Master in
Computer Vision
Barcelona

UAB UOC UPC upf.

T3: Singular Value Decomposition & Least Square Problems

Pablo Arias Martínez - ENS Paris-Saclay, UPF
pablo.arias@upf.edu
October 11, 2022

Optimization and inference techniques for Computer Vision

Pre-requisites I

Properties of dot product and norm

We will use the following dot product between vectors $\mathbf{x}, \mathbf{y}$ in $\mathbb{R}^n$:

$$\langle \mathbf{x}, \mathbf{y} \rangle = \mathbf{x}^T \mathbf{y} = \begin{pmatrix} x_1 & x_2 & \dots & x_n \end{pmatrix} \cdot \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix} = \sum_{i=1}^{n} x_i y_i$$

The dot product defines a norm (the Euclidean norm),

$$\|\mathbf{x}\|_2 = \sqrt{\langle \mathbf{x}, \mathbf{x} \rangle} = \sqrt{\sum_{i=1}^{n} x_i^2}.$$

If θ is the angle between vectors $\mathbf{x}$ and $\mathbf{y}$, we have that:

$$\langle \mathbf{x}, \mathbf{y} \rangle = \|\mathbf{x}\| \|\mathbf{y}\| \cos(\theta).$$

Thus, if $\mathbf{x}$ and $\mathbf{y}$ are orthogonal to each other, $\langle \mathbf{x}, \mathbf{y} \rangle = 0$.

Some useful algebraic properties

We will use the following properties of dot products:

- For all $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$: $\langle \mathbf{x}, \mathbf{y} \rangle = \langle \mathbf{y}, \mathbf{x} \rangle$.
- For all $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n, \alpha, \beta \in \mathbb{R}$: $\langle \mathbf{x}, \alpha\mathbf{y} + \beta\mathbf{z} \rangle = \alpha\langle \mathbf{x}, \mathbf{y} \rangle + \beta\langle \mathbf{x}, \mathbf{z} \rangle$.
- For all $\mathbf{x} \in \mathbb{R}^n$, $\mathbf{y} \in \mathbb{R}^m$ and $\mathbf{A}$ an $m \times n$ matrix: $\langle \mathbf{A}\mathbf{x}, \mathbf{y} \rangle = \langle \mathbf{x}, \mathbf{A}^T\mathbf{y} \rangle$.

If you have any doubts, remember that $\langle \mathbf{x}, \mathbf{y} \rangle = \mathbf{x}^T\mathbf{y}$, and apply the rules of matrix product. Example: $\langle \mathbf{A}\mathbf{x}, \mathbf{y} \rangle = (\mathbf{A}\mathbf{x})^T\mathbf{y} = \mathbf{x}^T\mathbf{A}^T\mathbf{y} = \langle \mathbf{x}^T, \mathbf{A}^T\mathbf{y} \rangle$.

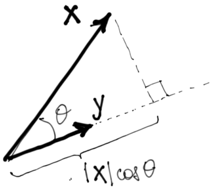
We will use the following properties of matrix product:

- For any matrices $\mathbf{A}$ $m \times n$ and $\mathbf{B}$ $n \times p$: $(\mathbf{AB})^T = \mathbf{B}^T\mathbf{A}^T$.
- For any $n \times n$ invertible matrices $\mathbf{A}$ and $\mathbf{B}$: $(\mathbf{AB})^{-1} = \mathbf{B}^{-1}\mathbf{A}^{-1}$.

Orthogonal projections

Let θ the angle between $\mathbf{x}$ and $\mathbf{y}$. Then the **signed length** of the projection of $\mathbf{x}$ over the direction defined by $\mathbf{y}$ is:

$$\|\mathbf{x}\| \cos(\theta) = \frac{\langle \mathbf{x}, \mathbf{y} \rangle}{\|\mathbf{y}\|} = \left\langle \mathbf{x}, \frac{\mathbf{y}}{\|\mathbf{y}\|} \right\rangle .$$



This is a number, and it is sometimes called the **scalar projection**. To compute the actual projection, we need a vector with that length in the direction of $\mathbf{y}$:

$$\mathrm{proj}_{\mathbf{y}}(\mathbf{x}) = \left\langle \mathbf{x}, \frac{\mathbf{y}}{\|\mathbf{y}\|} \right\rangle \frac{\mathbf{y}}{\|\mathbf{y}\|}.$$

If we project into a unit norm vector $\mathbf{u}$, with $\|u\| = 1$, we get:

$$\text{scalar projection: } \langle \mathbf{x}, \mathbf{u} \rangle = \mathbf{u}^T \mathbf{x}, \quad \text{and} \quad \mathrm{proj}_{\mathbf{u}}(\mathbf{x}) = \langle \mathbf{x}, \mathbf{u} \rangle \mathbf{u} = (\mathbf{u}^T \mathbf{x}) \mathbf{u}.$$

Least square problems

Review from linear algebra: systems of linear equations

Example: a system of 5 linear equations with 4 unkowns

$$\left\{\begin{array}{rrrrl} 4x_1 & +10x_2 & -3x_3 & +4x_4 & = 4 \\ 12x_1 & & +2x_3 & +0.2x_4 & = 0 \\ -7x_1 & -x_2 & -5x_3 & +20x_4 & = 2.51 \\ 3.2x_1 & +1.5x_2 & -2x_3 & & = -20 \\ 8x_1 & -20x_2 & +5x_3 & +3x_4 & = 3 \end{array}\right.$$

We can write it as a linear vectorial equation with unknown $\mathbf{x} \in \mathbb{R}^4$:

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

$$\mathbf{A} = \begin{pmatrix} 4 & 10 & -3 & 4 \\ 12 & 0 & 2 & 0.2 \\ -7 & -1 & -5 & 20 \\ 3.2 & 1.5 & -2 & 0 \\ 8 & -20 & 5 & 3 \end{pmatrix} \qquad \mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} \qquad \mathbf{b} = \begin{pmatrix} 4 \\ 0 \\ 2.51 \\ -20 \\ 3 \end{pmatrix}$$

Consider a linear equation with $\mathbf{A}$ $m \times n$ (thus, n unkowns $\mathbf{x} \in \mathbb{R}^n$ and m equations $\mathbf{b} \in \mathbb{R}^m$)

$$\mathbf{Ax} = \mathbf{b}.$$

We assume no equation can be written as a linear combination of the other equations. Then,

$m = n$: **determined** system: $\mathbf{A}$ is invertible. Unique solution $\mathbf{x}^* = \mathbf{A}^{-1}\mathbf{b}$

$m < n$: **under-determined** system: infinite solutions forming a hyperplane

$m > n$: **over-determined** system: there is no solution

Least squares problem

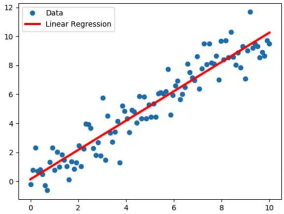
Suppose we have an over-determined system of equations. Since there is no exact solution, we can instead minimize the quadratic error between $\mathbf{Ax}$ and $\mathbf{b}$:

$$\begin{aligned} \mathbf{x}^* &= \operatorname{argmin} \|\mathbf{Ax} - \mathbf{b}\|^2 \\ &= \operatorname{argmin} \sum_{i=1}^{m} \left(\sum_{j=1}^{n} a_{ij} x_j - b_i \right)^2 . \end{aligned}$$

Motivation: linear fitting with noise

We work for an ice-cream company, and we want to determine the relation between average daily temperature and ice-cream consumption. We collect data for $m = 60$ days:

T_i	average temperature on ith day
n_i	ice-creams sold that day.



We observe a linear trend, and would like to determine its coefficients c, d

$$cT_i + d \approx n_i, \quad i = 1, ..., m.$$

In matrix notation: $\mathbf{Ax} \approx \mathbf{b}$, $\mathbf{A} = \begin{pmatrix} T_1 & 1 \\ T_2 & 1 \\ \vdots & \vdots \\ T_m & 1 \end{pmatrix}$ $\mathbf{x} = \begin{pmatrix} c \\ d \end{pmatrix}$ $\mathbf{b} = \begin{pmatrix} n_1 \\ n_2 \\ \vdots \\ n_m \end{pmatrix}$

However, we have only two variables and many equations (one for each observation). It is an over-complete system with not exact solution.

We can find the line that minimizes the vertical distance to the data points. This is the least squares solution:

$$(c^*, d^*) = \operatorname{argmin} \sum_{i=1}^{m} (T_i c + d - n_i)^2 = \operatorname{argmin} \|\mathbf{A}\mathbf{x} - \mathbf{b}\|^2.$$

$$\mathbf{A} = \begin{pmatrix} T_1 & 1 \\ T_2 & 1 \\ \vdots & \vdots \\ T_m & 1 \end{pmatrix} \quad \mathbf{x} = \begin{pmatrix} c \\ d \end{pmatrix} \quad \mathbf{b} = \begin{pmatrix} n_1 \\ n_2 \\ \vdots \\ n_m \end{pmatrix}$$

Least squares problem: geometrical interpretation

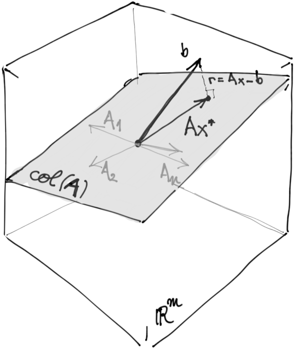
The matrix-vector multiplication $\mathbf{Ax}$ can be seen as a linear combination of the columns of $\mathbf{A}$, $\mathbf{A}_1, \ldots, \mathbf{A}_n$:

$$\mathbf{Ax} = \sum_{j=1}^{n} x_j \mathbf{A}_j .$$

Therefore, $\mathbf{Ax}$ lies in the column space of $\mathbf{A}$: the space generated by the columns of $\mathbf{A}$. The column space of $\mathbf{A}$ is a subspace of $\mathbb{R}^m$ of dimension n.

We are searching for the vector in the column space of $\mathbf{A}$ that minimizes the distance to $\mathbf{b}$.

Thus, $\mathbf{Ax}$ needs to be the *orthogonal projection* of $\mathbf{b}$ over the column space of $\mathbf{A}$.



Computing the least squares solution

We denote $f(\mathbf{x}) = \|\mathbf{A}\mathbf{x} - \mathbf{b}\|^2$. To minimize f, we need $\nabla f(\mathbf{x}) = \mathbf{0}$.

First, some algebraic manipulations. Recall that $\|\mathbf{v}\| = \sqrt{\langle \mathbf{v}, \mathbf{v} \rangle}$:

$$\begin{aligned} f(\mathbf{x}) = \|\mathbf{A}\mathbf{x} - \mathbf{b}\|^2 &= \langle \mathbf{A}\mathbf{x} - \mathbf{b}, \mathbf{A}\mathbf{x} - \mathbf{b} \rangle \\ &= \langle \mathbf{A}\mathbf{x}, \mathbf{A}\mathbf{x} \rangle - \langle \mathbf{A}\mathbf{x}, \mathbf{b} \rangle - \langle \mathbf{b}, \mathbf{A}\mathbf{x} \rangle + \langle \mathbf{b}, \mathbf{b} \rangle \\ &= \langle \mathbf{A}^T\mathbf{A}\mathbf{x}, \mathbf{x} \rangle - 2\langle \mathbf{A}^T\mathbf{b}, \mathbf{x} \rangle + \langle \mathbf{b}, \mathbf{b} \rangle \end{aligned}$$

Or equivalently: $f(\mathbf{x}) = \mathbf{x}^T\mathbf{A}^T\mathbf{A}\mathbf{x} - 2\mathbf{b}^T\mathbf{A}\mathbf{x} + \mathbf{b}^T\mathbf{b}$.

Parenthesis: useful gradients

f is a quadratic polynomial of $\mathbb{R}^n$ variables. The general form of quadratic polynomials is

$$p(\mathbf{x}) = \mathbf{x}^T \mathbf{Q} \mathbf{x} + \mathbf{d}^T \mathbf{x} + c,$$

where $\mathbf{Q}$ is $n \times n$, $\mathbf{d} \in \mathbb{R}^n$ and $c \in \mathbb{R}$. We will work more with them later.

The following are useful gradients that appear when working with quadratic functions in $\mathbb{R}^n$

f	∇f
$\langle \mathbf{b}, \mathbf{x} \rangle = \mathbf{b}^T \mathbf{x}$	$\mathbf{b}$
$\langle \mathbf{x}, \mathbf{A}\mathbf{x} \rangle = \mathbf{x}^T \mathbf{A} \mathbf{x}$	$2\mathbf{A}\mathbf{x}$
$\|\mathbf{x}\|^2 = \langle \mathbf{x}, \mathbf{x} \rangle = \mathbf{x}^T \mathbf{x}$	$2\mathrm{x}$
$\|\mathbf{A}\mathbf{x}\|^2 = \langle \mathbf{A}\mathbf{x}, \mathbf{A}\mathbf{x} \rangle = \mathbf{x}^T \mathbf{A}^T \mathbf{A} \mathbf{x}$	$2\mathbf{A}^T \mathbf{A} \mathbf{x}$
$\|\mathbf{A}\mathbf{x} - \mathbf{b}\|^2 = \langle \mathbf{A}\mathbf{x} - \mathbf{b}, \mathbf{A}\mathbf{x} - \mathbf{b} \rangle$	$2\mathbf{A}^T (\mathbf{A}\mathbf{x} - \mathbf{b})$

Computing the least squares solution

Going back to our least squares objective:

$$\begin{aligned}\nabla f(\mathbf{x}) &= \nabla\|\mathbf{A}\mathbf{x}-\mathbf{b}\|^2 = \nabla\langle \mathbf{A}^T\mathbf{A}\mathbf{x},\mathbf{x}\rangle - 2\nabla\langle \mathbf{A}^T\mathbf{b},\mathbf{x}\rangle + \nabla\langle \mathbf{b},\mathbf{b}\rangle \\ &= 2\mathbf{A}^T\mathbf{A}\mathbf{x} - 2\mathbf{A}^T\mathbf{b}\end{aligned}$$

Since it is a quadratic funcion, the global minimum is attained if and only if the gradient is zero:

$$\nabla f(\mathbf{x}^*) = \mathbf{0} \quad\Longrightarrow\quad 2\mathbf{A}^T\mathbf{A}\mathbf{x}^* - 2\mathbf{A}^T\mathbf{b} = \mathbf{0} \quad\Longrightarrow\quad \underbrace{\mathbf{A}^T\mathbf{A}\mathbf{x}^* = \mathbf{A}^T\mathbf{b}}_{\text{normal equations}}$$

If $\mathbf{A}^T\mathbf{A}$ is **invertible**, then there exists a unique solution:

$$\mathbf{x}^* = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{b}.$$

$\mathbf{A}^T\mathbf{A}$ is an $n \times n$ matrix. We have that

$$\operatorname{rank}(\mathbf{A}^T\mathbf{A}) = \operatorname{rank}(\mathbf{A}) = r \leq n.$$

If $r = n$, then there is a unique solution to the least squares problem.

If $r < n$, then there are multiple solutions to the least square problem, and they lie on a hyper-plane of $\mathbb{R}^n$ dimension $n - r$.

If $r < n$ one way of choosing one among all solutions is by choosing the smallest one:

$$\text{Regularized least-squares:} \qquad f(\mathbf{x}) = \|\mathbf{A}\mathbf{x} - \mathbf{b}\|^2 + \delta\|\mathbf{x}\|^2, \quad \text{with } \delta > 0.$$

Show that in this case there exists a unique solution: $\mathbf{x}^* = (\mathbf{A}^T\mathbf{A} + \delta I_n)^{-1}\mathbf{A}^T\mathbf{b}$.

Least squares and orthogonal projection

Remember: Least squares computes searches for $\mathbf{Ax}^*$ in the column space of $\mathbf{A}$ which is closest to $\mathbf{b}$.

Since the column space of $\mathbf{A}$ is a hyper-plane throuth the origin. The point that minimizes the distance to $\mathbf{b}$ needs to be orthogonal to the error $\mathbf{r} = \mathbf{Ax}^* - \mathbf{b}$. Let us verify:

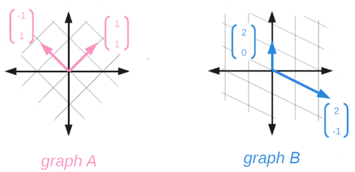
$$\begin{aligned}\langle \mathbf{Ax}^*, \mathbf{Ax}^* - \mathbf{b}\rangle &= \langle \mathbf{A}(\mathbf{A}^T\mathbf{A})^{-1}\mathbf{Ab}, \mathbf{A}(\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{b} - \mathbf{b}\rangle \\ &= \langle (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{Ab}, \mathbf{A}^T\mathbf{A}(\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{b} - \mathbf{A}^T\mathbf{b}\rangle \\ &= \langle (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{Ab}, \mathbf{A}^T\mathbf{b} - \mathbf{A}^T\mathbf{b}\rangle = 0\end{aligned}$$

Thus: $\mathbf{Ax}^* = \mathbf{A}(\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{b}$ is the orthogonal projection of $\mathbf{b}$ over the columns space of $\mathbf{A}$, and the matrix $\mathbf{A}(\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T$ projects orthogonally onto the column space of $\mathbf{A}$.

Pre-requisites II

In a vector space of dimension n we use basis to define coordinate systems. A basis is a set of n **linearly independent** vectors:

$$\mathcal{B} = \{\mathbf{v}_1, \dots, \mathbf{v}_n\}.$$



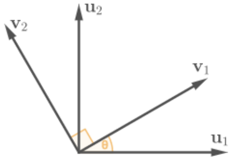
We can express any vector $\mathbf{x}$ as a **unique** linear combination of the basis vectors:

$$\mathbf{x} = \sum_{i=1}^{n} \alpha_i \mathbf{v}_i .$$

The coefficients α_i are the **coordinates** of $\mathbf{x}$ in the basis $\mathcal{B}$.

Orthonormal bases, orthonormal matrices and rotations

Orthonormal bases are bases where the vectors are orthogonal between each other and have unit norm:



$$\mathcal{B} = \{\mathbf{u}_1, \ldots, \mathbf{u}_n\}, \quad \text{with} \quad \langle \mathbf{u}_i, \mathbf{u}_j \rangle = \begin{cases} 0 & \text{if } i \neq j, \\ 1 & \text{if } i = j. \end{cases}$$

Computing the coordinates α_i of a vector $\mathbf{x} \in \mathbb{R}^n$ in an orthonormal basis is very simple: we just need to compute the scalar projections over each basis vector:

$$\boldsymbol{\alpha} = \begin{pmatrix} \alpha_1 \\ \vdots \\ \alpha_n \end{pmatrix} = \begin{pmatrix} \mathbf{u}_1^T \mathbf{x} \\ \vdots \\ \mathbf{u}_n^T \mathbf{x} \end{pmatrix} = \begin{pmatrix} \cdots \mathbf{u}_1^T \cdots \\ \vdots \\ \cdots \mathbf{u}_n^T \cdots \end{pmatrix} \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} = \mathbf{U}^T \mathbf{x}.$$

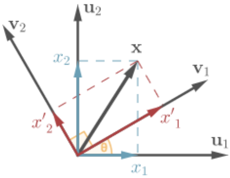
Where $\mathbf{U}$ is a matrix with the basis elements as columns.

If we have the coordinates $\boldsymbol{\alpha}$ and want to reconstruct $\mathbf{x}$ from them:

$$\mathbf{x} = \sum_{i=1}^{n} \alpha_i \mathbf{u}_i = \mathbf{U}\boldsymbol{\alpha}.$$

Orthonormal bases are bases where the vectors are orthogonal between each other and have unit norm:

$$\mathcal{B} = \{\mathbf{u}_1, ..., \mathbf{u}_n\}, \quad \text{with} \quad \langle \mathbf{u}_i, \mathbf{u}_j \rangle = \begin{cases} 0 & \text{if } i \neq j, \\ 1 & \text{if } i = j. \end{cases}$$



Computing the coordinates α_i of a vector $\mathbf{x} \in \mathbb{R}^n$ in an orthonormal basis is very simple: we just need to compute the scalar projections over each basis vector:

$$\boldsymbol{\alpha} = \begin{pmatrix} \alpha_1 \\ \vdots \\ \alpha_n \end{pmatrix} = \begin{pmatrix} \mathbf{u}_1^T\mathbf{x} \\ \vdots \\ \mathbf{u}_n^T\mathbf{x} \end{pmatrix} = \begin{pmatrix} \cdots \mathbf{u}_1^T \cdots \\ \vdots \\ \cdots \mathbf{u}_n^T \cdots \end{pmatrix} \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} = \mathbf{U}^T\mathbf{x}.$$

Where $\mathbf{U}$ is a matrix with the basis elements as columns.

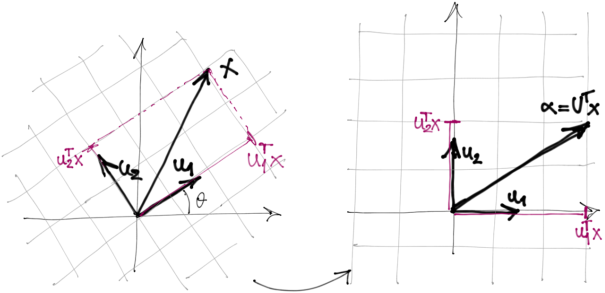
If we have the coordinates $\boldsymbol{\alpha}$ and want to reconstruct $\mathbf{x}$ from them:

$$\mathbf{x} = \sum_{i=1}^{n} \alpha_i \mathbf{u}_i = \mathbf{U}\boldsymbol{\alpha}.$$

Orthonormal bases, orthonormal matrices and rotations

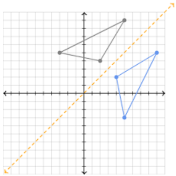
A square matrix with orthonormal columns is called an **orthonormal matrix**. Orthonormal matrices have several interesting properties: $\mathbf{U}$ is

- They are invertible, and $\mathbf{U}^{-1} = \mathbf{U}^T$. Thus $\mathbf{U}^T\mathbf{U} = \mathbf{U}\mathbf{U}^T = \mathbf{I}$.
- They preserve angles and lengths: $\langle \mathbf{Ux}, \mathbf{Uy} \rangle = \langle \mathbf{U}^T\mathbf{Ux}, \mathbf{y} \rangle = \langle \mathbf{x}, \mathbf{y} \rangle$.
- They correspond to rotations and reflections.
- They correspond to changes of coordinates between orthonormal bases.



Geometrical interpretation of matrices

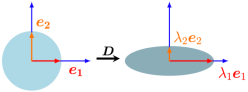
A matrix $\mathbf{A}$ can be seen as a transformation $\mathbf{x} \mapsto \mathbf{Ax}$. Some $n \times n$ matrices have a simple geometrical interpretation as transformations in $\mathbb{R}^n$. We will use examples in $\mathbb{R}^2$.



Rotation of angle θ $\mathbf{A} = \begin{pmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{pmatrix}$.

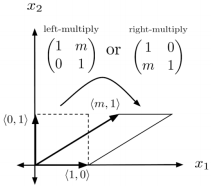
Reflections. E.g. around the horizontal axis and diagonal:

$$\mathbf{A} = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}, \quad \mathbf{A} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}.$$



Scalings. Diagonal matrices $\mathbf{A} = \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix}$.

Shear. E.g. horiz.: $\mathbf{A} = \begin{pmatrix} 1 & m \\ 0 & 1 \end{pmatrix}$. $\mathbf{Ax} = \begin{pmatrix} x_1 + m x_2 \\ x_2 \end{pmatrix}$.



We consider a **square matrix A** $n \times n$. An **eigenvector v** is a vector (and the corresponding direction) where **A** acts like a scaling, and the **eigenvalue** λ is the scaling factor:

$$\mathbf{A}\mathbf{v} = \lambda \mathbf{v}.$$

A square matrix can have any number from 0 to n of (linearly independent) eigenvectors. Some examples:

- Rotations in $\mathbb{R}^2$ have no eigenvectors
- Horizontal shears in $\mathbb{R}^2$ have a single eigenvector: $\mathbf{v} = (1, 0)^T$, $\lambda = 1$.
- Diagonal scalings have two eigenvectors: $(1, 0)^T$ and $(0, 1)^T$ with eigenvalues λ_1, λ_2, the scaling factors.

Eigendecomposition (or diagonalization)

In fact, any $n \times n$ matrix that has n linearly independent eigenvectors $\mathbf{v}_1, \ldots, \mathbf{v}_n$, **is a diagonal scaling matrix**, except that in the basis of eigenvectors!

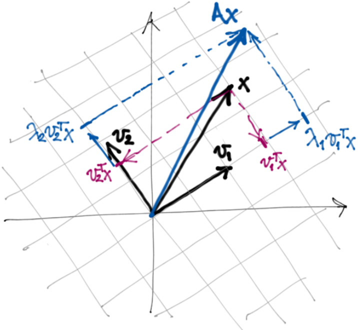
If $\mathbf{A}$, $n \times n$ has a n linearly indepedent eigengectors $\mathbf{v}_1, \ldots, \mathbf{v}_n$, then we can decompose $\mathbf{A}$ as:

$$\mathbf{A} = \underbrace{\mathbf{V}}_{\text{(3) back to standard coords}} \cdot \underbrace{\Lambda}_{\text{(2) diagonal scaling of } \mathbf{V} \text{ coords}} \cdot \underbrace{\mathbf{V}^{-1}}_{\text{(1) coords in basis } \mathbf{V}} .$$

Which matrices can be diagonalized?

Spectral theorem. Any symmetric matrix $\mathbf{A}$ has n eigenvectors forming an orthonomal basis. Therefore:

$$\mathbf{A} = \underbrace{\mathbf{V}}_{\text{(3) back to standard coords}} \cdot \underbrace{\Lambda}_{\text{(2) diagonal scaling of } \mathbf{V} \text{ coords}} \cdot \underbrace{\mathbf{V}^T}_{\text{(1) coords in basis } \mathbf{V}} .$$



Example in $\mathbb{R}^2$:

(1) $\mathbf{V}^T\mathbf{x} = \begin{pmatrix} \mathbf{v}_1^T\mathbf{x} \\ \mathbf{v}_2^T\mathbf{x} \end{pmatrix}$.

(2) $\boldsymbol{\Lambda}\mathbf{V}^T\mathbf{x} = \begin{pmatrix} \lambda_1\mathbf{v}_1^T\mathbf{x} \\ \lambda_2\mathbf{v}_2^T\mathbf{x} \end{pmatrix}$.

(3) $\mathbf{V}\boldsymbol{\Lambda}\mathbf{V}^T\mathbf{x} = \lambda_1(\mathbf{v}_1^T\mathbf{x})\mathbf{v}_1 + \lambda_2(\mathbf{v}_2^T\mathbf{x})\mathbf{v}_2$.

Eigen-decompositions are very useful because they give us a much simpler representation of a matrix: a diagonal matrix on an certain basis.

The problem is that only some square matrices have eigen-decompositions.


Eugenio
Beltrami
Camille
Jordan
James Joseph
Sylvester
Hermann
Weyl
Erhard
Schmidt
Gene
Golub

Hold my beer...

Singular Value Decomposition

The **singular value decomposition** or **SVD** or the $m \times n$ matrix $\mathbf{A}$ is given by

$$\mathbf{A} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T$$

where

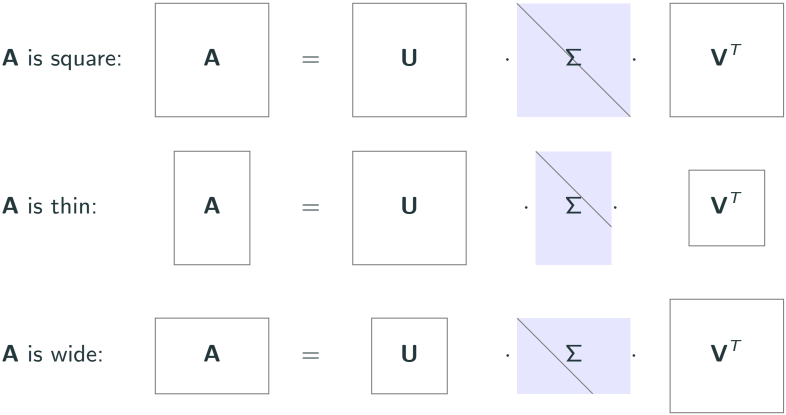
- $\mathbf{U}$ is an $m \times m$ orthonormal matrix,
- $\mathbf{V}$ is an $n \times n$ orthonormal matrix,
- $\boldsymbol{\Sigma}$ is an $m \times n$ diagonal matrix, with elements σ_i sorted in non-increasing order:

  $$\sigma_1 \geq \sigma_2 \geq ... \geq \sigma_k \geq 0,$$

  where $k = \min\{m, n\}$ is the smallest dimension of $\mathbf{A}$.

SVD theorem: Any matrix admits an SVD.

SVDs and matrix shapes



Singular values and vectors

The columns of $\mathbf{U}$ and $\mathbf{V}$ are orthonormal bases of $\mathbb{R}^m$ and $\mathbb{R}^n$. We denote them as follows:

- columns of $\mathbf{U}$: $\mathbf{u}_1, \ldots, \mathbf{u}_m$
- columns of $\mathbf{V}$: $\mathbf{v}_1, \ldots, \mathbf{v}_n$

We can rewrite the SVD as

$$\mathbf{A} = \mathbf{U\Sigma V}^T \quad \Longrightarrow \quad \mathbf{AV} = \mathbf{U\Sigma} \quad \text{or} \quad \mathbf{U}^T\mathbf{A} = \mathbf{\Sigma V}^T.$$

From these matrix equations we have that

$$\begin{aligned} \mathbf{Av}_i &= \sigma_i \mathbf{u}_i, \qquad \text{and} \qquad \mathbf{u}_i^T\mathbf{A} = \sigma_i \mathbf{v}_i^T, \qquad \text{for } i = 1, \ldots, k, \\ \mathbf{Av}_i &= \mathbf{0}_m, \qquad \text{and} \qquad \mathbf{u}_i^T\mathbf{A} = \mathbf{0}_n^T, \qquad \text{for } i \geq k. \end{aligned}$$

- $\mathbf{u}_1, \ldots, \mathbf{u}_m$ are called the **left singular vectors**,
- $\mathbf{v}_1, \ldots, \mathbf{v}_n$ are called the **right singular vectors**, and
- $\sigma_1, \ldots, \sigma_k$ are called the **singular values**.

Interpretation of SVD: bases of singular vectors

A is thin ($n < m$): $\mathbf{A} = \mathbf{U} \cdot \boldsymbol{\Sigma} \cdot \mathbf{V}^T$

(1) change to basis **V**

$$\mathbf{V}^T\mathbf{x} = \begin{pmatrix} \mathbf{v}_1^T\mathbf{x} \\ \vdots \\ \mathbf{v}_n^T\mathbf{x} \end{pmatrix}$$

(2) streching

$$\boldsymbol{\Sigma}\mathbf{V}^T\mathbf{x} = \begin{pmatrix} \sigma_1\mathbf{v}_1^T\mathbf{x} \\ \vdots \\ \sigma_n\mathbf{v}_n^T\mathbf{x} \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

(3) change from basis **U**

$$\mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T\mathbf{x} = \sum_{j=1}^{n}(\sigma_j\mathbf{v}_j^T\mathbf{x})\mathbf{u}_j$$

We don't use the last $m - n$ columns of **U**!

Interpretation of SVD: bases of singular vectors

$\mathbf{A}$ is wide ($n > m$): $\mathbf{A} = \mathbf{U} \cdot \Sigma \cdot \mathbf{V}^T$

(1) change to basis $\mathbf{V}$

$$\mathbf{V}^T\mathbf{x} = \begin{pmatrix} \mathbf{v}_1^T\mathbf{x} \\ \vdots \\ \mathbf{v}_m^T\mathbf{x} \\ \mathbf{v}_{m+1}^T\mathbf{x} \\ \vdots \\ \mathbf{v}_n^T\mathbf{x} \end{pmatrix}$$

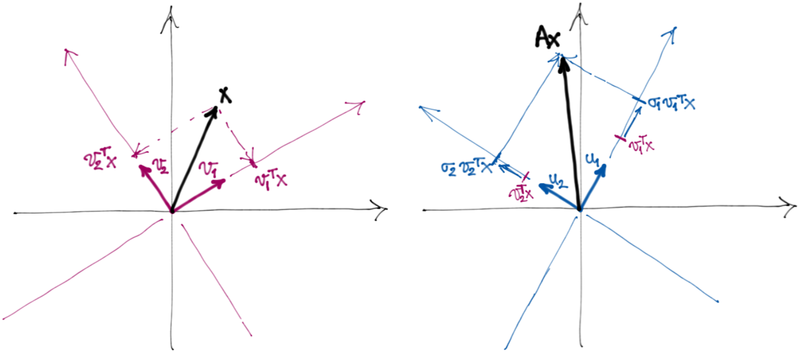
(2) streching

$$\mathbf{\Sigma}\mathbf{V}^T\mathbf{x} = \begin{pmatrix} \sigma_1\mathbf{v}_1^T\mathbf{x} \\ \vdots \\ \sigma_m\mathbf{v}_m^T\mathbf{x} \end{pmatrix}$$

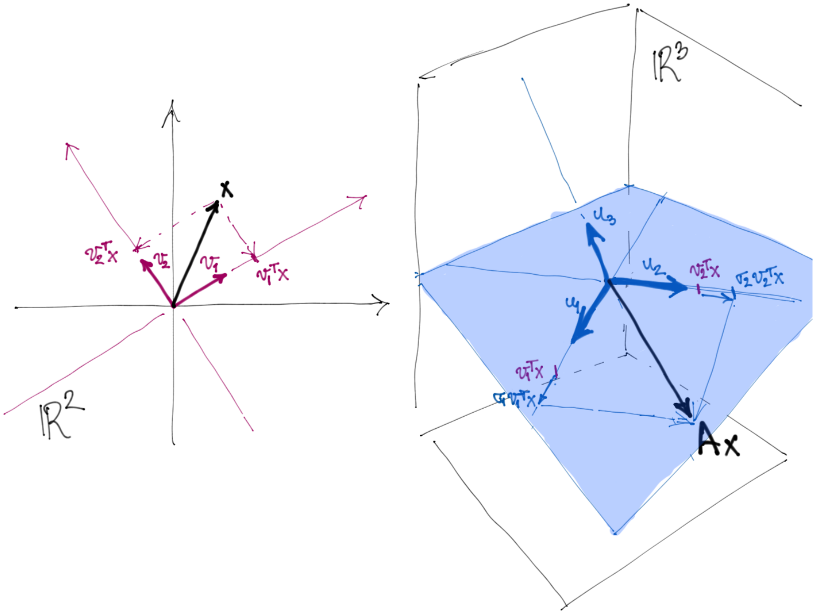
(3) change from basis $\mathbf{U}$

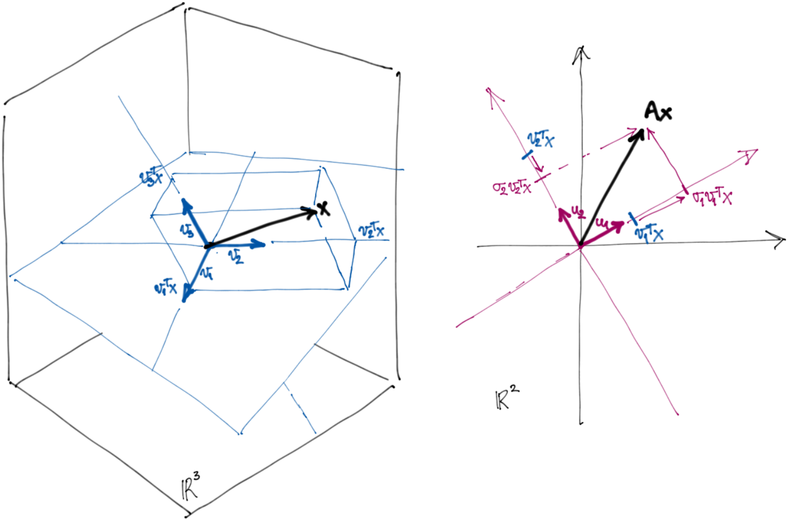
$$\mathbf{U}\mathbf{\Sigma}\mathbf{V}^T\mathbf{x} = \sum_{j=1}^m (\sigma_j \mathbf{v}_j^T\mathbf{x})\mathbf{u}_j$$

We don't use the last $n - m$ columns of $\mathbf{V}$.


x
$v_2^T x$
v_2
v_1
$v_1^T x$
Ax
$\sigma_1 v_1^T x$
$v_1^T x$
u_1
$\sigma_2 v_2^T x$
u_2
$v_2^T x$

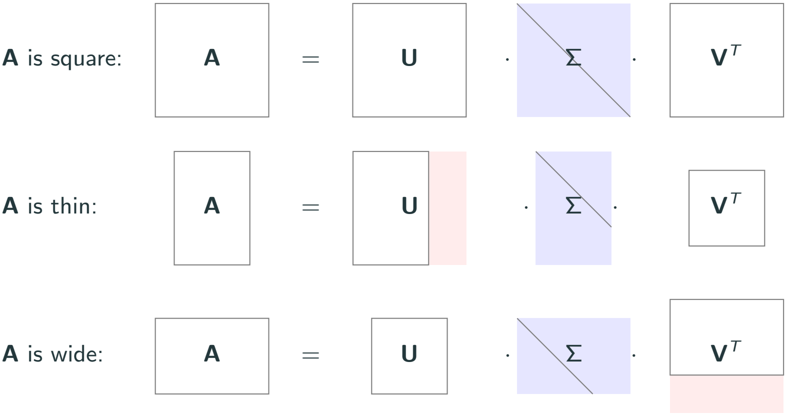
Geometric interpretation: $\mathbb{R}^2 \to \mathbb{R}^3$




$v_3^T x$
x
v_3
$v_2^T x$
v_2
v_1
$v_1^T x$
$\mathbb{R}^3$
Ax
$v_2^T x$
$\sigma_2 v_2^T x$
$\sigma_1 v_1^T x$
u_2
u_1
$v_1^T x$
$\mathbb{R}^2$

Economy-size SVD

The **economy-size SVD** is obtained by removing the columns from $\mathbf{U}$ or $\mathbf{V}$ beyond $k = \max\{n, m\}$ (because they are multiplied by zeros in $\mathbf{\Sigma}$).



Economy-size SVD

The **economy-size SVD** is obtained by removing the columns from **U** or **V** beyond $k = \max\{n, m\}$ (because they are multiplied by zeros in $\boldsymbol{\Sigma}$).

A is square: $\mathbf{A} = \mathbf{U} \cdot \Sigma \cdot \mathbf{V}^T$

A is thin: $\mathbf{A} = \mathbf{U}_{1:n} \cdot \Sigma_{1:n,:} \cdot \mathbf{V}^T$

A is wide: $\mathbf{A} = \mathbf{U} \cdot \Sigma_{:,1:m} \cdot \mathbf{V}^T_{1:m}$

Compact SVD

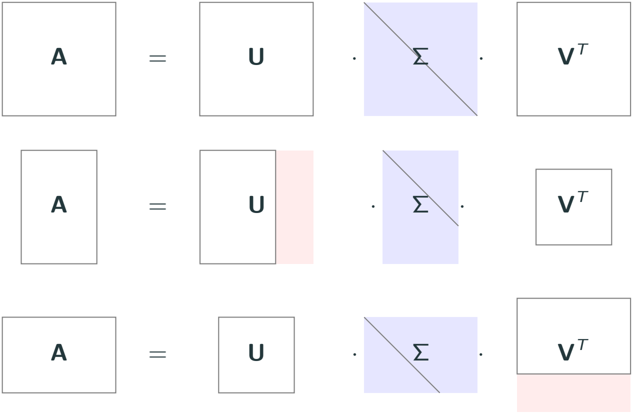
The **compact SVD** is obtained by removing the columns from $\mathbf{U}$ or $\mathbf{V}$ beyond $r = \text{rank}(\mathbf{A})$ (also because they are multiplied by zeros in $\mathbf{\Sigma}$).

In these examples, we suppose that $\mathbf{A}$ rank is deficient (i.e. $r < \min\{m, n\}$).

$\mathbf{A}$ is square:

$\mathbf{A}$ is thin:

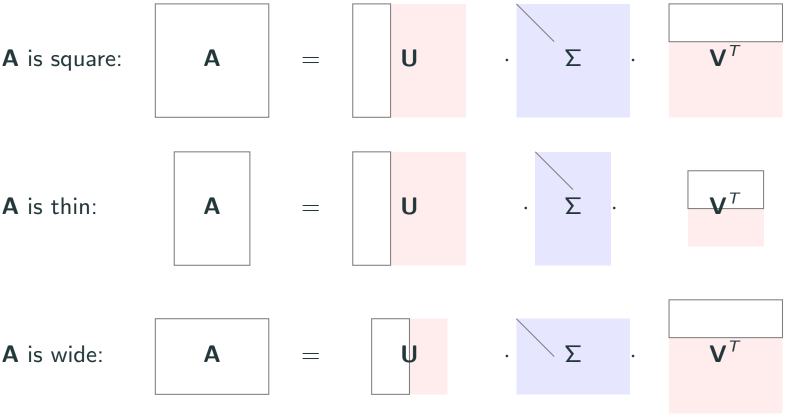
$\mathbf{A}$ is wide:



Compact SVD

The **compact SVD** is obtained by removing the columns from $\mathbf{U}$ or $\mathbf{V}$ beyond $r = \text{rank}(\mathbf{A})$ (also because they are multiplied by zeros in $\boldsymbol{\Sigma}$).

In these examples, we suppose that $\mathbf{A}$ rank is deficient (i.e. $r < \min\{m, n\}$).



Compact SVD

The **compact SVD** is obtained by removing the columns from $\mathbf{U}$ or $\mathbf{V}$ beyond $r = \text{rank}(\mathbf{A})$ (also because they are multiplied by zeros in $\boldsymbol{\Sigma}$).

In these examples, we suppose that $\mathbf{A}$ rank is deficient (i.e. $r < \min\{m, n\}$).

A is square: $\mathbf{A} = \mathbf{U}_{1:r} \cdot \Sigma_{1:r,1:r} \cdot \mathbf{V}_{1:r}^T$

A is thin: $\mathbf{A} = \mathbf{U}_{1:r} \cdot \Sigma_{1:r,1:r} \cdot \mathbf{V}_{1:r}^T$

A is wide: $\mathbf{A} = \mathbf{U}_{1:r} \cdot \Sigma_{1:r,1:r} \cdot \mathbf{V}_{1:r}^T$

Computing the SVD

Consder the SVD of $\mathbf{A}$, $m \times n$. Let us compute $\mathbf{A}^T\mathbf{A}$ and $\mathbf{A}\mathbf{A}^T$ in terms of the SVD. Suppose that $\mathbf{A}$ is a thin matrix:

$$\mathbf{A}^T \cdot \mathbf{A} = \mathbf{V} \cdot \Sigma^T \cdot \underbrace{\mathbf{U}^T \cdot \mathbf{U}}_{\mathbf{U}^T\mathbf{U}=\mathbf{I}_n} \cdot \Sigma \cdot \mathbf{V}^T$$

$$\mathbf{A} \cdot \mathbf{A}^T = \mathbf{U} \cdot \Sigma \cdot \underbrace{\mathbf{V}^T \cdot \mathbf{V}}_{\mathbf{V}^T\mathbf{V}=\mathbf{I}_m} \cdot \Sigma^T \cdot \mathbf{U}^T$$

Computing the SVD

Consder the SVD of $\mathbf{A}$, $m \times n$. Let us compute $\mathbf{A}^T\mathbf{A}$ and $\mathbf{A}\mathbf{A}^T$ in terms of the SVD. Suppose that $\mathbf{A}$ is a thin matrix:

$$\mathbf{A}^T\mathbf{A} = \mathbf{V} \cdot \underbrace{\Sigma^T \cdot \Sigma}_{\Sigma^T\Sigma=\Lambda_{\mathbf{V}}} \cdot \mathbf{V}^T$$

$$\mathbf{A}\mathbf{A}^T = \mathbf{U} \cdot \underbrace{\Sigma \cdot \Sigma^T}_{\Sigma\Sigma^T=\Lambda_{\mathbf{U}}} \cdot \mathbf{U}^T$$

Computing the SVD

Consder the SVD of $\mathbf{A}$, $m \times n$. Let us compute $\mathbf{A}^T\mathbf{A}$ and $\mathbf{A}\mathbf{A}^T$ in terms of the SVD. Suppose that $\mathbf{A}$ is a thin matrix:

$$\mathbf{A}^T\mathbf{A} = \mathbf{V} \cdot \Lambda_{\mathbf{V}} \cdot \mathbf{V}^T$$

$$\mathbf{A}\mathbf{A}^T = \mathbf{U} \cdot \Lambda_{\mathbf{U}} \cdot \mathbf{U}^T$$

These are **eigen-decompositions** of symmetric matrices!

Computing the SVD

- The eigenvectors of $\mathbf{A}^T\mathbf{A}$ are the right singular vectors $\mathbf{v}_i$ of $\mathbf{A}$

- The eigenvectors of $\mathbf{A}\mathbf{A}^T$ are the left singular vectors $\mathbf{u}_i$ of $\mathbf{A}$

- The eigenvalues λ_i of $\mathbf{A}\mathbf{A}^T$ (or of $\mathbf{A}^T\mathbf{A}$) are the singular vectors squared σ_i^2

$$\sigma_i = \sqrt{\lambda_i} \quad i = 1, ..., k = \min\{m, n\}.$$

Pseudo-inverse of a matrix

Can we use the SVD to invert a matrix?

We want to solve a linear equation $\mathbf{Ax} = \mathbf{b}$. Let's use the SVD: $\mathbf{U\Sigma V}^T\mathbf{x} = \mathbf{b}$.

1. We start from $\mathbf{x} \in \mathbb{R}^n$
2. Compute coordinates of $\mathbf{x}$ in basis $\mathbf{V}$
3. Scale them by singular values σ_i
4. Construct $\mathbf{b}$ using scaled coordinates on basis $\mathbf{U}$

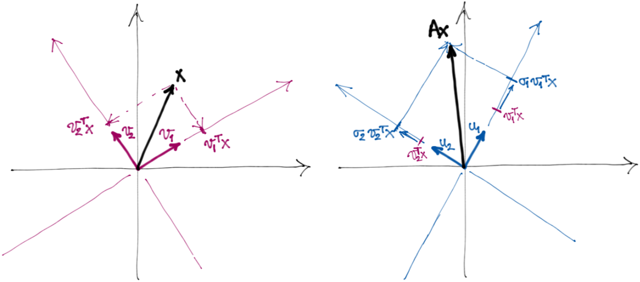
What if we do everything in the opposite direction?

1. We start from $\mathbf{b} \in \mathbb{R}^m$
2. Compute coordinates of $\mathbf{b}$ in basis $\mathbf{U}$
3. Scale them by inverse of singular values σ_i^{-1}
4. Construct $\mathbf{x}$ using scaled coordinates on basis $\mathbf{V}$

This works if $\mathbf{A}$ is square and invertible! Then we can compute $\mathbf{x}$ as follows:

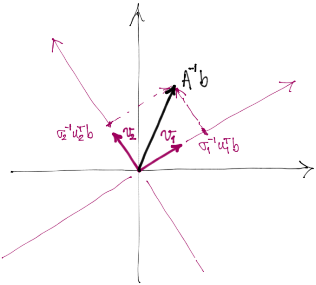
$$\mathbf{x} = \mathbf{V\Sigma}^{-1}\mathbf{U}^T\mathbf{b}.$$

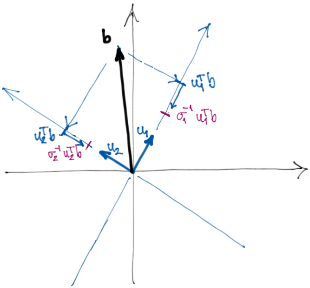
Computing the inverse using the SVD: $\mathbb{R}^2 \to \mathbb{R}^2$.



If $\mathbf{A}$ is square and invertible, $\mathbf{\Sigma}$ is square and invertible too, and we can compute $\mathbf{x}$ as follows:

$$\mathbf{x} = \mathbf{V}\mathbf{\Sigma}^{-1}\mathbf{U}^T\mathbf{b}.$$

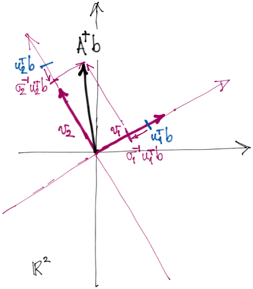


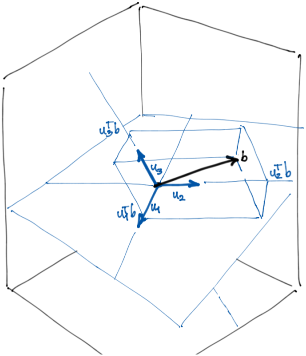


If $\mathbf{A}$ is square and invertible, $\mathbf{\Sigma}$ is square and invertible too, and we can compute $\mathbf{x}$ as follows:

$$\mathbf{x} = \mathbf{V}\mathbf{\Sigma}^{-1}\mathbf{U}^T\mathbf{b}.$$

Computing the inverse using the SVD: $\mathbb{R}^2 \to \mathbb{R}^3$.

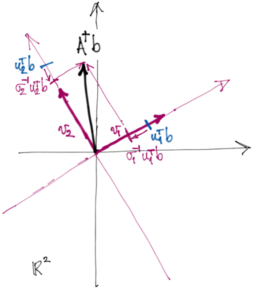


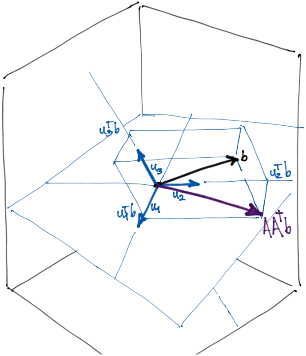


In this case we have a problem: the function $\mathbf{x} \mapsto \mathbf{Ax}$ is not **surjective** (it does not cover the entire output space). We can compute the **least squares solution** however, as follows:

$$\mathbf{x} = \mathbf{V} \underbrace{\begin{pmatrix} \sigma_1^{-1} & 0 & 0 \\ 0 & \sigma_2^{-1} & 0 \end{pmatrix}}_{\text{we'll call this matrix } \boldsymbol{\Sigma}^\dagger} \mathbf{U}^T\mathbf{b} = \underbrace{\mathbf{V}\boldsymbol{\Sigma}^\dagger\mathbf{U}^T}_{\text{...and this matrix } \mathbf{A}^\dagger} \mathbf{b} = \mathbf{A}^\dagger\mathbf{b}.$$

Computing the inverse using the SVD: $\mathbb{R}^2 \to \mathbb{R}^3$.

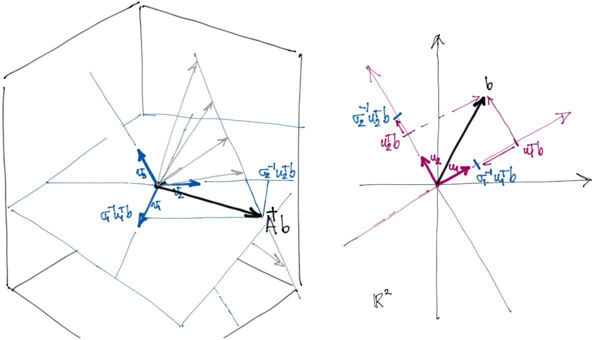




In this case we have a problem: the function $\mathbf{x} \mapsto \mathbf{Ax}$ is not **surjective** (it does not cover the entire output space). We can compute the **least squares solution** however, as follows:

$$\mathbf{x} = \mathbf{V} \underbrace{\begin{pmatrix} \sigma_1^{-1} & 0 & 0 \\ 0 & \sigma_2^{-1} & 0 \end{pmatrix}}_{\text{we'll call this matrix } \mathbf{\Sigma}^\dagger} \mathbf{U}^T \mathbf{b} = \underbrace{\mathbf{V}\mathbf{\Sigma}^\dagger \mathbf{U}^T}_{\text{...and this matrix } \mathbf{A}^\dagger} \mathbf{b} = \mathbf{A}^\dagger \mathbf{b}.$$

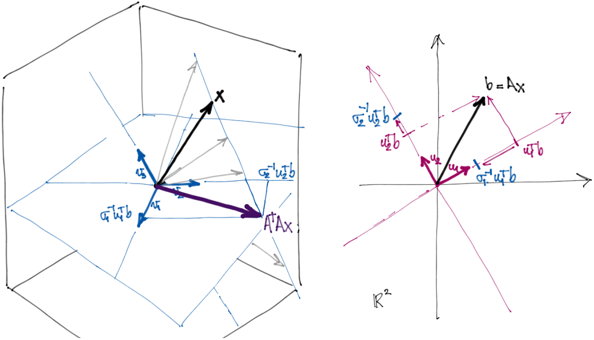
Computing the inverse using the SVD: $\mathbb{R}^3 \to \mathbb{R}^2$.



Now the function $\mathbf{x} \mapsto \mathbf{Ax}$ is not **injective** (there are many solutions, shown in gray). We can compute the **solution with smallest norm** however, as follows:

$$\mathbf{x} = \mathbf{V}\begin{pmatrix} \sigma_1^{-1} & 0 \\ 0 & \sigma_2^{-1} \\ 0 & 0 \end{pmatrix}\mathbf{U}^T\mathbf{b} = \mathbf{V}\boldsymbol{\Sigma}^\dagger\mathbf{U}^T\mathbf{b} = \mathbf{A}^\dagger\mathbf{b}.$$

Computing the inverse using the SVD: $\mathbb{R}^3 \to \mathbb{R}^2$.



Now the function $\mathbf{x} \mapsto \mathbf{Ax}$ is not **injective** (there are many solutions, shown in gray). We can compute the **solution with smallest norm** however, as follows:

$$\mathbf{x} = \mathbf{V}\begin{pmatrix} \sigma_1^{-1} & 0 \\ 0 & \sigma_2^{-1} \\ 0 & 0 \end{pmatrix}\mathbf{U}^T\mathbf{b} = \mathbf{V\Sigma}^\dagger\mathbf{U}^T\mathbf{b} = \mathbf{A}^\dagger\mathbf{b}.$$

Diagonal matrix $\mathbf{\Sigma}$ and its Penrose-Moore pseudo-inverse $\mathbf{\Sigma}^\dagger$:

$$\begin{array}{rl} \mathbf{\Sigma} \text{ is } m \times n \text{ diagonal} & \longrightarrow \mathbf{\Sigma}^\dagger \text{ is } {\color{red} n \times m} \text{ diagonal} \\ r \text{ non-zero diagonal elements } \sigma_1, ..., \sigma_r & \longrightarrow r \text{ non-zero diagonal elements } \frac{1}{\sigma_1}, ..., \frac{1}{\sigma_r} \end{array}$$

$$\mathbf{\Sigma} = \begin{pmatrix} \sigma_1 & \cdots & 0 & & \dots & 0 \\ \vdots & \ddots & \vdots & & & \vdots \\ 0 & \cdots & \sigma_r & & & \\ \vdots & & & & & \vdots \\ 0 & \dots & & & \dots & 0 \end{pmatrix} \longrightarrow \mathbf{\Sigma}^\dagger = \begin{pmatrix} \sigma_1^{-1} & \cdots & 0 & \dots & 0 \\ \vdots & \ddots & \vdots & & \vdots \\ 0 & \cdots & \sigma_r^{-1} & & \\ & & & & \\ \vdots & & & & \vdots \\ 0 & \dots & & \dots & 0 \end{pmatrix}$$

Definition of pseudo-inverse II: any matrix

The psuedo-inverse of any matrix is defined via the SVD.

Suppose that **A** is an $m \times n$ with SVD given by

$$\mathbf{A} = \mathbf{U} \cdot \Sigma \cdot \mathbf{V}^T$$

then its pseudo inverse is given by

$$\mathbf{A}^\dagger = \mathbf{V} \cdot \Sigma^\dagger \cdot \mathbf{U}^T$$

Least squares and the pseudo-inverse

Let's plug the SVD of $\mathbf{A}$: $\mathbf{A} = \mathbf{U\Sigma V}^T$, in the solution of the least squares.

$$
\begin{aligned}
\mathbf{x}^* &= (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{b} \\
&= (\mathbf{V\Sigma}^T\mathbf{U}^T\mathbf{U\Sigma V}^T)^{-1}\mathbf{A}^T\mathbf{b} \\
&= (\mathbf{V\Sigma}^T\mathbf{\Sigma V}^T)^{-1}\mathbf{A}^T\mathbf{b} \\
&= (\mathbf{V}^T)^{-1}(\mathbf{\Sigma}^T\mathbf{\Sigma})^{-1}\mathbf{V}^{-1}\mathbf{A}^T\mathbf{b} \\
&= \mathbf{V}(\mathbf{\Sigma}^T\mathbf{\Sigma})^{-1}\mathbf{V}^T\mathbf{A}^T\mathbf{b} \\
&= \mathbf{V}(\mathbf{\Sigma}^T\mathbf{\Sigma})^{-1}\mathbf{V}^T\mathbf{V\Sigma}^T\mathbf{U}^T\mathbf{b} \\
&= \mathbf{V}(\mathbf{\Sigma}^T\mathbf{\Sigma})^{-1}\mathbf{\Sigma}^T\mathbf{U}^T\mathbf{b} \\
&= \mathbf{V\Sigma}^\dagger\mathbf{U}^T\mathbf{b} \\
&= \mathbf{A}^\dagger\mathbf{b}
\end{aligned}
$$

We just showed that **if $\mathbf{A}^T\mathbf{A}$ is invertible** $\mathbf{A}^\dagger = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}$.

Properties about the pseudo-inverse

Let $\mathbf{A}$ be a $m \times n$ matrix:

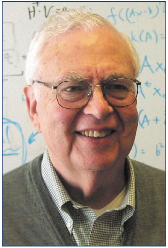
- If $\mathbf{A}$ is invertible, then $\mathbf{A}^\dagger = \mathbf{A}^{-1}$
- If $\mathbf{A}$ is a matrix of zeros, then $\mathbf{A}^\dagger = \mathbf{A}^T$
- $\left(\mathbf{A}^\dagger\right)^\dagger = \mathbf{A}$
- $(\alpha \mathbf{A})^\dagger = \alpha^{-1}\mathbf{A}^\dagger$
- If $\mathbf{A}$ is full rank and $m \geq n$, then $\mathbf{A}^\dagger = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T$
- $\mathbf{A}^\dagger = (\mathbf{A}^T\mathbf{A})^\dagger\mathbf{A}^T$
- In general $(\mathbf{AB})^\dagger \neq \mathbf{A}^\dagger\mathbf{B}^\dagger$
- $\mathbf{AA}^\dagger$ projects onto the column space of $\mathbf{A}$
- $\mathbf{A}^\dagger\mathbf{A}$ projects onto the row space of $\mathbf{A}$ (the orthogonal complement of the kernel of $\mathbf{A}$)

More about the SVD

The SVD has a lot of applications in data analysis. We barely scratched the surface.

- Low-rank matrix approximation
- Principal component analysis (correlation structure of data)
- Google page-rank
- Recommendation systems

Recommended: watch Steve Brunton SVD series



Any questions?