

The General Linear Model II

Olaf Hauk

MRC Cognition and Brain Sciences Unit

olaf.hauk@mrc-cbu.cam.ac.uk

We are still trying to solve linear equations

$$\mathbf{y} = \begin{pmatrix} y_1 \\ \dots \\ y_j \\ \dots \\ y_C \end{pmatrix} = \mathbf{M}\mathbf{x} = \begin{pmatrix} M_{11} & \dots & M_{1j} & \dots & M_{1C} \\ \dots & \dots & \dots & \dots & \dots \\ M_{i1} & \dots & M_{ij} & \dots & M_{iC} \\ \dots & \dots & \dots & \dots & \dots \\ M_{R1} & \dots & M_{Rj} & \dots & M_{RC} \end{pmatrix} \begin{pmatrix} x_1 \\ \dots \\ x_j \\ \dots \\ x_C \end{pmatrix} = \begin{pmatrix} \sum_{j=1}^C x_j * M_{1j} \\ \dots \\ \sum_{j=1}^C x_j * M_{ij} \\ \dots \\ \sum_{j=1}^C x_j * M_{Rj} \end{pmatrix} = \sum_{j=1}^C x_j * \begin{pmatrix} M_{1j} \\ \dots \\ M_{ij} \\ \dots \\ M_{Rj} \end{pmatrix} = \sum_{j=1}^C x_j \mathbf{M}_{.j}$$

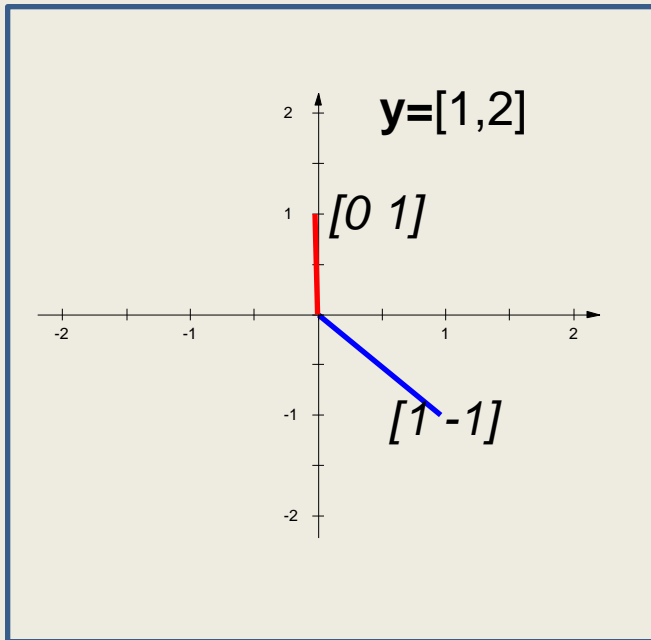
$\mathbf{M}_{.j}$ stands for the j -th column of \mathbf{M} .

$$\begin{array}{l} 1*a + 0*b = 1 \\ -1*a + 1*b = 2 \end{array} \quad \longrightarrow \quad a * \begin{pmatrix} 1 \\ -1 \end{pmatrix} + b * \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ -1 & 1 \end{pmatrix} * \begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} 1 \\ 2 \end{pmatrix}$$

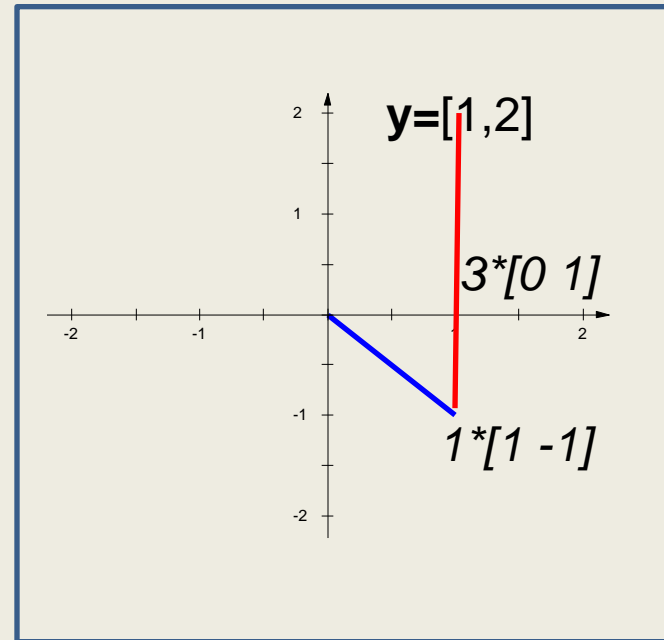
$$\mathbf{M}\mathbf{x} = \mathbf{y}$$

Geometric interpretation of linear equations

Problem



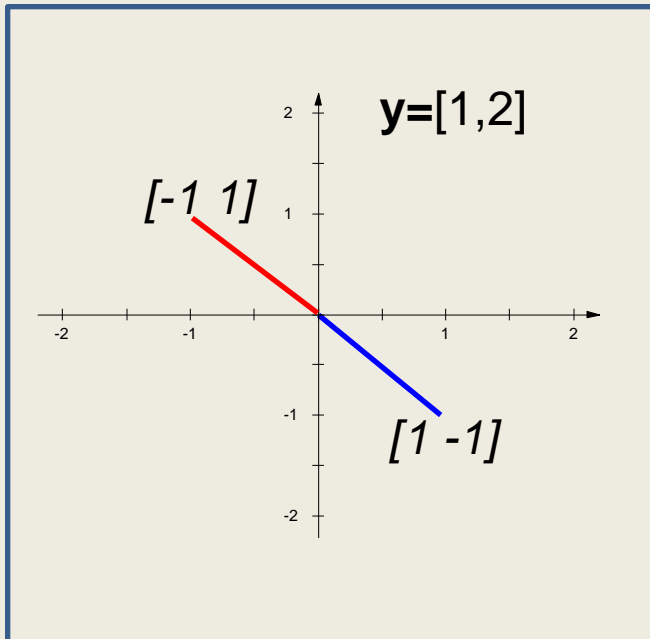
Solution



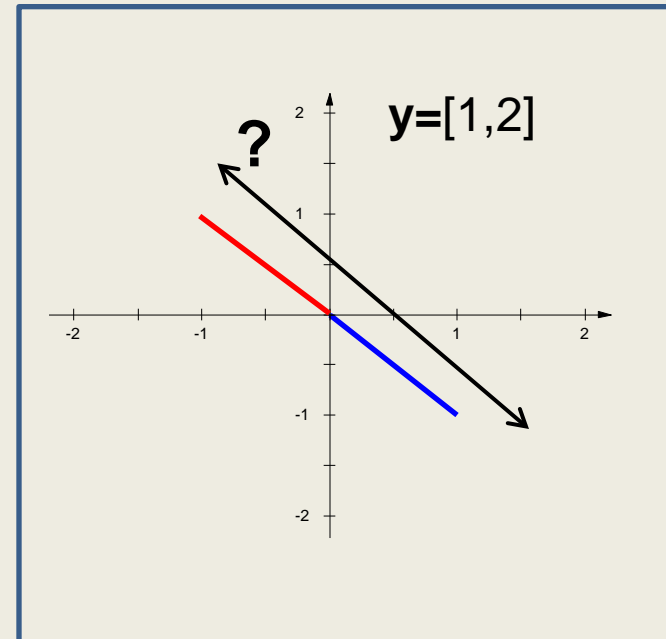
$$a * \begin{pmatrix} 1 \\ -1 \end{pmatrix} + b * \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ -1 & 1 \end{pmatrix} * \begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} 1 \\ 2 \end{pmatrix}$$

Geometric interpretation of linear equations

Problem



No Solution

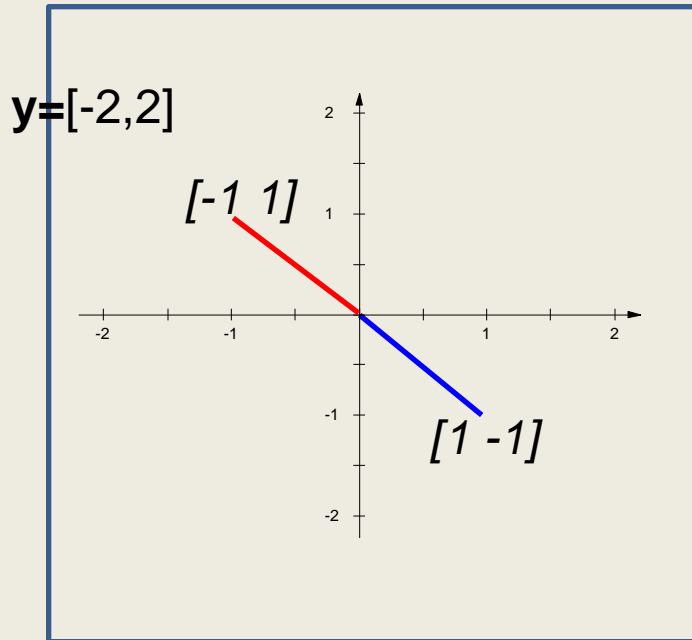


$$a * \begin{pmatrix} 1 \\ -1 \end{pmatrix} + b * \begin{pmatrix} -1 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix} * \begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} 1 \\ 2 \end{pmatrix}$$

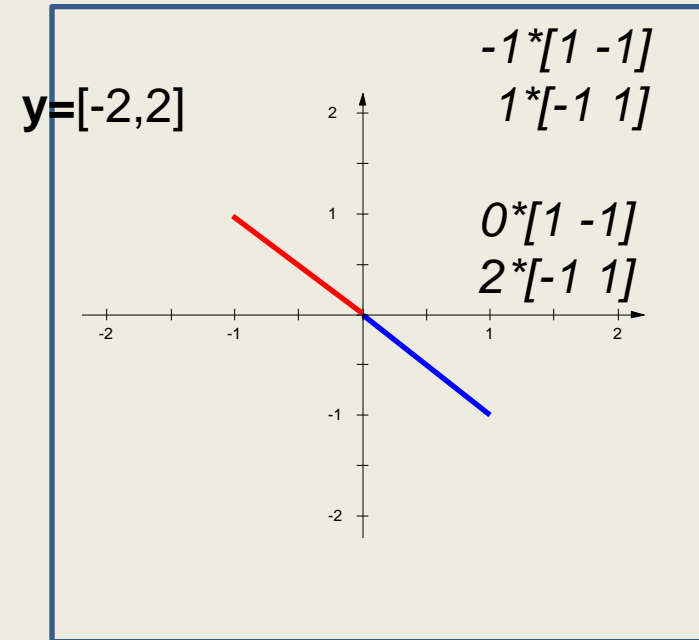
Columns are parallel – matrix not invertible

Geometric interpretation of linear equations

Problem



Infinitely many solutions

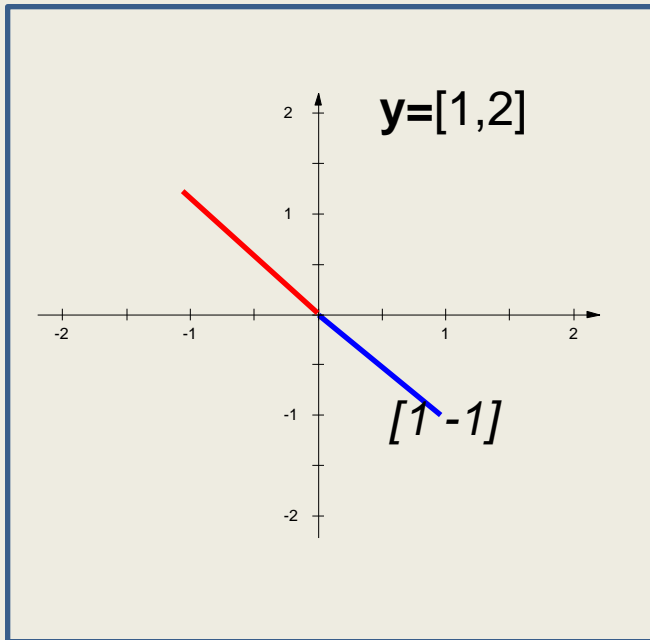


$$a * \begin{pmatrix} 1 \\ -1 \end{pmatrix} + b * \begin{pmatrix} -1 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix} * \begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} -2 \\ 2 \end{pmatrix}$$

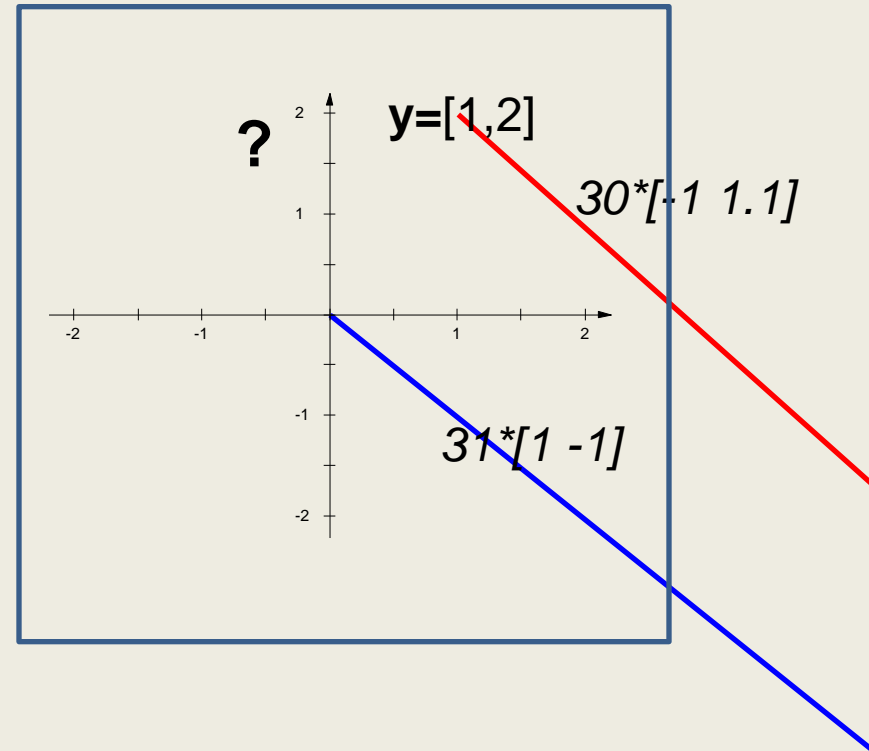
Columns are parallel – matrix not invertible

Geometric interpretation of linear equations

Problem



Solution



$$a * \begin{pmatrix} 1 \\ -1 \end{pmatrix} + b * \begin{pmatrix} -1 \\ 1.1 \end{pmatrix} = \begin{pmatrix} 1 & -1 \\ -1 & 1.1 \end{pmatrix} * \begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} 1 \\ 2 \end{pmatrix}$$

Matrix invertible – but possibly instable

Example

The “rank” of a matrix

In order for an equation to have a unique solution, the vectors in that equation cannot be parallel, i.e. they have to be “linearly independent”

The number of linearly independent (row or column) vectors in a matrix is its “rank”

If a matrix is square, and its rank is equal to its dimension, then it’s invertible and the equation has a unique solution

Example

A problem with rank

The rank of a matrix does not tell us how independent the vectors in the equation are

Shouldn't these cases (both rank = 2) be different?

$$a * \begin{pmatrix} 1 \\ -1 \end{pmatrix} + b * \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ -1 & 1 \end{pmatrix} * \begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} 1 \\ 2 \end{pmatrix}$$

$$a * \begin{pmatrix} 1 \\ -1 \end{pmatrix} + b * \begin{pmatrix} -1 \\ 1.1 \end{pmatrix} = \begin{pmatrix} 1 & -1 \\ -1 & 1.1 \end{pmatrix} * \begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} 1 \\ 2 \end{pmatrix}$$

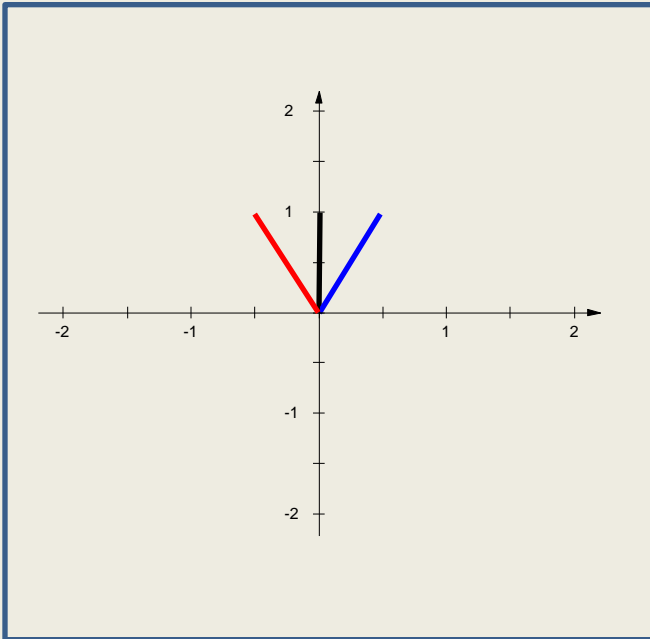
The idea behind Singular Value Decomposition (SVD)

don't worry about the name yet

Let's find out how much variation there is among vectors

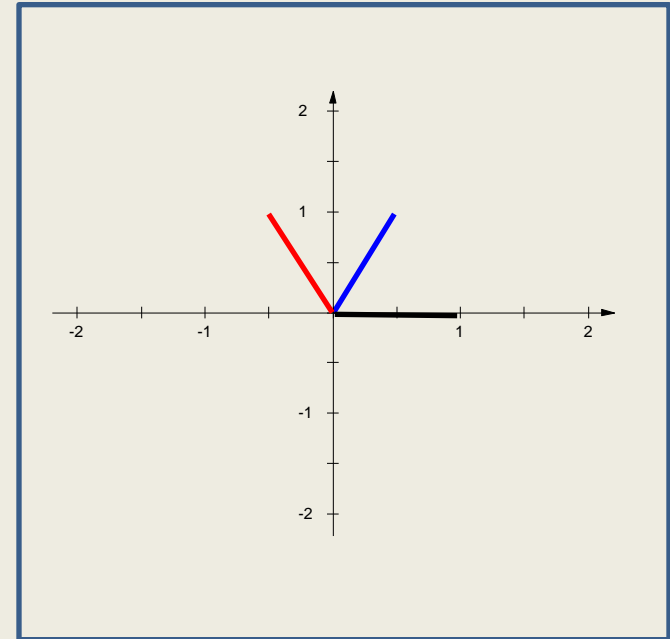
Step 1:

Find a vector 1 that "explains as much as possible" of all vectors



Step 2:

Find an orthogonal vector that explains as much as possible of what vector 1 couldn't



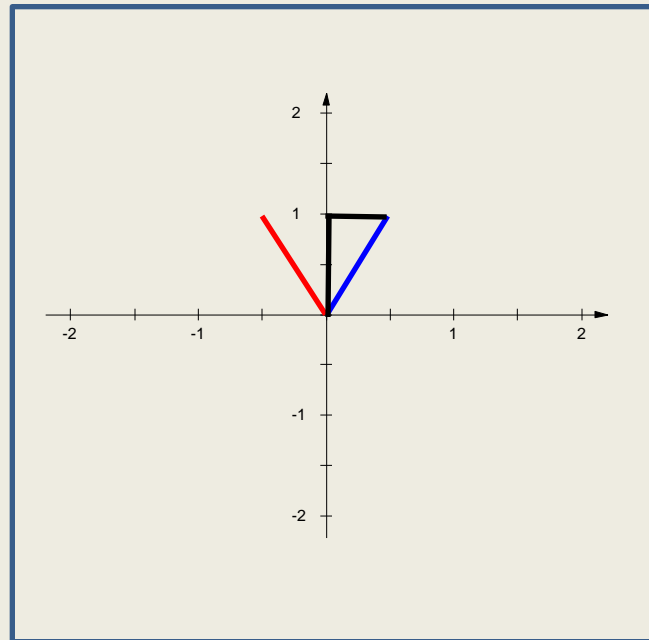
Which of these vectors is "more important"?

The idea behind Singular Value Decomposition (SVD)

don't worry about the name yet

Step 3:

Find out how much the two vectors
contribute relative to each other

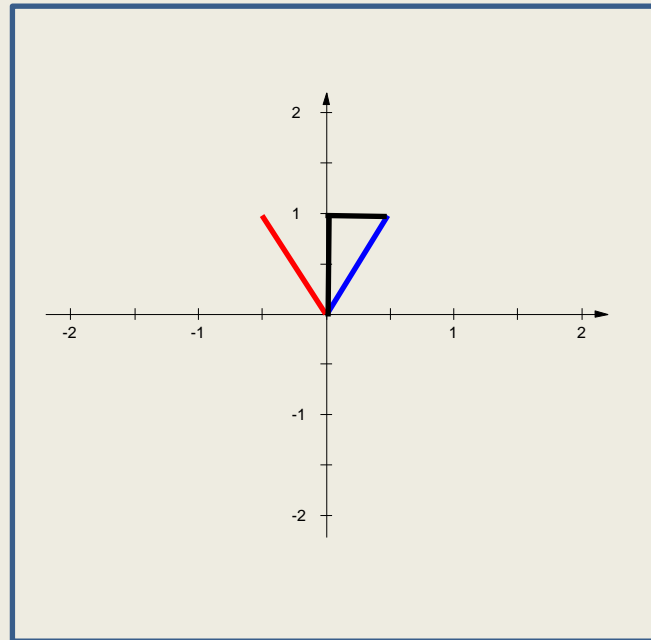


If the relative contributions are (more or less equal), then the original vectors
are very independent

If the relative contributions are very different, then there is some similarity
among vectors

The idea behind Singular Value Decomposition (SVD)

don't worry about the name yet

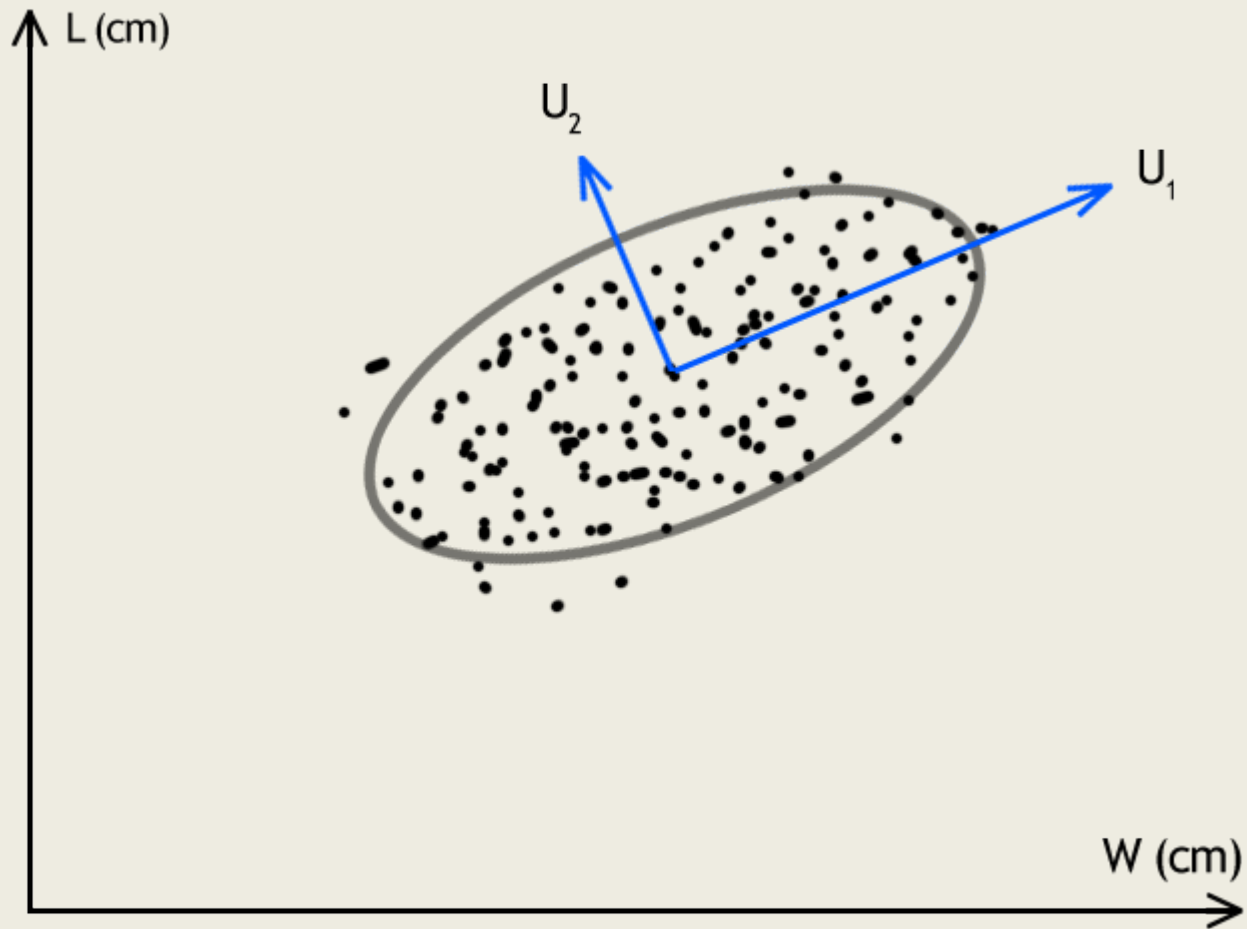


The ratio of the contributions of the two (black) vectors tells us something about how “separable” the two (red and blue) vectors are.

This will lead to the “condition number” of a matrix.

The idea behind Singular Value Decomposition (SVD)

don't worry about the name yet



SVD in Matlab

Let's start with a square matrix:

$$[\mathbf{U}, \mathbf{S}] = \text{svd}(\mathbf{M})$$

U: “Singular” or “Eigen”-vectors of **M**

S: contains “singular” values” on diagonal

The columns of **U** correspond to the new (black) vectors in the previous examples.
The values in **S** reflect their “relative importance”.

The ratio between the largest and smallest values in **S** is the condition number.

If it's “infinity” then the matrix is not invertible.

Large values ($\sim > 30$) mean the matrix is “unstable”, i.e. inversion may not produce accurate results.

Small values ($\sim 1-10$) mean the vectors in the matrix are quite independent.

Example

What if matrices are not square?

$$a * \begin{pmatrix} 2 \\ 2 \\ 0 \end{pmatrix} + b * \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 2 & 0 \\ 2 & 0 \\ 0 & 1 \end{pmatrix} * \begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$$

Rows and columns have different dimensions

There is still one “rank”:

The maximum number of independent rows and columns is the same

$$[\mathbf{U}, \mathbf{S}, \mathbf{V}] = \text{svd}(\mathbf{M})$$

U: Eigenvectors of columns **M**

V: Eigenvectors of rows of **M**

S: contains “singular values” on diagonal

This leads to a very useful decomposition of **M**:

$$\mathbf{M} = \mathbf{U} * \mathbf{S} * \mathbf{V}'$$

Example

What is this talk about “singular” values...

If you have a matrix \mathbf{M} with a singular value decomposition (SVD):

$$\mathbf{M} = \mathbf{U} * \mathbf{S} * \mathbf{V}'$$

\mathbf{U} and \mathbf{V} are unitary, i.e. $\mathbf{U} * \mathbf{U}' = \mathbf{I}$ and $\mathbf{V} * \mathbf{V}' = \mathbf{I}$

\mathbf{S} is diagonal, i.e. zero everywhere, except on the diagonal

Then with a bit of matrix abracadabra... I mean algebra :

$$\mathbf{M} * \mathbf{V} = (\mathbf{U} * \mathbf{S} * \mathbf{V}') \mathbf{V} = \mathbf{U} * \mathbf{S}$$

and

$$\mathbf{M}' * \mathbf{U} = (\mathbf{U} * \mathbf{S} * \mathbf{V}')' \mathbf{U} = (\mathbf{V} * \mathbf{S} * \mathbf{U}') \mathbf{U} * \mathbf{S} = \mathbf{V} * \mathbf{S}$$

Vectors in \mathbf{U} and \mathbf{V} are projected onto each other

What's this talk about "eigen"values?

For a matrix \mathbf{M} , one can compute the covariance matrix by row

$$\mathbf{C} = \mathbf{M} * \mathbf{M}' = (\mathbf{U} * \mathbf{S} * \mathbf{V}') * (\mathbf{U} * \mathbf{S} * \mathbf{V}')' = (\mathbf{U} * \mathbf{S} * \mathbf{V}') * (\mathbf{V} * \mathbf{S} * \mathbf{U}') = \mathbf{U} * \mathbf{S}^2 * \mathbf{U}'$$

(because \mathbf{V} is unitary)

Then :

$$\mathbf{C} * \mathbf{U} = \mathbf{M} * \mathbf{M}' * \mathbf{U} = \mathbf{U} * \mathbf{S}^2 * \mathbf{U}' * \mathbf{U} = \mathbf{U} * \mathbf{S}^2$$

(because \mathbf{U} is unitary)

In other words:

If you put a column vector of \mathbf{U} through $\mathbf{C} = \mathbf{M} * \mathbf{M}'$,
then you get the same vector back, multiplied by a factor

$$\mathbf{C} * \mathbf{U}_{.i} = \mathbf{S}_{ii}^2 \mathbf{U}_{.i}$$

What's this talk about "eigen"values?

For a matrix \mathbf{M} , one can compute the covariance matrix by column

$$\mathbf{C} = \mathbf{M}' * \mathbf{M} = (\mathbf{U} * \mathbf{S} * \mathbf{V}')' * (\mathbf{U} * \mathbf{S} * \mathbf{V}') = (\mathbf{V} * \mathbf{S} * \mathbf{U}') * (\mathbf{U} * \mathbf{S} * \mathbf{V}') = \mathbf{V} * \mathbf{S}^2 * \mathbf{V}'$$

(because \mathbf{U} is unitary)

Then :

$$\mathbf{C} * \mathbf{V} = \mathbf{M}' * \mathbf{M} * \mathbf{V} = \mathbf{V} * \mathbf{S}^2 * \mathbf{V}' * \mathbf{V} = \mathbf{V} * \mathbf{S}^2$$

(because \mathbf{V} is unitary)

In other words:

If you put a column vector of \mathbf{U} through $\mathbf{C} = \mathbf{M}' * \mathbf{M}$,
then you get the same vector back, multiplied by a factor

$$\mathbf{C} * \mathbf{V}_{.i} = \mathbf{S}_{ii}^2 * \mathbf{V}_{.i}$$

What's this talk about "eigen"vectors?

"Eigen" means something like "self" in German

Eigenvectors of a matrix are projected on themselves (or each other)

Link to PCA:

U and **V** contain the principal components or eigenvectors,
and **S** the factor loadings or **S**² the eigenvalues

Example

Why is this useful?

Filtering:

Once you have decomposed your matrix/data/equation into components, you can remove those that annoy you
(this can also be done using PCA or ICA etc.)

Example

Why is this useful?

Remember the stuff about inverse matrices?

Problem:

- We have an equation $\mathbf{Mx}=\mathbf{y}$
- We know \mathbf{M} and \mathbf{y}
- We want to know \mathbf{x}

If only we had a matrix \mathbf{M}^{-1} with the property

$$\mathbf{M}^{-1}*\mathbf{M} = \mathbf{I}$$

(\mathbf{I} is the identity matrix)

because then:

$$\mathbf{M}^{-1}*\mathbf{Mx} = \mathbf{I}*\mathbf{x} = \mathbf{x} = \mathbf{M}^{-1}\mathbf{y}$$

\mathbf{M}^{-1} is the “inverse matrix” of \mathbf{M}

(Not every matrix has a unique inverse matrix. If it does, it’s called “invertible”)

Why is this useful?

Imagine \mathbf{M} is square:

$$\mathbf{M} = \mathbf{U} * \mathbf{S} * \mathbf{V}'$$

\mathbf{U} and \mathbf{V} are “unitary” matrices. Then:

(this is a bit hairy)

$$\begin{aligned} (\mathbf{U} * \mathbf{S}^{-1} * \mathbf{V}')' * (\mathbf{U} * \mathbf{S} * \mathbf{V}') &= (\mathbf{V} * \mathbf{S}^{-1} * \mathbf{U}') * (\mathbf{U} * \mathbf{S} * \mathbf{V}') \\ &= \mathbf{V} * \mathbf{S}^{-1} * \mathbf{U}' * \mathbf{U} * \mathbf{S} * \mathbf{V}' = \mathbf{V} * \mathbf{S}^{-1} * \mathbf{S} * \mathbf{V}' = \mathbf{V} * \mathbf{V}' = \mathbf{I} \end{aligned}$$

In other words:

Once we have the singular value decomposition, we can simply compute the (pseudo)inverse by inverting the individual eigenvalues

Small eigenvalues are dangerous:

They hardly contribute to \mathbf{M} , but their inverse will contribute hugely to \mathbf{M}^{-1} .
If they are not reliable (e.g. modelling errors), they can completely distort any result.

Solution:

Ignore or “dampen” those eigenvalues that you consider unreliable (“regularisation”).

Example

