EngSci 721

Inverse Problems and Learning From Data

Oliver Maclaren (oliver.maclaren@auckland.ac.nz)

1. *Basic concepts* [5 lectures + 1 Tutorial]
   Forward vs inverse problems. Well-posed vs ill-posed problems. Algebra and calculus of inverse problems (left and right inverses, generalised and pseudo inverses, resolution operators, matrix calculus). Representing higher dimensional problems (image data etc).
2. *Instability and regularisation in linear and nonlinear problems* [6 lectures + 1 Tutorial]
   Instability and related issues for generalised inverses. Introduction to regularisation and trade-offs. Tikhonov regularisation. Higher-order Tikhonov regularisation. Sparsity and regularisation using different norms. Truncated singular value decomposition. Iterative regularisation, including stochastic/mini-batch gradient descent.
3. *Further topics* [3 lectures + 1 Tutorial]
   Regularisation parameter choice, including statistical and machine learning views of regularisation. Confidence sets for linear and nonlinear models. Physics-informed machine learning and neural networks.

Module overview

Inverse Problems and Learning From Data (*Oliver Maclaren*)
[~14 lectures/3 tutorials]

Lecture 9: Singular value decomposition

Topics:

- Singular Value Decomposition
  - The ‘crown jewel’ of linear algebra!
  - Generalises eigenvalue analysis to general (non-square etc) matrices
- Truncated Singular Value Decomposition
  - As regularisation scheme
  - Relation to Tikhonov regularisation

EngSci 721 : Lecture 9

Regularisation in linear problems:
Matrix factorisation & the (truncated) Singular Value Decomposition (SVD)

- Matrix factorisations in general
- SVD (type of matrix factorisation)
  - extension of eigen analysis
  - insight/calculation for inverses, resolution, effect of regularis.
  - the 'crown jewel' of linear algebra
- Truncated SVD.
  - as regularisation scheme.
  - connection to Tikhonov.

Bonus: rank factorisation & extension to nonlinear epi-mono factorisations

Matrix factorisations

Matrices play a central role in (linear) inverse problems, eg representing linear forward maps, multidimensional data, etc.

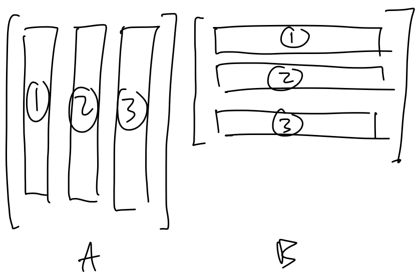
It is frequently useful to decompose or factorise a matrix A into two or more factors, eg

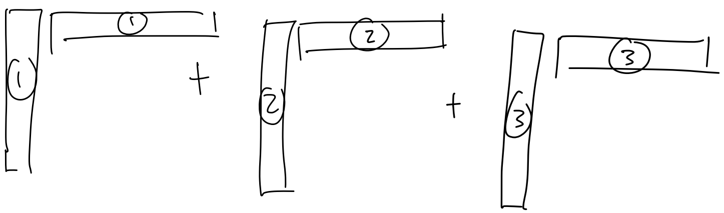
$$\boxed{A = BC} \text{ or } \boxed{A = BCD} \text{ etc.}$$

These are useful if B &/or C, D etc have particular structure that give us insight into A or allow us to operate with A, solve eq^{n}s involving A etc.

Outer products

Recall that we can think of an arbitrary dimensionally consistent matrix multiplication AB in terms of outer products, eg

$AB =$ 

$=$ 

$=$ sum of outer products:

$$\mathrm{col}_i(A)\,\mathrm{row}_i(B) = c_i(A)\, \tilde{r}_i^T(B)$$

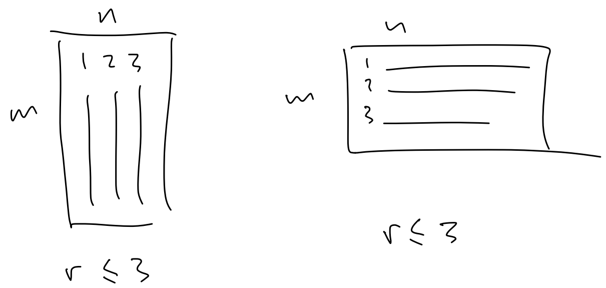
recall r^T & row convention.

Rank

The rank of a matrix is the number of linearly independent columns of the matrix

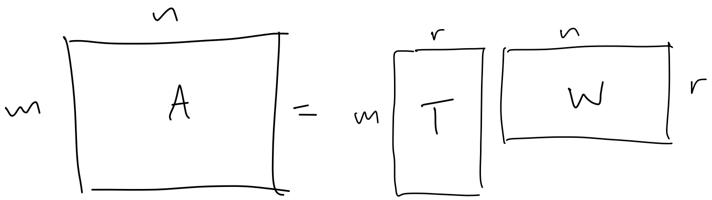
A fundamental theorem of linear algebra is that this is also equal to the number of linearly independent rows of the matrix

we have $\mathrm{rank}(A) \leq \min\{m, n\}$ for an $m \times n$ matrix A



Rank factorisation

A <u>rank factorisation</u> for a matrix A of <u>shape</u> $m\times n$ & <u>rank</u> r is a factorisation into an <u>$m\times r$ 'tall/square'</u> matrix T & an <u>$r\times n$ 'wide/square'</u> matrix W:



$$\Rightarrow \boxed{A = \sum_{i=1}^{r} t_i \underbrace{w_i^T}_{\text{rows of } W}} \quad \Big\} \text{ sum of outer products}$$

↳ <u>expansion</u> form

→ It turns out these <u>always</u> exist

→ are <u>not</u> unique: many useful <u>types</u> exist

→

Simple example

Suppose A has linearly indep. col.

eg $A = \begin{bmatrix} ① & ② & ③ \end{bmatrix}$ & all L.I.

obv. $A = AI$

$$= \underbrace{\begin{bmatrix} ① & ② & ③ \end{bmatrix}}_{T} \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{W}$$

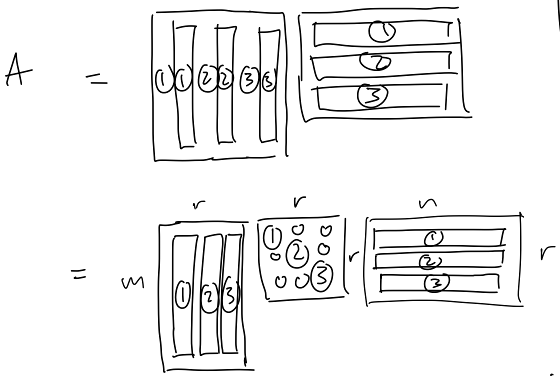
$$= \begin{bmatrix} ① & 0 & 0 \\ & 0 & 0 \\ & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & ② & 0 \\ 0 & & 0 \\ 0 & & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & ③ \\ 0 & 0 & \\ 0 & 0 & \end{bmatrix}$$

Suppose we rescale 'T' columns, $t_i = s_i \bar{t}_i$

get $\boxed{A = \sum_{i=1}^{r} t_i w_i^T = \sum_{i=1}^{r} s_i \bar{t}_i w_i^T}$

→

leads to eg



Note:

AS ← diag.
= scale A cols by corresp. diag.

diag → SA = scale rows of A by corresp. diag.

$\Rightarrow \boxed{A = TSW} \iff \boxed{A = \sum_i^r t_i s_i w_i^T}$ } expansion form

vector

'tall' 'scale' 'wide' scalar vector

More generally,

$\boxed{A = TMW}$ ← tho' no simple expansion version since M <u>not</u> diag.

'tall' 'mixing' 'wide'
(not diagonal, but <u>in</u>vertible)

[These ideas form foundation of
- singular value decomposition (SVD)
- CUR factorisation & matrix sketches.]

Note: <u>extending/padding</u> a rank factorisation

<u>Given:</u>

$$\underset{m\times n}{A} = \underset{m\times r}{T}\;\underset{r\times r}{S}\;\underset{r\times n}{W} \qquad (1)$$

<u>can</u> write as:

$$A = \underbrace{\begin{bmatrix} T & T_0 \end{bmatrix}}_{m\times m} \underbrace{\begin{bmatrix} S & 0 \\ 0 & 0 \end{bmatrix}}_{m\times n} \underbrace{\begin{bmatrix} W \\ W_0 \end{bmatrix}}_{n\times n} \qquad (2)$$

zeros

arbitrary.

arbitrary

---

call (1) 'reduced'
(2) 'full' or 'padded'

Side note:

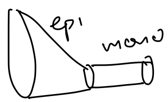
we will now consider various forms of these matrix/linear factorisations

→ First though, note that we can actually extend concept of $A = \text{Tall} \cdot \text{Wide}$ to nonlinear $f = \text{mono} \circ \text{epi}$!

monomorphism = 1-1 nonlinear function ~ 'tall, LI col' matrix

epimorphism = onto nonlinear function ~ 'wide, LI row' matrix

f:



→ I haven't seen much on this tho'!

↳ research topic ??

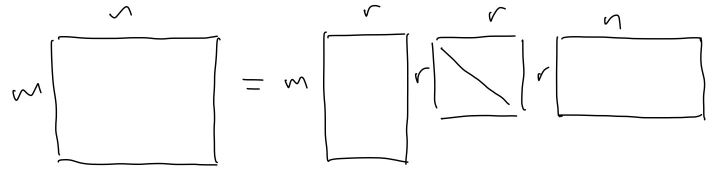
SVD (reduced)

The singular value decomposition (SVD) of an arbitrary $m \times n$ matrix A with rank r always exists and has the (reduced) form:

$$\boxed{A = U_r \Lambda_r V_r^T}$$

(some use Σ instead of Λ but we want sum!)

with shapes:



A arbitrary

U_r tall, cols are orthonormal & linearly indep.

Λ_r square & diagonal

V_r^T wide, rows are orthonormal & linearly indep.

Expansion form:

$$\boxed{A = \sum_{i=1}^{r} u_i \lambda_i v_i^T}$$

number

col vector

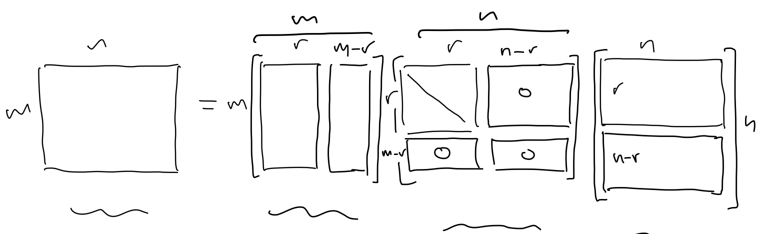
row vector

SVD (full - usually what people mean, tho' ref?.)

The 'full' SVD is a 'padded' version of the reduced SVD with form

$$\boxed{A = U \Lambda V^T}$$

with shapes:



$\underset{\sim}{A}$ arbitrary $m \times n$

$\underset{\sim}{U}$ square $m \times m$; cols are orthonormal & linearly indep. & form basis for $\mathbb{R}^m$

$\underset{\sim}{\Lambda}$ $m \times n$, zero padded

$\underset{\sim}{V^T}$ square $n \times n$, rows orthonormal & linearly indep. & form basis for $\mathbb{R}^n$

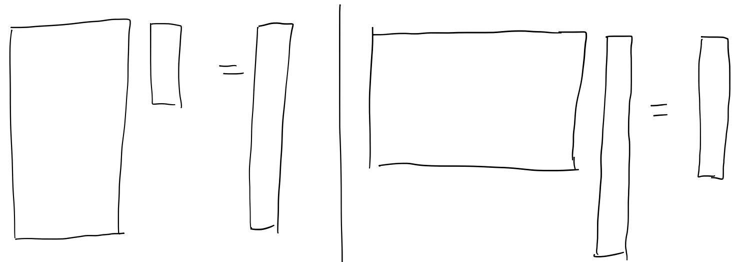
Origin of SVD?

→ generalisation of eigen-analysis to non-square matrices

Recall that for a square matrix A, eigenvalues solve

$$\boxed{Ax = \lambda x}$$

However, we are interested in non-square matrices in inverse problems (& statistics etc)!



Tall/overdetermined systems	Wide/underdetermined systems
– 'classical statistics'	– 'inverse problems' – 'nonparametric statistics – 'machine learning' etc

→ eigenvalues don't make sense for non-square!

Eigenvalues?

$$\left.\begin{array}{ll} A & m \times n \\ x & n \\ y & m \end{array}\right] \quad Ax = y$$

$Ax = \lambda x$ doesn't make sense

$$\left.\underbrace{Ax}_{\in \mathbb{R}^m} \quad \underbrace{\lambda x}_{\in \mathbb{R}^n}\right\}$$ live in different spaces!

Solutions?

related $\Big\{$
- Consider different bases for each space
- Consider eigenvalues of square matrices like A^TA & AA^T:

$$\circlearrowleft \; \underset{A^T}{\overset{A}{\rightleftarrows}} \; \circlearrowleft$$

| A^TA: $\mathbb{R}^n \to \mathbb{R}^n$ | AA^T: $\mathbb{R}^m \to \mathbb{R}^m$ |

Model space: A^TA

First consider A^TA, $A: m \times n$

$A^T: n \times m$

$A^TA: n \times n$

$$\circlearrowleft \; \underset{② A^+}{\overset{① A}{\rightleftarrows}}$$

$\mathbb{R}^n$

$A^TA : \mathbb{R}^n \to \mathbb{R}^n$

Properties of A^TA

- square: $n \times n$
- symmetric: $(A^TA)^T = A^TA$
- positive semi-definite: x^TA^TAx
  $= (Ax)^TAx = \|Ax\|^2 \geq 0$
- $\text{rank}(A^TA) = \text{rank}(A)$

Facts about symmetric / p.s.d matrices

1. If an $n \times n$ matrix has real entries & is symmetric then it has n distinct, real, linearly independent & orthogonal eigenvectors : basis for $\mathbb{R}^n$!

2. If an $n \times n$ matrix is symmetric & positive semi-definite then its eigenvalues are ≥ 0

3. The rank of a symmetric $n \times n$ matrix is equal to the number of non-zero (here, positive) eigenvalues

---

Imply: we can use normalised eigenvectors of A^TA as a basis for $\mathbb{R}^n$ (model space)

↳ solve $(A^TA)v_i = \lambda_i^2 v_i$ ($\lambda_i^2 \geq 0$)

↳ get n sol^ns $\{(\lambda_i, v_i)\}$ $i = 1, \dots, n$

↳ orthonormal basis $\{v_i\}$ $i = 1, \dots, n$

$$v_i^T v_j = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases}$$

Data space

Before considering $AA^T : \mathbb{R}^m \to \mathbb{R}^m$

lets consider the image of our $\mathbb{R}^n$ basis vectors, ie $Av_i \in \mathbb{R}^n$

Are these orthogonal?

$$(Av_i)^T(Av_j) = v_i^T A^TA v_j$$
$$= v_i^T \lambda_j^2 v_j \quad \text{eig.}$$
$$= \lambda_j^2 \underbrace{v_i^T v_j}_{i \neq j} \quad v \text{ orthog.}$$
$$= \begin{cases} \lambda_i^2 & i = j \\ 0 & i \neq 0 \end{cases}$$

Yes!

To make orthonormal, divide Av_i by $\lambda_i \neq 0$:

$$u_i = \frac{Av_i}{\lambda_i} \iff \boxed{Av_i = \lambda_i u_i}$$

for $i = 1, \dots, \text{rank}$

($\lambda_i > 0$ for $i = 1, \dots, r$)

Properties :

$$u_i = \frac{Av_i}{\lambda_i}, \quad \lambda_i > 0$$

Consider

$$AA^T u_i = AA^T \frac{Av_i}{\lambda_i}$$

$$= A\left(\frac{\lambda_i^2 v_i}{\lambda_i}\right) \quad [\text{eig } v]$$

$$= A\lambda_i v_i$$

$$= \lambda_i (Av_i) \quad [u_i \text{ def}^n]$$

$$= \lambda_i^2 u_i$$

$\Rightarrow$ u_i are eigenvectors of AA^T with eigenvalues λ_i^2 !

For $\lambda_i = 0$, still define u_i as eigenvector solving $A^TAu = \lambda^2 u = 0$

Get m solutions $\{(\lambda_i, u_i)\}$ $i = 1, \ldots, m$

$\{u_i\}$ basis for $\mathbb{R}^m$

Singular values & singular vectors I

Instead of $\boxed{Av = \lambda v}$ } eigenvector v, eig value λ

Consider $\boxed{Av = u\lambda}$

$$\underset{\text{matrix}}{A}\ \underset{\text{vector}}{v} = \underset{\text{vector}}{u}\ \underset{\text{scalar}}{\lambda}$$

where : $\boxed{\begin{array}{l} A^TAv = \lambda^2 v \\ AA^Tu = \lambda^2 u \end{array}}$ $\begin{array}{l} \} \; i = 1, \ldots n \\ \} \; i = 1, \ldots m \end{array}$

Sol^{n}s : For $\lambda_i > 0$, get triples solving all eq^{n}s,

$$\{(\underset{\substack{\text{left singular vector} \\ \text{(data space)}}}{u_i}, \underset{\text{singular value}}{\lambda_i}, \underset{\substack{\text{right singular vector} \\ \text{(model space)}}}{v_i})\} \quad i = 1, \ldots, r,$$

Also get extra u, v to 'fill out' $\mathbb{R}^m, \mathbb{R}^n$ corresponding to $\lambda = 0$ eig. values of A^TA &/or AA^T

Singular values & singular vectors II

In $Av_i = u_i \lambda_i$

ie $\{(u_i, \lambda_i, v_i)\}$ $i = 1, \ldots, \text{rank}$

we have:

$\{u_i\}$ are normalised eigenvectors of AA^T for $i = 1, \ldots \text{rank}$

$\{v_i\}$ are normalised eigenvectors of A^TA for $i = 1, \ldots \text{rank}$

Both sets are <u>orthonormal</u>

$$u_i^T u_j = \begin{cases} 0 & i \neq j \\ 1 & i = j \end{cases}$$

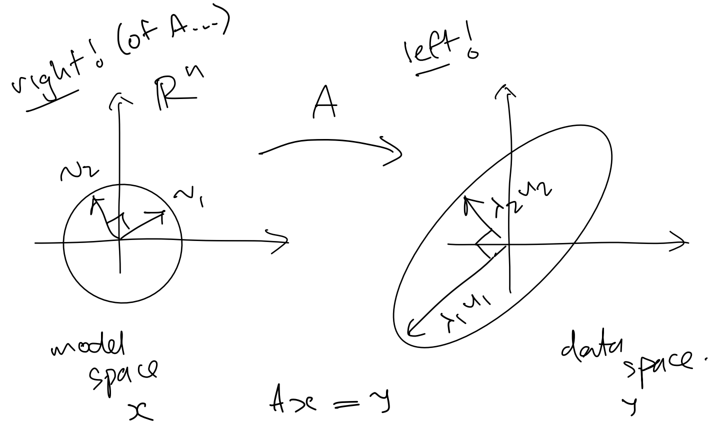
$$v_i^T v_j = \begin{cases} 0 & i \neq j \\ 1 & i = j \end{cases}$$

Our <u>triplets</u> <u>here</u> are defined for $\boxed{\lambda_i > 0}$ but $\{u_i\}$ for all eigenvectors of AA^T & $\{v_i\}$ for all eig. vectors <u>span</u> their respective spaces:

$$\text{Span}\{u_i\} = \mathbb{R}^m \quad \text{for } i = 1, \ldots, m$$
$$\text{span}\{v_i\} = \mathbb{R}^n \quad \text{for } i = 1, \ldots, n$$

Interpretation:

$$\boxed{Av_i = \lambda_i u_i}$$



Matrix form: reduced.

$A: m \overset{n}{\square}$

$$\boxed{Av_i = u_i \lambda_i}, \quad i = 1, \dots \text{rank}$$

$v_i \in \mathbb{R}^n$, $u_i \in \mathbb{R}^m$

$\Longleftrightarrow$

$$A \begin{bmatrix} v_1 & v_2 & \cdots & v_r \\ & & & \end{bmatrix} = \begin{bmatrix} u_1 & u_2 & \cdots & u_r \\ & & & \end{bmatrix} \begin{bmatrix} \lambda_1 & & [0] \\ & \ddots & \\ [0] & & \lambda_r \end{bmatrix}$$

$\Longleftrightarrow$

$$\boxed{AV_r = U_r \Lambda_r}, \quad V_r : n \times r, \quad U_r : m \times r, \quad \Lambda_r : r \times r$$

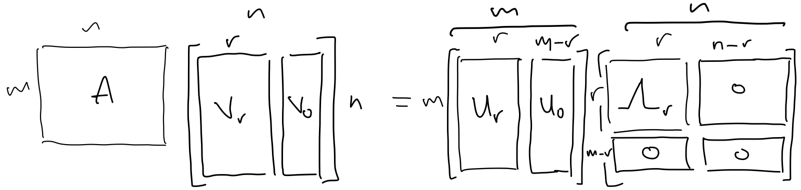
$\longrightarrow$

full/padded:

$m \overset{n}{\square}$

$$A \begin{bmatrix} v_1 & \cdots & v_n \\ & & \end{bmatrix} = \begin{bmatrix} u_1 & \cdots & u_m \\ & & \end{bmatrix} \begin{bmatrix} \begin{bmatrix} \lambda_1 & & \\ & \lambda_2 & \\ & & \ddots \; \lambda_r \end{bmatrix} & [0] \\ [0] & [0] \end{bmatrix}$$

ie



$\Longleftrightarrow$

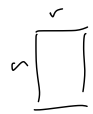
$$\boxed{AV = U\Lambda}$$

Note: 'inefficient' but $V^{-1} = V^T$, $U^{-1} = U^T$
because have full orthogonal basis

$$\Rightarrow AVV^T = U\Lambda V^T$$

$$\Rightarrow \boxed{A = U\Lambda V^T} \quad \text{(full SVD)}$$

Reduced SVD



It can be verified that

$$\boxed{A = U \Lambda V^T} \quad \text{full SVD of } A$$

expands to

$$\boxed{A = U_r \Lambda_r V_r^T} \quad \text{reduced SVD of } A$$

However, we don't have $V_r V_r^T = I$ in general (tho do have $V_r^T V_r = I$ in general) so can't so easily go from $AV_r = U_r \Lambda_r$ to

$A = U_r \Lambda V_r^T$

→ Can instead show eg $\boxed{AV_r V_r^T = A}$

even if $V_r V_r^T \neq I$!

SVD key properties I.

$A: \mathbb{R}^n \to \mathbb{R}^m$
$m \times n$

$A = U \Lambda V^T$ full

$A = U_r \Lambda_r V_r^T$ reduced.

where: $U^T U = I_m = U_r^T U_r$

$V^T V = I_n = V_r^T V_r$

$UU^T = I_m$, $U_r U_r^T \neq I$ in gen

$VV^T = I_n$, $V_r V_r^T \neq I$ in gen.

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

$= V_r \Lambda_r U_r^T \quad (\Lambda_r^T = \Lambda_r)$

$A^T A = V \Lambda^T \Lambda V^T$

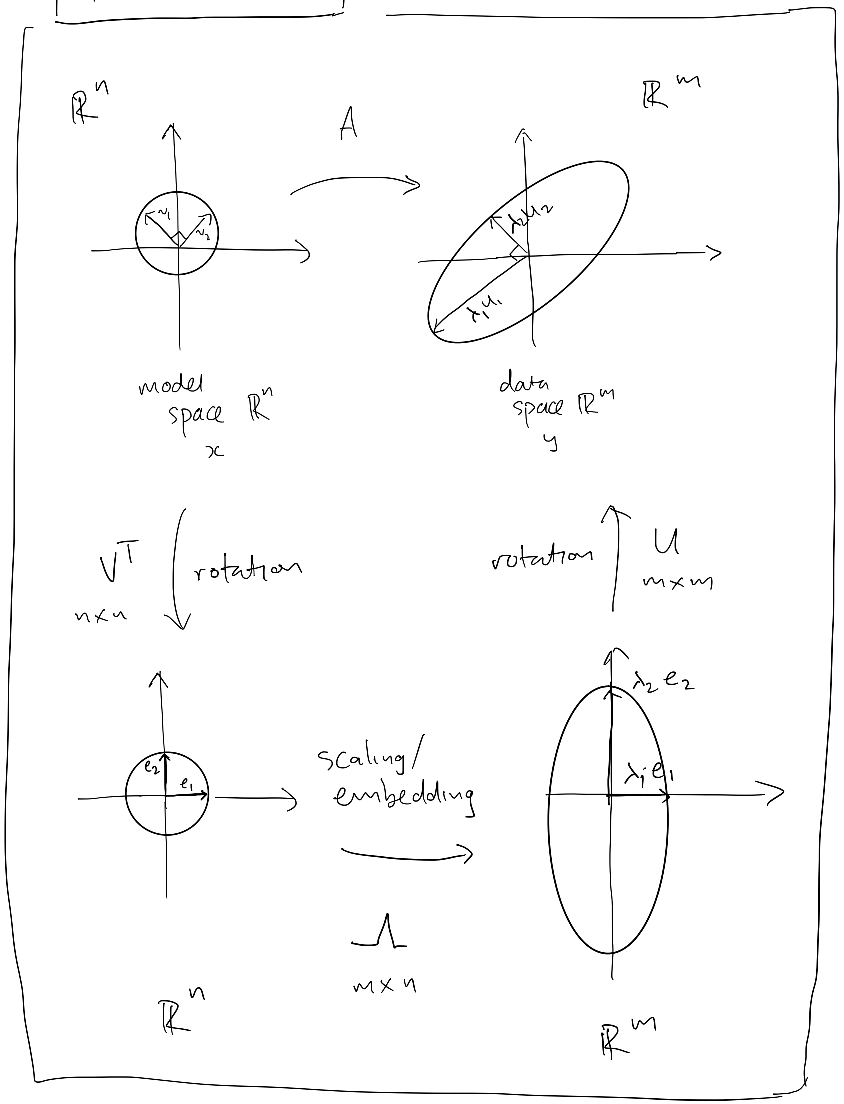
$= V_r \Lambda_r^2 V_r^T$

$AA^T = U \Lambda \Lambda^T U^T$

$= U_r \Lambda_r^2 U_r^T$

SVD : interpretation : U, V 'orthogonal' matrices = rotations!

$$\boxed{A = U \Lambda V^T} \quad \text{(full)}$$



SVD & inverses

Given:

$$A = U \Lambda V^T = U_r \Lambda_r V_r^T$$

The pseudo inverse is given by

$$A^+ = V \Lambda^+ U^T = V_r \Lambda_r^{-1} U_r^T$$

Pseudoinverse : $\boxed{A^+ = V_r \Lambda_r^{-1} U_r^T}$

- Check $AA^+A = A$?

$$(U_r \Lambda_r \underbrace{V_r^T)(V_r}_{I} \Lambda_r^{-1} \underbrace{U_r^T)(U_r}_{I} \Lambda_r V_r^T)$$

$$= U_r \Lambda_r \underbrace{\Lambda_r^{-1} \Lambda_r}_{I} V_r^T$$

$$= U_r \Lambda_r V_r^T = A \checkmark$$

- $(A^+A)^T = A^+A$?

$$A^+A = V_r \Lambda_r^{-1} \underbrace{U_r^T U_r}_{I} \Lambda_r V_r^T$$

$$= V_r V_r^T$$

$$A^T (A^+)^T = V_r \Lambda_r U_r^T U_r \Lambda_r^{-1} V_r^T$$

$$= V_r V_r^T \checkmark$$

etc!

Useful properties of $A = U_r \Lambda_r V_r^T$ } $V_r : n \times r$, $U_r : m \times r$

$n \times n$ { $\boxed{R_M = A^+A = V_r V_r^T}$ } projection on model space (row space of A)

$m \times m$ { $\boxed{R_D = AA^+ = U_r U_r^T}$ } projection on data space (col space of A)

$\boxed{A = A V_r V_r^T,}$ since $AA^+A = AR_M = AV_rV_r^T$

$\boxed{A = U_r U_r^T A,}$ since $AA^+A = R_D A = U_r U_r^T A$

& $\boxed{A = U_r U_r^T A V_r V_r^T}$ by above.

$\longrightarrow$

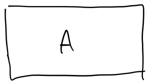
SVD & Resolution : Explicit cases

Now : $\boxed{R_D = U_r U_r^T}$ $m \times m$ $\begin{cases} U_r & m \times r \\ U_r^T & r \times m \end{cases}$ (r $m$vectors)

$\boxed{R_M = V_r V_r^T}$ $n \times n$ $\begin{cases} V_r & n \times r \\ V_r^T & r \times n \end{cases}$ (r $n$vectors)

- If rank $\underline{r = m < n}$

  $\Rightarrow R_D = I_m$

  (recover data exactly)

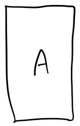
  (models are 'reduced')

  But $\underline{R_M \neq I_n}$

  Though $R_M^2 = V_r \underbrace{V_r^T V_r}_{} V_r^T = V_r V_r^T = R_M$

  $\Rightarrow$ <u>R_M is model projection operator</u>

- If rank $\underline{r = n < m}$

  (data are 'reduced')

  (models recovered exactly.)

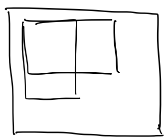
  $R_D \neq I_m$

  $\underline{R_M = I_n}$

  though $R_D = U_r \underbrace{U_r^T U_r}_{} U_r^T = U_r U_r^T = R_D$

  $\Rightarrow$ <u>R_D is data projection</u>

- if rank $\underline{r < m \,\&\, n}$

  $R_D \neq I_m$

  $\underline{R_M \neq I_n}$

  $\Rightarrow$ <u>Both</u> are projection operators.

  ($U_r^T U_r = I$, $V_r^T V_r = I$ still)

SVD : Big picture

Key advantage : $\boxed{\text{explicit calculation}}$

- inverses (left, right, pseudo)
- model/data resolution operators

coming up. $\begin{cases} \text{– stability/instability depending on singular values} \\ \text{– stablised approximations via truncation} \\ \text{– effect of Tikhonov (etc) regularisation on singular values} \end{cases}$

<u>Disadvantage</u> : though some intutions transfer to nonlinear, essentially a <u>linear</u> concept.

↳ But see <u>rank factorisation</u> & mono-<u>epi</u> factorisation

↳ can apply to eg linearisation

So.... regularisation!

→ singular values may be positive but effectively zero (machine tol. etc)

⇒ cause: $\boxed{\text{effective rank } p < \text{rank } r}$

→ small singular values cause instability

Key: inverse leads to $\boxed{\text{dividing by small } \lambda_i \text{ values}}$

SVD as basis expansion:

$$A^\dagger = V_r \Lambda_r^{-1} U_r^T = \sum_{i=1}^{r} v_i \frac{1}{\lambda_i} u_i^T$$

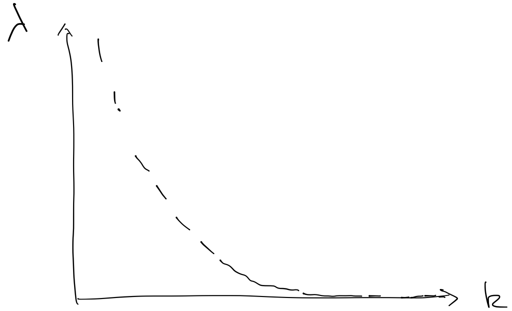
$$\& \quad x^\dagger = A^\dagger y = \left(\sum_{i=1}^{r} v_i \frac{1}{\lambda_i} u_i^T \right) y$$

$$= \sum_{i=1}^{r} \underbrace{\left(\frac{u_i^T y}{\lambda_i} \right)}_{\text{coeff.}} \underbrace{v_i}_{\text{basis vector in model space}}$$

coeff. ↙ Large for $\lambda_i \to 0$

Spectrum

Plot of singular values in decreasing order:



Key: $\boxed{\text{ill-posed}}$

→ no clear gap

→ decreases to zero

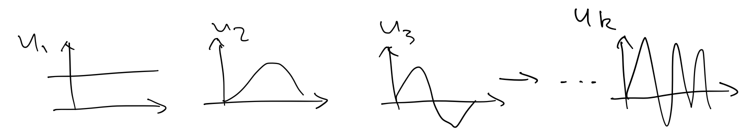
} $\boxed{\text{rank is hard to define}}$

$\boxed{\text{cf: rank deficient: clear gap. } A^\dagger \text{ OK then?}}$

$\boxed{\text{Also}}$: singular $\boxed{\text{vectors}}$ typically $\boxed{\text{'oscillate'}}$ more (sign changes in elements) for smaller values for ill-posed problems

↳ Like Fourier bases (see example)

note: ※ sign changes {



Truncate ~ low pass filter, but basis determined by A.

Truncated SVD:

$$A_k^+ = V_k \Lambda_k^{-1} U_k^T$$ } Truncated SVD inverse

→ remove small singular values/ oscillatory basis vectors

→ a form of regularisation/ stabilisation

Resolution?

$$R_M = A_k^+ A$$

$$R_D = A A_k^+$$

$$\xrightarrow{A}$$
$$\xleftarrow[A_k^+]{}$$

Stability

consider $x^+ = A^+ y$

& $x^{+\prime} = A^+ y'$

for small data perturbation

$$\| y - y' \|_2 < \delta$$

Then $x^+ - x^{+\prime} = A^+ (y - y')$

& $\| x^+ - x^{+\prime} \|_2 \leq \| A^+ \|_2 \, \| y - y' \|_2$

$$\boxed{\text{where } \| A \|_2 := \max_{\|x\|_2 = 1} \| A x \|_2 = \lambda_1 = \text{largest singular value}}$$

Leads to (with other details---)

$$\boxed{\frac{\| x^+ - x^{+\prime} \|_2}{\| x^+ \|_2} \leq \frac{\lambda_1}{\lambda_r} \frac{\| y - y' \|_2}{\| y \|_2}}$$

λ_1: largest singular value

λ_r: smallest singular value.

Stability : Key point

Stability (continuity modulus) of A^+ governed by

$$\boxed{\mathrm{cond}(A^+) = \frac{\lambda_1}{\lambda_r}}$$

(condition number)

Key trade-off :

truncate singular value expansion

↳ more stable (less 'variance')

↳ biased (model resolution less like identity

→ favour particular models)

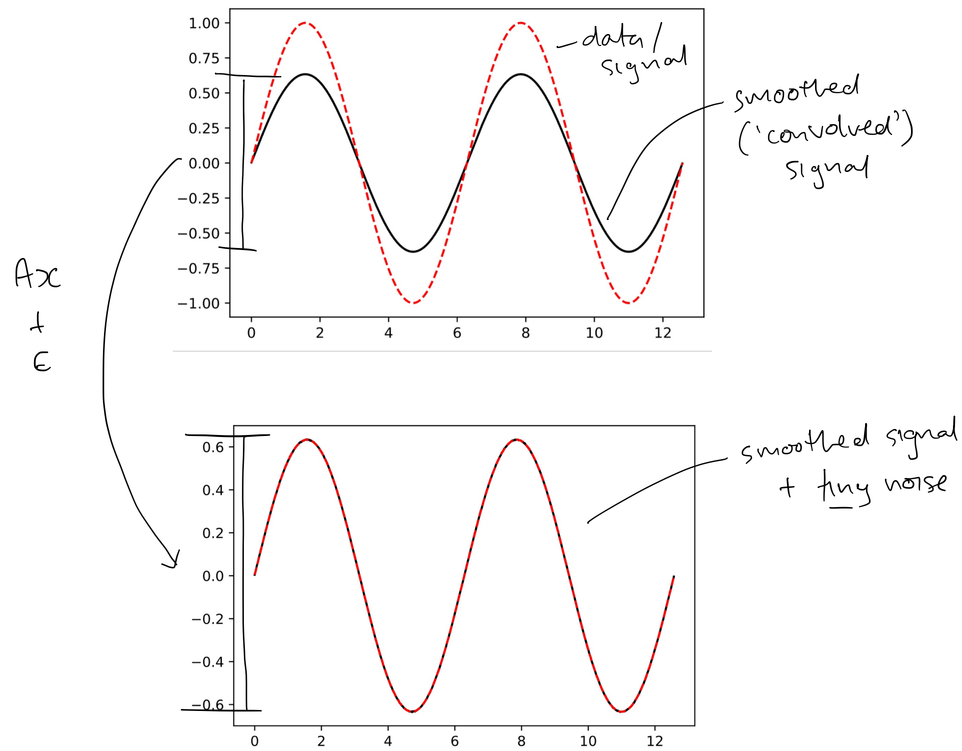
(stats : Bias - Variance tradeoff)

Example

Return to deconvolution example from L1.

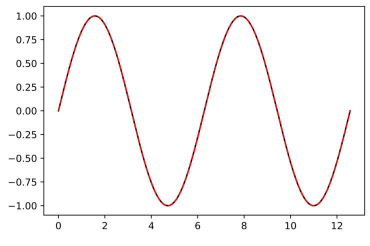
(convolution ≈ window averaging

deconvolution ≈ undoing ↑ !)



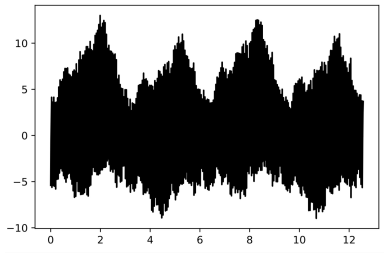
Example

Deconvolution no noise



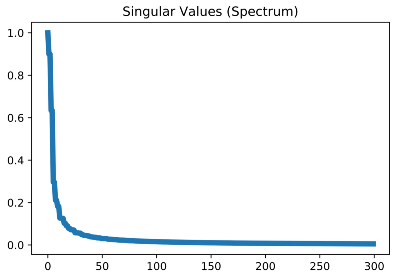
yay!

Deconvolution with noise

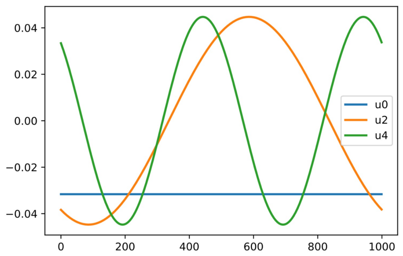


noo!

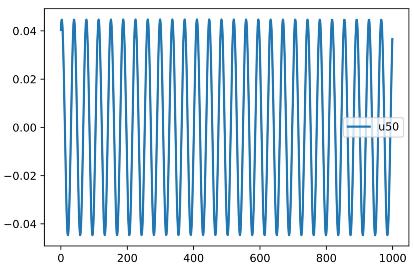
SVD: spectrum



u vectors (v similarly)



u vector for small λ_i:

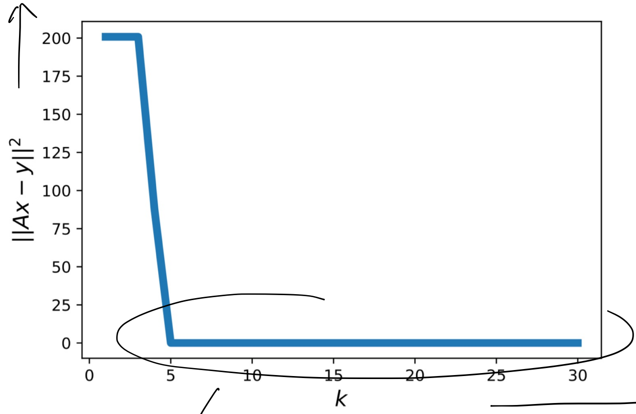


Think: Fourier components.

('but 'tailored' to A)

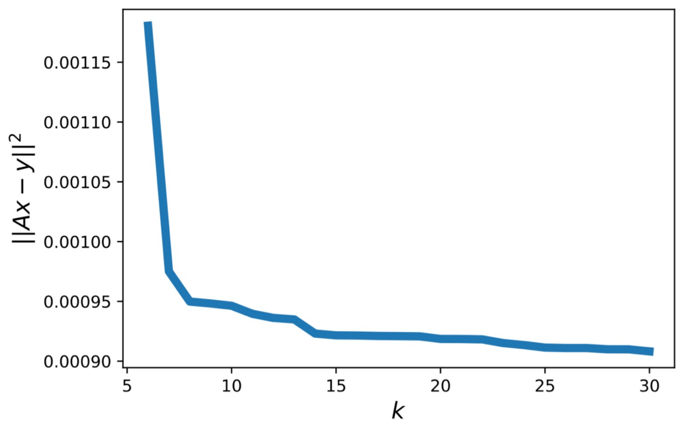
Pareto (trade-off) curve:

Fit



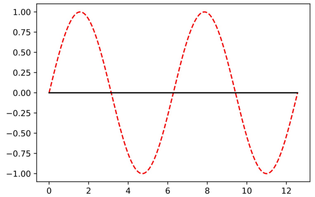
(number of retained singular values)

zoom



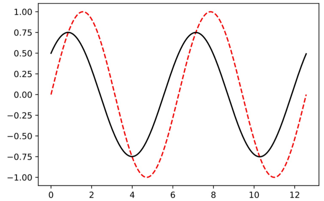
Solutions as depending on k (number retained singular values)

Stable / Biased

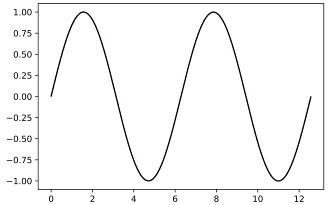


k = 1

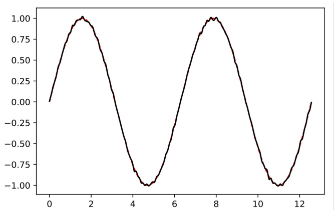
— recovered
- - - true



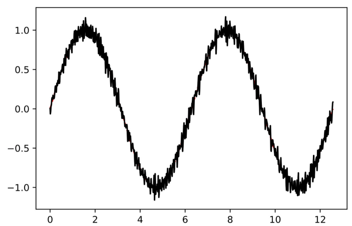
k = 4



k = 5 ← sweet spot



k = 100



k = 300

(this is a very smooth problem ... typically much worse, faster)

Unstable / Unbiased

Choosing truncation?

- Pareto: Smallest number of singular values giving adequate fit, beyond which 'flattens' — 'L-curve'-like
- Picard condition

Consider series rep. of SVD:

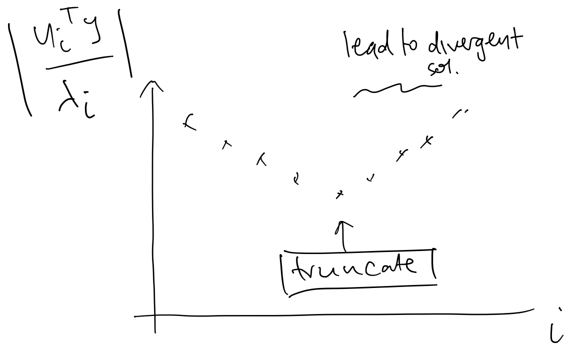
$$x = \sum_{i=1}^{r} \left[\underbrace{\left(\frac{u_i^T y}{\lambda_i} \right)}_{\text{coeff.}} \underbrace{v_i}_{\text{model basis}} \right]$$

series converge? ⇒ need coeff → 0!

Plot $\left| \frac{u_i^T y}{\lambda_i} \right|$ vs i [exercise: do for convolution!]

Expect: $u_i^T y$ decay faster than λ_i at first, then λ_i faster & ratio starts to increase

→ truncate here



Note: relative decay is what matters, not abs. magnitude

Note:

TSUD & optimality (Eckart-Young)

- The optimal rank k approx. to a matrix A in L_2 norm is given by the rank k truncation of the SVD

$$\Rightarrow \underset{\{\hat{A} \mid \hat{A} \text{ rank } k\}}{\operatorname{argmin}} \| A - \hat{A} \|_2 = U_k \Lambda_k V_k^T$$

= rank k SVD

= rank k truncation of SVD

Tikhonov & SVD

Finally, lets return to Tikhonov regularisation & see if SVD can help understand.

Zeroth order: Normal eq^ns

instead of λ to simplify

$$\boxed{(A^T A + \alpha^2 I) x = A^T y}$$

Consider SVD of original & augmented..

$\longrightarrow$

... Tikhonov & SVD ... Spectral filtering.

can show

$$x = \sum_{i=1}^{r} \underbrace{\left(\frac{u_i^T y}{\lambda_i}\right)}_{\text{coeff.}} \underbrace{v_i}_{\text{model basis}} \qquad \text{(SVD)}$$

becomes for Tikhonov regularised:

$$\boxed{x_\alpha = \sum_{i=1}^{r} \left(f_i \cdot \frac{u_i^T y}{\lambda_i}\right) v_i}$$

} general form of 'spectral filtering' methods

where

$$\boxed{f_i = \frac{\lambda_i^2}{\lambda_i^2 + \alpha^2}}$$

are the $\boxed{\text{filter factors}}$

Note:

$$\left(\begin{array}{l} \alpha = 0 \Rightarrow f_i = 1 \\ \lambda_i \ll \alpha \Rightarrow f_i \to \left(\frac{\lambda_i}{\alpha}\right)^2 \to 0 \\ \lambda_i \gg \alpha \Rightarrow f_i \to 1 \end{array}\right)$$

$\Rightarrow$ Tikhonov regularisation implements (continuous version of) truncated SVD !

$\to$ diff. f_i give diff. spectral regularisations! Eg TSVD: $f_i = \begin{cases} 1 & i \leq k \\ 0 & i > k \end{cases}$

Exercises

1. Compare & contrast the standard eigenvalue problem for a matrix A with the problem giving the singular values/vectors of A

2. Write down the SVD in the forms
   - full
   - reduced
   - series expansion

3. Write down the expression for A^+ given the SVD of A.

4. Verify A^+ is a left/right inverse when one exists, using the SVD representation

5. Write down expressions for R_D & R_M resolution matrices in terms of SVD components

Hints

4. Left inverse exists for tall A with linearly independent cols.

Consider $A = U_r \Lambda_r V_r^T$

$A^+ = V_r \Lambda_r^{-1} U_r^T$

$\Rightarrow \quad A^+A = V_r V_r^T$

But A: $\underbrace{m \times n}_{n=r}$ } linearly indep. cols.

full V = basis vectors for $\mathbb{R}^n$ (model space)

= basis for $\mathbb{R}^r$ ($r = n$ as lin. indep)

= V_r

full V always satisfies $VV^T = I$ (orthog. basis)

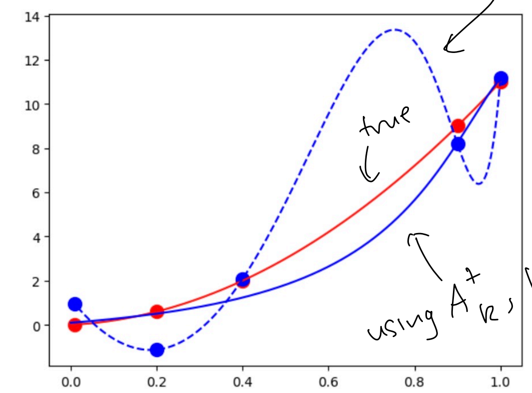
Here $V_r = V \Rightarrow A^+A = V_r V_r^T = VV^T = I$

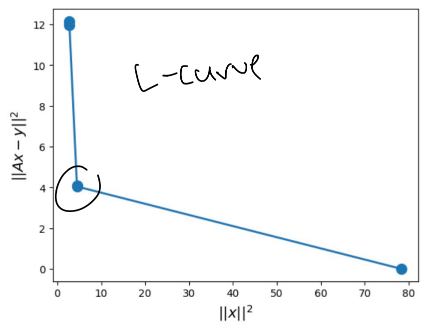
$\Rightarrow A^+$ is left inverse

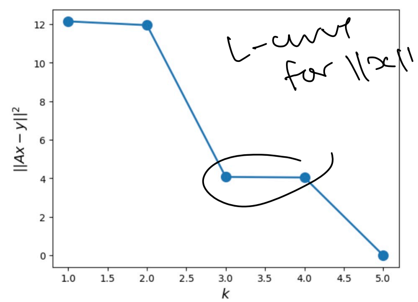
ie left inverse $\Leftrightarrow V_r = V$

$\Leftrightarrow$ linearly indep. cols

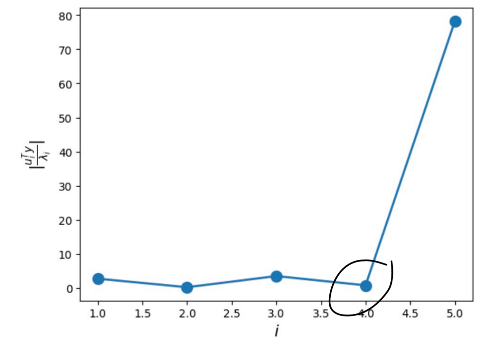
SVD: regression example cont'd (x^{19})







Picard:



Model resolution

A^+ $\qquad$ A_q^+