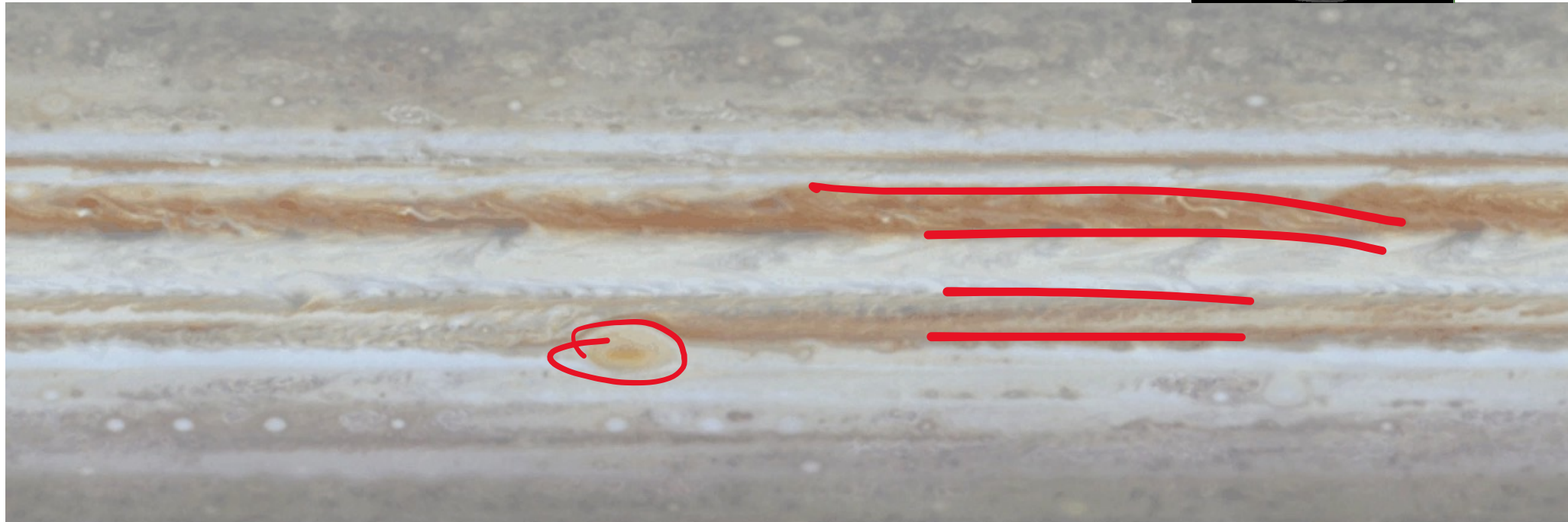
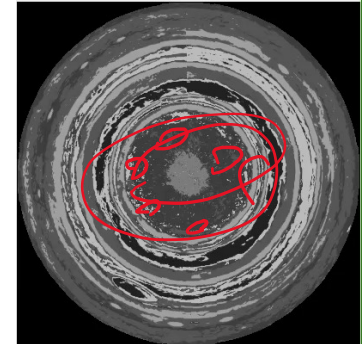
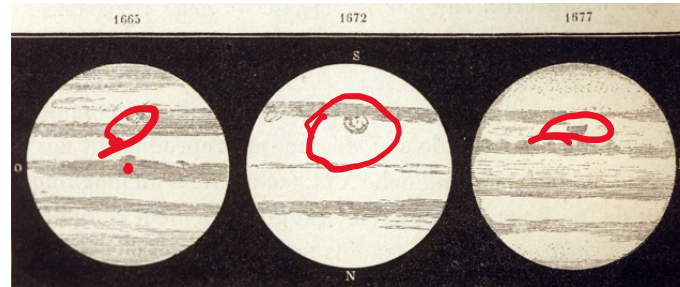


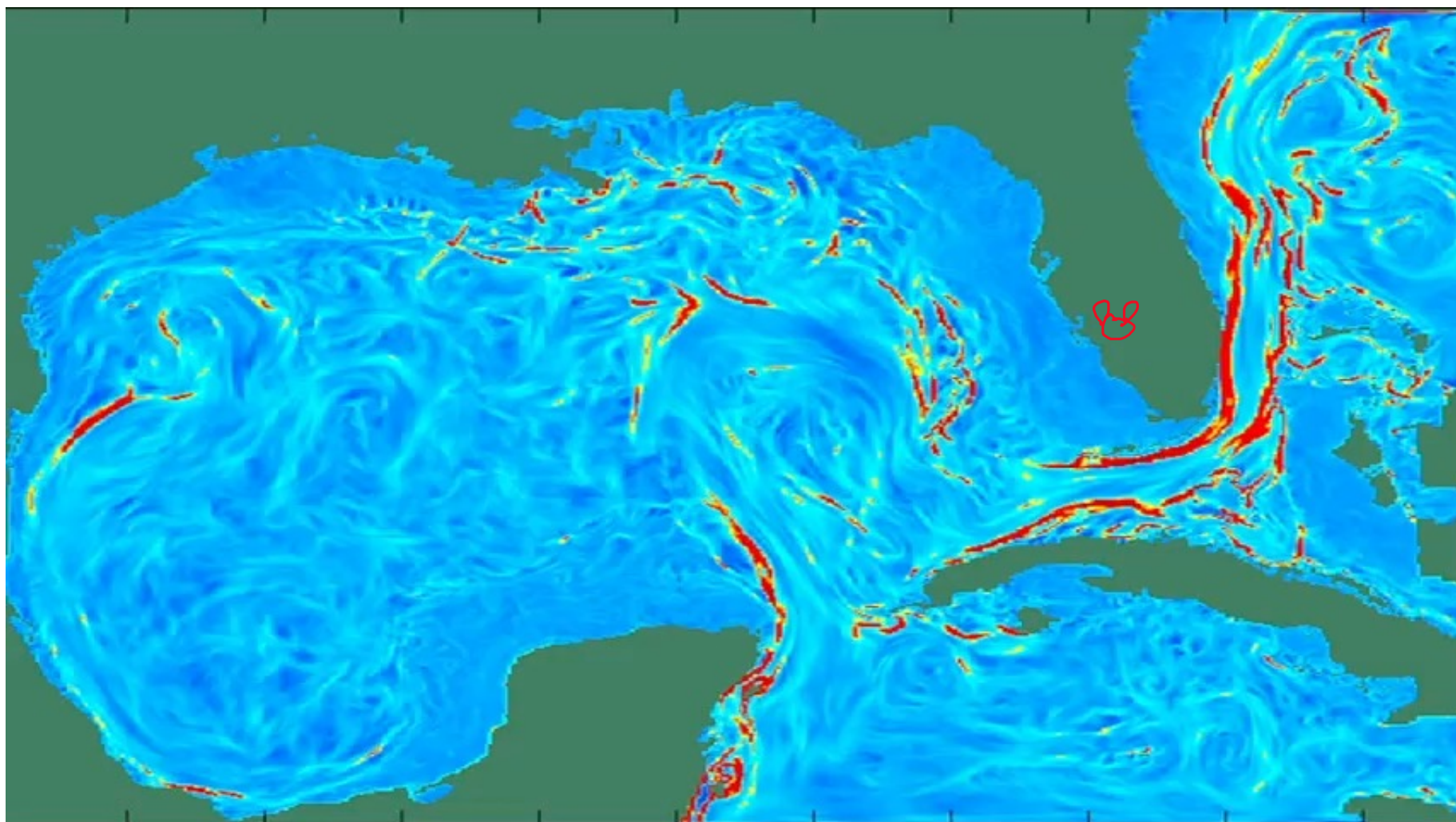
EE520 Data Driven Analysis of Complex Systems

Erik Bollt

~

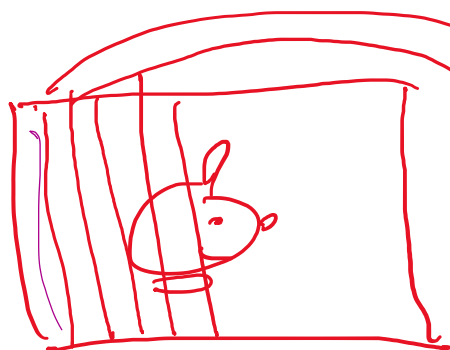
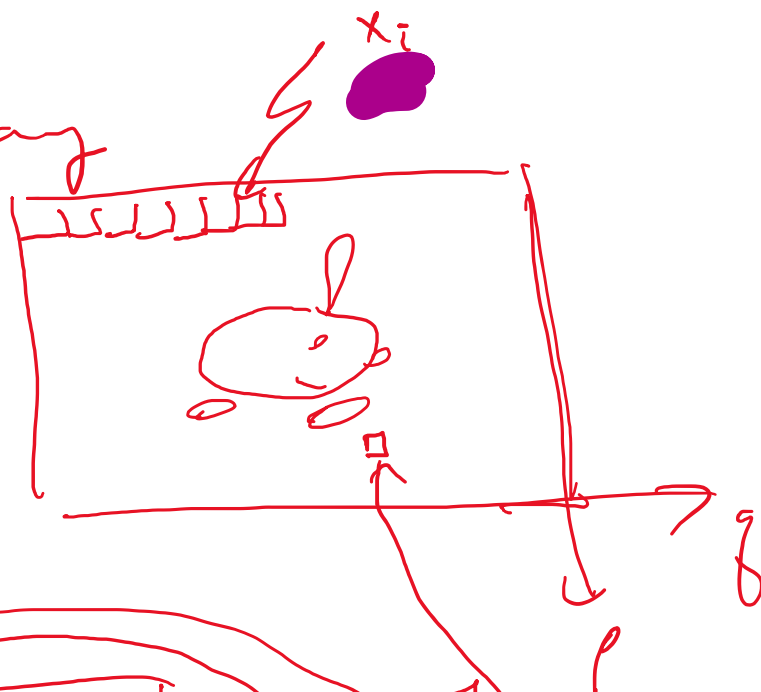
~



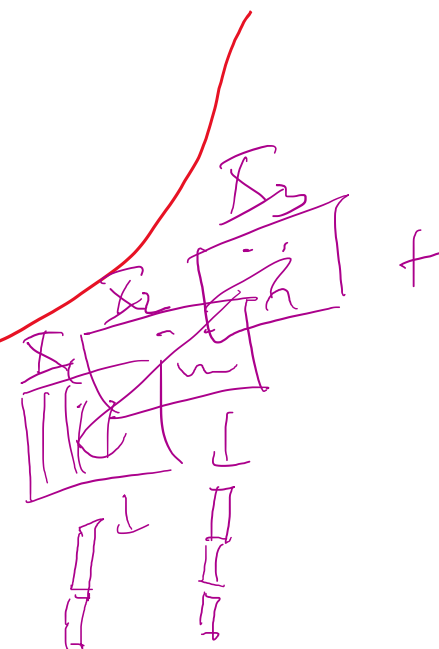


Data as
an array

$X =$



slice & stack



$X_{p \times q} \in \mathbb{R}^{p \times q}$
matrix

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

Data as an array

On Matrix Multiplication

$$L(\mathbf{z}) : \mathbb{R}^n \rightarrow \mathbb{R}^m$$

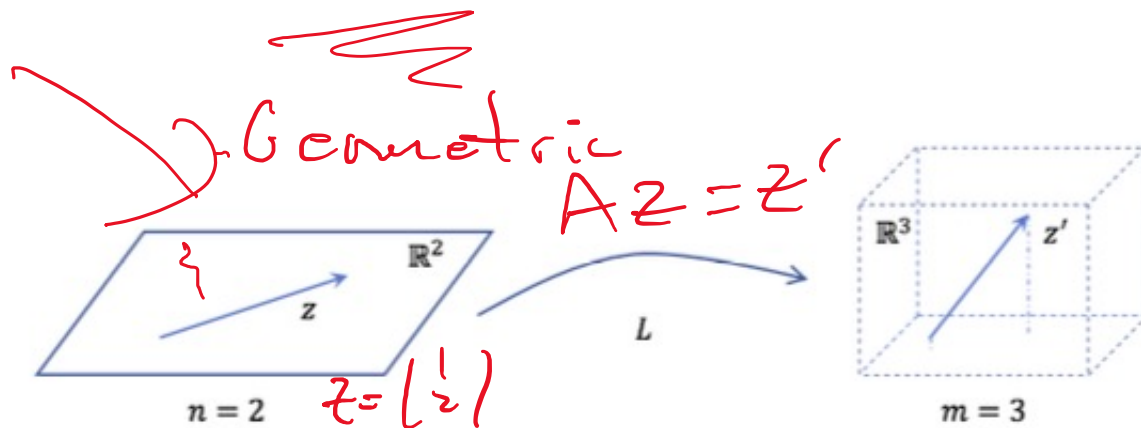
$$\mathbf{z} \mapsto \mathbf{z}' = A\mathbf{z},$$

in terms of the usual matrix times vector multiplication,

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

a vector has length & direction.

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

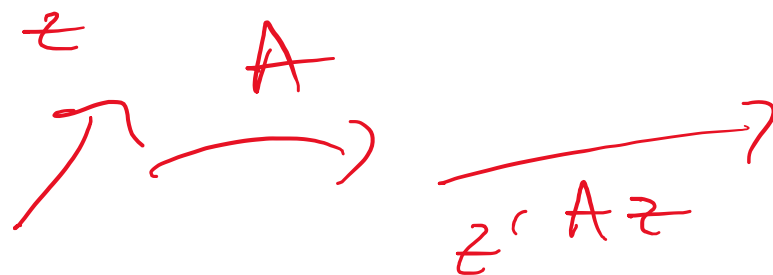


But as linear algebra

$$A_{2 \times 2} = \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}; \text{ matrices } \times \text{ vectors}$$

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

$$z' = Az$$

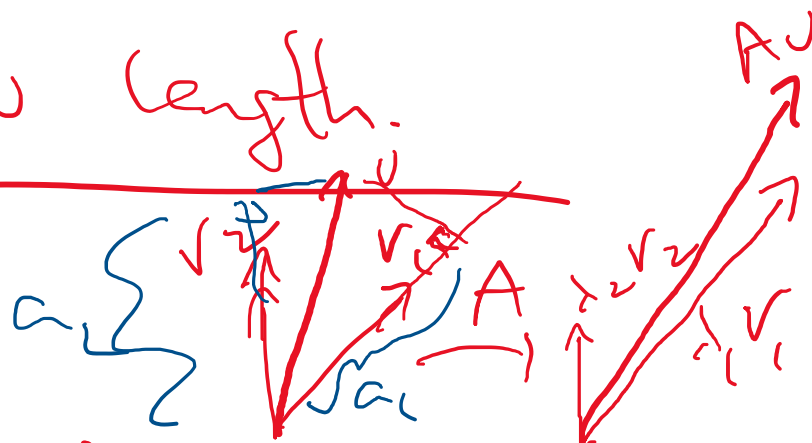


new direction, new length.

Eig for square -

$$AV = \lambda V$$

2×2

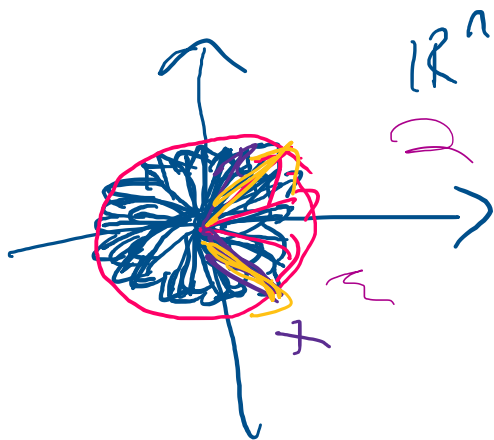


Characterize matrices by knowing just these
 • Eig. special directions

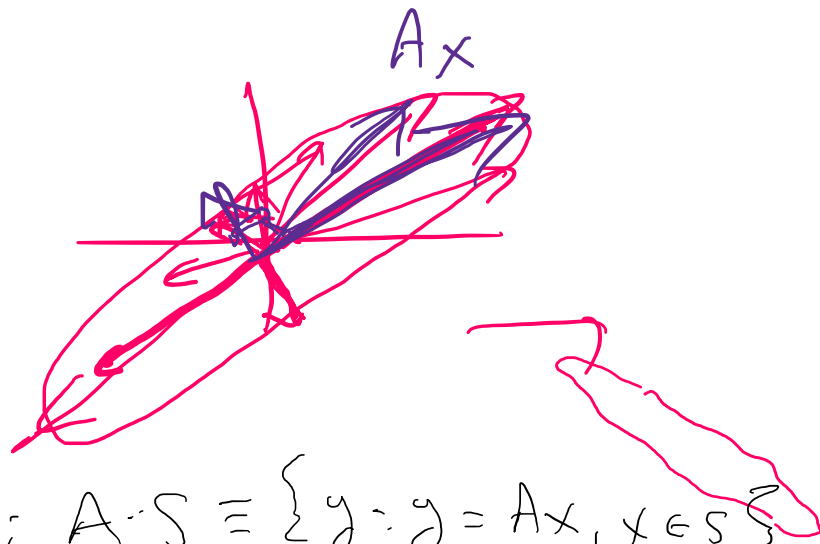
$$\begin{aligned}
 \bullet \quad Av &= A(a_1v_1 + a_2v_2) = a_1Av_1 + a_2Av_2 \\
 &= a_1\lambda_1v_1 + a_2\lambda_2v_2
 \end{aligned}
 \quad
 \begin{aligned}
 \det(A - \lambda I) &= 0 \\
 (A - \lambda I)v &= 0
 \end{aligned}$$

Matrix time circle =
all vectors of length 1.

? Matrix $\times$ circle ? But matrix
 times vector. $\nearrow =$



$$\begin{matrix} A \\ \hline Ax \end{matrix}$$



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

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:

$$A_{m \times n} = U_{m \times m} \Sigma_{m \times n} V_{n \times n}^* \quad (2.5)$$

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 nonnegative values: $\sigma_i \geq 0, i = 1, \dots, n$,

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

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

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 nonnegative values: $\sigma_i \geq 0, i = 1, \dots, n$,

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

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

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

$$AV = U\Sigma V^*V = U\Sigma, \quad (2.8)$$

$$\Sigma := \text{diag}(\sigma_1, \sigma_2, \dots, \sigma_p), p = \min(m, n),$$

$$\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_p \geq 0.$$

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

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

$$u_i^T u_i = u_i \cdot u_i$$

$$u_i^T u_j = \begin{pmatrix} u_1^T & u_2^T & \dots \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \\ \vdots \end{pmatrix} = \delta_{ij} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$U^T U = I$$

$$U \text{ unitary} \Leftrightarrow U^* U = U U^* = I$$

$$U^T U = \begin{pmatrix} u_1^T & u_2^T & \dots & u_m^T \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_m \end{pmatrix} = I = \begin{pmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1 \end{pmatrix}$$

Fri 08/21/20

$$A x = b$$

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

if A is square

$$A \underline{V} = \underline{U} \underline{\Sigma}$$

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

■ **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:

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

$$= \begin{pmatrix} \frac{1}{\sqrt{5}} & \frac{2}{\sqrt{5}} \\ \frac{-2}{\sqrt{5}} & \frac{1}{\sqrt{5}} \end{pmatrix} \begin{pmatrix} \sqrt{70} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} \frac{1}{\sqrt{14}} & \sqrt{\frac{2}{7}} & \frac{3}{\sqrt{14}} \\ \frac{-3}{\sqrt{10}} & 0 & \frac{1}{\sqrt{10}} \\ \frac{-1}{\sqrt{35}} & \sqrt{\frac{5}{7}} & \frac{-3}{\sqrt{35}} \end{pmatrix}. \quad (2.28)$$

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:

$$u_1 \sigma_1 v_1^T = \begin{pmatrix} \frac{1}{\sqrt{5}} \\ \frac{-2}{\sqrt{5}} \end{pmatrix} (\sqrt{70}) \begin{pmatrix} \frac{1}{\sqrt{14}} & \sqrt{\frac{2}{7}} & \frac{3}{\sqrt{14}} \end{pmatrix}$$

$$\approx \begin{pmatrix} 1 & 2 & 3 \\ 2 & 4 & 6 \end{pmatrix}$$

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

$$\begin{aligned} \textcircled{A^T A} &= \underline{V} \underline{\Sigma}^T \underline{\Sigma} \underline{V}^T \quad (2.29) \\ \underline{A^T A} \underline{V} &= \underline{\Sigma}^T \underline{\Sigma} \underline{V} \\ \underline{V} &= \begin{pmatrix} v_1 & v_2 & \cdots & v_n \end{pmatrix} \end{aligned}$$

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,

$$A_{m \times n} = \hat{U}_{m \times n} \hat{\Sigma}_{n \times n} V_{n \times n}^* \quad (2.21)$$

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

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

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

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

$$A_{m \times n} = \hat{U}_{m \times r} \hat{\Sigma}_{n \times n} V_{n \times r}^* \quad (2.23)$$

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

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

Rank ill conditioned

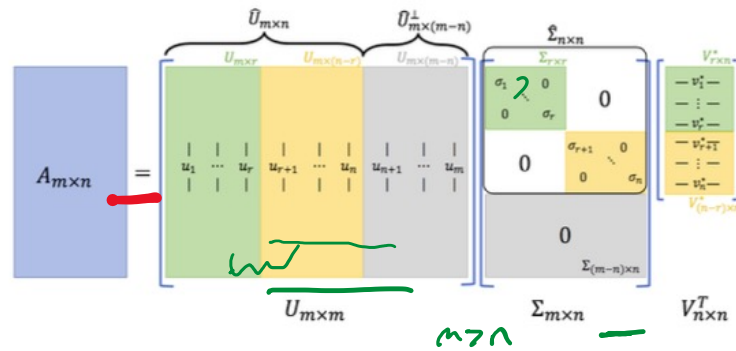


Figure 2.3: $m > n$ tall skinny

Recall that,

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

$$= \begin{bmatrix} | & | & | & | \\ u_1 & u_2 & \dots & u_n \\ | & | & | & | \end{bmatrix} \begin{bmatrix} \sigma_1 & 0 & \dots & 0 \\ 0 & \sigma_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \sigma_n \end{bmatrix} \begin{bmatrix} - & v_1^T & - \\ - & v_2^T & - \\ - & \vdots & - \\ - & v_n^T & - \end{bmatrix} \quad (2.24)$$

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

$$A_{m \times n} \hat{V}_{n \times n} = \hat{U}_{m \times n} \hat{\Sigma}_{n \times n} \quad (2.25)$$

so,

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

but this just states n -matrix times vector statements:

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

Full,
Economy,
Truncated
SVD

Handwritten notes:

$$b_{r+1} = 0$$

$$b_r = 10^{-6}$$

$$b_{r+1} \approx 10^{-13} < \epsilon$$

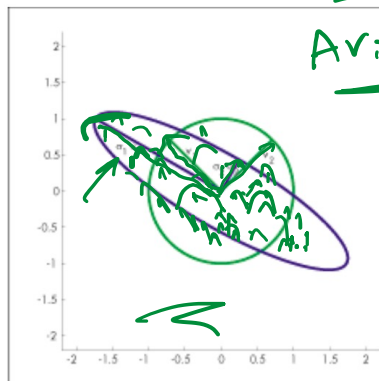
$$A = \underline{U} \underline{\Sigma} \underline{V}^T$$

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

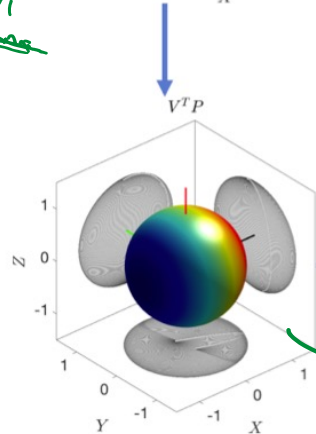
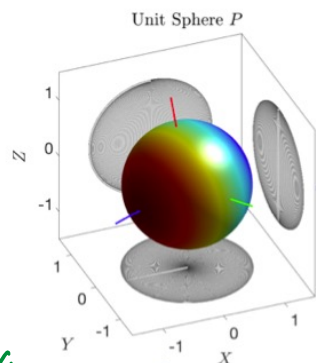
$$A \underline{V}_i = \underline{\sigma}_i \underline{U}_i$$

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 .

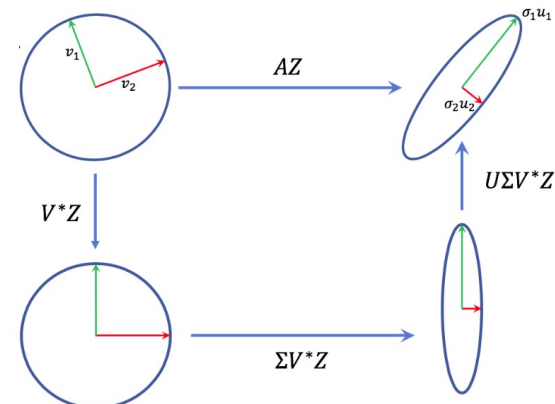
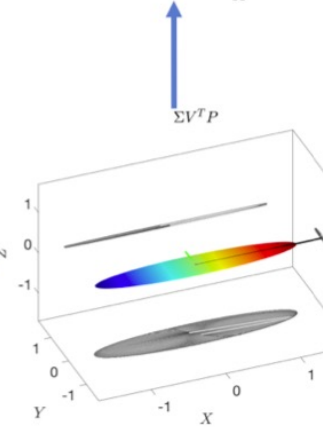
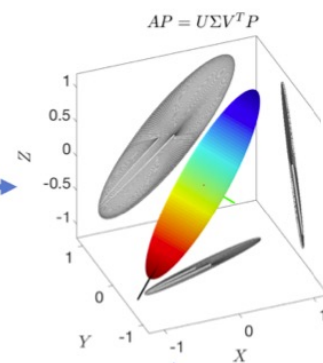


$\sigma_1 \sim 1.5$
 $\sigma_2 \sim 0.5$
 $\sigma_3 = 2$
 $\sigma_4 = 0.0003$



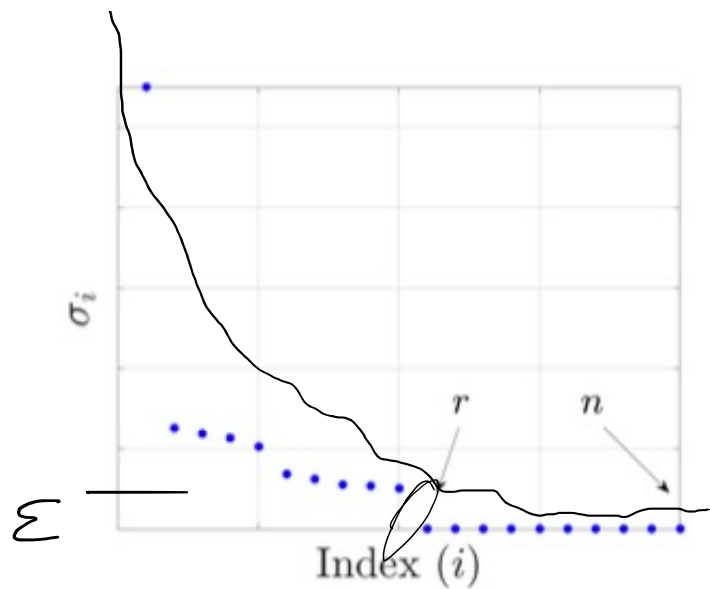
$\text{disp}(A)$
 $\begin{bmatrix} 0.8929 & 0.0417 & 0.1000 \\ 0.3320 & 0.1077 & 0.0000 \\ 0.8212 & 0.5951 & 0.0000 \end{bmatrix}$
 $[U, S, V] = \text{svd}(A)$

$U = 3 \times 3$
 $\begin{bmatrix} -0.6330 & 0.7469 & 0.2035 \\ -0.2590 & 0.0435 & -0.9649 \\ -0.7296 & -0.6635 & 0.1659 \end{bmatrix}$
 $S = 3 \times 3$
 $\begin{bmatrix} 1.3438 & 0 & 0 \\ 0 & 0.3912 & 0 \\ 0 & 0 & 0.0208 \end{bmatrix}$
 $V = 3 \times 3$
 $\begin{bmatrix} -0.9304 & 0.3488 & -0.1126 \\ -0.3635 & -0.9176 & 0.1612 \\ -0.0471 & 0.1909 & 0.9805 \end{bmatrix}$

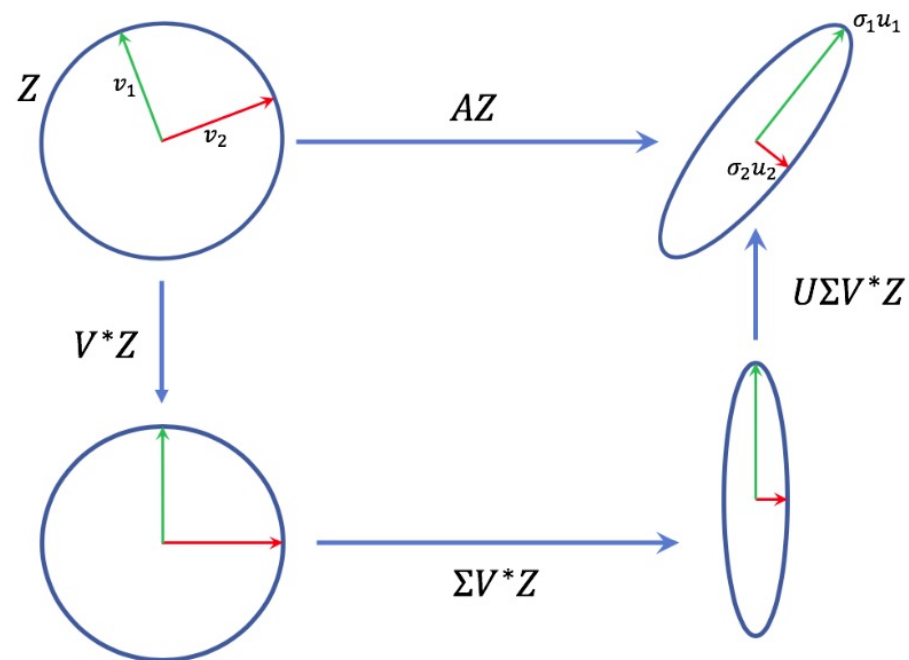


$A \underline{w} = \underline{U} \underline{\Sigma} \underline{V}^T \underline{w}$
 $\underline{v}_i^T \underline{w} = \underline{r}_i \cdot \underline{w}$
 $\underline{X} = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 4 & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots \end{bmatrix}$
 $\underline{w} = \begin{bmatrix} 1 \\ 4 \\ \dots & \dots \end{bmatrix}$
 $\underline{w}$ is redundant if $\sigma_i \approx 0$

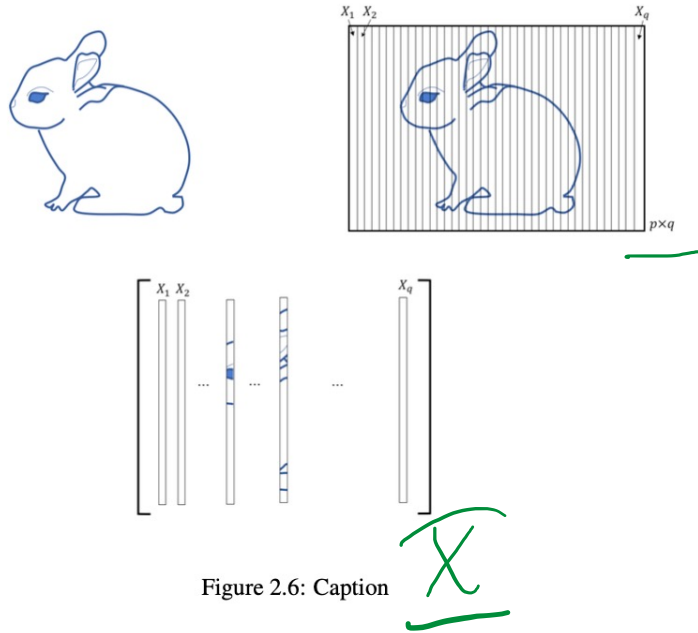
And ROM



$$G \subset \mathcal{E} \supset G_{\text{cut}} \sim 0$$



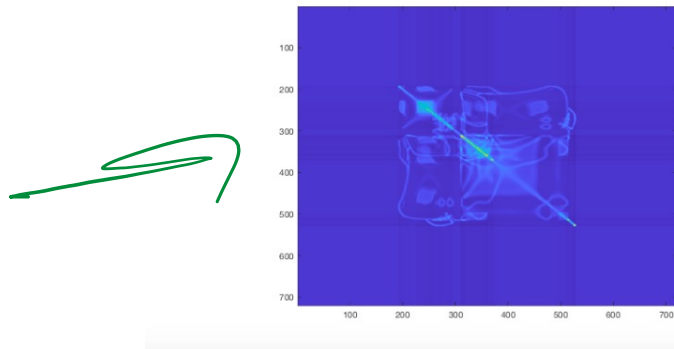
Bunny Compression



Covariance – notice the demean step

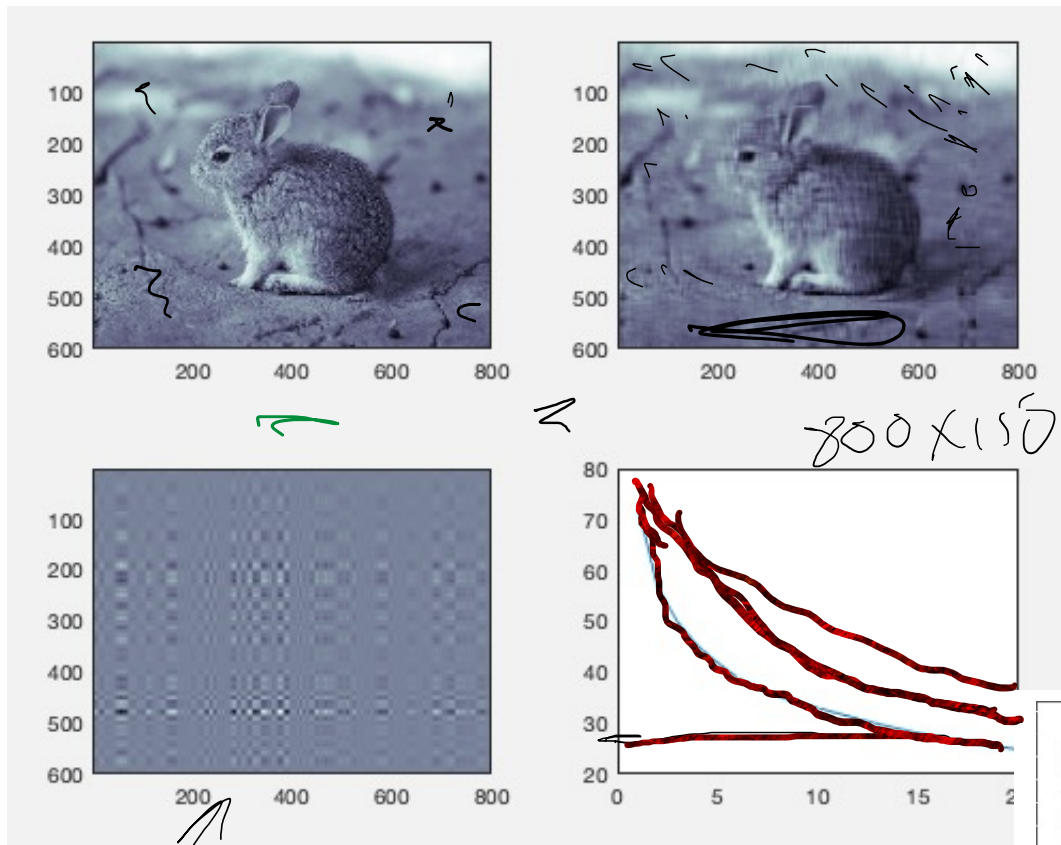
$$C_I = \frac{1}{n-1} (X - \tilde{X})^T (X - \tilde{X})$$

A A



$$\underline{X^T X}$$

$$\underline{X - \tilde{X}}$$



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

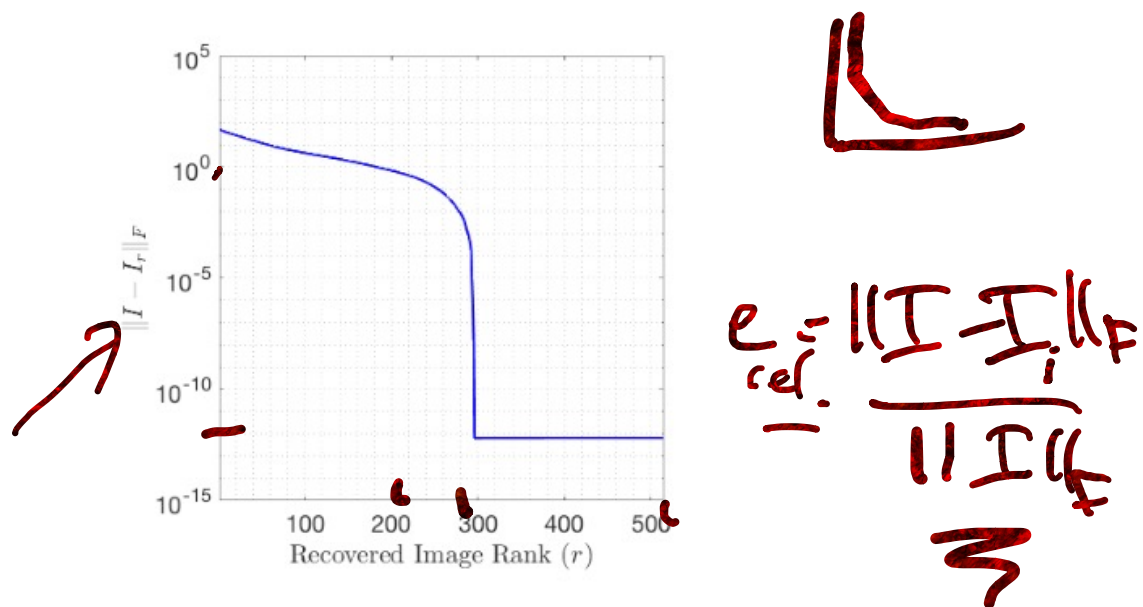
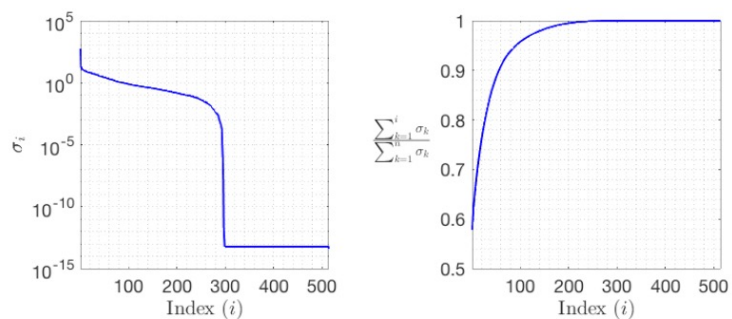
$\uparrow$
 $i \rightarrow \infty$
 $|a_i(t)| \rightarrow 0$
 as fast as possible.

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

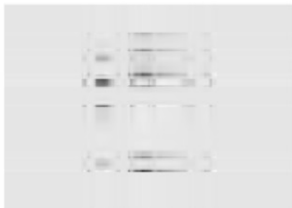
$\approx$
 $\sum_{i=1}^n \sigma_i u_i v_i^T$

```

1 I = imread('Bunny.jpg');
2
3 figure
4 subplot(1,2,1)
5 imshow(I)
6 xticks({}); yticks({});
7 pbaspect([1 1 1])
8 title('RGB Image')
9
10 I = rgb2gray(I); %Convert the 3D RGB color to 1D grayscale
11 I = im2double(I); %Convert integer value to double (scaled ...
    from 0 to 1)
12
13 subplot(1,2,2)
14 imshow(I)
15 xticks({}); yticks({});
16 pbaspect([1 1 1])
17 title('Grayscale Image')
  
```



Rank $r = 2$



Rank $r = 10$



Rank $r = 20$



Rank $r = 100$



Code 2.1: Read, convert, and display images.

```
1 I = imread('Bunny.jpg');
2
3 figure
4 subplot(1,2,1)
5 imshow(I)
6 xticks({}); yticks({});
7 paspect([1 1 1])
8 title('RGB Image')
9
10 I = rgb2gray(I); %Convert the 3D RGB color to 1D grayscale
11 I = im2double(I); %Convert integer value to double (scaled ...
    from 0 to 1)
12
13 subplot(1,2,2)
14 imshow(I)
15 xticks({}); yticks({});
16 paspect([1 1 1])
17 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.

Lots of Machine learning

& Data Analysis

is solving an ill-posed

- optimize a cost function.

$$AA^T = U \Sigma V^T (V \Sigma^T U^T)$$

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

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

$$\underline{U} = \begin{pmatrix} \underset{|}{u_1} & \underset{|}{u_2} & \dots & \underset{|}{u_n} \end{pmatrix}$$

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:

$$\|A\|_p = \sup_{x \neq 0} \frac{\|Ax\|_p}{\|x\|_p} \quad (2.14)$$

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

$$\|A\|_p = \sup_{\|x\|_p=1} \|Ax\|_p \quad (2.15)$$

Some Special (Simple) Matrix Norms

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

- For $p = 1$:

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

- For $p = \infty$:

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

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

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

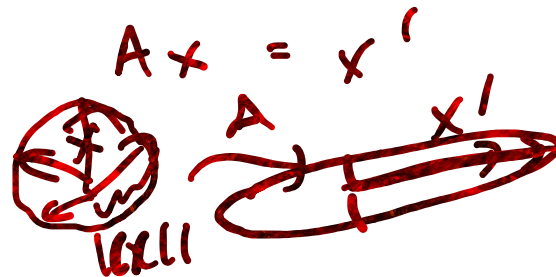
where $\sigma_{\max}$ is the maximum singular value of the matrix A .

- The Frobenius norm is given by:

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

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

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



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

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

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

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

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

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

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:

$$\|A\|_p = \sup_{x \neq 0} \frac{\|Ax\|_p}{\|x\|_p} \quad (2.14)$$

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

$$\|A\|_p = \sup_{\|x\|_p=1} \|Ax\|_p \quad (2.15)$$

Some Special (Simple) Matrix Norms

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

- For $p = 1$:

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

- For $p = \infty$:

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

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

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

where $\sigma_{\max}$ is the maximum singular value of the matrix A .

- The Frobenius norm is given by:

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

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

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

Fun facts about matrix estimation (data estimation)

$$\text{If } A, \quad b_1 \geq \dots \geq b_r > b_{r+1} = 0$$

$$\bullet \text{ range}(A) = \text{span}(u_1, u_2, \dots, u_r)$$

$$\bullet \text{ null}(A) = \text{span}(v_{r+1}, v_{r+2}, \dots, v_n)$$

$$\left. \begin{array}{l} \bullet \text{ range}(A) = \text{span}(u_1, u_2, \dots, u_r) \\ \bullet \text{ null}(A) = \text{span}(v_{r+1}, v_{r+2}, \dots, v_n) \end{array} \right\} \begin{array}{l} \text{A} \\ 0 \rightarrow \end{array}$$

$$b_2 = 0$$

$$\bullet \|A\|_2 = b_1 \quad ; \quad \|A\|_F = \sqrt{b_1^2 + b_2^2 + \dots + b_r^2}$$

$$\bullet A = \sum_{i=1}^r b_i u_i v_i^* = \underbrace{b_1 u_1 v_1^*}_{\text{rank-1}} + \underbrace{b_2 u_2 v_2^*}_{\text{outer products}} + \dots + b_r u_r v_r^*$$

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

$$\begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_m \end{pmatrix} \begin{pmatrix} v_1^T & v_2^T & \dots & v_n^T \end{pmatrix} \begin{pmatrix} b_1 & b_2 & \dots & b_r \end{pmatrix} \begin{pmatrix} -v_1^T \\ -v_2^T \\ \vdots \end{pmatrix}^{1 \times n}$$

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

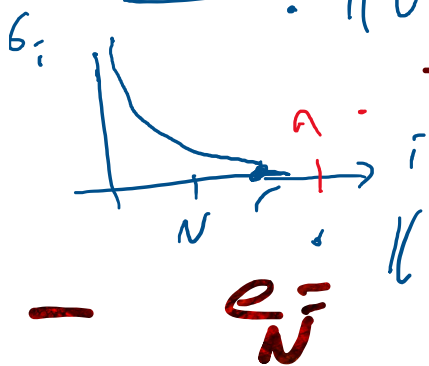
$$w_1^T w_2 = w_1 \cdot w_2 = \|w_1\| \|w_2\| \cos \theta$$

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

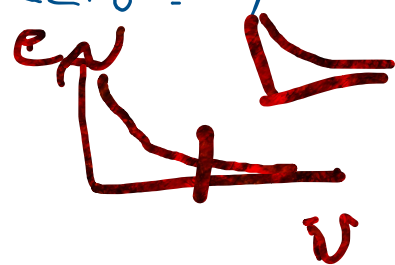
Let $0 \leq N \leq r$ and $A_N = \sum_{i=1}^N \sigma_i u_i v_i^*$
 (so we may be skipping some of them ... $\sum_{i=N+1}^r$)

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

(what if it zero?)



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

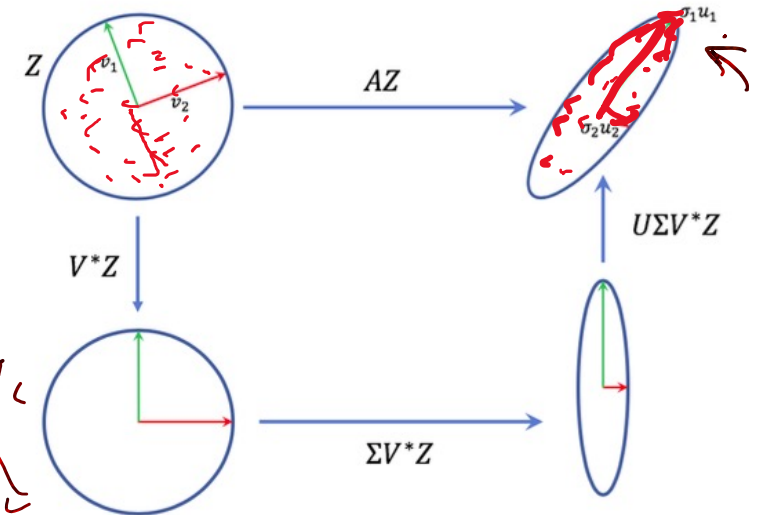


On PCA Principal Component Analysis, Eigenface

-On Raleigh Ritz Quotient

-On Spectral Decomposition Theorem

-On Data Clouds



$$\overline{X} = [x_1 | x_2 | \dots | x_n] \quad \vec{x}_i = a_1^i \vec{v}_1 + a_2^i \vec{v}_2 + \dots + a_r^i \vec{v}_r$$

$\vec{v}_i$ as basis set.

$x_i = \text{PCA gives basis set}$
 where $\langle a_j \rangle_i \downarrow$ as fast as possible vs any other basis set.

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

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

Ex. $\underline{X}_i \sim 4000 \times 1$ gene expression table for each i .
 $i = 1 \dots 216$ patients

$$X_i = \begin{bmatrix} x_i^1 \\ x_i^2 \\ \vdots \\ x_i^{4000} \end{bmatrix}$$

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



$$\mathbb{Z}_2 = \{0, 1\}$$

• Supervised vs. unsupervised.

just x $\xrightarrow{\text{unsupervised}}$ - just input - just structural geometry.
 "Data Cloud" $\sim$ distribution of R.V. $x \sim \underline{X}$
 $\xrightarrow{\text{level sets of } p}$ $p(x): \mathbb{R}^{4000} \rightarrow \mathbb{R}^+$
 $\xrightarrow{\text{supervised learning is descriptive}}$ $f: x \rightarrow y$ $\leftarrow$ 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 $u_1, \dots, u_n$ 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,

$$A u_j = \lambda_j u_j \quad \leftarrow \text{real}$$

Then

$$A = P D P^{-1} = P D P^T$$

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

$$\begin{aligned} A &= [u_1 \ \dots \ u_n] \begin{bmatrix} \lambda_1 & & \\ & \dots & \\ & & \lambda_n \end{bmatrix} \begin{bmatrix} u_1^T \\ \vdots \\ u_n^T \end{bmatrix} \\ &= [\lambda_1 u_1 \ \dots \ \lambda_n u_n] \begin{bmatrix} u_1^T \\ \vdots \\ u_n^T \end{bmatrix} \\ &= \lambda_1 (u_1 u_1^T) + \dots + \lambda_n (u_n u_n^T) \end{aligned}$$

outer

$n \times n \cdot (n \times n)$
 $n \times n$

$$\|u_i\|^2 = u_i \cdot u_i = u_i^T u_i \quad \text{scalar = inner product}$$

The expression

$$A = \lambda_1 u_1 u_1^T + \dots + \lambda_n u_n u_n^T$$

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 .

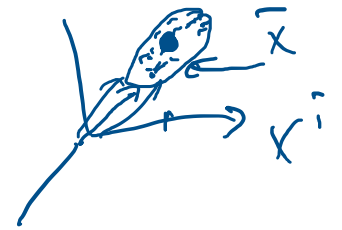
$A = B^T B$ is symmetric
 $\Rightarrow$ spectral decomp. theorem
i.e. also covariance matrices.

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

PCA as algorithm

• Data $\underline{X} = \begin{pmatrix} x_1 & x_2 & \dots & x_n \\ \vdots & \vdots & & \vdots \end{pmatrix}^{ith}$
 $n \times 1$

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



• what if $x_i \sim \mathcal{N}(\bar{x}, \Sigma)$ —

covariance matrix.

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

$$\bar{x}_i = \frac{1}{n} \sum_{i=1}^n x_i$$



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

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 i^2

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

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

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

$p < 1$, $p = 1$ harmonic



b_i^2 the power spectrum

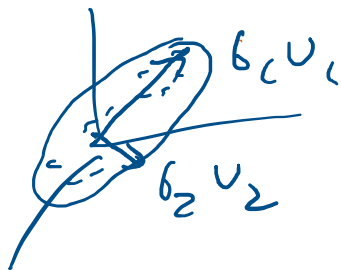
$$= 1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \dots \quad p = 1$$

$$B = \frac{1}{n} \bar{X}^T$$

let $C = \frac{1}{n-1} B^T B$ covariance matrix
is covariance matrix

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

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



$$\begin{aligned} b_3 &\approx 0 \\ b_2 &\geq 0 \end{aligned}$$

$$u_1 = \underset{\|u\|=1}{\operatorname{argmax}} \frac{u^T B^T B u}{\|B u\|_2^2} = \underset{u}{\operatorname{argmax}} \frac{u^T B^T B u}{\underbrace{u^T u}_{=1}}$$

Rayleigh-Ritz gradient

$$B u \cdot B u = \|B u\|_2^2$$

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

$$\begin{aligned}
\|x_i - \text{proj}_w x_i\|_2^2 &= \|x_i(w \cdot x_i)w\|_2^2 = (x_i - (w^T x_i)w)^T (x_i - (w^T x_i)w) \\
&= (x^T x_i) - (w^T x_i)^2 = \|x_i\|_2^2 - (w^T x_i)^2.
\end{aligned} \tag{2.43}$$

To minimize this residual with respect to the unknown vector w , averaged across the data set, it is sufficient to maximize the second term since the first term does not depend on w . Thus we wish to maximize,

$$\mathcal{L}_1(\mathcal{D}; \Theta) = \frac{1}{n} \sum_{i=1}^N (w^T x_i)^2, \tag{2.44}$$

The Eigs of $C=B'B$ give optimal projection – thus PCA and.... KL

$CV = VD$ eigenvectors of C all stacked.

Ax optimize $x^T Ax$, maximize.

$x = \begin{pmatrix} x^1 \\ \vdots \\ x^j \\ \vdots \\ x^n \end{pmatrix}$

$r(x) = \frac{x^T Ax}{x^T x}$

$\left(A = B^T B \right)$
 $x \neq 0$

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

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

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

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

Conclude

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

• S.D.T. for $A = B^T B = \sum_{i=1}^r \lambda_i \underbrace{U_i U_i^T}_{\substack{\text{cre} \\ \text{the} \\ v_i^T}}$

let $B = \underbrace{U \Sigma V^T}_{\text{svd}}$

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

$$D = \left(\begin{array}{ccc|c} b_1 & \dots & b_r & \\ \hline & & & 0 \end{array} \right) \quad \left(\begin{array}{ccc|c} b_1 & \dots & b_r & 0 \\ \hline & & & 0 \end{array} \right) \quad \left(\begin{array}{ccc|c} b_1^2 & & & \\ b_2^2 & \dots & & \\ \hline & & & 0 \end{array} \right)$$

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

$$\mathcal{L}_1(\mathcal{D}; \Theta) = \frac{1}{n}(X^T w)^T (X^T w) = \frac{1}{n}w^T (X X^T) w, \quad (2.45)$$

and the matrix $\frac{1}{n}(X X^T)$ is familiar in statistics as a covariance matrix. To optimize $\mathcal{L}_1$, subject to a constraint,¹²

$$\|w\|_2 = 1, \quad (2.46)$$

we can use the Lagrange multiplier method by defining an expanded loss function(cost function) with the equality constraint built in with a Lagrange multiplier. Let,

$$\mathcal{L}(\mathcal{D}; \Theta, \lambda) = \mathcal{L}_1(\mathcal{D}; \Theta) - \lambda(w^T w - 1) = \frac{1}{n}w^T (X X^T) w - \lambda(w^T w - 1). \quad (2.47)$$

To minimize this, we take derivatives and set them equal to zero.

$$\frac{\partial \mathcal{L}}{\partial w} = \frac{2}{n} X X^T w - 2\lambda w \implies \frac{1}{n} (X X^T) w = \lambda w \quad (2.48)$$

$$\frac{\partial \mathcal{L}}{\partial \lambda} = w^T w - 1 \implies \|w\|_2 = 1. \quad (2.49)$$

Theorem 2.2.1 — PCA foundations. Let A be a symmetric $d \times d$ matrix. Then its (real) eigenvalues $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_d$, associate with orthogonal eigenvectors $w_1, w_2, \dots, w_d$. Furthermore,

$$\begin{aligned} \lambda_1 &= \max_{\|w\|=1} w^T A w, \text{ with } w_1 = \arg \max_{\|w\|=1} w^T A w. & (2.50) \\ \lambda_2 &= \max_{\|w\|=1, w \perp w_1} w^T A w \text{ with } w_2 = \arg \max_{\|w\|=1, w \perp w_1} w^T A w. \\ &\vdots \\ \lambda_d &= \max_{\|w\|=1, w \perp w_1, w_2, \dots, w_{d-1}} w^T A w \text{ with } w_d = \arg \max_{\|w\|=1, w \perp w_1, w_2, \dots, w_{d-1}} w^T A w. \end{aligned}$$

Theorem 2.2.2 — Spectral Decomposition. If A is a symmetric positive semi-definite matrix, then there is an orthogonal set of eigenvectors, u_i , each with non-negative eigenvalues, $\lambda_i \geq 0$. Furthermore, the decomposition of A has the following representation by rank one matrices that describe the action of A as a weighted sum of simple projections onto the subspaces spanned by each u_i ,

$$A = \sum_{i=1}^N \lambda_i u_i u_i^T \quad (2.51)$$

Which functions are most efficient?

That is, we write a linear combination,

$$u(x, t) = \sum_k a_k(t) \varphi_k(x), \quad (2.57)$$

of functions $\varphi_k(x)$, where the time varying (component projection) values,

$$a_k(t) = \frac{(u(x, t), \varphi_k(x))}{\|\varphi_k(x)\|_2}, \quad (2.58)$$

or better yet, we can effectively skip the denominator by choosing the basis set of functions such that,

$$\|\varphi_k(x)\|_2^2 = (\varphi_k(x), \varphi_k(x)) = 1. \quad (2.59)$$

Given a spatiotemporal data sample as an array, (for example, typically from a solution derived from a computational solver for a PDE):

$$\mathbf{U} = \begin{pmatrix} u(\vec{x}, t_1) & u(\vec{x}, t_2) & \dots & u(\vec{x}, t_T) \end{pmatrix}. \quad (2.61)$$

then develop the demeaned array $U - \bar{U}$. **Find:**

$$\Phi = \begin{pmatrix} \varphi_1(x) & \varphi_2(x) & \dots & \varphi_k(x) \end{pmatrix}. \quad (2.62)$$

by the singular value decomposition. This basis yields the fastest decaying power spectrum, *in time average*, versus all other possible basis.

$$\begin{aligned}
a(t) &= \frac{(u, \varphi)}{\|\varphi\|^2} = \frac{\|u\| \|\varphi\| \cos \theta}{\|\varphi\|^2} = \|u\| \cos \theta \\
&= \int_{\Omega} u(x, t) \varphi(x) dx
\end{aligned} \tag{2.63}$$

when $\|\varphi\| = 1$.

So, we will also write time average using brackets $\langle \cdot, \cdot \rangle$, so define:

$$\begin{aligned}
\langle |a(t)| \rangle &= \frac{1}{T} \int_0^T |a(t)| dt \\
&= \frac{1}{T} \int_0^T |(u, \varphi)| dt \\
&= \frac{1}{T} \int_0^T \left| \int_{\Omega} u(x, t) \varphi(x) dx \right| dt
\end{aligned} \tag{2.64}$$

★★ The goal is to choose a basis with fastest decaying power spectrum, in time average ★★

Our goal can be summarized by the following loss function.

$$\mathcal{L}(\varphi) = \frac{\langle |(u, \varphi)|^2 \rangle}{\|\varphi\|^2}. \tag{2.65}$$

This very compact notation, encodes two integrations. We remind that the round brackets describe the inner product, (f, g) , meaning integration in the “space” variable. Now we have introduced the pointy brackets to describe time average. So,

$$\mathcal{L}(\varphi) = \frac{\frac{1}{T} \int_0^T \left| \int_{\Omega} u(x, t) \varphi(x) dx \right|^2 dt}{\int_0^L \varphi^2(x) dx} \equiv \langle a\varphi^2(t) \rangle \tag{2.66}$$

or

$$\max_{\|\varphi\|=1} \frac{1}{T} \int_0^T \left| \int_0^L u(x, t) \varphi(x) dx \right|^2 dt \tag{2.67}$$

Theorem 2.3.1 — Parseval's Like Identity. If $f \in L^2([0, L])$, then if $\{\varphi_k(x)\}$ is an orthonormal basis set, then:

$$\|f\|_2^2 \leq \sum_{k=0}^{\infty} |a_k|^2 \quad (2.68)$$

where $a_k = f, \varphi_k$.

$$\mathcal{L}(\varphi) = \langle |(u, \varphi)|^2 \rangle - \lambda (\|\varphi\|^2 - 1)$$

$$C\vec{\varphi}(x) = \lambda\vec{\varphi}(x), C = \mathbf{U}\mathbf{U}^T,$$

$$\frac{\partial}{\partial \delta} \mathcal{L}(\varphi + \delta\psi) = 0:$$

$$\max_{\begin{cases} \|\varphi_1\| = 1 \\ \varphi \in \mathcal{H} \end{cases}} \langle |(u, \varphi)| \rangle$$

and $\varphi_2(x)$ solves:

$$\max_{\begin{cases} \|\varphi_2\| = 1 \\ \varphi \in \mathcal{H} \\ \varphi_2 \perp \varphi_1 \end{cases}} \langle |(u, \varphi)| \rangle$$

Theorem 2.3.2 — Spectral Decomposition. If A is a symmetric positive semi-definite matrix (i.e. $\forall u \in \mathbb{R}^n, u^T A u \geq 0$), then there is an orthogonal set of (column) eigenvectors, v_i , each with non-negative eigenvalues, $\lambda_i \geq 0$, and furthermore the decomposition of A as the following rank one matrices describes the action of A as a weighted sum of simple projections onto the subspaces spanned by each v_i ,

$$A = \sum_{i=1}^N \lambda_i v_i v_i^T \quad (2.71)$$

Solve $\mathbf{U}\mathbf{U}^t$ for eigs; solve fastest decaying time averaged power spectrum

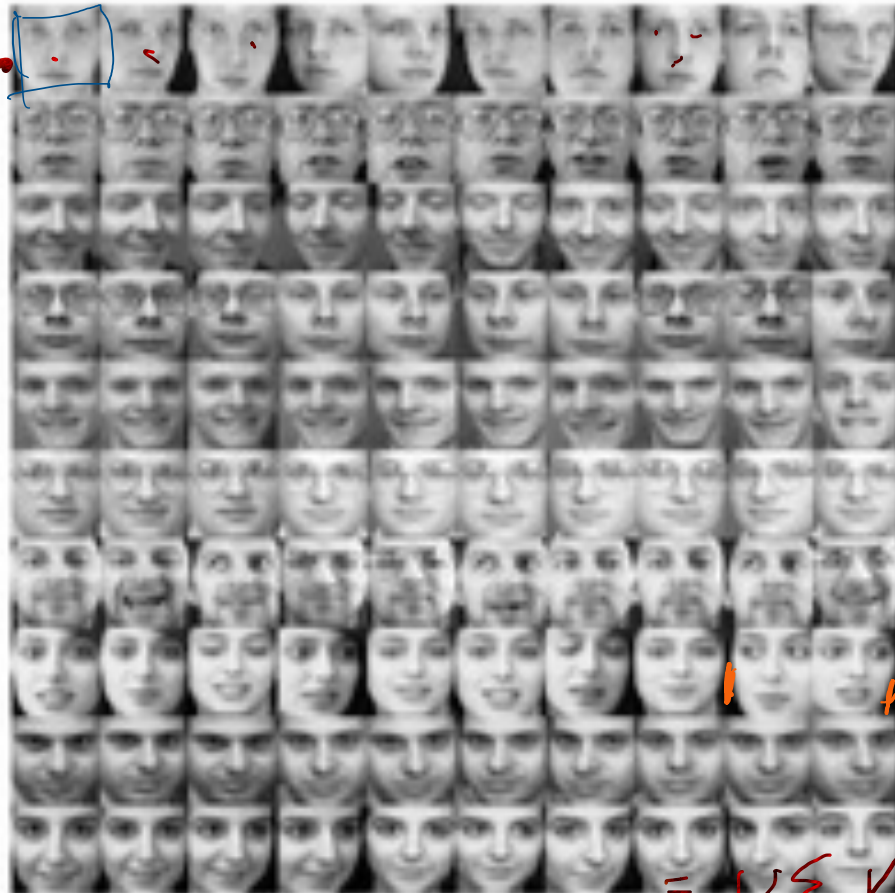
$$\mathbf{U}\mathbf{U}^t\mathbf{V}^t = \Lambda\mathbf{V}^t, \quad (2.74)$$

we use $\mathbf{U} = \mathbf{U}\Sigma\mathbf{V}^t$, with $\Lambda = \Sigma^2$

Eigenface



72 picture



i 
 $p \times q$
 array

reshape
as vector

$$\begin{bmatrix} 1 \\ \vdots \\ x_i \end{bmatrix}_{n \times 1} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

$$M = P^T Q$$

$$\underline{X} = [x_1, x_2, \dots, x_n]_{m \times n}$$

$$= U \Sigma V^T$$

$$P = 1000, q = 2000$$

08/31/20

remove
variance



Register



E_1, E_2, E_3

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

$\|R^T \hat{y}\| = \hat{R}$

b_i^2

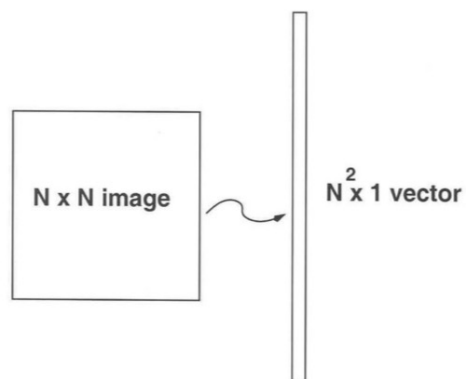
$\underline{X}$

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

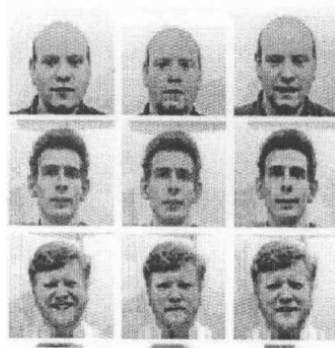
- Suppose Γ is an $N^2 \times 1$ vector, corresponding to an $N \times N$ face image I .
- The idea is to represent Γ ($\Phi = \Gamma$ - mean face) into a low-dimensional space:

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

Computation of the eigenfaces

Step 1: obtain face images $I_1, I_2, \dots, I_M$ (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 :

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

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

$$C = \frac{1}{M} X^T X$$

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 $A^T A$ ($M \times M$ 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 u_i and v_i ?

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

$$CAv_i = \mu_i Av_i \text{ or } Cu_i = \mu_i u_i \text{ where } u_i = Av_i$$

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)

Representing faces onto this basis

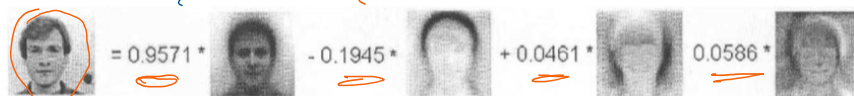
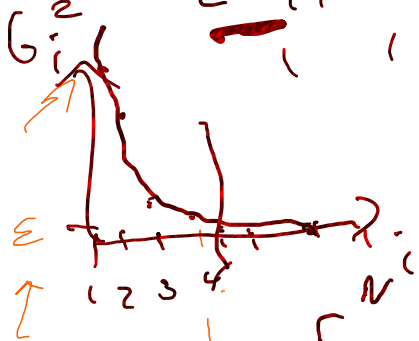
- Each face (minus the mean) Φ_i in the training set can be represented as a linear combination of the best K eigenvectors:

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

$\bullet$ images $\text{size}(g)$
 $\bullet g = \text{reshape}(U(1:i), p, q)$

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

$$U = [u_1 | u_2 | \dots | u_5 | u_{s+1} | \dots | u_r | u_{s+1} | \dots | u_n] \quad (\text{we call the } u_j \text{'s eigenfaces})$$

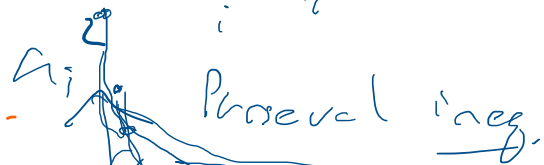


Each normalized training face Φ_i is represented in this basis by a vector:

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



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



$$\|\vec{a}\|_2^2 = \sum_i a_i^2 = \sum_i (f(x_i))^2 = \|f(x)\|_2^2$$

Data $\Rightarrow \begin{pmatrix} | & | & | & | \\ x_1 & x_2 & \dots & x_N \\ | & | & | & | \end{pmatrix}_{n \times n} \rightarrow \underline{X} = \begin{bmatrix} x_1^T & x_2^T & \dots & x_N^T \end{bmatrix}$

$\swarrow$ one column
 $= U \Sigma V^T$



$x_i = a_1^i u_1 + a_2^i u_2 + \dots + a_N^i u_N$

$\swarrow$
 a_1^i $\swarrow$ a_2^i $\swarrow$ a_N^i
 bases
 full of as
 possible for
 the x_i set

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

On basis, functions, and Hilbert space.
Fourier, Taylor, Wavelet, POD-KL

lead in 3 \$ ② - in 5 min. signals analysis, harmonic.

• Historically favorite basis set. $B = \{v_1, w_1, \dots\}$
 (vs. energy favorite basis set comes from PEA)

• Taylor polynomials. $f(x) = a_0 + a_1 x + a_2 x^2 + \dots + a_n x^n$

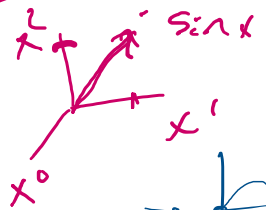
• Fourier modes. $B = \{x^0, x^1, x^2, \dots\}$

$$a_i = \frac{f^{(i)}(0)(x-0)^i}{i!}$$

$$f(x) = a_1 \sin x + a_2 \sin 2x + a_3 \sin 3x + \dots$$

$$B = \{\sin x, \sin 2x, \dots\}$$

H : best space



$$a_1 = (f(x), \sin x)$$

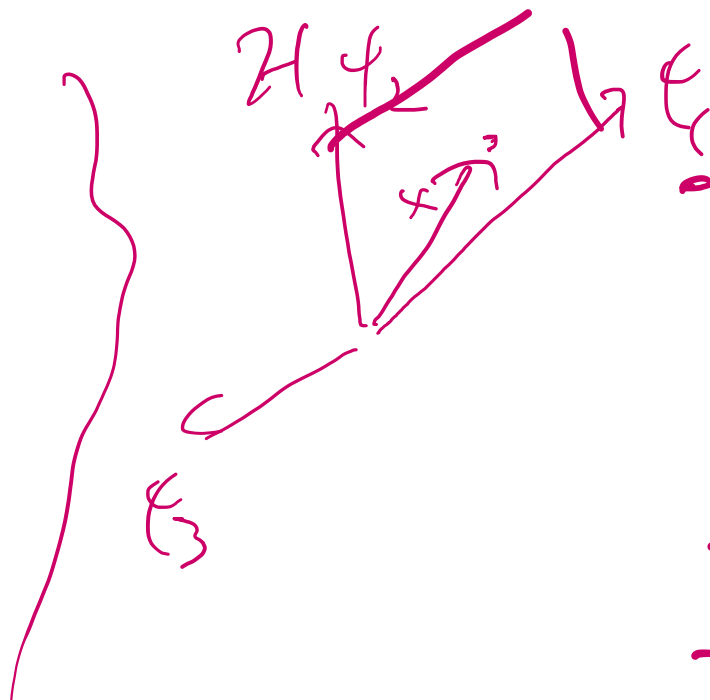
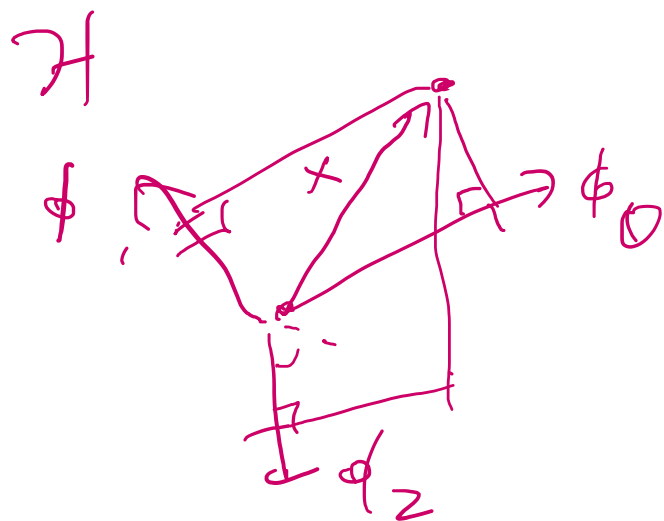
$$= \left(\frac{1}{2\pi} \right) \int_0^{2\pi} f(x) \sin x \, dx$$

• Wavelet basis.
 • Chebyshev polys, Legendre ...

$$\sin x = 1x - \frac{x^3}{3!} + \frac{x^5}{5!} - \dots$$

$\text{complex } \sin x = 1$

Changing bases is a sort of coord-rot.



$$\begin{aligned}\phi_0(y) &= y^0 \\ \phi_1(y) &= y^1 \\ \phi_2(y) &= y^2 \\ &\vdots\end{aligned}$$

$$\rightarrow \phi_1(y) = \sin(y)$$

$$\rightarrow \phi_2(y) = \sin(2y)$$

$$\rightarrow \phi_3(y) = \sin(3y)$$

On basis,
functions,
and Hilbert
space.
Fourier,
Taylor,
Wavelet,
POD-KL

15.2.1

Hilbert space - a complete inner product space.

- An inner product space is a "vector space" E together with a function called "inner product" $\langle \cdot, \cdot \rangle : E \times E \rightarrow \mathbb{C}$ with properties.

Gift: you get geometry in E as on a plane $\langle u, v \rangle = \frac{\langle u, v \rangle}{\|u\| \|v\|}$

• and projection.

• vector space.

set of objects "like vectors"

that have a + and scalar multiplication - including commutative, associative, add. ident, inverse, distributive w.r. scalar identity for scalar.

• conjugate $\langle u, v \rangle = \overline{\langle v, u \rangle} \quad \forall u, v \in E$ (sym.)

• linear $\langle au + bv, v \rangle = a \langle u, v \rangle + b \langle v, v \rangle \quad \forall a, b \in \mathbb{C}, u, v \in E$

• pos. defn: $\langle u, u \rangle \geq 0 \quad \forall u \in E, u \neq 0$

Ex: arrays of real numbers that are 2×1

$$\begin{bmatrix} 1 \\ 2 \end{bmatrix} + \begin{bmatrix} 3 \\ 4 \end{bmatrix} = \begin{bmatrix} 4 \\ 6 \end{bmatrix}$$

(Norm: $\|x\|^2 = \langle x, x \rangle$)

Ex functions in $C([0, 1])$ e.g. $3x^2 + 4x^3 + 7 \sin x = f(x) \in C([0, 1])$
 $\phi_1(x) \quad \phi_2(x) \quad \phi_3(x)$

Ex. Vector space of functions

Name on inner product space $L^2([0, 1]) = \{f \mid \int_0^1 |f(x)|^2 dx < \infty\}$
($L^2([0, 1]) \subset C([0, 1])$)

$$\text{let } \langle f, g \rangle_{L^2([0, 1])} = \int_0^1 f(x) \overline{g(x)} dx \quad \text{3. btw } \|f\|_{L^2([0, 1])} = \left(\int_0^1 |f(x)|^2 dx \right)^{1/2}$$

Basis of unit vectors. • orthogonal

$$B = \{\phi_i \mid \phi_i \in B \subseteq E \text{ and } \langle \phi_i, \phi_j \rangle = \delta_{ij} \text{ and } \|\phi_i\| = 1\}$$

Ex: $L([0, 1])$; $B = \left\{ \cos \frac{2\pi k x}{2}, \sin \frac{2\pi k x}{2}, 1 \mid k \in \mathbb{N} \right\} = \{1, \cos x, \sin x, \dots\}$
if $l=1$; $f(x) = \frac{a_0}{2} + \sum_{k=1}^{\infty} a_k \cos kx + b_k \sin kx$; $a_k = \langle f(x), \cos kx \rangle = \frac{1}{\pi} \int_{-\pi}^{\pi} f(x) \cos kx dx$
 $b_k = \langle f(x), \sin kx \rangle = \frac{1}{\pi} \int_{-\pi}^{\pi} f(x) \sin kx dx$

$$f(x) = \frac{a_0}{2} + \sum_{k=1}^{\infty} a_k \cos kx + b_k \sin kx + \dots$$

$$\phi_k(x) = \frac{2 \cos kx}{2} \text{ or } \frac{2 \sin kx}{2}$$

$$\text{or similar if } \phi_k(x) = e^{ikx}$$

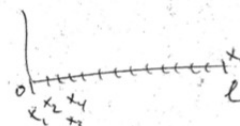
finite if "trig. poly."

$$\text{or maybe } B = \{x^k\} \quad \text{Taylor poly.}$$

$$\text{or maybe } B = \{\Delta, \dots, \Delta\} \quad \text{wavelets.}$$

Heat "2.13" function

Infinite dimensional inner product space of functions.



$x_1, x_2, \dots, x_n$ a grid on $[0, 1]$
 $G = \{x_i \mid i=1, \dots, n\}$

and "function values."

$f: G \rightarrow \mathbb{R}$.
let $f_i = f(x_i) \Rightarrow \vec{f}$
• connect dots if you like.

$$\text{Ex let } \langle \vec{f}, \vec{g} \rangle = \sum_{k=1}^n f_k \bar{g}_k$$

• this inner product space is "isomorphic" to vectors in $\mathbb{C}^n$.

• separable. there exists a countable basis B .
Ex: $L^2([0, 1])$ vs. $L^2(\mathbb{R})$

$\circ B = \{ \text{---}, \text{~}, \text{~}, \text{~}, \text{~} \}$ [only] Fourier
 $\Rightarrow$ Fourier $\{ 1, \sin x, \cos x, \sin 2x, \cos 2x, \dots \}$

$\circ B' = \{ \text{---}, /, \text{~}, \text{~}, \text{~}, \dots \} = \{ 1, x, x^2, x^3, \dots \}$
 Taylor ... Legendre, Taylor

$\circ B'' = \{ \text{---}, \text{~}, \text{~}, \text{~}, \text{~}, \dots \}$
 Wavelet - $\phi_0(x)$

$$f(x) \in L^2(\mathbb{R})$$

$$f(x) = \sum a_i \phi_i(x);$$

$\circ$  Here....

$$a_i = \langle f, \phi_i(x) \rangle$$

$\Rightarrow \|\phi_i(x)\|$

On Compressed Sensing and on to Sparsity

$$f(x) = \sin x + 3 \sin 3x + 4 \sin 7x \in L^2([0, \pi])$$

$$= \sum_{n=1}^{\infty} a_n \sin nx$$

$$f(x) = \begin{pmatrix} a_1 \\ a_3 \\ \vdots \end{pmatrix}$$

$$= \begin{pmatrix} 1 \\ 3 \\ 0 \\ 0 \\ 0 \\ 4 \\ 0 \\ \vdots \end{pmatrix}$$

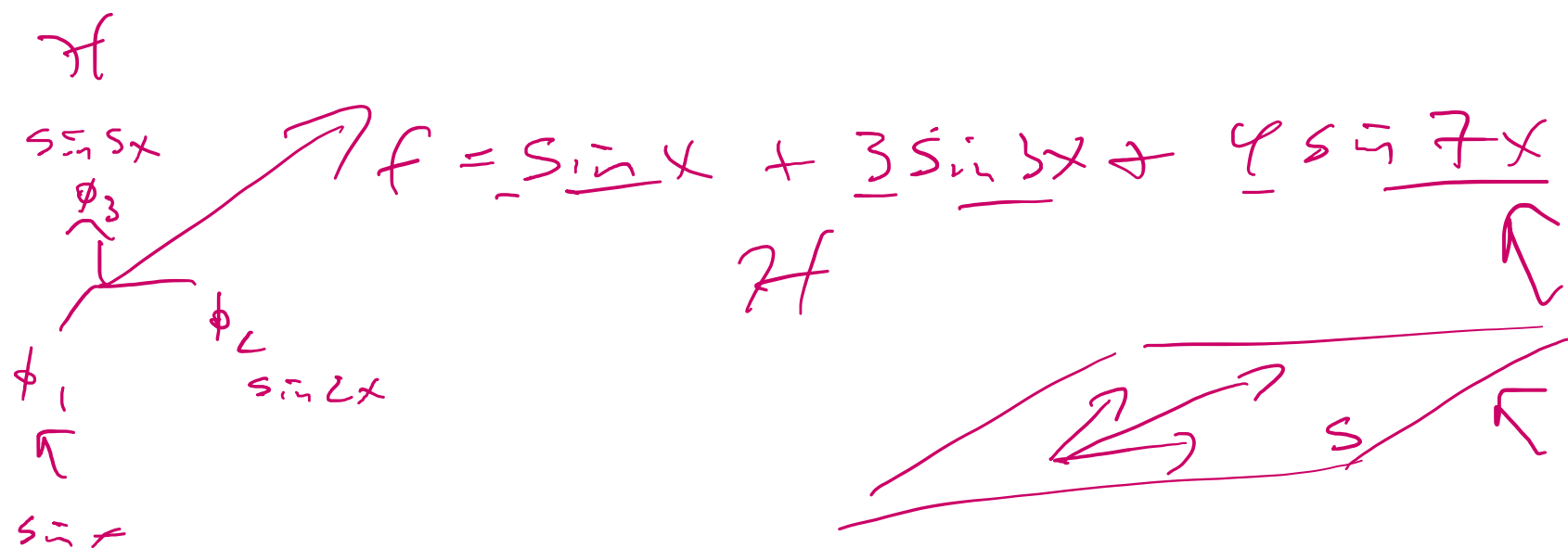
$$f_T = \begin{pmatrix} 38 \\ 0 \\ 0 \\ 0 \\ 0 \\ 5 \\ \vdots \end{pmatrix}$$

$$f(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \dots$$

$$+ 3 \left(3x - \frac{(3x)^3}{3!} + \frac{(3x)^5}{5!} - \dots \right)$$

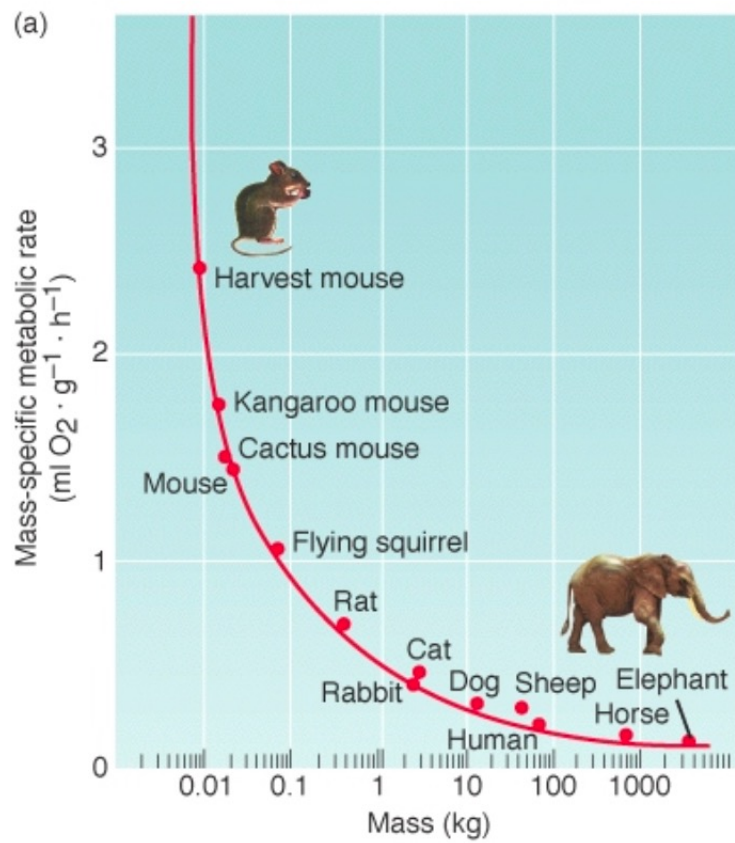
$$+ 4 \left(7x - \frac{(7x)^3}{3!} + \frac{(7x)^5}{5!} - \dots \right)$$

$$= 38x - \left(\frac{1}{5!} + \frac{3^3}{2!} + \frac{4 \cdot 7^3}{3!} \right) x^3 + \dots$$



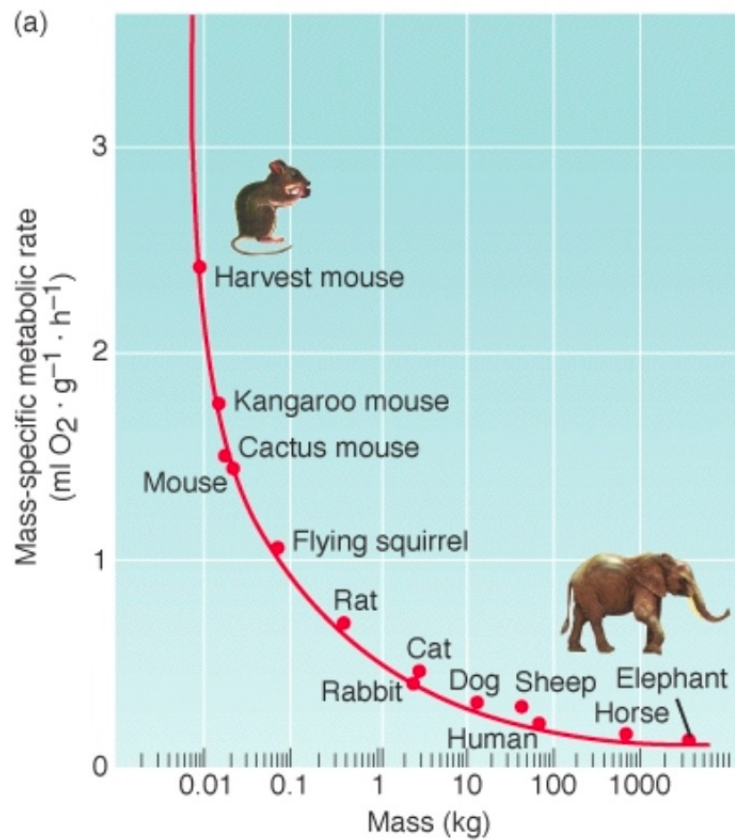
- A vector $v \in E$ is k -sparse, if $[v]$ has exactly k -nonzero values, and $k \leq \dim(E)$

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



$$y_i = \beta_0 + \beta_1 x_i$$

$$y_i = \beta_0 + \beta_1 x_i + \epsilon_i$$



$$y_i = \beta_0 + \beta_1 x_i$$

$$y_i = \beta_0 + \beta_1 x_i + \epsilon_i$$

$$y_0 = \beta_0 + \beta_1 x_0 + \epsilon_0$$

$$y_1 = \beta_0 + \beta_1 x_1 + \epsilon_1$$

$$\vdots$$

$$y_{N-1} = \beta_0 + \beta_1 x_{N-1} + \epsilon_{N-1}$$

$$Y = X\beta + \epsilon$$

$$e_i = (f(x_i) - y_i)^2$$

$$\begin{aligned} E &= \sum_{i=1}^N e_i \\ &= \sum_{i=1}^N (f(x_i) - y_i)^2 \\ &= \sum_{i=1}^N (\beta_0 + \beta_1 x_i - y_i)^2. \end{aligned}$$

$$e_i = (f(x_i) - y_i)^2$$

$$\begin{aligned} E &= \sum_{i=1}^N e_i \\ &= \sum_{i=1}^N (f(x_i) - y_i)^2 \\ &= \sum_{i=1}^N (\beta_0 + \beta_1 x_i - y_i)^2. \end{aligned}$$

$$\frac{\partial E}{\partial \beta_0} = 0 \quad \text{and} \quad \frac{\partial E}{\partial \beta_1} = 0,$$

$$e_i = (f(x_i) - y_i)^2$$

$$\begin{aligned} E &= \sum_{i=1}^N e_i \\ &= \sum_{i=1}^N (f(x_i) - y_i)^2 \\ &= \sum_{i=1}^N (\beta_0 + \beta_1 x_i - y_i)^2. \end{aligned}$$

$$\frac{\partial E}{\partial \beta_0} = 0 \quad \text{and} \quad \frac{\partial E}{\partial \beta_1} = 0,$$

$$\begin{aligned}
\frac{\partial E}{\partial \beta_1} &= \sum_{i=1}^N 2x_i (\beta_0 + \beta_1 x_i - y_i) \\
&= \sum_{i=1}^N (2\beta_0 x_i) + \sum_{i=1}^N (2\beta_1 x_i^2) - \sum_{i=1}^N (2x_i y_i) = 0
\end{aligned}$$

and

$$\begin{aligned}
\frac{\partial E}{\partial \beta_0} &= \sum_{i=1}^N 2 (\beta_0 + \beta_1 x_i - y_i) \\
&= \sum_{i=1}^N (2\beta_0) + \sum_{i=1}^N (2\beta_1 x_i) - \sum_{i=1}^N (2y_i) = 0.
\end{aligned}$$

From the above two equations we have:

$$\begin{aligned}
\sum_{i=1}^N (x_i y_i) &= \sum_{i=1}^N (\beta_0 x_i) + \sum_{i=1}^N (\beta_1 x_i^2) \\
\sum_{i=1}^N (y_i) &= \sum_{i=1}^N (\beta_0) + \sum_{i=1}^N (\beta_1 x_i)
\end{aligned}$$

again, which can be written in matrix form as:

$$\begin{pmatrix} \sum_{i=1}^N (x_i y_i) \\ \sum_{i=1}^N (y_i) \end{pmatrix} = \begin{pmatrix} \sum_{i=1}^N x_i & \sum_{i=1}^N x_i^2 \\ \sum_{i=1}^N 1 & \sum_{i=1}^N x_i \end{pmatrix} \begin{pmatrix} \beta_0 \\ \beta_1 \end{pmatrix}$$

and then

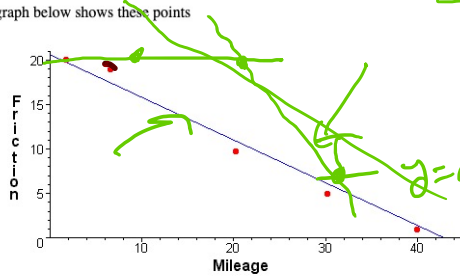
$$\begin{pmatrix} b_0 \\ b_1 \end{pmatrix} = \begin{pmatrix} \sum_{i=1}^N x_i & \sum_{i=1}^N x_i^2 \\ \sum_{i=1}^N 1 & \sum_{i=1}^N x_i \end{pmatrix}^{-1} \begin{pmatrix} \sum_{i=1}^N (x_i y_i) \\ \sum_{i=1}^N (y_i) \end{pmatrix}.$$

Example

An engineer is tracking the friction index over mileage of a braking system of a vehicle. She expects that the mileage-friction relationship is approximately linear. She collects five data points that are shown 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 $\mathbf{b}$ is the vector of friction index data values and $\mathbf{y}$ 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 $\mathbf{b}$ and $\mathbf{y}$. If the equation of the line is

$$ax + b = y$$

then we get the five equations

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

The corresponding matrix equation is

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

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

$$y = -0.48x + 20.6$$

$$\text{Cost} = \sum_{i=1}^n ||y_i - (ax_i + b)||^2$$



$$\begin{pmatrix} x_1 & 1 \\ x_2 & 1 \\ \vdots & \vdots \\ x_n & 1 \end{pmatrix} \begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}$$

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

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

$$\vdots$$

$$\begin{pmatrix} 1 & 2 \\ 2 & 4 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 3 \\ 4 \end{pmatrix}$$

$$A\mathbf{x} = \mathbf{y} \quad \mathbf{x} = \begin{pmatrix} a \\ b \end{pmatrix} \quad \mathbf{x} = A^{-1} \mathbf{y}$$

$$\mathbf{x} = A^+ \mathbf{y} \quad \text{vs} \quad A^T$$

Least Squares

Definition and Derivations

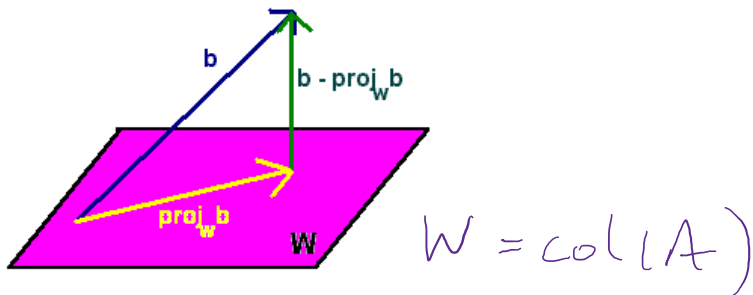
We have already spent much time finding solutions to

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

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

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

where W is the column space of A .



Notice that $\mathbf{b} - \text{proj}_W \mathbf{b}$ 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

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

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

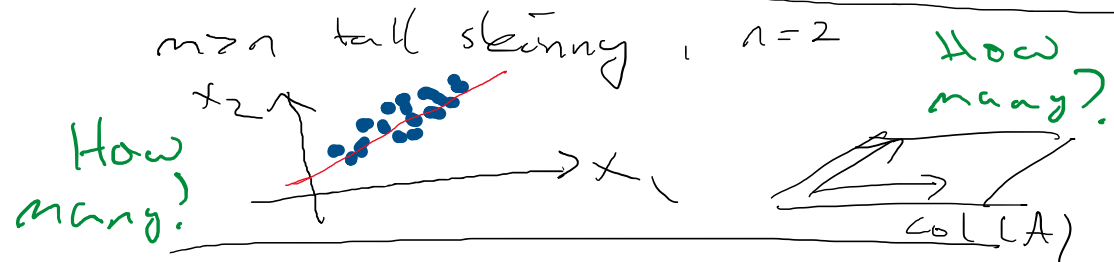
Lemma

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

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

$m - \text{eqns}$
 $n - \text{unknowns}$
 x

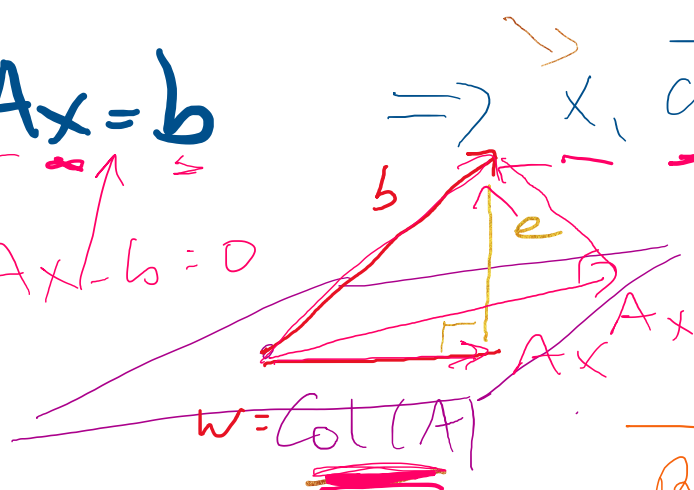
$$\Leftrightarrow \begin{aligned} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n &= b_1 \\ \vdots \\ a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n &= b_m \end{aligned}$$



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

$$Ax = b$$

$$Ax - b = 0$$



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

$$A \tilde{x} = \text{proj}_W b$$

$$\tilde{x} = \underset{x}{\text{argmin}} \|Ax - b\|_2^2$$

LS solution

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

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

$$\Rightarrow A^T (A \tilde{x} - b) = 0$$

$$\begin{pmatrix} -a'_1 \\ -a'_2 \\ \vdots \\ -a'_n \end{pmatrix} (Ax - b) = 0 \leftarrow \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

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

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

$$\Rightarrow (A \tilde{x} - b) \perp \text{every vector in } \text{Col}(A) \Leftrightarrow \text{solve "normal eqns" } A^T A \tilde{x}$$

Theorem

Let A be an $m \times n$ matrix of rank n , then the system

$$Ax = b$$

has the unique *least squares* solution

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

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

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

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

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.

$Ax = b$ $m \times n$ • Square & nonsing. A^{-1} d-ne. $x = A^{-1}b$ unique. • Not, $m > n$

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

$n \times n$ $n \times n$ normal equations

Covariance matrix (if demeaned).

$$\begin{pmatrix} 1 & 2 & 1 \\ 3 & 4 & 6 \end{pmatrix}^T \begin{pmatrix} 1 & 3 \\ 2 & 4 \\ 1 & 6 \end{pmatrix}^{-1} \begin{pmatrix} 1 & 2 & 1 \\ 3 & 4 & 6 \end{pmatrix} \begin{pmatrix} 4 \\ 1 \\ 3 \end{pmatrix} = x$$

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.

Example

A bioengineer is studying the growth of a genetically engineered bacteria culture and suspects that it is 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

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

This gives six equations with four unknowns

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

The corresponding matrix equation is

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

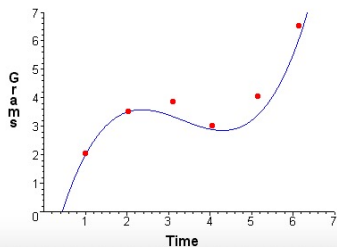
We can use the least squares equation to find the best solution

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

So that the best fitting cubic is

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

The graph is shown below



Matrix Formulation of

LS slide

for general Models



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

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

$$\begin{aligned} Ax &= b \\ x &= A^{-1}b \end{aligned}$$

$$\begin{pmatrix} x_1^3 & x_1^2 & x_1 & 1 \\ x_2^3 & x_2^2 & x_2 & 1 \\ \vdots & \vdots & \vdots & \vdots \\ x_n^3 & x_n^2 & x_n & 1 \end{pmatrix} \begin{pmatrix} a \\ b \\ c \\ d \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}$$

$$A_{m \times n} = \begin{bmatrix} \overbrace{U_{m \times r}}^{U_{m \times n}} & \overbrace{U_{m \times (n-r)}}^{U_{m \times (n-r)}} & \overbrace{U_{m \times (m-n)}}^{U_{m \times (m-n)}} \\ \hline u_1 & \dots & u_r & u_{r+1} & \dots & u_n & u_{n+1} & \dots & u_m \end{bmatrix} \begin{bmatrix} \overbrace{\Sigma_{r \times r}}^{\Sigma_{n \times n}} & 0 \\ \hline 0 & \overbrace{\Sigma_{(n-r) \times (n-r)}}^{\Sigma_{(n-r) \times (n-r)}} \\ \hline 0 & 0 & \Sigma_{(m-n) \times n} \end{bmatrix} \begin{bmatrix} V_{r \times n}^* \\ \hline V_{(n-r) \times n}^* \end{bmatrix}$$

$U_{m \times m}$ $\Sigma_{m \times n}$ $V_{n \times n}^T$

$$(\hat{V} \hat{\Sigma}^T \hat{V})^T (\hat{U} \hat{\Sigma} \hat{V})^T x$$

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

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

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

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

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

$\hat{\Sigma}$

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

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

$\hat{\Sigma} \hat{\Sigma}^T$ when exist > 0 $= 0$

$$A_{m \times n} = \begin{bmatrix} \overbrace{U_{m \times r}}^{U_{m \times n}} & \overbrace{U_{m \times (n-r)}}^{U_{m \times (m-n)}} & \overbrace{U_{m \times (m-n)}}^{\hat{U}_{m \times (m-n)}^\perp} \end{bmatrix} \begin{bmatrix} \Sigma_{r \times r} & 0 \\ 0 & \Sigma_{(m-n) \times n} \end{bmatrix} \begin{bmatrix} V_{r \times n}^* \\ V_{(n-r) \times n}^* \end{bmatrix}$$

$U_{m \times m}$
 $\Sigma_{m \times n}$
 $V_{n \times n}^T$

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

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

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

if $\hat{\Sigma}$ is invertible

• if $r = n$

if $\sigma_i > 0, 1 \leq i < n$

But if $r < n$
 $\sigma_r > 0, \sigma_{r+1} = 0$

$$(\hat{\Sigma}^T \hat{\Sigma}) = \begin{pmatrix} \sigma_1^2 & 0 & \dots & 0 \\ 0 & \sigma_2^2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \sigma_r^2 \\ 0 & 0 & \dots & 0 \end{pmatrix}$$

$\frac{1}{\sigma} \dots = 0$
 $\frac{1}{\sigma} \dots = 0$
 $\frac{1}{\sigma} \dots = 0$

LS soln = solve normal equations

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

When inverse exists

$$\begin{aligned} \tilde{x} &= (A^T A)^{-1} A^T b \\ &\equiv A^+ b \end{aligned}$$

Moore -
Penrose
Pseudo-Inverse

$$A^+$$

$$Ax = b$$

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

$$Ax = b$$

$$U^T U \Sigma V^T x = U^T b$$

$$(\Sigma^T U^T V^T x = \Sigma^T V^T b$$

$$V^T x = (\Sigma^T \Sigma)^{-1} \Sigma^T V^T b$$

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

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

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

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

$$\equiv \begin{pmatrix} 1/\sigma_1 & \dots & 1/\sigma_r \\ 0 & \dots & 0 \end{pmatrix}$$

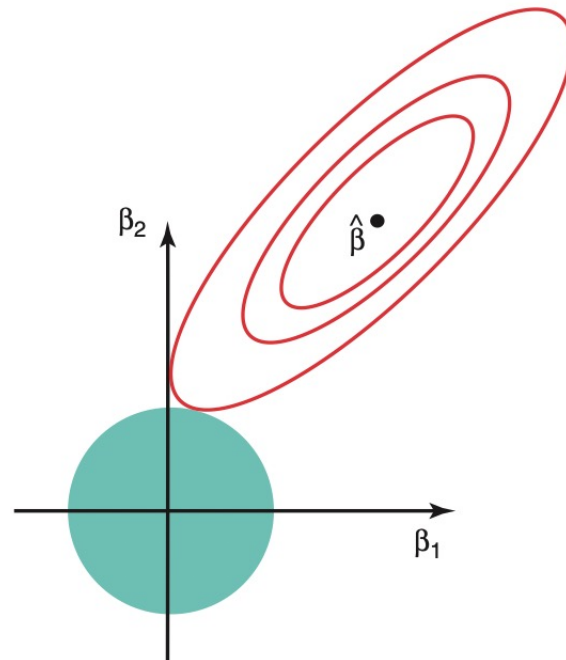
$$\begin{cases} r \neq 0 \\ \sigma_r \neq 0 \Rightarrow \frac{1}{0} = 0 \end{cases}$$

if it exists

Tikhonov Regularization for inverse problems and preventing over fitting

-Ridge regression

-Lasso regression



K-fold leave 1 out cross validation – A version cross validation



Then average scores of testing out of training sample

K-fold leave 1 out cross validation



Compressed sensing and L1 regularization – Sparse Signal Recovery

Exact sparsity vs. approximate

$$f(x) = \sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \frac{x^9}{9!} - \frac{x^{11}}{11!} + \frac{x^{13}}{13!} + \dots$$



$$\sin x = \sum_{n=1}^{\infty} a_n \phi_n(x) \approx \sum_{n=1}^N a_n \phi_n(x)$$

$\phi_n(x) = x^n$
 $f(x)$ $f_N(x)$

$f = \begin{pmatrix} 1 \\ 0 \\ 1/6 \\ 0 \\ 1/120 \\ 0 \\ 1/720 \end{pmatrix}$ vs. $f_N = \begin{pmatrix} 1 \\ 0 \\ 1/6 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$

$N=7$ $N=3$

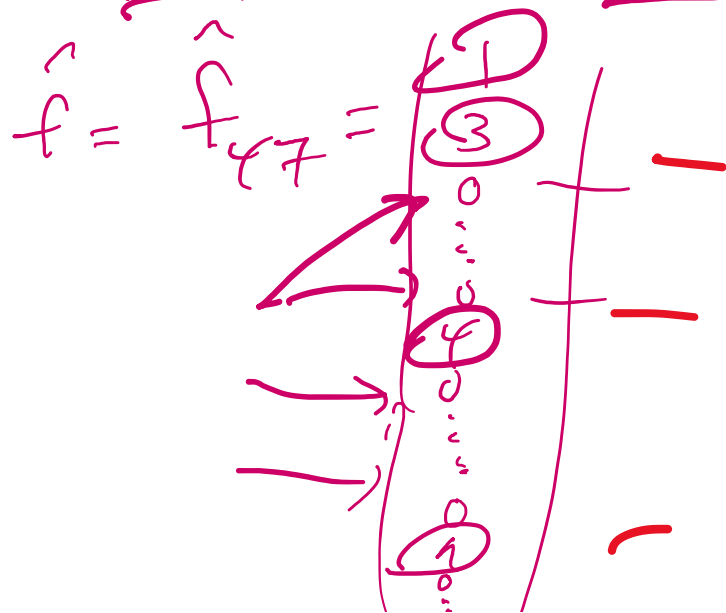
$$\|f - f_N(x)\|$$

$$= \left\| \sin x - \left(x - \frac{x^3}{3!} \right) \right\|$$

$N=3$

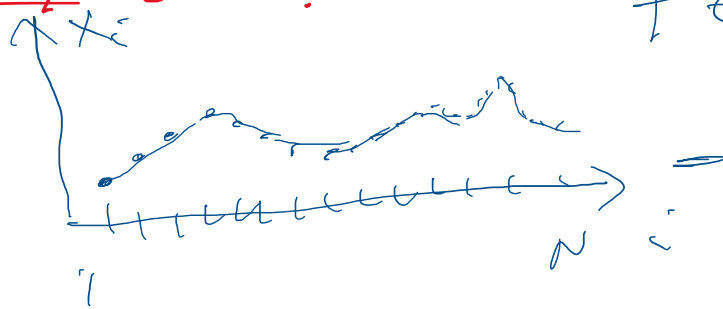
- Truncation impose sparsity -
(just truncate after a large number.)

$$\bullet \underline{f(x) = \sin x + 3 \sin 3x + 9 \sin 12x + 9 \sin 47x}$$

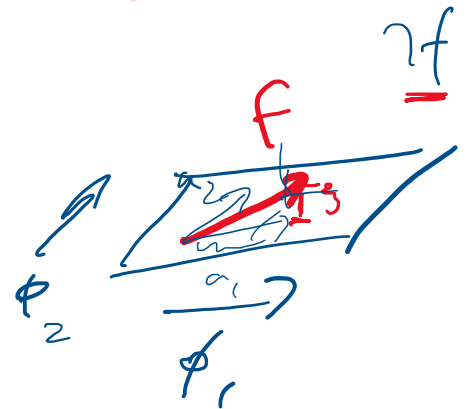


just 1, 3, 4, 9 and
tell guy where to
put them. }

3 Truncation as projection 3 projection
by matrix mult.



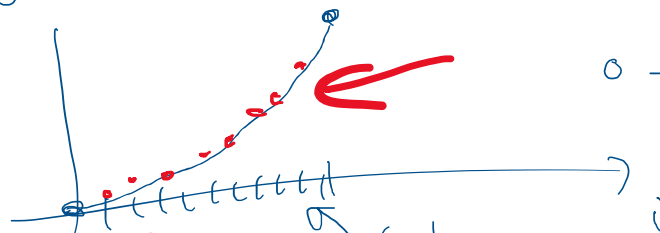
$$f \in \mathcal{F}$$



(say from $f(s) = \bar{s}^3$)

$$x_i = f(s_i)$$

$$0 \leq \bar{s} \leq 2$$



$$N = 100, \text{ uniform}$$

$$h = \frac{2-0}{N} = \frac{2}{100} = \frac{1}{50}$$

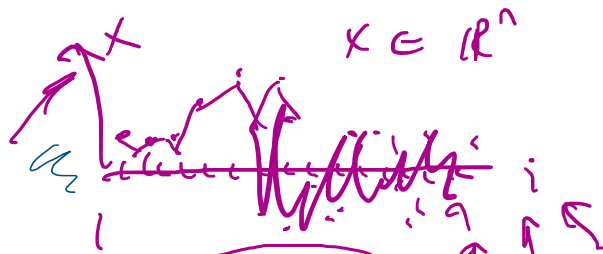
$$x_i = [x]_i$$

$$[x]_3 = x_3 = 0$$

$$\vec{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{pmatrix} = \begin{pmatrix} 0 \\ 1/50^3 \\ \vdots \\ (2/50)^3 \end{pmatrix}$$

$$s_1 = 0, s_2 = \frac{1}{50}, s_3 = \frac{2}{50}, \dots, s_{100} = 2$$

• Suppose signal



$$x_i; \vec{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}$$

FFT
or WT
or other
PCA/KL

$$\langle x, \phi_j \rangle = s_j$$

basis functions

$$\vec{x} = s_1 \vec{\phi}_1 + s_2 \vec{\phi}_2 + \dots + s_k \vec{\phi}_k$$

$k = ? = \infty$

column vector
project on its
side

$$\phi_j(i) = \sin j i \pi$$

i is the index var.

$$\begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} s_1 \phi_1(1) & s_2 \phi_2(1) & \dots & s_k \phi_k(1) \\ s_1 \phi_1(2) & s_2 \phi_2(2) & \dots & s_k \phi_k(2) \\ \vdots & \vdots & \ddots & \vdots \\ s_1 \phi_1(n) & s_2 \phi_2(n) & \dots & s_k \phi_k(n) \end{pmatrix}$$

$$\begin{pmatrix} \phi_1 & \phi_2 & \dots & \phi_k \\ \vdots & \vdots & \ddots & \vdots \end{pmatrix} \begin{pmatrix} s_1 \\ s_2 \\ s_3 \\ \vdots \\ s_k \end{pmatrix}$$

$\Phi_{n \times k}$ $\vec{s}_{k \times 1}$ $\vec{x}_{n \times 1}$

$$\boxed{x = \Phi s}$$

$\|S\|_0 = \# \text{ of non-zero values.}$

$$\|S\|_1 = |S_1| + |S_2| + \dots + |S_n|$$

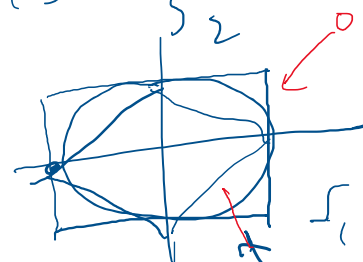
$$\|S\|_2 = (S_1^2 + S_2^2 + \dots + S_n^2)^{1/2}$$

$$\|S\|_\infty = \max_i |S_i|$$

$$\|S\|_p = (S_1^p + S_2^p + \dots + S_n^p)^{1/p}$$

$S \in \mathbb{R}^n$

"Balls" $\|S\|_p = 1$



Euclidean - $\|S\|_2 = 1$

want to solve $\diamond$ w.r. 0 - no cm - But have to test all vlds NP-complete.

⑧



$$y = \Theta S$$

$$\rho = 1$$

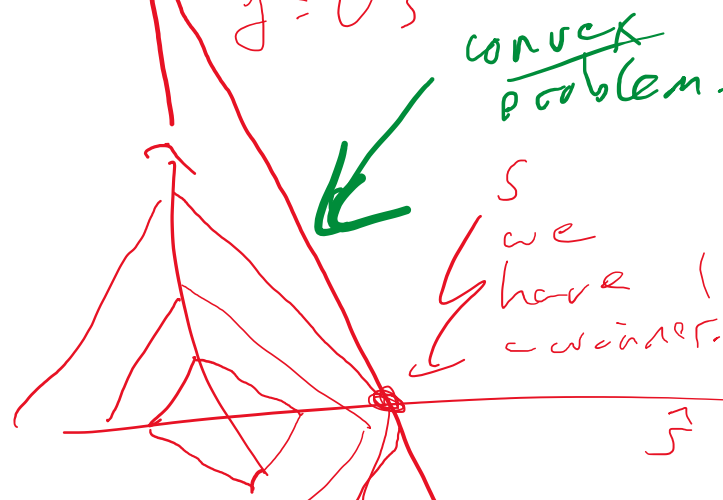
$$S = \underset{S \in \mathcal{S}}{\operatorname{argmin}} \|\hat{S}\|_2$$

basis pursuit --- $y = \Theta \hat{S}$ convex optimization.

$$S = \underset{S \in \mathcal{S}}{\operatorname{argmin}} \|\hat{S}\|_1$$

$$\Rightarrow y = \Theta \hat{S}$$

on the circle - $y = \Theta S$



$$\vec{x} = \Psi \vec{s} \quad \begin{matrix} n \times 1 & n \times n & n \times 1 \end{matrix} = \begin{pmatrix} \psi_1 & \psi_2 & \dots & \psi_n \\ \vdots & \vdots & & \vdots \\ \vdots & \vdots & & \vdots \end{pmatrix} \begin{pmatrix} s_1 \\ s_2 \\ \vdots \\ s_n \end{pmatrix} \quad \begin{matrix} n \times n & n \times 1 \end{matrix} \quad 3.1$$

$$\vec{x} \in \text{col}(\Psi) \\ \equiv \text{linear combo of columns}$$

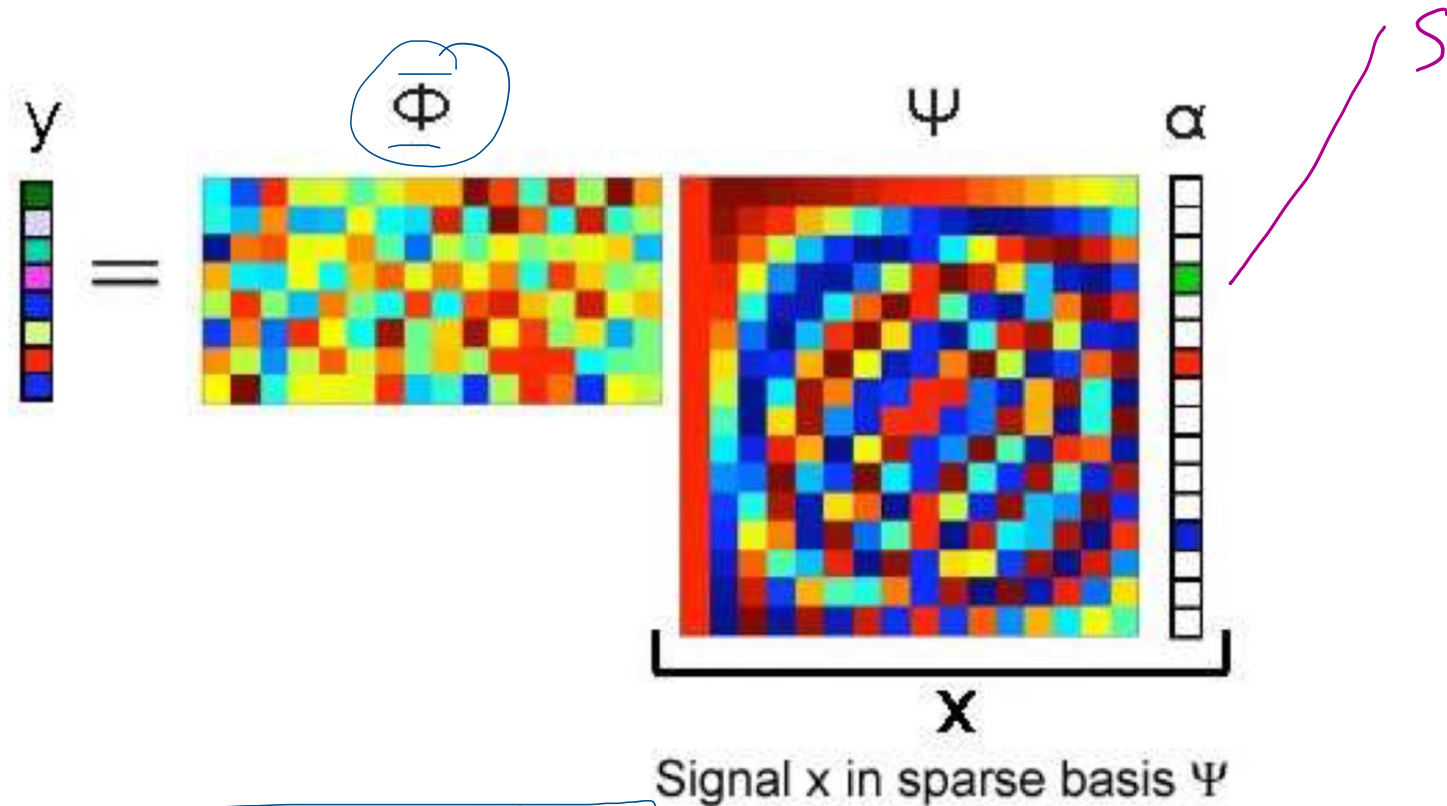
$\uparrow$
 $n \times 1$
column

$$= s_1 \psi_1 + s_2 \psi_2 + \dots + s_n \psi_n \\ (\text{linear combo of columns})$$

$$\vec{x} = \sum_{i=1}^n s_i \psi_i \approx \sum_{i=1}^n s_i \psi_i, \quad n < \infty$$

$$\vec{x} = \left(\psi_1 \psi_2 \dots \psi_n \right) \begin{pmatrix} s_1 \\ s_2 \\ \vdots \\ s_n \end{pmatrix} \rightarrow \underline{\Psi_n} \text{ vs. } \underline{\Psi_n} = \begin{pmatrix} \psi_1 & \dots & \psi_n \end{pmatrix}$$

Random measurements can be used for signals sparse in any basis.



Φ good enough if "coherent" to high probability

$$\Phi_{n \times n} \sim \text{randn}(p, n)$$

$$X = \Phi S$$

$$\begin{matrix} \Phi & S \\ n \times m & m \times 1 \\ X & m \times 1 \end{matrix}$$

- choose $k = m$ modes
- a sampler of space for X BT:

$$\begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix} = \begin{pmatrix} | & | & \Phi & | & | \end{pmatrix} \begin{pmatrix} s_1 \\ \vdots \\ s_k \\ \vdots \\ s_m \end{pmatrix}$$

Shannon-Nyquist

- Choose k at least twice as big as frequency content of X

$\left. \begin{matrix} s_1 \\ \vdots \\ s_k \end{matrix} \right\} \begin{matrix} \text{not zero} \\ 20,000 \end{matrix}$
 $\left. \begin{matrix} s_{k+1} \\ \vdots \\ s_m \end{matrix} \right\} 40,000$

- if S is full (non zeros.) then I need at least to truncate "late!"

$$X = \begin{bmatrix} \vdots \\ s_4 \\ \vdots \\ s_75 \\ \vdots \\ s_{127} \\ \vdots \\ \vdots \end{bmatrix}$$

$\frac{1}{S}$ vector

S_i is a scalar

- $\Rightarrow$ is
- k -sparse if $\vec{s}$ has exactly k non zero values
 - Call sparse if $k \ll n$
 $3 \ll 40,000$

3

~~Any~~ x

• sparse basis but linearly full in another basis.

$$\begin{aligned} & \frac{1}{\pi} \sin 4i\pi \quad \frac{1}{\pi} \sin 75i\pi \\ & \downarrow \quad \downarrow \\ & = s_4 \phi_4 + s_{75} \phi_{75} + s_{127} \phi_{127} \\ & = \hat{s}_1 \hat{\phi}_1 + \hat{s}_2 \hat{\phi}_2 + \dots + \hat{s}_n \hat{\phi}_n \end{aligned}$$

• What if x is not sparse w.r.t. $B = \{\phi_i\}_{i=1}^m$
 it may be almost sparse w.r.t. B .

$$\vec{x} = s_1 \phi_1 + s_2 \phi_2 + s_3 \phi_3 + \dots + s_m \phi_m$$

• and all or most $s_i \neq 0$ but

$s_4 > s_{75} > s_{127} \gg s_i \approx 0$
 most of the $s_i \approx 0$.

$$S = \begin{pmatrix} 1 \cdot 10^{-6} \\ 1 \cdot 10^{-8} \\ 3 \cdot 10^{-7} \\ 14 \\ 2 \cdot 10^{-6} \\ \vdots \\ 3 \cdot 10^{-11} \\ 5 \end{pmatrix}$$

$x = \Phi s$ s coefficients
~~signal~~ assume s is k -sparse or at
 least almost k -sparse.

$\|s - \hat{s}\|_2 < \delta$ for $\delta > 0$ small $\hat{s}$ is sparse
 projection matrix C .

Observe

$$y = \underset{p \times 1}{C} \underset{p \times n}{X}$$

$$x_{n \times 1}$$

$$p < n$$

signal I want

$$y = \begin{pmatrix} \ominus \end{pmatrix} s \quad \checkmark$$

$p \times 1 \quad \uparrow \quad p \times n \quad n \times 1$

Solve for s

$x = \Phi s$

~~$$s = \begin{pmatrix} \ominus \end{pmatrix}^{-1} y$$

$p < n$~~

singular problem $\Rightarrow$ no unique solution
 $\Rightarrow$ inverse problems tend to be ill-posed

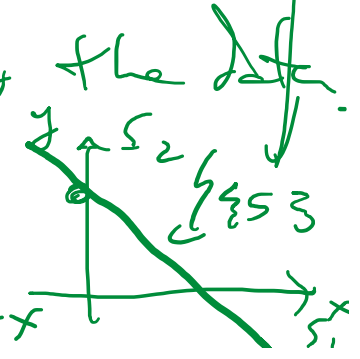
Hadamard define a well posed problem as
 on where
 else call ill-posed

- solution exists
- uniquely
- and ctsly w.r.t. to the data.

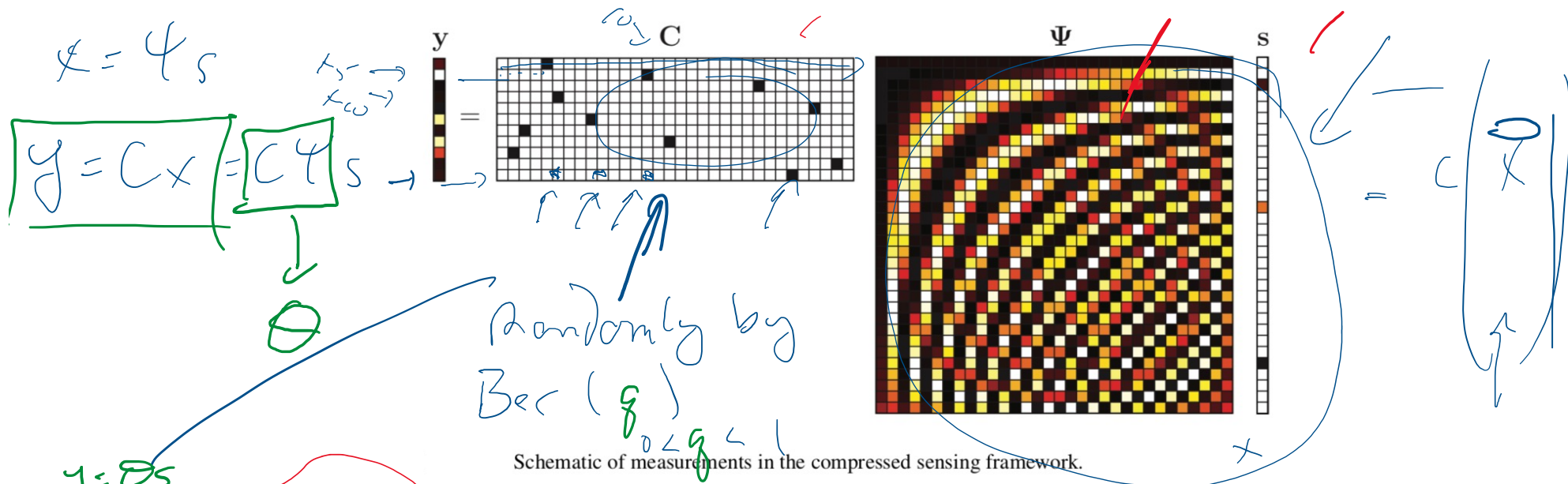
Ex: $y = \begin{pmatrix} 7 \\ 1 \end{pmatrix}$ if $\begin{pmatrix} 1 & 1 \end{pmatrix} \begin{pmatrix} s_1 \\ s_2 \end{pmatrix} = \begin{pmatrix} 7 \\ 1 \end{pmatrix}$

$l=1, n=2$

$$\begin{pmatrix} 1 & 1 \end{pmatrix} \begin{pmatrix} s_1 \\ s_2 \end{pmatrix} = \begin{pmatrix} 7 \\ 1 \end{pmatrix} \Rightarrow \begin{aligned} 7 &= s_1 + s_2 \\ s_2 &= 7 - s_1 / y = 7 - x \end{aligned}$$



$\uparrow \ominus = s$



$y = \Theta s$

$\hat{s}_1, \hat{s}_2, \dots, \hat{s}_n$

$\arg \min \|s\|_1$

$y = \Theta s$

$\leftarrow$ Lasso Regression

(a) Sparse $s (\ell_1)$

$\forall s, J(s) = \|y - \Theta s\|_2^2 + \lambda \|s\|_1$

$\arg \min \|y - \Theta s\|_2^2$

$\forall s, J(s) = \|y - \Theta s\|_2^2$

(b) Least-squares $s (\ell_2)$

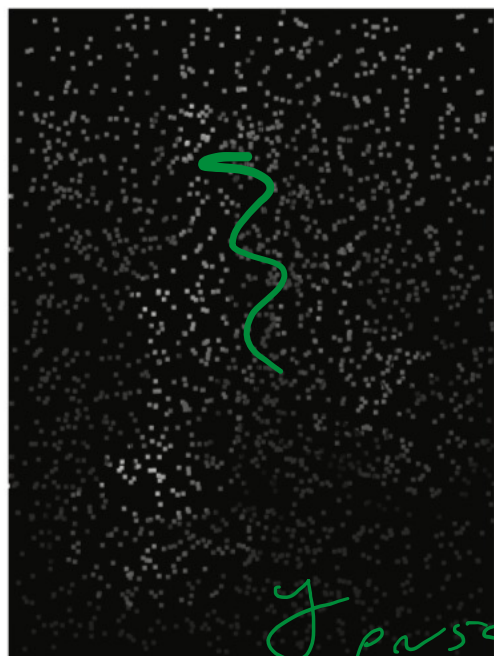
$\leftarrow$ Ridge Regress

$\leftarrow$ Tikhonov Regular

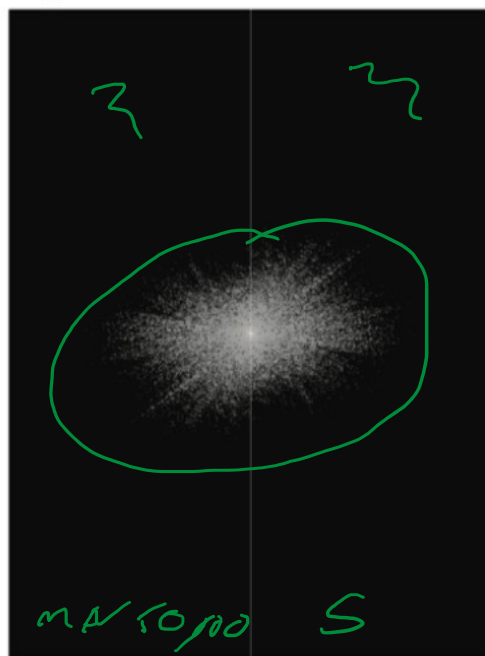
$\theta < 1$

$$y = \Phi s$$

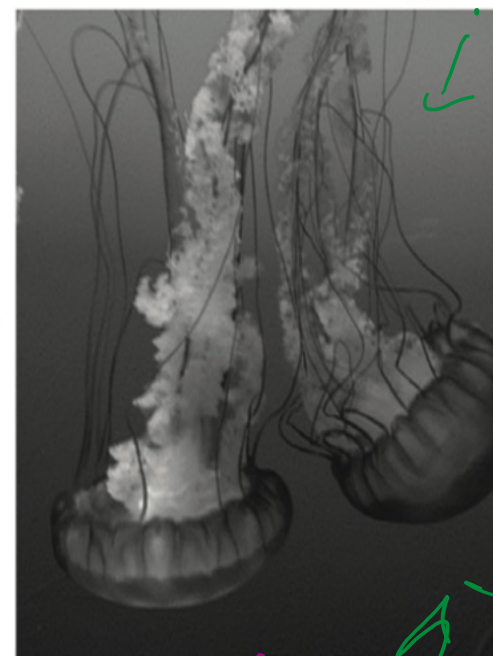
Measurements, y



Sparse Coefficients, s



Reconstructed Image, x



ℓ_1

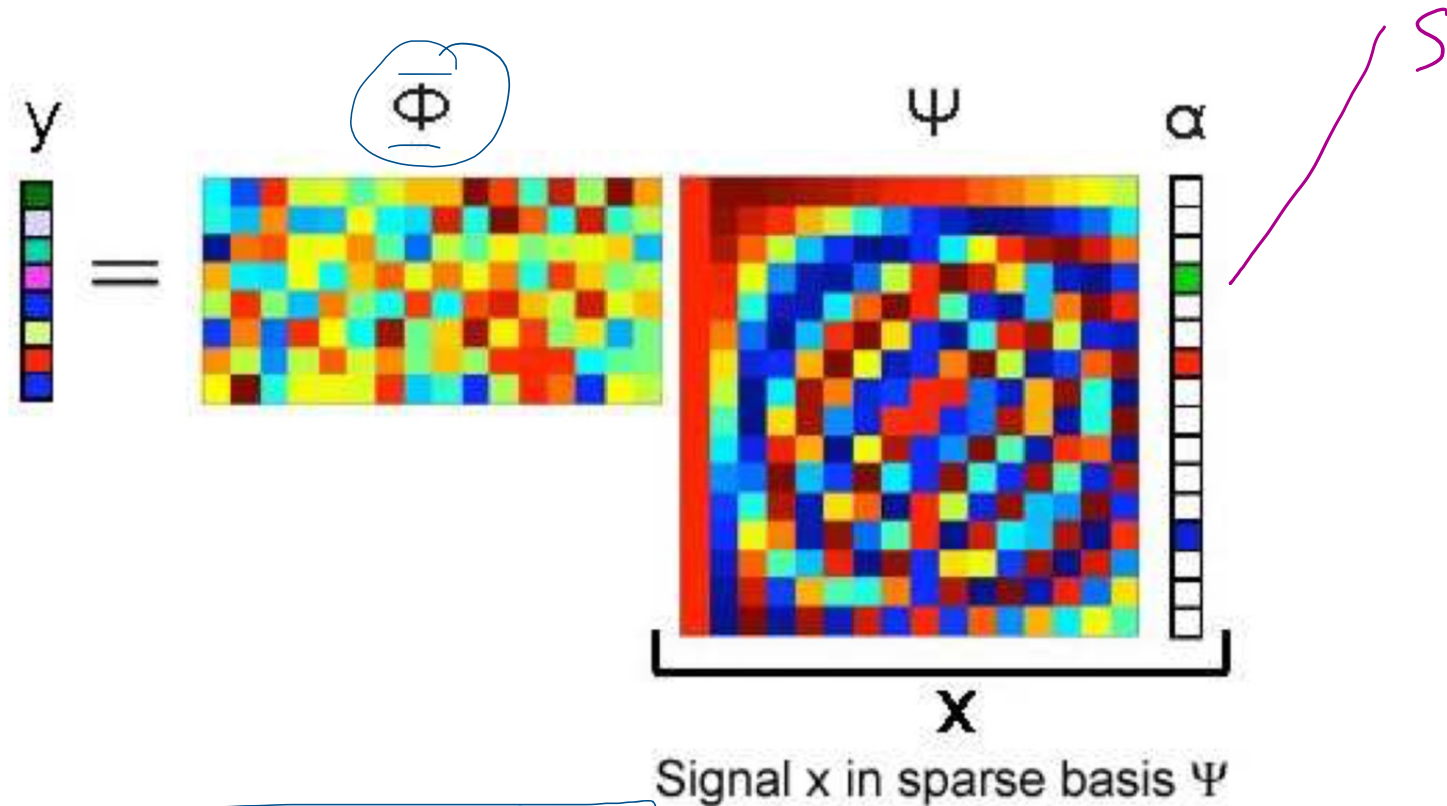
$\mathcal{F}^{-1}$

$$y_{pxl} = C \Phi s$$

$$= \Phi s_{n \times l}$$

$$x = \Phi s$$

Random measurements can be used for signals sparse in any basis.



Φ good enough if "coherent" to high probability

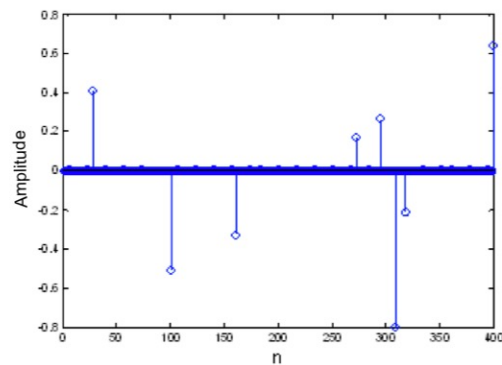
$$\Phi_{n \times n} \sim \text{randn}(p, n)$$

$$y_k = \langle \phi_k, x \rangle; \quad k = 1, \dots, M; \quad \text{with } M \ll N$$

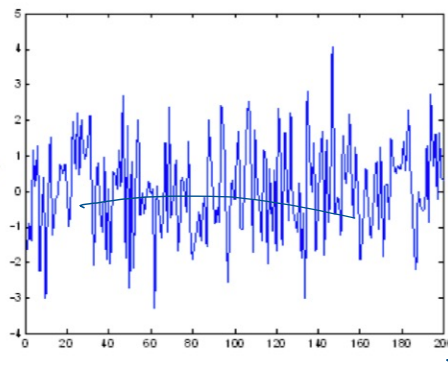
- Need to solve an under determined system of equations $y = \Phi x$.
- Infinitely solutions for the system since $M \ll N$.
- A sparse solution x is recovered from y by solving the following inverse problem

$$(P0) : \min_x \|x\|_{\ell_0} \quad s.t. \quad y = \Phi x$$

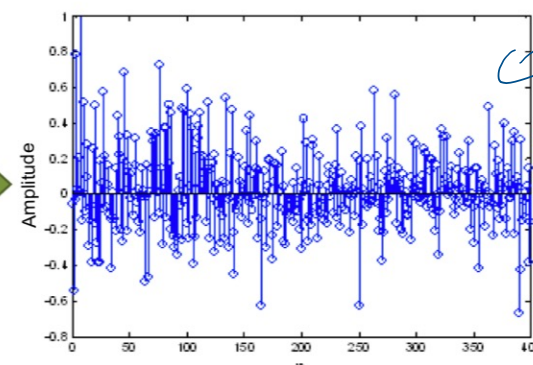
Example of the recovery of an under determined system of equations:



Original sparse signal



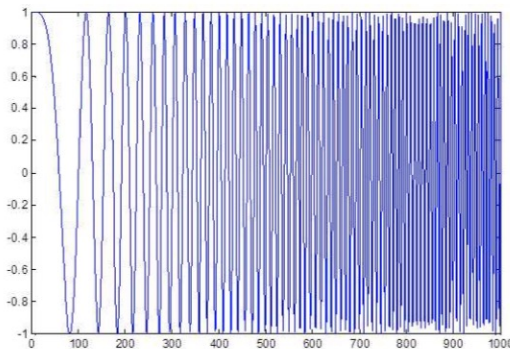
Compressed measurements



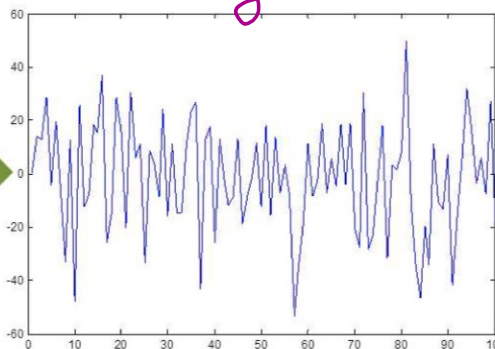
Reconstructed signal using least-squares.
Solution not sparse



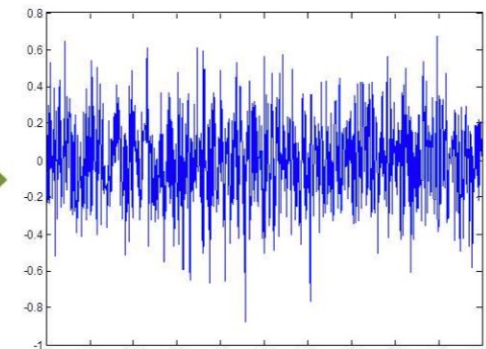
Handwritten notes and arrows pointing to the reconstructed signal plot, indicating a solution that is not sparse.



Original sparse signal
in Time-Frequency basis



Compressed Measurements



Reconstructed signal using
least-squares

- Sparsity is what makes it possible to recover a signal from undersampled data.
- The number of measurements we need for successful reconstruction depends on the nature of the waveforms ϕ_k , and S

1. Incoherent Orthobasis

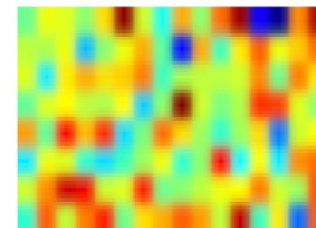


Ordered Hadamard
Ensemble



Scrambled Block
Hadamard Ensemble

2. Random waveforms ϕ_k



Gaussian Random
Ensemble

Incoherent Orthobasis Example

Example of incoherent basis: the "spike" basis (identity) and the Fourier basis.

Consider the case where the dictionary is the union of two orthobasis:

- I : the "spike" basis (identity).
- F : the Fourier basis (sinusoids).

$$\Phi = [I; F]$$

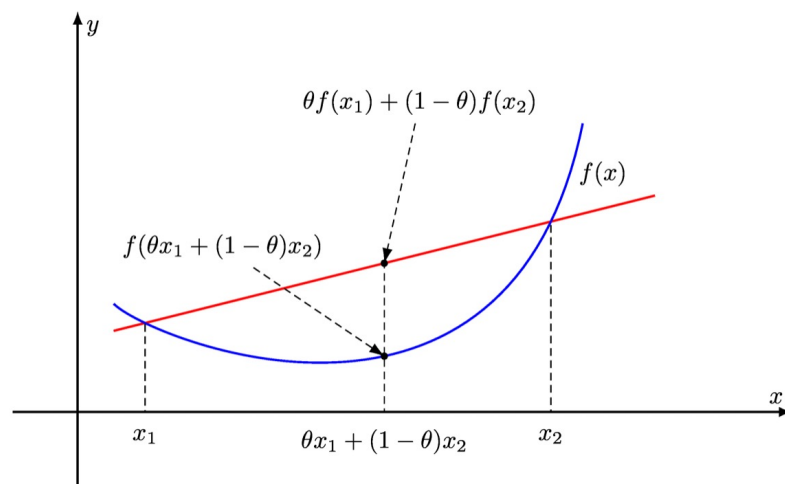
where I is a $N \times N$ matrix and F is a $N \times N$ matrix with

$$f_{m,\ell} = \frac{1}{\sqrt{(N)}} e^{j2\pi(m-1)(\ell-1)/N}$$

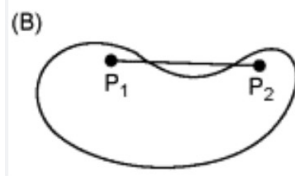
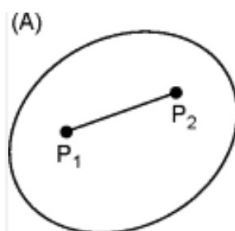
Convex Optimization – optimize a convex objective function over a convex domain

Convex function

$$f(tx_1 + (1-t)x_2) \leq tf(x_1) + (1-t)f(x_2)$$



Convex Domain



Compressive Sensing

[Candes, Romberg, Tao; Donoho]

- Signal x is K -sparse in basis/dictionary Ψ
 - WLOG assume sparse in space domain $\Psi = I$
- Replace samples with *few linear projections* $y = \Phi x$

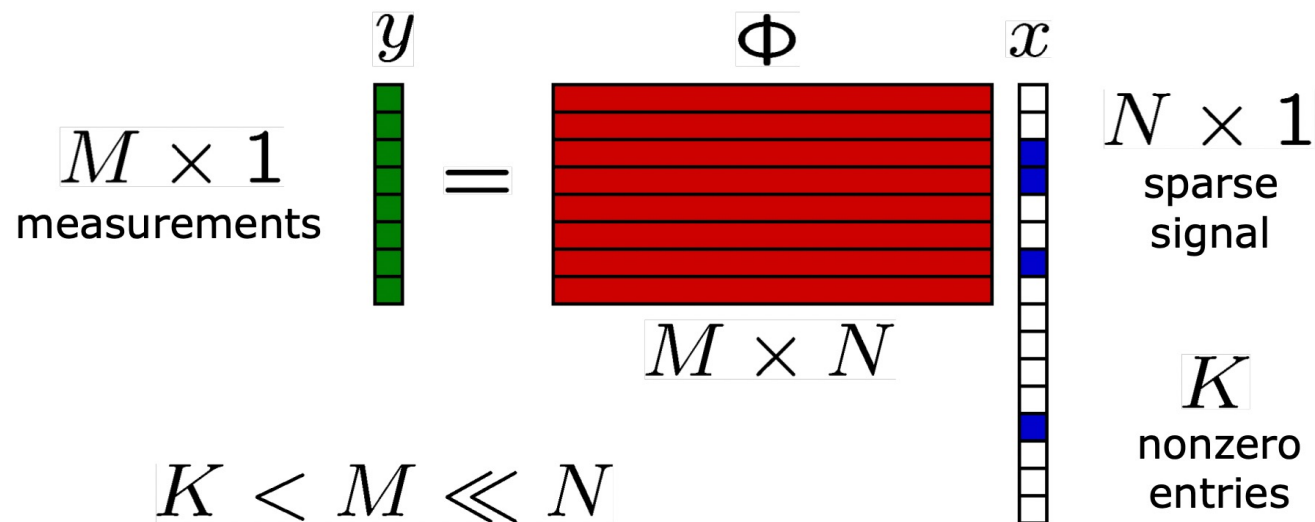
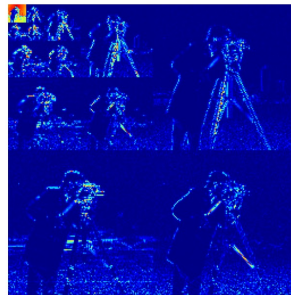


Image Representation



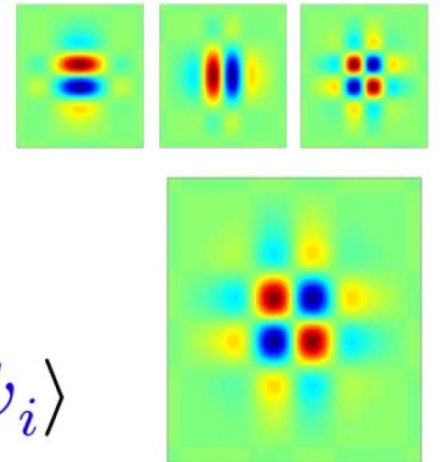
N
pixels

$K \ll N$
large
wavelet
coefficients



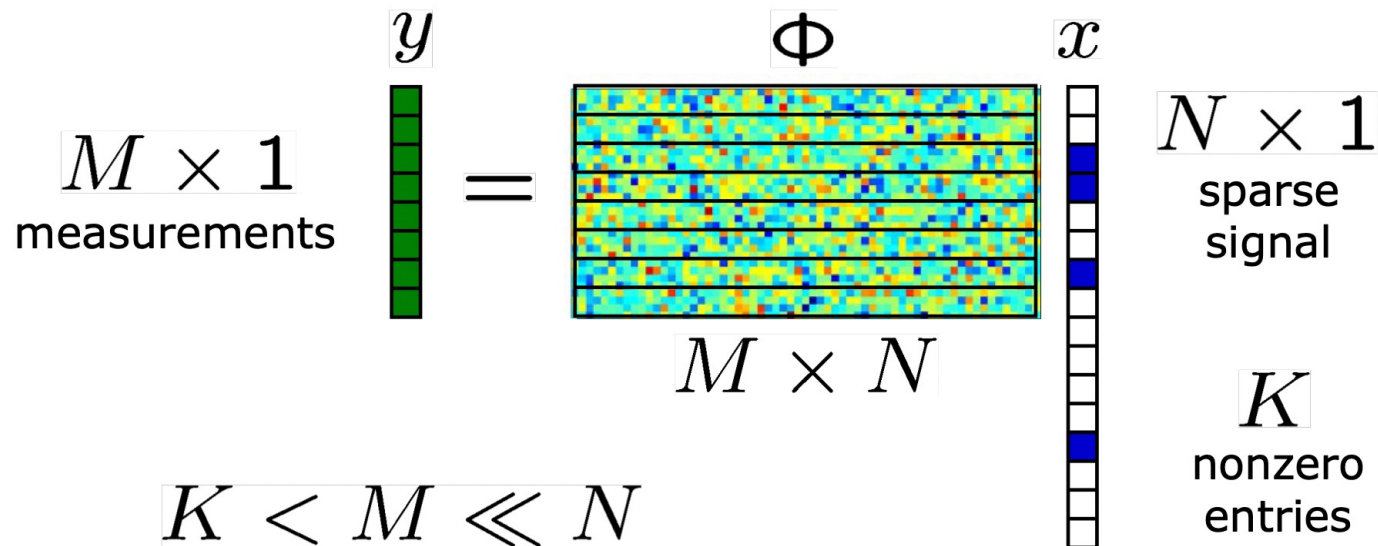
$$x = \sum_{i=1}^N \alpha_i \psi_i$$

$$\alpha_i = \langle x, \psi_i \rangle$$



CS Signal Recovery

- Reconstruction/decoding: given $y = \Phi x$
(ill-posed inverse problem) find x



- Reconstruction/decoding: given $y = \Phi x$
(ill-posed inverse problem) find x

- L_2 fast, wrong

$$\hat{x} = \arg \min_{y=\Phi x} \|x\|_2$$

- L_0 **correct, slow**
only $M=K+1$
measurements
required to
perfectly reconstruct
 K -sparse signal

$$\hat{x} = \arg \min_{y=\Phi x} \|x\|_0$$

*number of
nonzero
entries*



[Bresler; Wakin et al]

- Reconstruction/decoding: given $y = \Phi x$
(ill-posed inverse problem) find x

- L_2 fast, wrong

$$\hat{x} = \arg \min_{y=\Phi x} \|x\|_2$$

- L_0 correct, slow

$$\hat{x} = \arg \min_{y=\Phi x} \|x\|_0$$

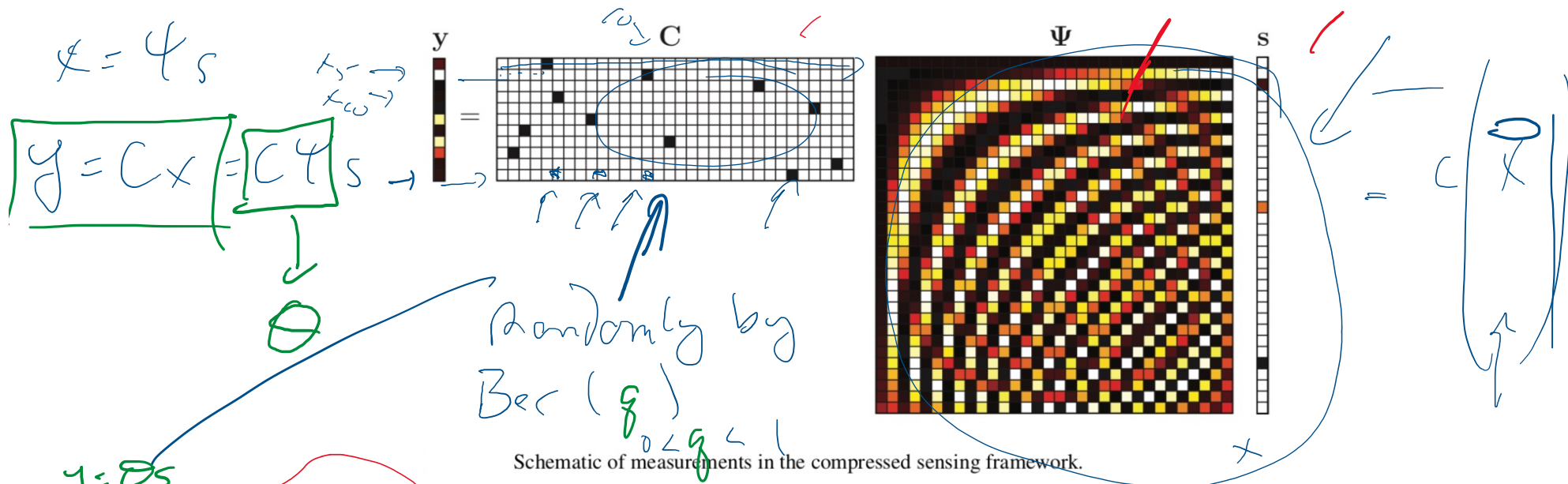
- L_1 **correct,
mild oversampling**

[Candes et al, Donoho]

$$\hat{x} = \arg \min_{y=\Phi x} \|x\|_1$$

linear program

$$M \approx K \log N \ll N$$



$y = \Theta s$

$\hat{s}_1, \hat{s}_2, \dots, \hat{s}_n$

$\arg \min \|s\|_1$

$y = \Theta s$

$\leftarrow$ Lasso Regression

(a) Sparse $s (\ell_1)$

$\text{vs. } J(s) = \|y - \Theta s\|_2^2 + \lambda \|s\|_1$

$\arg \min \|y - \Theta s\|_2^2$

vs.

(b) Least-squares $s (\ell_2)$

$\text{vs. } J(s) = \|y - \Theta s\|_2^2 + \lambda \|s\|_2^2$

Ridge Regress

Tikhonov Regular

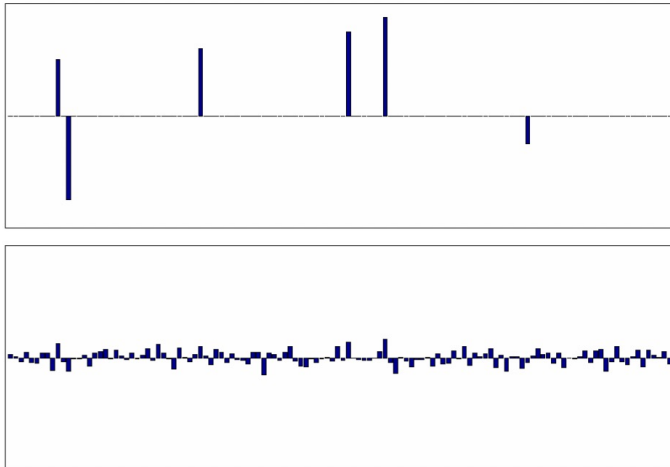
$\theta < 1$

CS Signal Recovery

- Reconstruction/decoding: given $y = \Phi x$
(ill-posed inverse problem) find x

- L_2 fast, **wrong**

$$\hat{x} = \arg \min_{y=\Phi x} \|x\|_2$$



x

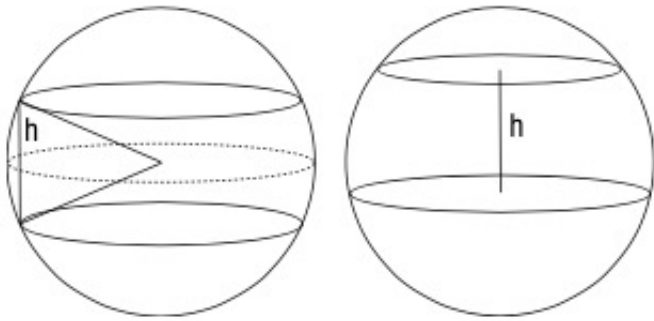
$$\hat{x} = (\Phi^T \Phi)^{-1} \Phi^T y$$

Another view on **where all the random goes** ...geometry of data in high dimensions

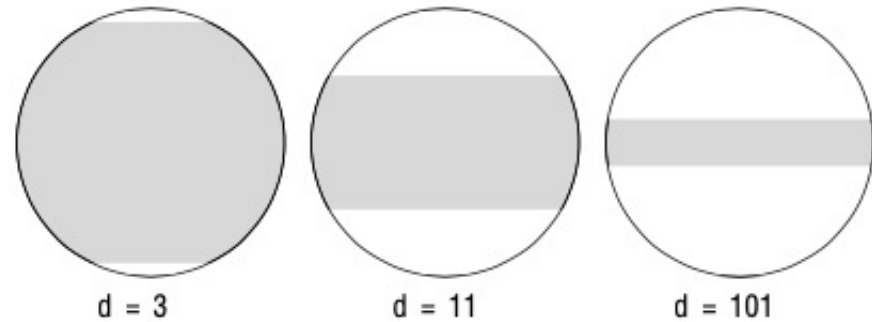
highly important topic in Machine Learning, Data Science, and ROM

-story one – random gets soaked up by time.

-story two – its about random projection (Johnson-Lindenstrass) which is an application of concentration of measure



a strip of width h



(width of the) strip around the equator that contains 90% of the area

Several different important concepts follow from this simple idea

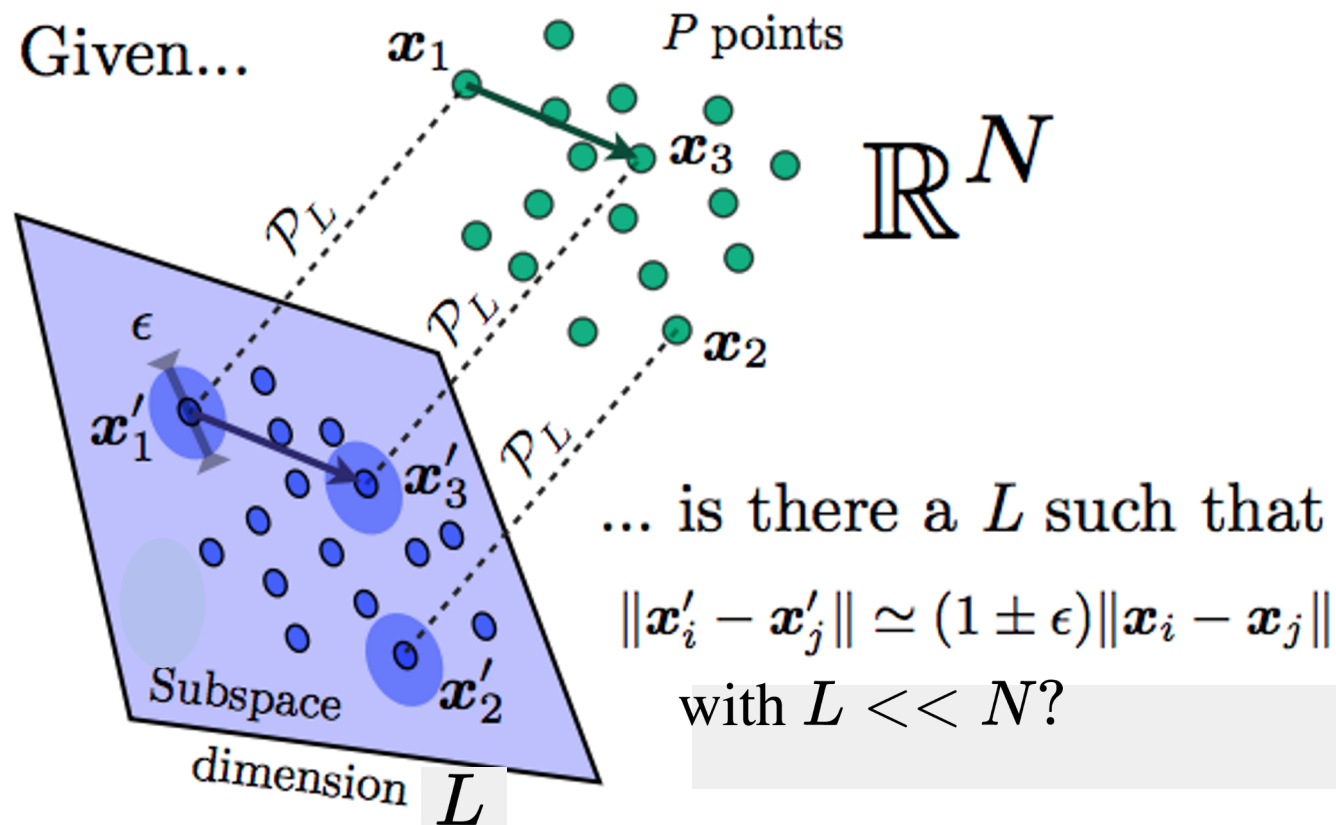
-Markov inequalities, Chernoff inequality

-for us – in high dimensions, samples of data “look” almost like they live in a “flatter” space – Johnson-Lindenstrass theorem

See Ortega for CoM/JL random projection interpretation of RC

Matoušek's book [Mat02]

Random Projection



Random Projection – on projection and approximate isometry

- The random projection method is based on the Johnson-Lindenstrauss lemma.

Theorem[Johnson-Lindenstrauss Lemma]:

For any $0 < \epsilon < 1$ and any integer $M > 1$, let L be a positive integer such that $L \geq L_0$ with $L_0 = \frac{C \ln M}{\epsilon^2}$, where C is a suitable constant ($C \approx 8$ in practice, $C = 2$ is good enough). Then for any set X of M data points in $\mathbb{R}^N$, there exists a map $f : \mathbb{R}^N \rightarrow \mathbb{R}^L$ such that

$$\text{for all } x_1, x_2 \in X, (1 - \epsilon) \|x_1 - x_2\|^2 \leq \|f(x_1) - f(x_2)\|^2 \leq (1 + \epsilon) \|x_1 - x_2\|^2.$$

[Johnson et. al, 1984]

Theorem[Random Projection]

For any $0 < \epsilon, \delta < \frac{1}{2}$ and positive integer N ,

there exists a random matrix of B of size $L \times N$ such that for $L \geq L_0$ with $L_0 = \frac{C \ln(1/\delta)}{\epsilon^2}$

and for any unit-length vector $x \in \mathbb{R}^N$, $Pr\{||Bx||^2 - 1 > \epsilon\} \leq \delta$

$$\text{or } Pr\{||Bx||^2 - 1 > \epsilon\} \leq e^{-CL\epsilon^2}$$

[Papadimitriou et. al., 1998]

