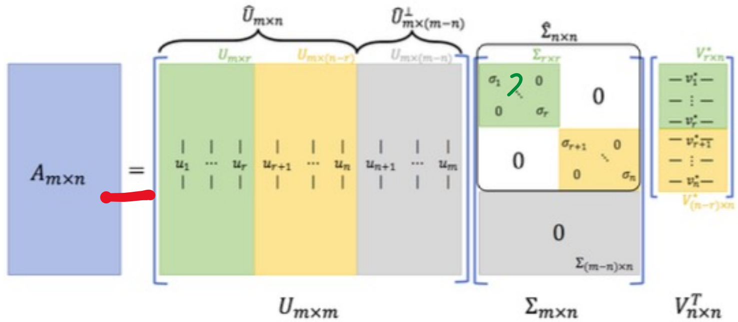
then the SVD can be written in terms of an economy form as smaller matrices,

$$A_{m\times n} = \hat{U}_{m\times r}\hat{\Sigma}_{n\times n}V^*_{n\times r}, \tag{2.23}$$

and related to the general SVD Eq. (2.5) by $U = [\hat{U}_{m\times r}|\hat{U}_{(n-r)\times n}]$, but $r < n$.

Rank ill conditioned

Full,
Economy,
Truncated
SVD



m>n

Figure 2.3: $m > n$ tall skinny

$\sigma_{r+1} = 0$

$\sigma_r = 10^{-6}$

$\sigma_{r+1} \sim 10^{-13}$

$< \varepsilon$

m<n

$A = [\quad]$

Recall that,

$$
\begin{aligned}
A_{m\times n} &= \hat{U}_{m\times n}\hat{\Sigma}_{n\times n}\hat{V}^T_{n\times n} \\
&= \begin{bmatrix} | & | & & | \\ u_1 & u_2 & \dots & u_n \\ | & | & & | \end{bmatrix} \begin{bmatrix} \sigma_1 & 0 & \dots & 0 \\ 0 & \sigma_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \sigma_n \end{bmatrix} \begin{bmatrix} - & v_1^T & - \\ - & v_2^T & - \\ - & \vdots & - \\ - & v_n^T & - \end{bmatrix}
\end{aligned} \tag{2.24}
$$

but $V^TV = I$, orthogonality allows:

$$A_{m\times n}\hat{V}_{n\times n} = \hat{U}_{m\times n}\hat{\Sigma}_{n\times n} \tag{2.25}$$

so,

$$A_{m\times n}\begin{bmatrix} | & | & & | \\ v_1 & v_2 & \dots & v_n \\ | & | & & | \end{bmatrix} = \begin{bmatrix} | & | & & | \\ u_1 & u_2 & \dots & u_n \\ | & | & & | \end{bmatrix}\begin{bmatrix} \sigma_1 & 0 & \dots & 0 \\ 0 & \sigma_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \sigma_n \end{bmatrix} \tag{2.26}$$

but this just states n-matrix times vector statements:

$$
\begin{aligned}
Av_1 &= \sigma_1 u_1 \\
Av_2 &= \sigma_2 u_2 \\
&\vdots \\
Av_n &= \sigma_n u_n
\end{aligned} \tag{2.27}
$$

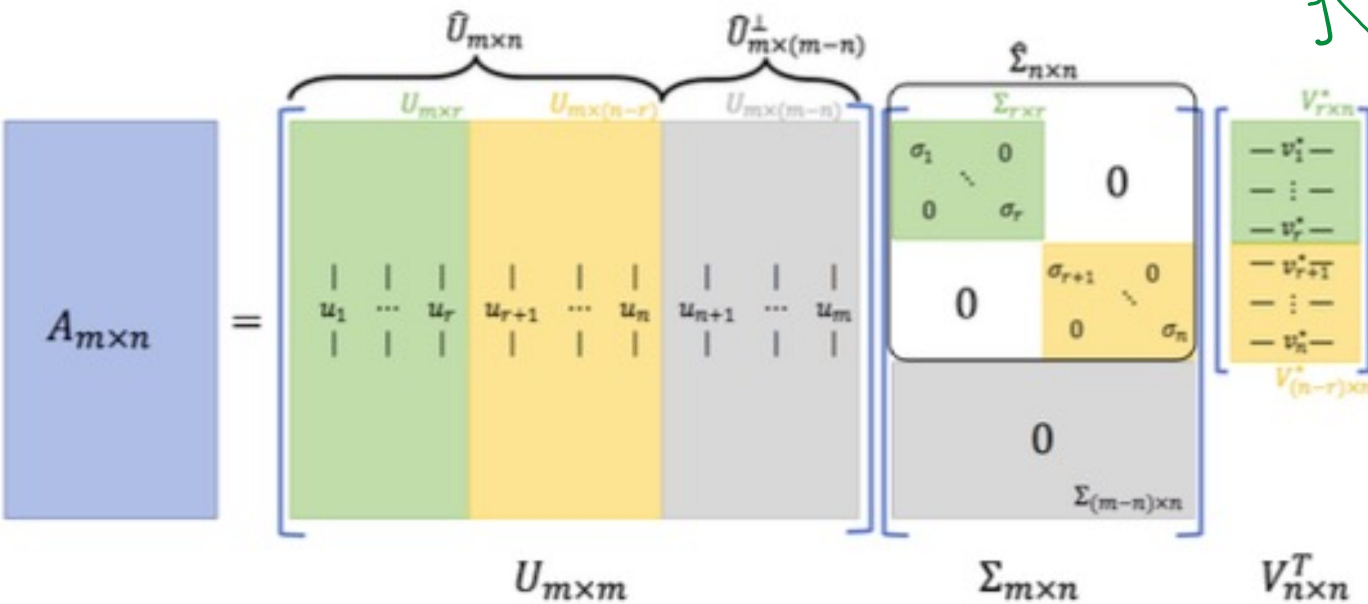
Full,
Economy,
Truncated
SVD

$\sigma_{r+1} = 0$

$\sigma_r = 10^{-6}$

$\sigma_{r+1} \approx 10^{-13}$

$< \varepsilon$



$m > n$

Figure 2.3: $m > n$ tall skinny

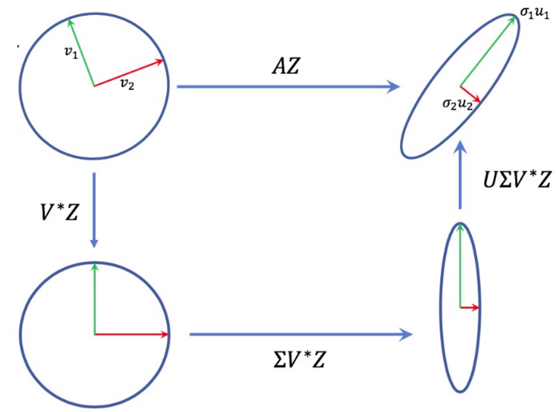
Geometry:.

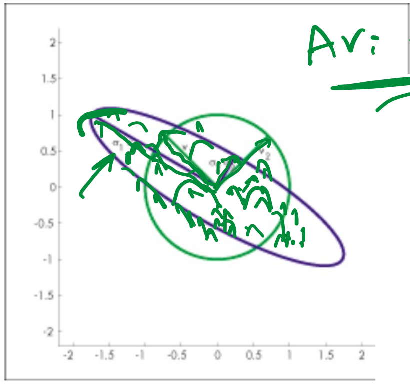
1. V^* rotates to a standard configuration.
2. Σ stretches each orthogonal axis to the major covariance axis of the corresponding ellipsoid, and
3. U rotates results back to the configuration that associates with A.

$A = U \Sigma V^T$

$A V = U \Sigma$

$A v_i = \sigma_i u_i$



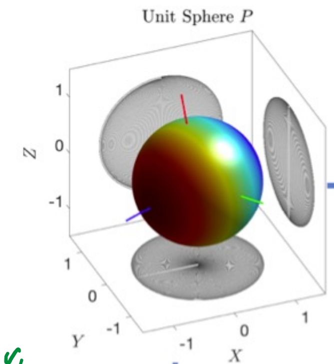


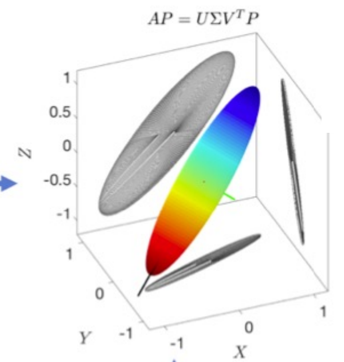
$\sigma_1 \sim 1.5$

$\sigma_2 \sim 0.5$?

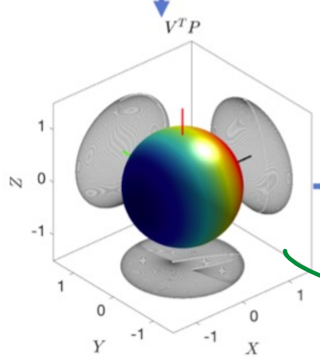
what if $\sigma_1 = 2$

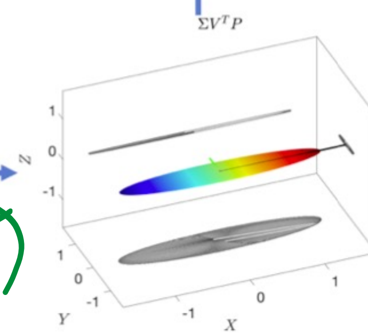
$\sigma_2 = 0.0003$






```
disp(A)

    0.8929    0.0417    0.1000
    0.3320    0.1077    0.0000
    0.8212    0.5951    0.0000

[U,S,V] = svd(A)

U = 3x3
   -0.6330    0.7469    0.2035
   -0.2590    0.0435   -0.9649
   -0.7296   -0.6635    0.1659

S = 3x3
    1.3438         0         0
         0    0.3912         0
         0         0    0.0208

V = 3x3
   -0.9304    0.3488   -0.1126
   -0.3635   -0.9176    0.1612
   -0.0471    0.1909    0.9805
```





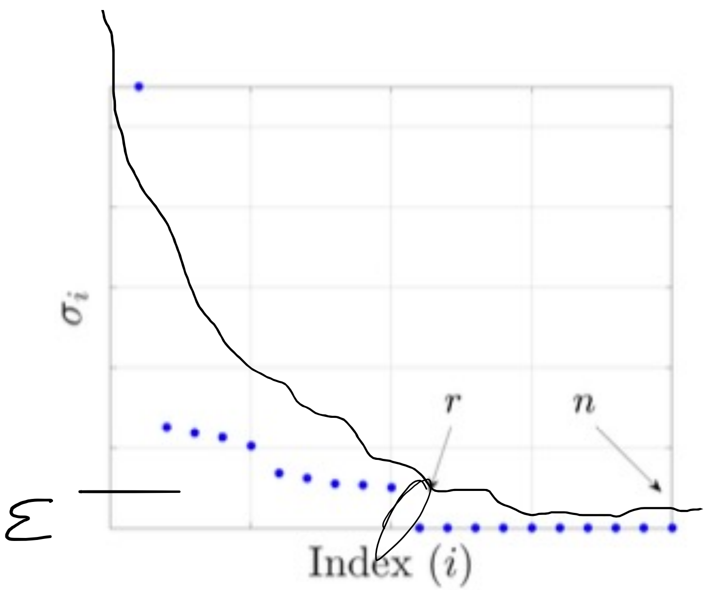
A

$A w = U \Sigma V^T w$

$v_i^T w$

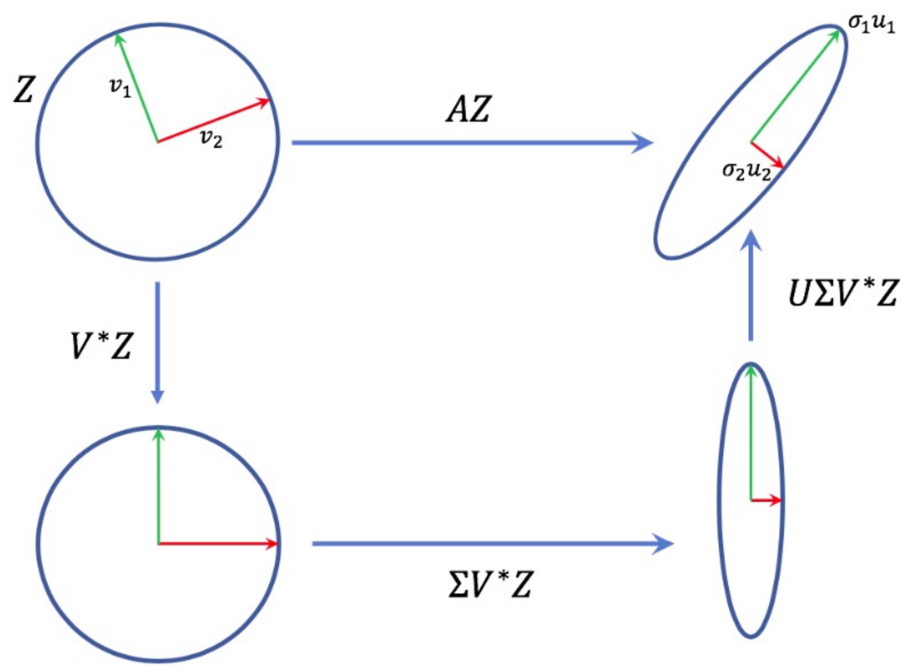
$= v_i \cdot w$

And ROM



$\sigma_r > \varepsilon > \sigma_{r+1}$

$\sim$
0



Bunny Compression

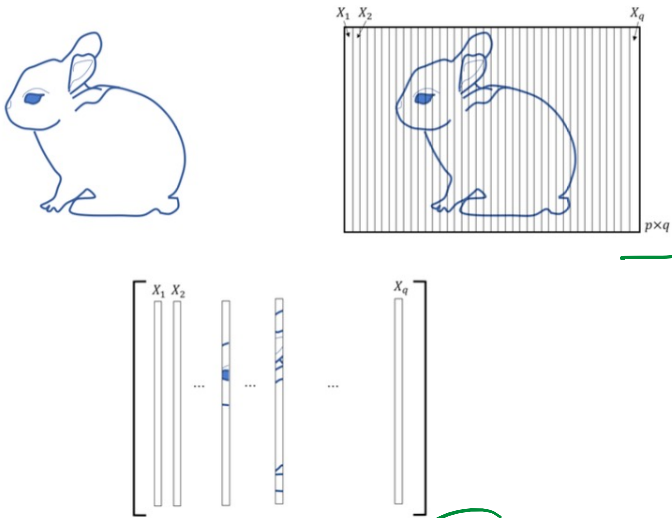
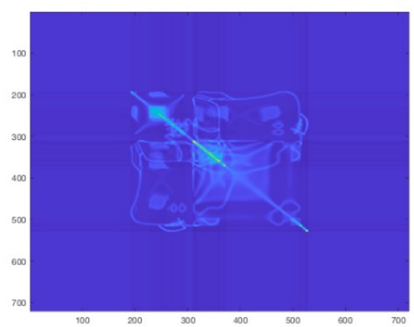


Figure 2.6: Caption

X



$X^T X$

Covariance – notice the demean step

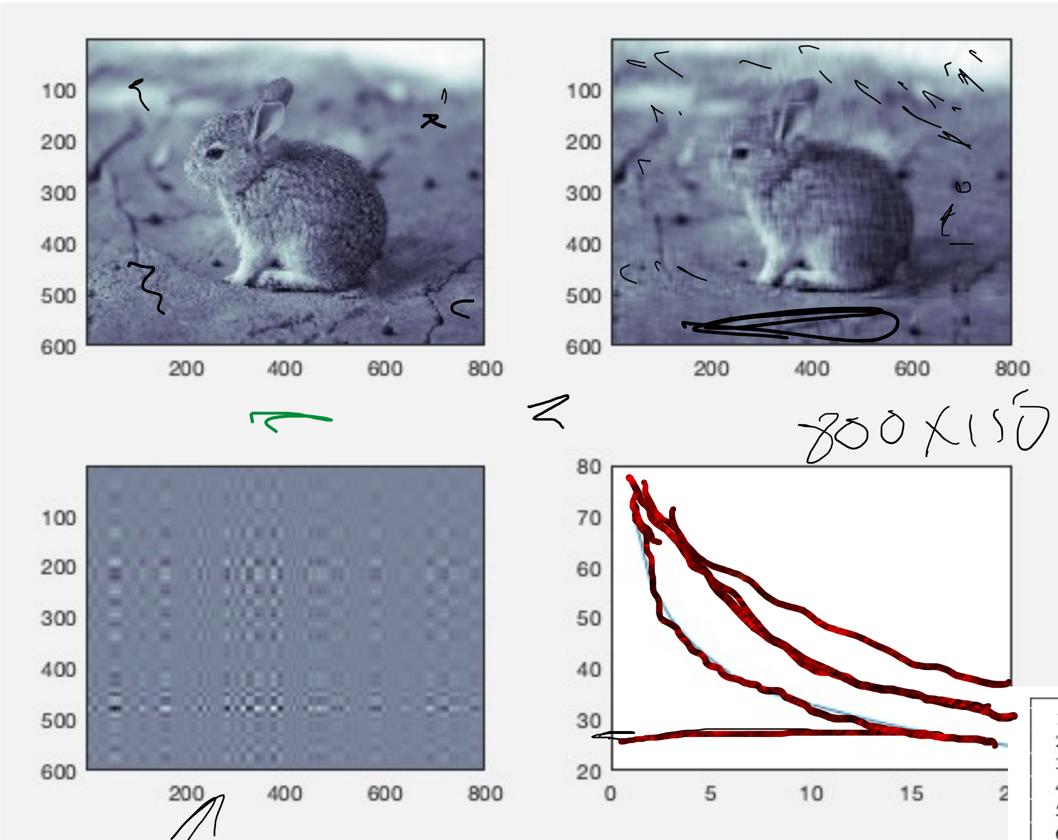
$$C_I = \frac{1}{n-1}\left(X - \tilde{X}\right)^T \left(X - \tilde{X}\right)$$

$A^T \quad A$

$\bar{X}$

$\bar{X}$

$\bar{X} - \tilde{\bar{X}}$



800 x 150

$$U(x,t) = \sum a_i(t)\,\phi_i(x)$$

$$|a_i(t)| \xrightarrow{i \to \infty} 0$$

as fast as possible.

$$I = U \Sigma V^T$$

800 x 150

$$D \approx \hat{U} \hat{\Sigma} \hat{V}^T$$

800 x 0

```matlab
I = imread('Bunny.jpg');

figure
subplot(1,2,1)
imshow(I)
xticks({}); yticks({});
pbaspect([1 1 1])
title('RGB Image')

I = rgb2gray(I);  %Convert the 3D RGB color to 1D grayscale
I = im2double(I); %Convert integer value to double (scaled ...
    from 0 to 1)

subplot(1,2,2)
imshow(I)
xticks({}); yticks({});
pbaspect([1 1 1])
title('Grayscale Image')
```

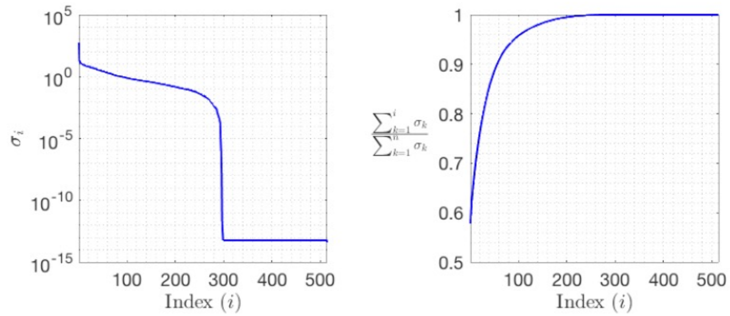
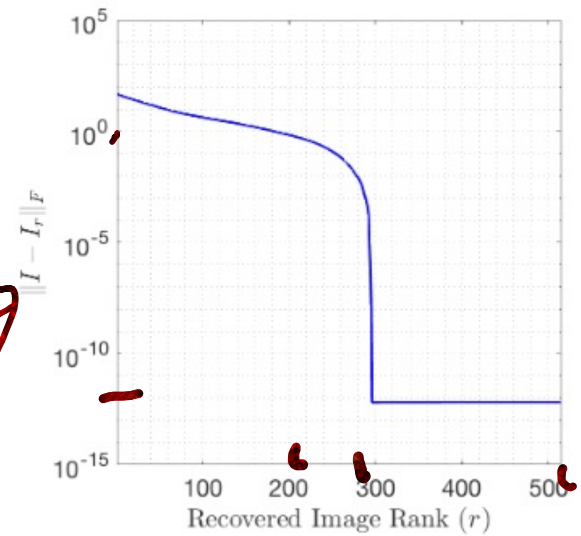
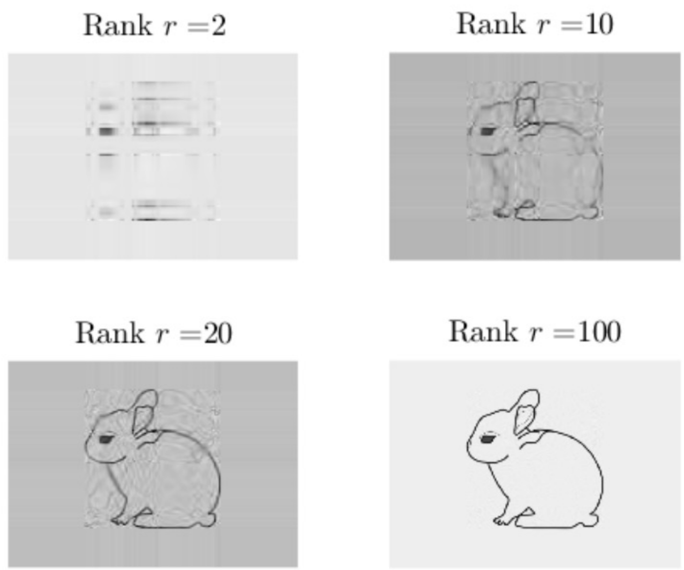


Figure 2.8: (Left) Singular Values. (Right) Energy



$$e_{rel} = \frac{\|I - I_i\|_F}{\|I\|_F}$$



: Distance $\|I - I_r\|_F$, where I_r is the recovered image using the reduced

Code 2.1: Read, convert, and display images.

```
I = imread('Bunny.jpg');

figure
subplot(1,2,1)
imshow(I)
xticks({}); yticks({});
pbaspect([1 1 1])
title('RGB Image')

I = rgb2gray(I);  %Convert the 3D RGB color to 1D grayscale
I = im2double(I); %Convert integer value to double (scaled ...
    from 0 to 1)

subplot(1,2,2)
imshow(I)
xticks({}); yticks({});
pbaspect([1 1 1])
title('Grayscale Image')
```

History

Gene Golub's license plate, photographed by Professor P. M. Kroonenberg of Leiden University.Gene Howard Golub (February 29, 1932 – November 16, 2007), Fletcher Jones Professor of Computer Science at Stanford University. His work made fundamental contributions that have made the singular value decomposition practical as one of the most powerful and widely used tools in modern matrix computation.

Lot's of Machine learning & Data Analysis is solving an ill-posed - optimize a cost function.

Definition 2.1.2 — Induced Norm. Suppose a vector norm $\|\cdot\|$ on $\mathcal{K}^m$ is given. Any matrix $A_{m\times n}$ induces a linear operator from $\mathcal{K}^n$ to $\mathcal{K}^m$ with respect to the standard basis, and one defines the corresponding induced norm or operator norm on the space $\mathcal{K}^{m\times n}$ of all $m \times n$ matrices as follows:

$$\|A\|_p = \sup_{x\neq 0} \frac{\|Ax\|_p}{\|x\|_p} \tag{2.14}$$

or, taking a vector x such that $\|x\|_p = 1$, then we have

$$\|A\|_p = \sup_{\|x\|_p=1} \|Ax\|_p \tag{2.15}$$

Some Special (Simple) Matrix Norms

The first 3 of these are induced norms, but the 4th is not.

- For $p = 1$:

$$\|A\|_1 = \max_{1\le j\le n} \sum_{i=1}^{m} |a_{ij}| \tag{2.16}$$

- For $p = \infty$:

$$\|A\|_\infty = \max_{1\le i\le m} \sum_{j=1}^{n} |a_{ij}| \tag{2.17}$$

- A special case is the spectral norm when $p = 2$, in which we have:

$$\|A\|_2 = \sqrt{\lambda_{max}(A^T A)} = \sigma_{max} \tag{2.18}$$

where σ_{max} is the maximum singular value of the matrix A.

- The Frobenius norm is given by:

$$\|A\|_F = \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n} |a_{ij}|^2} = \sqrt{\sum_{i=1}^{\min\{m,n\}} \sigma_i^2} \tag{2.19}$$

Theorem 2.1.2 For a matrix A, the product of the singular values of A, equals the absolute value of its determinant:

$$|det(A)| = \prod_{i=1}^{n} \sigma_i \tag{2.20}$$

$Ax = x'$

$p=1: \|(x_1, x_2)\|_1 = |x_1| + |x_2|$

$p=\infty: \|(x_1, x_2)\|_\infty = \max_i |x_i|$

$p=2: \|(x_1, x_2)\|_2 = \sqrt{x_1^2 + x_2^2}$

$x = \begin{pmatrix}1\\3\end{pmatrix}$; $\|x\|_1 = 1+3 = 4$

$\|x\|_\infty = 3$

$\|x\|_2 = \sqrt{1^2+3^2} = \sqrt{10}$

Definition 2.1.2 — Induced Norm. Suppose a vector norm $\|\cdot\|$ on $\mathcal{K}^m$ is given. Any matrix $A_{m\times n}$ induces a linear operator from $\mathcal{K}^n$ to $\mathcal{K}^m$ with respect to the standard basis, and one defines the corresponding induced norm or operator norm on the space $\mathcal{K}^{m\times n}$ of all $m \times n$ matrices as follows:

$$\|A\|_p = \sup_{x\neq 0} \frac{\|Ax\|_p}{\|x\|_p} \tag{2.14}$$

or, taking a vector x such that $\|x\|_p = 1$, then we have

$$\|A\|_p = \sup_{\|x\|_p=1} \|Ax\|_p \tag{2.15}$$

Some Special (Simple) Matrix Norms

The first 3 of these are induced norms, but the 4th is not.

- For $p = 1$:

$$\|A\|_1 = \max_{1\leq j\leq n} \sum_{i=1}^{m} |a_{ij}| \tag{2.16}$$

- For $p = \infty$:

$$\|A\|_\infty = \max_{1\leq i\leq m} \sum_{j=1}^{n} |a_{ij}| \tag{2.17}$$

- A special case is the spectral norm when $p = 2$, in which we have:

$$\|A\|_2 = \sqrt{\lambda_{max}(A^T A)} = \sigma_{max} \tag{2.18}$$

  where σ_{max} is the maximum singular value of the matrix A.
- The Frobenius norm is given by:

$$\|A\|_F = \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n} |a_{ij}|^2} = \sqrt{\sum_{i=1}^{\min\{m,n\}} \sigma_i^2} \tag{2.19}$$

Theorem 2.1.2 For a matrix A, the product of the singular values of A, equals the absolute value of its determinant:

$$|det(A)| = \prod_{i=1}^{n} \sigma_i \tag{2.20}$$

Fun facts about matrix estimation (data estimation)

If A, $\sigma_1 \geq \cdots \geq \sigma_r > \sigma_{r+1} = 0$

- $\text{range}(A) = \text{span}(u_1, u_2, \ldots, u_r)$
- $\text{null}(A) = \text{span}(v_{r+1}, v_{r+2}, \ldots, v_n)$
- $\|A\|_2 = \sigma_1$; $\|A\|_F = \sqrt{\sigma_1^2 + \sigma_2^2 + \cdots + \sigma_r^2}$
- $A = \sum_{i=1}^{r} \sigma_i u_i v_i^* = \sigma_1 u_1 v_1^* + \sigma_2 u_2 v_2^* + \cdots + \sigma_r u_r v_r^*$

rank-1 outer products.

$\} 0 \xrightarrow{A}$

$\sigma_2 = 0$

$r = 1$

$A = U \Sigma V^T$

$$\begin{pmatrix} u_1 & u_2 & \cdots & u_m \end{pmatrix}_{m \times r} \begin{pmatrix} \sigma_1 & \sigma_2 & \cdots & \sigma_r \end{pmatrix} \begin{pmatrix} - v_1^T - \\ - v_2^T - \\ \vdots \end{pmatrix} \quad 1 \times n$$

$w_1 \to w_2$

$w_1^T w_2 = w_1 \cdot w_2$

$= \|w_1\| \|w_2\| \cos\theta$

$$\begin{pmatrix} 1 & 2 \\ 2 & 4 \end{pmatrix}$$

Matrix Estimation / Data Estimation. $A_{m\times n}$

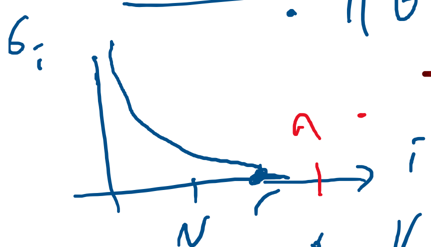
A

- let $0 \le N \le r$ and $A_N = \sum_{i=1}^{N} \sigma_i u_i v_i^*$

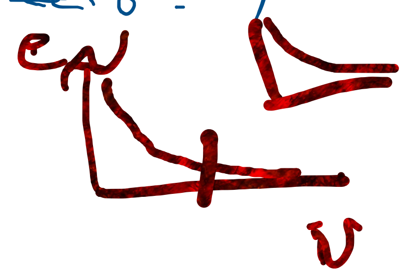
(so we may be skipping some of them ... $\sum_{i=N+1}^{r}$

Then. $\|A - A_N\|_2 = \sigma_{N+1}$ (first one skipped)

(what if it zero?)



e_N $\|A - A_N\|_F = \sqrt{\sigma_{N+1}^2 + \sigma_{N+2}^2 + \dots + \sigma_r^2}$

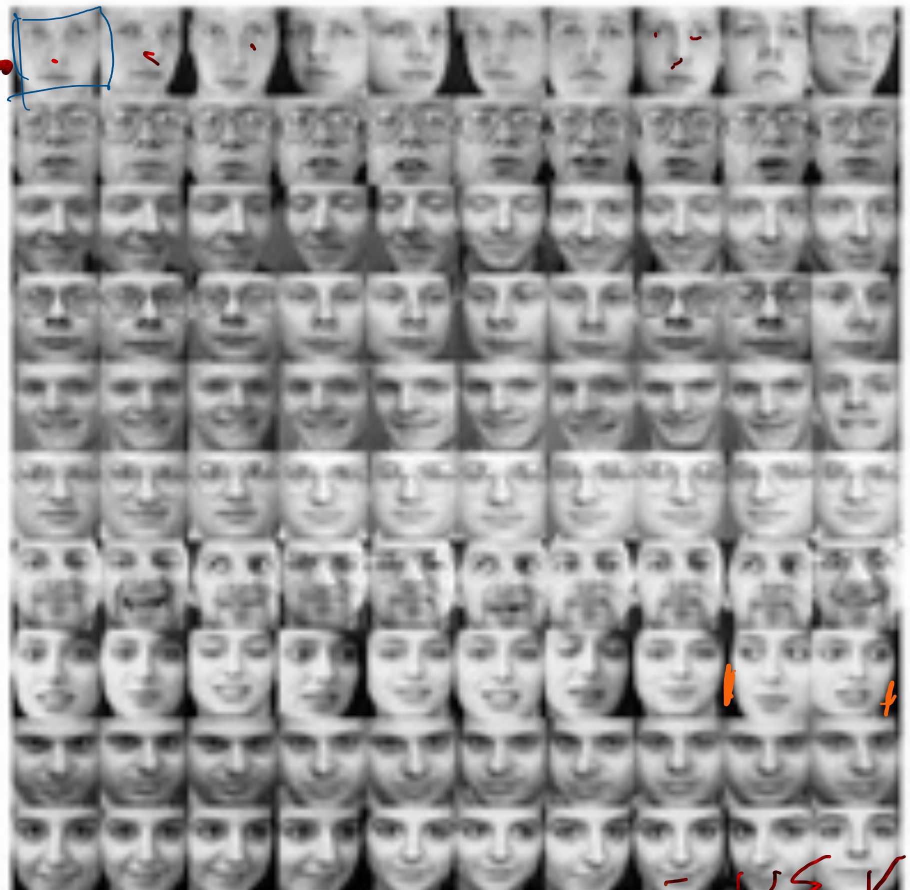


- EigenFace 1st Present

Eigenface

08/31/20

i-th picture



i

$p \times q$ array

reshape as vector

x_i $\quad e_1 = [1, 0, 0, 0, 0]^T$

$M \times 1$

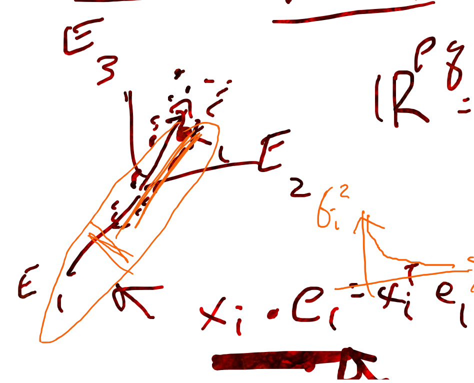
$M = pq$

$X = [x_1 | x_2 | \cdots | x_n]_{M \times n} = U \Sigma V^T$

$p = 1000, q = 200$

- remove variance

Register



$\mathbb{R}^{pq} = \mathbb{R}^M$

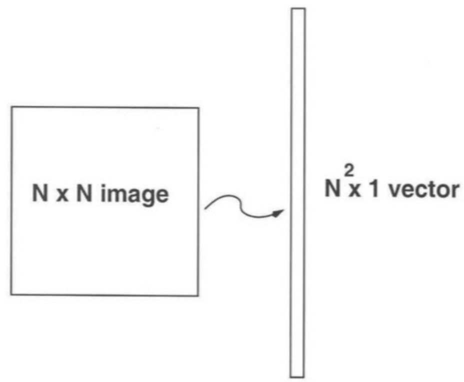
$x_i \cdot e_1 = x_i^T e_1 = E$

Eigenfaces for Face Detection/Recognition

(M. Turk and A. Pentland, "Eigenfaces for Recognition", *Journal of Cognitive Neuroscience*, vol. 3, no. 1, pp. 71-86, 1991, hard copy)

• Face Recognition

- The simplest approach is to think of it as a template matching problem:



- Problems arise when performing recognition in a high-dimensional space.

- Significant improvements can be achieved by first mapping the data into a *lower-dimensionality* space.

- How to find this lower-dimensional space?

• Main idea behind eigenfaces

- Suppose Γ is an N^2x1 vector, corresponding to an NxN face image I.

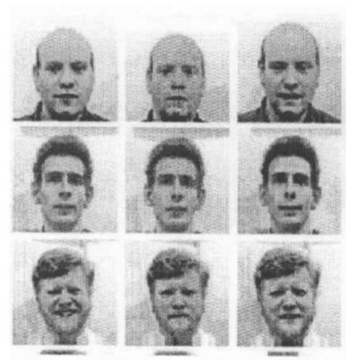
- The idea is to represent Γ (Φ=Γ - mean face) into a low-dimensional space:

$$\hat{\Phi} - mean = w_1 u_1 + w_2 u_2 + \cdots w_K u_K \ (K << N^2)$$

Computation of the eigenfaces

Step 1: obtain face images $I_1, I_2, ..., I_M$ (training faces)

(**very important:** the face images must be *centered* and of the same *size*)



Step 2: represent every image I_i as a vector Γ_i

Step 3: compute the average face vector Ψ:

$$\Psi = \frac{1}{M} \sum_{i=1}^{M} \Gamma_i$$

Step 4: subtract the mean face:

$$\Phi_i = \Gamma_i - \Psi$$

Step 5: compute the covariance matrix C:

$$C = \frac{1}{M} \sum_{n=1}^{M} \Phi_n \Phi_n^T = AA^T \quad (N^2 \text{x} N^2 \text{ matrix})$$

$$\text{where } A = [\Phi_1 \; \Phi_2 \cdots \Phi_M] \quad (N^2 \text{x} M \text{ matrix})$$

$C = X^T X$

Step 6: compute the eigenvectors u_i of AA^T

The matrix AA^T is very large --> not practical !!

Step 6.1: consider the matrix $A^T A$ (MxM matrix)

Step 6.2: compute the eigenvectors v_i of $A^T A$

$$A^T A v_i = \mu_i v_i$$

What is the relationship between us_i and v_i?

$A^T A v_i = \mu_i v_i$ => $AA^T A v_i = \mu_i A v_i$ =>

$CAv_i = \mu_i A v_i$ or $Cu_i = \mu_i u_i$ where $u_i = Av_i$

Thus, AA^T and $A^T A$ have the same eigenvalues and their eigenvectors are related as follows: $u_i = Av_i$!!

Note 1: AA^T can have up to N^2 eigenvalues and eigenvectors.

Note 2: $A^T A$ can have up to M eigenvalues and eigenvectors.

Note 3: The M eigenvalues of $A^T A$ (along with their corresponding eigenvectors) correspond to the M *largest* eigenvalues of AA^T (along with their corresponding eigenvectors).

Step 6.3: compute the M best eigenvectors of AA^T: $u_i = Av_i$

(**important:** normalize u_i such that $\|u_i\| = 1$)

Step 7: keep only K eigenvectors (corresponding to the K largest eigenvalues)

$\underline{X} = U \Sigma V^T$

$U = [u_1 | u_2 | \dots | u_s | u_{s+1} | \dots | u_r | u_{r+1} \dots u_n]$

Representing faces onto this basis

- Each face (minus the mean) Φ_i in the training set can be represented as a linear combination of the best K eigenvectors:

$$\hat{\Phi}_i - mean = \sum_{j=1}^{K} w_j u_j, \quad (w_j = u_j^T \Phi_i)$$

(we call the u_j's *eigenfaces*)

- Each normalized training face Φ_i is represented in this basis by a vector:

$$\Omega_i = \begin{bmatrix} w_1^i \\ w_2^i \\ \dots \\ w_K^i \end{bmatrix}, \quad i = 1, 2, \dots, M$$

• imagesc(g)

• $g = \text{reshape}(U(:,i), p, q)$

$\|\vec{a}\|_2^2 = \sum a_i^2$

Parseval ineq.

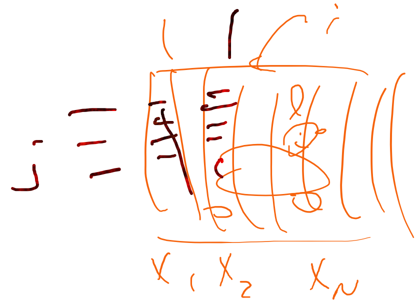
$\|\vec{a}\|_2^2 \cong \|f\|_2^2 = \int |f(x)|^2 dx$

$= a_1 \cdot \; + a_2 \; + a_3 \; + \dots$

Fourier

Taylor.

Bunny



$\rightarrow \bar{X} = [x_1 \,|\, x_2 \,|\, \dots \,|\, x_N]$ ← one column

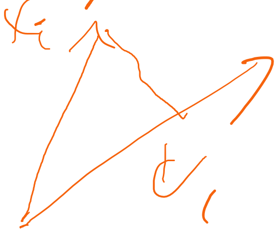
$= U \Sigma V^T$

PCA of the bunny

$x_i = a_1^i \vec{u}_1 + a_2^i \vec{u}_2 + \dots + a_N^i \vec{u}_N$

basis

full of as possible for the data set



$a_1^i = x_i^T u_1 = \mathrm{comp}_{u_1} x_i = \frac{u_1 \cdot x_i}{\|u_1\|} = \|x_i\| \cos\theta$

- On PCA

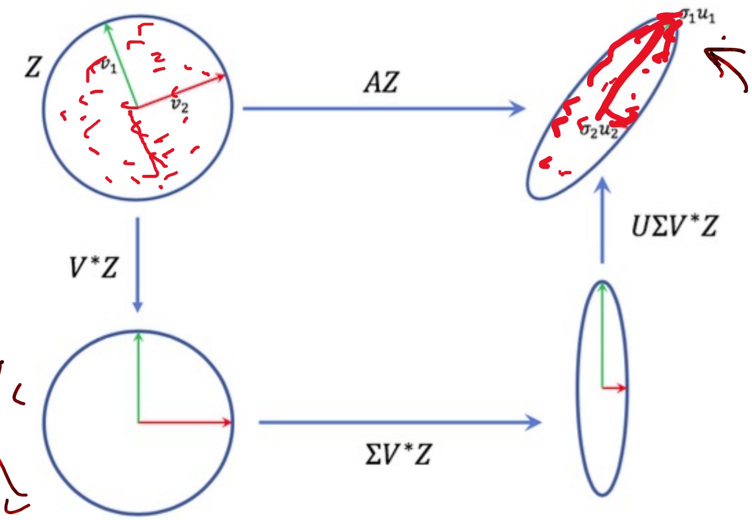
- Optimal

On PCA Principal Component Analysis, Eigenface

-On Raleigh Ritz Quotient

-On Spectral Decomposition Theorem

-On Data Clouds



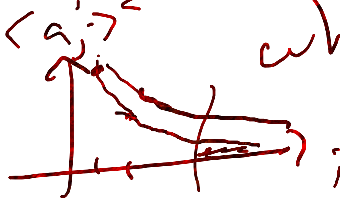
$$\underline{\overline{X}} = [x_1 \mid x_2 \mid \ldots \mid x_n]$$

$$\vec{x}_i = \underline{a^i_1 \vec{v}_1} + \underline{a^i_2} \vec{v}_2 + \ldots + \underline{a^i_r} \vec{v}_r$$

v's as basis set.

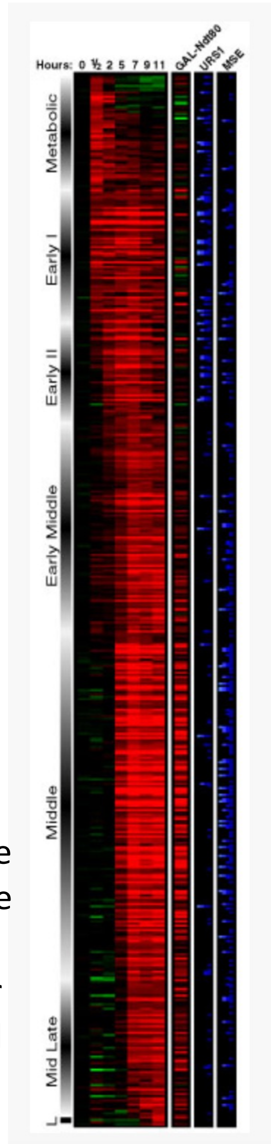
x_i - PCA gives basis set where $\langle a^i_j \rangle_i \downarrow$ as fast as possible vs any other basis set.

$$\langle a^i_1 \rangle_i = \frac{1}{n} \sum_{i=1}^{n} a^i_1$$



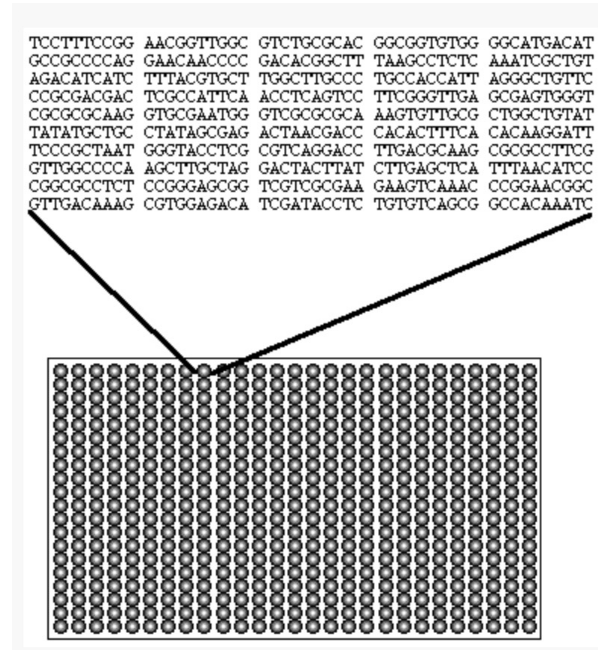
DNA Microarrays

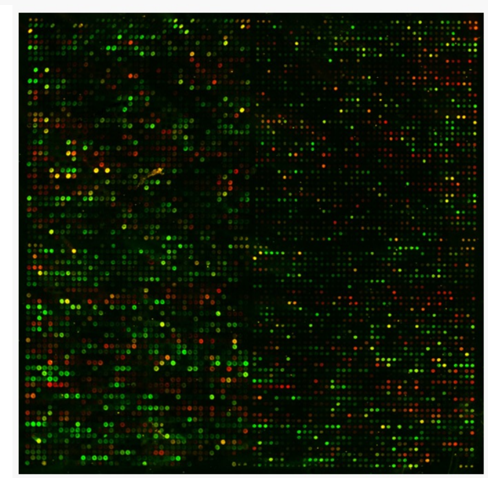
Gene Expression



Microarray results that have been analyzed such that the colors were linked with expression and then similar gene profiles were grouped Together – budding yeast

cartoon illustrating an array of DNA snippets on a chip. The top portion depicts a possible nucleotide sequence for the DNA segment immobilized in the position indicated.

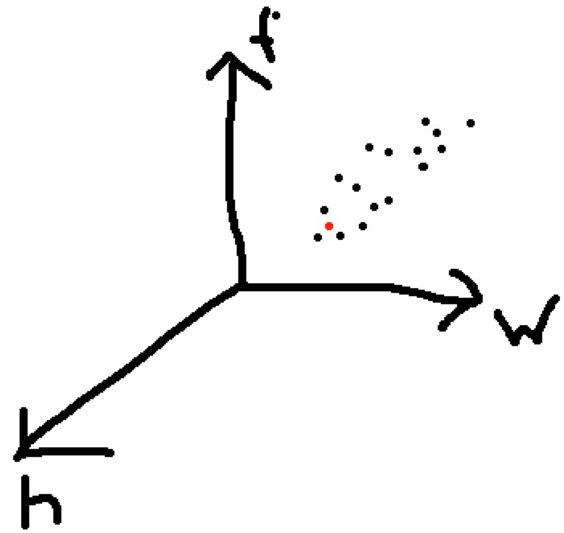




DNA Microarray chip containing the entire yeast genome

Morphological

- Height
- Weight
- Footsize
- Belt (waist) size
- Hand size
- Forearm size
- Head circumference
- Femur length



Interpreting as an ellipsoid in the high dimensional space is the simplest geometric interpretation of the data cloud and leads to simplification as major and minor axis, and even Reduced order model (ROM) (meaning a lower dimensional representation).

PCA, SVD, SDT – is optimal

Data for PCA - "Pretend data looks like an ellipsoid"

Ex: $X_i \sim 4500 \times 1$ gene expression table for each i.

$i = 1 \ldots 216$ patients

$$X_i = \begin{pmatrix} x_i^1 \\ x_i^2 \\ \vdots \\ x_{4500} \end{pmatrix}$$

$y_i = 0$ or 1 "0" if not cancer "1" if cancer.

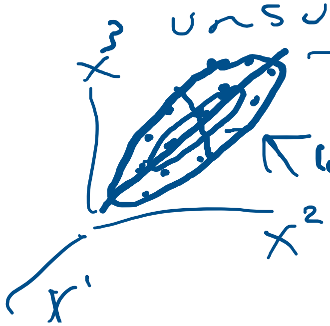
$f : \mathbb{R}^{4500} \to \mathbb{Z}_2$ $\qquad \mathbb{Z}_2 = \{0, 1\}$.

$x \in \mathbb{R}^{4500}$

- Supervised vs. unsupervised.

unsupervised - just input - just structure / geometry.

just x



$\leftarrow$ "Data Cloud" ~ distribution

level sets of ρ

$\rho(x) : \mathbb{R}^{4500} \to \mathbb{R}^+$ of R.V. $x \sim X$

- supervised learning is descriptive $f : x \to y$ labels

The spectral decomposition

Let A be a $n \times n$ symmetric matrix. From the spectral theorem, we know that there is an orthonormal basis $u_1, \cdots, u_n$ of $\mathbb{R}^n$ such that each u_j is an eigenvector of A. Let λ_j be the eigenvalue corresponding to u_j, that is,

$$Au_j = \lambda_j u_j.$$

Then

$$A = PDP^{-1} = PDP^T$$

where P is the orthogonal matrix $P = [u_1 \ \cdots \ u_n]$ and D is the diagonal matrix with diagonal entries $\lambda_1, \cdots, \lambda_n$. The equation $A = PDP^T$ can be rewritten as:

$$A = [u_1 \ \cdots \ u_n] \begin{bmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_n \end{bmatrix} \begin{bmatrix} u_1^T \\ \vdots \\ u_n^T \end{bmatrix}$$

$$= [\lambda_1 u_1 \ \cdots \ \lambda_n u_n] \begin{bmatrix} u_1^T \\ \vdots \\ u_n^T \end{bmatrix}$$

outer

$$= \lambda_1 u_1 u_1^T + \cdots + \lambda_n u_n u_n^T.$$

$n \times 1 \cdot 1 \times n$
$n \times n$

$\|v_i\|^2 = v_i \cdot v_i = v_i^T v_i$ scalar - inner product

The expression

$$A = \lambda_1 u_1 u_1^T + \cdots + \lambda_n u_n u_n^T.$$

is called the spectral decomposition of A. Note that each matrix $u_j u_j^T$ has rank 1 and is the matrix of projection onto the one dimensional subspace spanned by u_j. In other words, the linear map P defined by $P(x) = u_j u_j^T x$ is the orthogonal projection onto the subspace spanned by u_j.

- $A = B^T B$ is symmetric
$\Rightarrow$ spectral decomp. theorem
i.e. also covariance matrices.

- A is pos. definite if $\lambda_i > 0$ all i.

PCA as algorithm

(say $m = 4000$, $n = 216$)

- Data $\underline{\overline{X}} = \begin{pmatrix} | & | & & | \\ x_1 & x_2 & \dots & x_n \\ | & | & & | \end{pmatrix}_{m \times n}$ ith

- what if $x_i \sim \underline{\mathcal{N}(\bar{x}, \Sigma)}$ — covariance matrix.

$B = \underline{X} - \bar{B}$; $\bar{B} = \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix}_{n \times 1} \bar{x}^T_{1 \times n} = \underline{1}\,\bar{x}^T$. $\bar{x} = \begin{pmatrix} \bar{x}_1 \\ \bar{x}_2 \\ \vdots \\ \bar{x}_n \end{pmatrix}$

$\bar{x}_i = \frac{1}{n} \sum_{i=1}^{n} \underline{X}_{ij}$

$B = U \Sigma V^T$; $U = [u_1\ u_2 \dots u_n]$

and u_1 is major axis — most energetic — feature that most explains data.

u_2 is first minor axis

Side note:

$\frac{1}{i^2}$ is slowest converging to zero σ_i^2

i.e. $\frac{1}{i}$ is slowest converging σ_i

$\sigma_i \leq \frac{1}{i}$

$$\sum_{i=1}^{\infty} \frac{1}{i^p} < \infty \text{ if } \cancel{p > 0} \text{ bad)}$$

$p \leq 1$, $p = 1$ harmonic

σ_i^2 the power spectrum

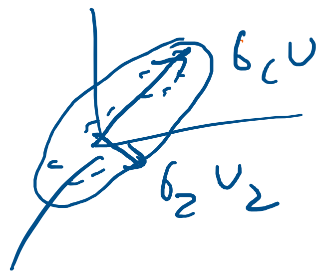
$= 1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \cdots \quad p = 1$

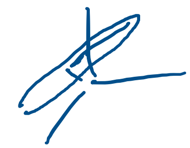
$B = I\bar{X}^T$; let $C = \frac{1}{n-1} B^T B$ everywhere

is covariance matrix.

$B = U \Sigma V^T$

$U = [u_1 | u_2 \ldots | u_n]$





- $u_1 = \underset{\|u\|=1}{\operatorname{argmax}}\ u^T B^T B u = \underset{u}{\operatorname{argmax}} \frac{(u^T B^T)(B u)}{u^T u}$

$\|Bu\|_2^2$

Raleigh - Ritz quotient.

$Bu \cdot Bu = \|Bu\|_2^2$

- $u_2 = \underset{\substack{\|u\|=1 \\ u \perp u_1}}{\operatorname{argmax}}\ u^T B^T B u$

∘ $CV = VD$ eigenvectors of C all stacked.

Ax optimize $x^T A x$, maximize. $\left(A = B^T B,\ x^T x = 1 \right)$

$$x = \begin{pmatrix} x^1 \\ \vdots \\ x^n \end{pmatrix} \leftarrow j \qquad r(x) = \frac{x^T A x}{x^T x}$$

$$\frac{\partial r}{\partial x^j} = \frac{\partial}{\partial x^j}\left(r(x)\right) = \frac{\frac{\partial}{\partial x^j}\left(x^T A x\right) - x^T A x \frac{\partial}{\partial x^j}\left(x^T x\right)}{(x^T x)^2}$$

$$= \frac{2(Ax)_j}{x^T x} - \frac{(x^T A x)\, 2x_j}{(x^T x)^2} = \frac{2}{x^T x}\left(Ax - r(x)x\right)_j$$

$$\nabla r(x) = \frac{2}{x^T x}\left(Ax - r(x)x\right) = \frac{2}{x^T x}\left(A - r(x)I\right)x = 0$$

$$Ax = \underbrace{r(x)}_{\lambda}\, x \quad \Rightarrow \quad Ax = \lambda x$$

Conclude

The x that optimizes $r(x) = \dfrac{x^T A x}{x^T x}$ is an eigenvector and $r(x)$ is its eigenvalue.

• S.D.T. for $A = B^T B = \sum_{i=1}^{n} \lambda_i u_i u_i^T$

u_i are the v_i's

let $B = U \Sigma V^T$ svd.

$$\underline{B^T B} = V \Sigma^T \cancel{U^T U} \Sigma V^T = V \underbrace{\Sigma^T \Sigma} V^T$$

$$D = \begin{pmatrix} \sigma_1 & & & \\ & \ddots & & \\ & & \sigma_r & \\ & & & \ddots \; 0 \end{pmatrix} \begin{pmatrix} \begin{matrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_r \\ & & & \ddots \; 0 \end{matrix} & 0 \end{pmatrix} = \begin{pmatrix} \sigma_1^2 & & & \\ & \sigma_2^2 & & \\ & & \ddots \; \sigma_r^2 & \\ & & & \ddots \; 0 \end{pmatrix}$$

$$= \underline{V D V^T} = \sum_{i=1}^{r} \underline{\sigma_i^2}\, \underline{v_i}\, \underline{v_i^T}$$

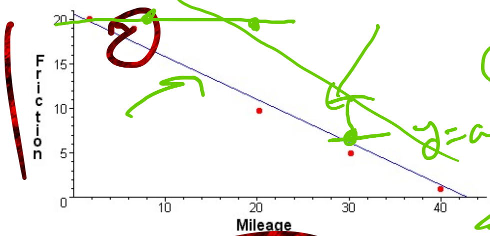
On Moore Penrose Pseudo Inverse, Matrix Least Squares, Geometric Least Squares.

Example

An engineer is tracking the friction index over mileage of a breaking system of a vehicle. She expects that the mileage-friction relationship is approximately linear. She collects five data points that are show in the table below.

Mileage	2000	6000	20,000	30,000	40,000
Friction Index	20	18	10	6	2

The graph below shows these points



We are interested in the line that best fits the data. More specifically, if **b** is the vector of friction index data values and **y** is the vector consisting of y values when we plug in the mileage data for x and find y by the equation of the line, then we want the line that minimizes the distance between **b** and **y**. If the equation of the line is

$$ax + b = y$$

then we get the five equations

$$2a + b = 20$$
$$6a + b = 18$$
$$20a + b = 10$$
$$30a + b = 6$$
$$40a + b = 2$$

The corresponding matrix equation is

$$Ax = y$$

or

$$\begin{pmatrix} 2 & 1 \\ 6 & 1 \\ 20 & 1 \\ 30 & 1 \\ 40 & 1 \end{pmatrix} \begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} 20 \\ 18 \\ 10 \\ 6 \\ 2 \end{pmatrix}$$

Although this does not have an exact solution, it does have a closest solution. We have

$$\begin{pmatrix} a \\ b \end{pmatrix} = (A^T A)^{-1} A^T \mathbf{b} = \begin{pmatrix} -0.48 \\ 20.6 \end{pmatrix}$$

We can conclude that the equation of the regression line is

$$y = -0.48x + 20.6$$

$$J(a,b) = \sum_{i=1} \| y_i - (ax_i + b) \|_2^2$$

$(2000, 20)$

$y = ax + b$ $= (x_i, y_i)$

$y_i = ax_i + b$

$y = ax + b = ax + b \cdot 1$

$2 \cdot a + 1 \cdot b = 20$

$6 \cdot a + 1 \cdot b = 18$

$A x = y$

$x = \begin{pmatrix} a \\ b \end{pmatrix}$

wish $x = A^{-1} y$

$x = A^{+} y$ vs A^T

Theorem

Let A be an m x n matrix or rank n, then the system

$$Ax = b$$

has the unique *least squares* solution

$$\hat{x} = (A^TA)^{-1}A^Tb$$

$Ax = b$
$m \times n$

• Square & nonsing.

$x = A^{-1}b$ unique

A^{-1} d.n.e.

• Not, $m > n$

$(A^TA)^{-1}$ - never form that

$$\boxed{\underbrace{A^TA}_{n \times n}\underset{m \times n}{x} = A^Tb \quad \text{normal equations}}$$

Covariance matrix (+ demeaned.)

Example

Find the least squares solution to

$$Ax = b$$

with

$$A = \begin{pmatrix} 1 & 3 \\ 2 & 4 \\ 1 & 6 \end{pmatrix} \quad b = \begin{pmatrix} 4 \\ 1 \\ 3 \end{pmatrix}$$

Solution

$m > n$ - tall skinny

We can quickly check that A has rank 2 (the first two rows are not multiples of each other). Hence we can compute

$$x = (A^TA)^{-1}A^Tb = \begin{pmatrix} -0.377 \\ .662 \end{pmatrix}$$

$$\left(\begin{pmatrix} 1 & 2 & 1 \\ 3 & 4 & 6 \end{pmatrix}\begin{pmatrix} 1 & 3 \\ 2 & 4 \\ 1 & 6 \end{pmatrix}\right)^{-1}\begin{pmatrix} 1 & 2 & 1 \\ 3 & 4 & 6 \end{pmatrix}\begin{pmatrix} 4 \\ 1 \\ 3 \end{pmatrix}$$

Notice that

$$Ax = \begin{pmatrix} 1.61 \\ 1.90 \\ 3.60 \end{pmatrix}$$

not exactly b, but as close as we are going to get.

Least Squares

Definition and Derivations

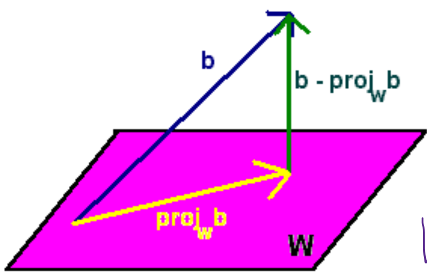
We have already spent much time finding solutions to

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

If there isn't a solution, we attempt to seek the $\mathbf{x}$ that gets closest to being a solution. The closest such vector will be the $\mathbf{x}$ such that

$$A\mathbf{x} = \mathrm{proj}_W \mathbf{b}$$

where W is the column space of A.



$W = \mathrm{col}(A)$

Notice that $\mathbf{b}$ - $\mathrm{proj}_W\mathbf{b}$ is in the orthogonal complement of W hence in the null space of A^T. Hence if $\mathbf{x}$ is a this closest vector, then

$$A^T(\mathbf{b} - A\mathbf{x}) = 0 \qquad A^TA\mathbf{x} = A^T\mathbf{b}$$

Now we need to show that A^TA nonsingular so that we can solve for $\mathbf{x}$.

Lemma

If A is an $m \times n$ matrix of rank n, then A^TA is nonsingular.

$$\underset{m\times n}{\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & & \\ a_{31} & & & \\ \vdots & & & \\ a_{m1} & \cdots & \cdots & a_{mn} \end{pmatrix}} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{pmatrix}$$

m - eqns

n - unknowns

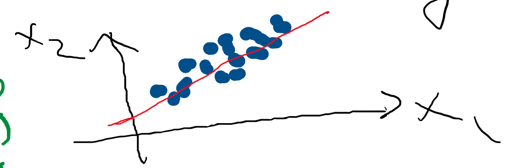
x

$$\Longleftrightarrow a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n = b_1$$
$$\vdots$$
$$a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n = b_n$$

$m > n$ tall skinny, $n = 2$

How many?

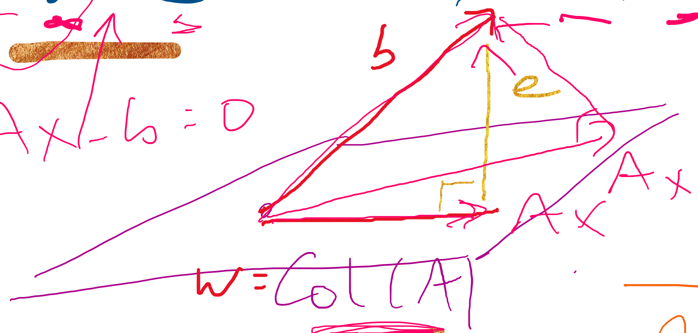
How many?



$$x_1 \begin{pmatrix} a_{11} \\ a_{21} \\ \vdots \\ a_{m1} \end{pmatrix} + x_2 \begin{pmatrix} a_{12} \\ \vdots \\ a_{m2} \end{pmatrix} + \cdots + x_n \begin{pmatrix} a_{1n} \\ \vdots \\ a_{mn} \end{pmatrix} = \begin{pmatrix} b_1 \\ \vdots \\ b_m \end{pmatrix}$$

$Ax = b$ $\Rightarrow x_1 \vec{a}_1 + x_2 \vec{a}_2 + \cdots + x_n \vec{a}_n = \vec{b}$

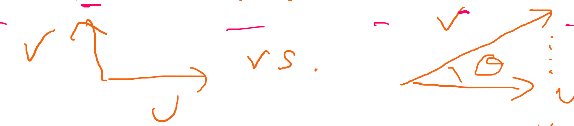
$Ax - b = 0$



$A\tilde{x} = \text{proj}_W b$; $\tilde{x} = \underset{x}{\text{argmin}} \underbrace{\|Ax - b\|_2^2}_{e}$

$\Rightarrow$ LS solution

Recall $u \perp v$ iff $u \cdot v = u^T v = 0$



$u \cdot v = \|u\|\|v\| \cos\theta$

$\underbrace{\vec{e}}_{} = A\tilde{x} - b$

$\Rightarrow A^T(A\tilde{x} - b) = 0 \Longleftrightarrow \begin{pmatrix} -a_1' - \\ -a_2' - \\ \vdots \\ -a_n' - \end{pmatrix} \underbrace{(Ax - b)}_{e} = 0 \text{ " } \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$

$A^T(Ax - b) = 0$

$\Rightarrow \vec{a}_i \perp (A\tilde{x} - b) \ \forall i$

$\Rightarrow (A\tilde{x} - b) \perp$ every vector in Col(A) $\Longleftrightarrow$ Solve "normal eqns"

$A^T A \tilde{x}$

$A^T b$

LS soln = solve normal equations

$$A^T A \tilde{x} = A^T b$$

When inverse exists

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

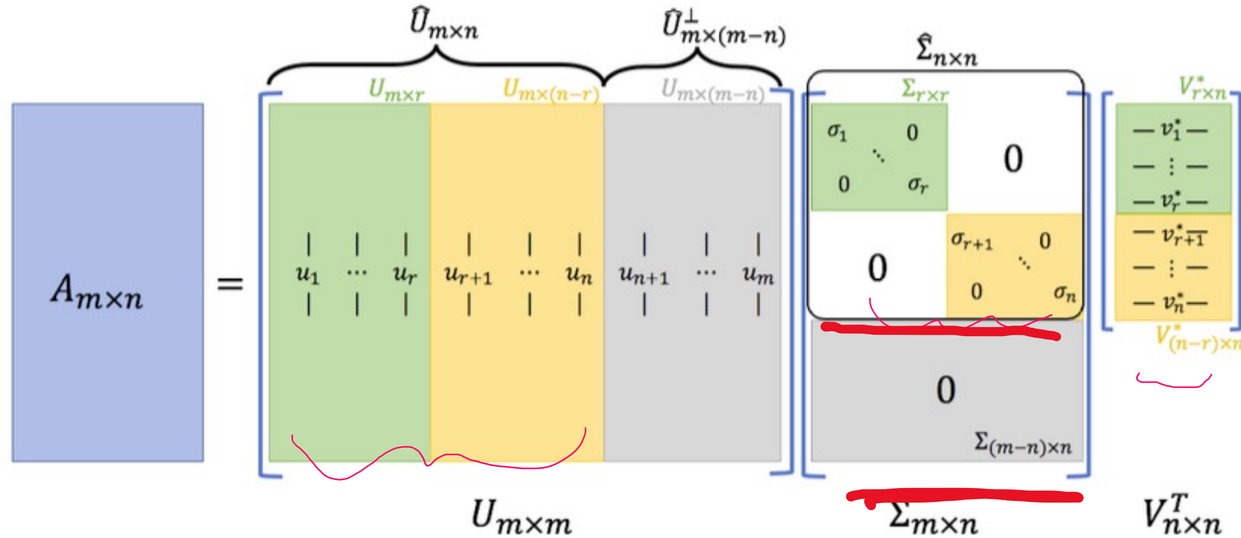
$$= A^+ b$$

A^+

Moore-Penrose; Pseudo-Inverse

$Ax = b$

- In terms of SVD?
- and what if inverse doesn't exist.



$$(\hat V \hat\Sigma^T \cancel{\hat U^T})(\cancel{\hat U} \hat\Sigma \hat V^T)\tilde x = \hat V \hat\Sigma^T \hat U^T b$$

$$\cancel{\hat V^T \hat V} \hat\Sigma^T \hat\Sigma V^T \tilde x = \cancel{\hat V^T \hat V} \hat\Sigma^T \hat U^T b$$

$$(\hat\Sigma^T \hat\Sigma) V^T \tilde x = \hat\Sigma^T \hat U^T b$$

$$\cancel{V V^T} \tilde x = V(\hat\Sigma^T \hat\Sigma)^{-1} \hat\Sigma^T \hat U^T b$$

$$\boxed{\tilde x = V\underbrace{(\hat\Sigma^T \hat\Sigma)^{-1} \hat\Sigma^T U^T}_{A^+} b}$$

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

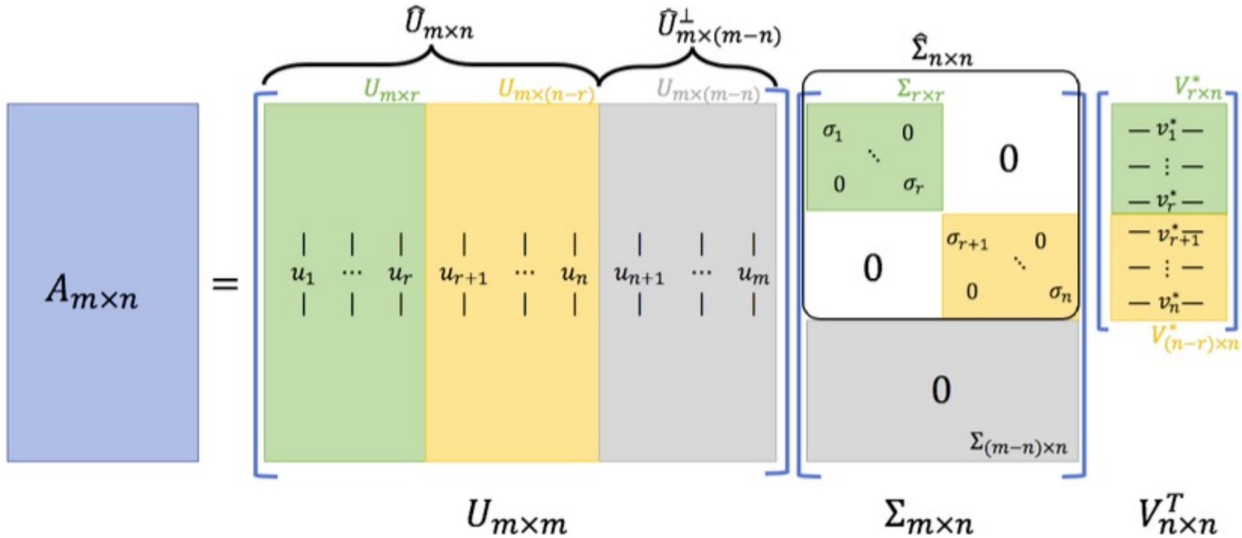
$$\hat\Sigma^T \hat\Sigma = \sigma_1^2 \ \ldots \ \sigma_r^2 \ \ldots \ \sigma_n$$

when exist

$>0 \quad =0$

$$A_{m\times n} = \hat U_{m\times n} \hat\Sigma_{n\times n} \hat V^T_{n\times n}$$

$$= \begin{bmatrix} | & | & & | \\ u_1 & u_2 & \ldots & u_n \\ | & | & & | \end{bmatrix} \underbrace{\begin{bmatrix} \sigma_1 & 0 & \ldots & 0 \\ 0 & \sigma_2 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & \sigma_n \end{bmatrix}}_{\hat\Sigma} \begin{bmatrix} - & v_1^T & - \\ - & v_2^T & - \\ & \vdots & \\ - & v_n^T & - \end{bmatrix}$$



$$\hat{\Sigma}^+ = (\hat{\Sigma}^T\hat{\Sigma})^{-1}\hat{\Sigma}$$

if $\hat{\Sigma}$ is invertible

- if $r = n$

  if $\sigma_i > 0 \quad 1 < i < n$

- But if $r < n$

  $\sigma_r > 0, \ \sigma_{r+1} = 0$

$$(\hat{\Sigma}^T\hat{\Sigma}) = \begin{pmatrix} 1/\sigma_1^2 & & 0 \\ & \ddots \ 1/\sigma_r^2 & \\ 0 & & \text{''} = \frac{1}{0} \ \text{''} = 1/0 \end{pmatrix}$$

$$\boxed{\text{''}\ \frac{1}{0}\ \text{''} := 0}$$

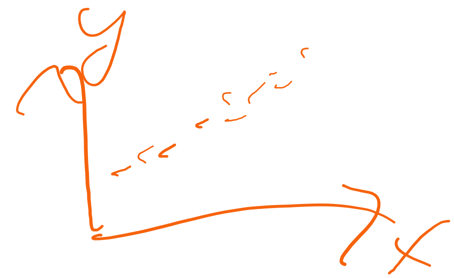
$$A_{m\times n} = \hat{U}_{m\times n}\hat{\Sigma}_{n\times n}\hat{V}^T_{n\times n}$$

$$= \begin{bmatrix} | & | & & | \\ u_1 & u_2 & \dots & u_n \\ | & | & & | \end{bmatrix} \begin{bmatrix} \sigma_1 & 0 & \dots & 0 \\ 0 & \sigma_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \sigma_n \end{bmatrix} \begin{bmatrix} - & v_1^T & - \\ - & v_2^T & - \\ - & \vdots & - \\ - & v_n^T & - \end{bmatrix}$$

Best Fitting Curves

Often, a line is not the best model for the data. Fortunately the same technique works if we want to use other nonlinear curves to fit the data. Here we will explain how to find the least squares cubic. The process for other polynomials is similar.

Matrix Formulation of LS slick for general Models



Example

A bioengineer is studying the growth of a genetically engineered bacteria culture and suspects that is it approximately follows a cubic model. He collects six data points listed below

Time in Days	1	2	3	4	5	6
Grams	2.1	3.5	4.2	3.1	4.4	6.8

He assumes the equation has the form

$$ax^3 + bx^2 + cx + d = y$$

Grams = f(time)

y = f(x)

This gives six equations with four unknowns

$$\begin{aligned} a + b + c + d &= 2.1 \\ 8a + 4b + 2c + d &= 3.5 \\ 27a + 9b + 3c + d &= 4.2 \\ 64a + 16b + 4c + d &= 3.1 \\ 125a + 25b + 5c + d &= 4.4 \\ 216a + 36b + 6c + d &= 6.8 \end{aligned}$$

$2.1 = a \cdot 1^3 + b \cdot 1^2 + c \cdot 1 + d$ $\quad i = 1$

$3.5 = a \cdot 2^3 + b \cdot 2^2 + c \cdot 2 + d$ $\quad i = 2$

The corresponding matrix equation is

$$\begin{pmatrix} 1 & 1 & 1 & 1 \\ 8 & 4 & 2 & 1 \\ 27 & 9 & 3 & 1 \\ 64 & 16 & 4 & 1 \\ 125 & 25 & 5 & 1 \\ 216 & 36 & 6 & 1 \end{pmatrix} \begin{pmatrix} a \\ b \\ c \\ d \end{pmatrix} = \begin{pmatrix} 2.1 \\ 3.5 \\ 4.2 \\ 3.1 \\ 4.4 \\ 6.8 \end{pmatrix}$$

$Ax = b$

$X\beta = y$

$$\underbrace{\begin{pmatrix} x_1^3 & x_1^2 & x_1 & 1 \\ x_2^3 & x_2^2 & x_2 & 1 \\ \vdots & \vdots & \vdots & \vdots \\ x_n^3 & x_n^2 & x_n & 1 \end{pmatrix}}_{X = A} \underbrace{\begin{pmatrix} a \\ b \\ c \\ d \end{pmatrix}}_{\beta = x} = \underbrace{\begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}}_{y = b}$$

We can use the least squares equation to find the best solution

$$\begin{pmatrix} a \\ b \\ c \\ d \end{pmatrix} = (A^T A)^{-1} A^T \mathbf{b} = \begin{pmatrix} 0.2 \\ -2.0 \\ 6.1 \\ -2.3 \end{pmatrix}$$

So that the best fitting cubic is

$$y = 0.2x^3 - 2.0x^2 + 6.1x - 2.3$$

The graph is shown below