

SVD: Splitting the matrix A into
the product of three matrices

$$A = U W V^T$$

Matrix $M \times N$ U $N \times N$
 diagonal $N \times N$ V^T

U and V are orthogonal matrices:

$$\begin{aligned} U^T U &= I \\ V^T V &= I \end{aligned} \quad \Rightarrow$$

All singularities are
 collected in the diagonal
 matrix W .

SVD is always possible and is
unique. (except for permutations
 of columns in U and rows in V^T)

If $M < N$, $w_j = 0$ for $j = M+1, \dots, N$.

Suppose in the following that $M = N$:

$$A = UWV^T \Rightarrow$$

$$\begin{aligned} A^{-1} &= V W^{-T} U^T \\ &= \text{diag} \left(\frac{1}{w_i} \right) \end{aligned}$$

This gives trouble if $w_i = 0$
— or close to zero.

Condition number:

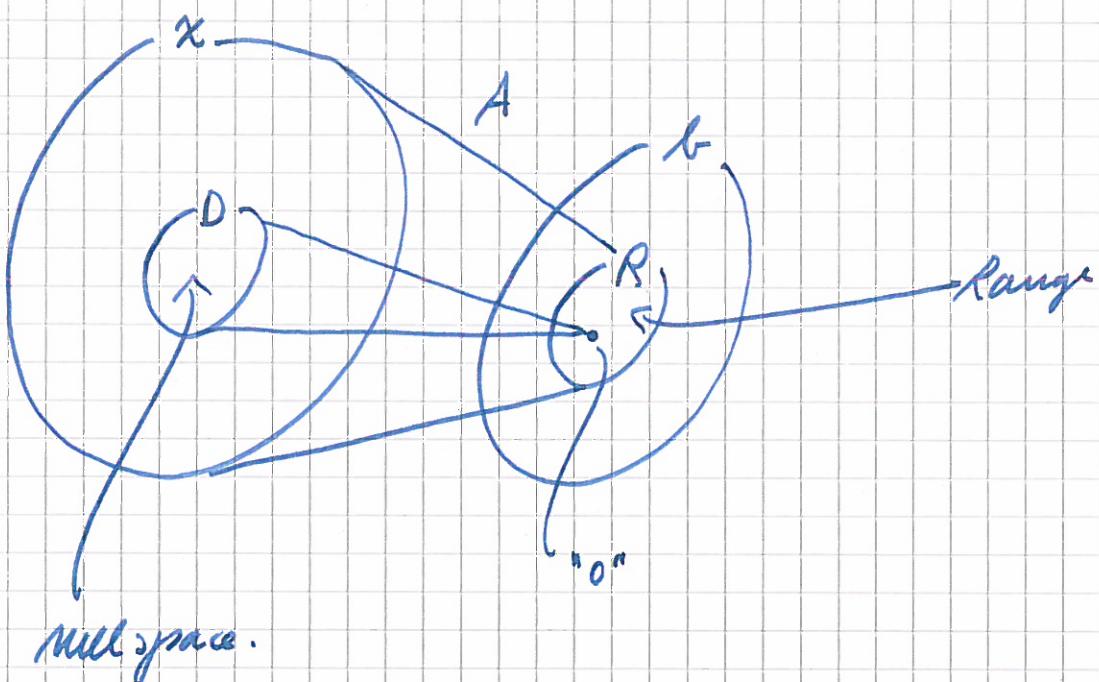
$$\boxed{\frac{\max_i |w_i|}{\min_i |w_i|}}$$

determines how singular A is.

$\sim 10^6$ for single precision (32 bits)

$\sim 10^{12}$ for double precision (64 bits)

When the condition number is of this order, the matrix is singular.



Dimension of nullspace = multiplicity = D_0

Dimension of range = rank = D_R

If A is not singular, $D_R = N$.

To

$$\text{nullity} + \text{rank} = N$$

SVD constructs explicitly an orthonormal basis for nullspace and range.

Columns of U corresponding to $w_i \neq 0$ span the range.

Columns of V corresponding to $w_i = 0$ span nullspace.

If A is singular,

$$x = x_p + \sum_{i=1}^{D_0} c_i x_i^0$$

($Ax = b$ is not unique,
see page 29.)

Find the x such that $Ax = b$
has the smallest norm.

If $w_i = 0$, substitute $1/w_i$ by 0.

Then
$$x = \sqrt{\text{diag}(1/w_i)} u^T b.$$

Eigenvalue Problem

$$A a_i = \lambda_i a_i \quad i=1, \dots, N$$

Orthonormal λ_i, a_i .

General advice: Use library routines.

But, if only one λ_i is to be determined
 - or just a few - including corresponding
 eigenvectors, one may use simple
iterative methods.

Largest eigenvalue and
 corresponding eigenvector.

A has real eigenvalues.

Let λ_m be largest eigenvalue.

Horde:

$$\begin{array}{|c|} \hline X'_k = AX_{k-1} \\ X_k = X'_k / \|X'_k\| \\ \hline \end{array}$$

$$X_{k-1} = \sum_{i=1}^N C_{k-1}^{(i)} q_i$$

$$x_k' = Ax_{k-1} = \sum_{i=1}^N c_{k-1}^i \lambda_i a_i$$

$$x_k = \frac{x_k'}{\|x_k'\|} = \sum_{i=1}^N \frac{c_{k-1}^i \lambda_i}{\left(\sum_{j=1}^N (c_{k-1}^j \lambda_j)^2 \right)^{1/2}} q_i$$

$$= \sum_{i=1}^N c_k^i a_i$$

$$c_k^i = \frac{c_{k-1}^i \lambda_i}{\left(\sum_{j=1}^N (c_{k-1}^j \lambda_j)^2 \right)^{1/2}} \rightarrow \begin{cases} 1 & \text{if } i = m \\ 0 & \text{otherwise.} \end{cases}$$

When

$$A_m = x_{k-1} \cdot x_k'$$

$$x_k \rightarrow a_m.$$

Determining any eigenvalue and corresponding eigenvectors.

A has no eigenvalues.

$$(A - \lambda I)^{-1} a_j = \frac{1}{\lambda_j - \lambda} a_j$$

unit matrix

maximum value
projected out.

$$\boxed{x_k' = (A - \lambda I)^{-1} x_{k-1}}$$

$$x_k = x_k' / \|x_k'\|$$

Converges to the eigenvalue closest to λ
in value.

$$\lambda_j = \lambda + \frac{1}{x_{k-1} \cdot x_k'}$$

Lambert - Weaire algorithm

Lambert and Weaire, Phys. Stat. Solidi B,
101, 591 (1980).

Builm and Lambert, Europhys. Lett. 5, 461 (1988).

This is a little-known but efficient algorithm to count the number of eigenvalues being λ smaller than some specified value λ .

A is assumed to have real eigenvalues.

This makes it possible to construct a histogram of the eigenvalues.

The problem is again

$$A \mathbf{a}_i = \lambda_i \mathbf{a}_i \\ i=1, \dots, N.$$

The aim is to eliminate one by one the N directions of the original problem, by transforming

the $N \times N$ matrix A into an $(N-1) \times (N-1)$ matrix A' having

λ_i , $i=1, \dots, N-1$ eigenvalues, all equal to the eigenvalues of the original matrix A .

To clarify. Let $N=2$.

$$\begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \end{pmatrix} = \lambda \begin{pmatrix} a_1 \\ a_2 \end{pmatrix} \Rightarrow$$

$$(A_{11} - \lambda)a_1 + A_{12}a_2 = 0 \quad (1)$$

$$A_{21}a_1 + (A_{22} - \lambda)a_2 = 0 \quad (2)$$

We eliminate a_2 from this equation set, and we are left with

$$(A_{11} - \lambda)a_1 - \frac{A_{12} A_{21}}{A_{22} - \lambda} a_1 = 0$$

(Subtracting (2) $\cdot \frac{A_{12}}{A_{22} - \lambda}$ from (1)).

We rewrite this as:

$$\underbrace{\left(A_{11} - \frac{A_{12} A_{21}}{A_{22} - \lambda} \right) a_1}_{\text{This is now a } 1 \times 1 \text{ matrix and}} = \lambda a_1$$

This is now a 1×1 matrix and
 λ in this case corresponds to one
of the original eigenvalues.

$$\underbrace{A_{11}' a_1 = \lambda a_1; A_{11}' = A_{11} - \frac{A_{12} A_{21}}{A_{22} - \lambda}}$$

In the case of an $N \times N$ matrix,
the eigenvalue equation reads

$$\sum_{j=1}^N A_{ij} a_j = \lambda a_i$$

We eliminate a_N and find

$$\sum_{j=1}^{N-1} A'_{ij} a_j = \lambda a_i$$

where

$$A'_{ij} = A_{ij} - \frac{A_{iN} A_{Nj}}{A_{NN} - \lambda}$$

We do this recursively:

$$A_{ij}^{(k)} = A_{ij}^{(k-1)} - \frac{A_{i,N-k+1}^{(k-1)} A_{N-k+1,j}^{(k-1)}}{A_{N-k+1,N-k+1}^{(k-1)} - \lambda} \quad (\#)$$

If we now choose a value λ ,

the negative eigenvalue theorem tells us that

The number of eigenvalues λ_i less than d is equal to the number of times the denominator in $(*)$, $(A_{N-k+1, N-k+1}^{(k-1)} - \lambda)$, has been negative for $k=1, \dots, N$.

This is a simple, graceful and largely forgotten algorithm!

Analysis: From derivatives to finite differences.

A tedious but important chapter: How to approximate derivatives $\frac{df^k}{dx^k}$ on the computer when we only know f