SANS-MIRI: Review of linear algebra and applications to data science

Jorge Garcia Vidal, Jose M. Barcelo Ordinas and Pau Ferrer Cid

July 17, 2023

The course SANS (Statistical Analysis of Networks and Systems) belongs to the Master MIRI (Master of Innovation and Research in Computer Science) of the Faculty of Computer Science of Barcelona. The course is an introduction to some mathematical foundations used in data science. The course content includes an introduction to probability, linear algebra, and estimation.

These lecture notes of the course are devoted to linear algebra concepts applied to data science, and is divided into the following topics:

1. Basics in Linear Algebra: vector spaces, matrices and applications (linear equations and least squares equations);
2. Eigendecomposition of square matrices: eigenvectors and eigenvalues, positive definite matrices and the trace operator;
3. Quadratic forms. Multivariate Gaussian distribution;
4. Eigendecomposition of square matrices: singular value decomposition, pseudoinverses, matrix norms, Eckart-Young approximation (low-rank approximation of matrices);
5. Principal component analysis (PCA), the eigenfaces problem;
6. Fourier Transform and its applications;
7. Graph signal processing (GSP) and its applications.

Linear Algebra is a classical topic, and there are many very good books covering the material that we need for our course at different levels of deep. Two books that we especially like are: ”Introduction to Linear Algebra” by Gilbert Strang, and ”Linear Algebra and Learning from Data” by Strang, Gilbert. Other books related to this course is ”Data-Driven Science and Engineering: Machine Learning, Dynamical Systems, and Control” by Steven L. Brunton and J. Nathan Kutz, or ” Introduction to Applied Linear Algebra – Vectors, Matrices, and Least Squares” by Stephen Boyd and Lieven Vandenberghe.

1 Some basic facts that you probably know about vectors and linear maps

1.1 Vector spaces and sub-spaces

Let V be a set in which we have defined the addition operation and the multiplication by a scalar (in this course, the scalars will be usually real numbers). We say that V is a **vector space** if the addition and scalar multiplication operations satisfy the following properties:

$\forall \boldsymbol{u}, \boldsymbol{v}, \boldsymbol{w} \in V$:

- Associativity: $\boldsymbol{u} + (\boldsymbol{v} + \boldsymbol{w}) = (\boldsymbol{u} + \boldsymbol{v}) + \boldsymbol{w}$,
- Commutativity: $\boldsymbol{u} + \boldsymbol{v} = \boldsymbol{v} + \boldsymbol{u}$,
- Additive identity: $\exists \mathbf{0} \in V$ such that $\boldsymbol{v} + \mathbf{0} = \boldsymbol{v}$, ,
- Existence of inverse: $\forall \boldsymbol{v} \in V, \exists -\boldsymbol{v} \in V$ such that $\boldsymbol{v} + (-\boldsymbol{v}) = \mathbf{0}$.

$\forall\ a, b \in \mathbb{R}$ and $\forall \boldsymbol{u}, \boldsymbol{v} \in V$:

- Associativity of scalar multiplication: $a(b\boldsymbol{v}) = (ab)\boldsymbol{v}$,
- Scalar multiplication identity: $1\boldsymbol{v} = \boldsymbol{v}$,
- Distributivity of scalar sums: $(a + b)\boldsymbol{v} = a\boldsymbol{v} + b\boldsymbol{v}$,
- Distributivity of vector sums: $a(\boldsymbol{u} + \boldsymbol{v}) = a\boldsymbol{u} + a\boldsymbol{v}$.

The elements of a vector space are called **vectors**. Some examples of **vector spaces** are:

Example 1.1 (Coordinate space) *The set of vectors* $\mathbf{x} \in \mathbb{R}^n$ *with coordinates* $\mathbf{x} = (x_1, ..., x_n)$ *and* $t \in \mathbb{R}$ *is a vector space.*

Example 1.2 (Set of matrices) *The set of matrices* $\mathbf{A} \in \mathbb{R}^{m \times n}$ *is a vector space over* $\mathbb{R}$*, (addition of matrices and multiplication of scalar over matrices).*

Example 1.3 (Set of polynomials) *The set of polynomials* $\mathbf{P}_n$ *over* $\mathbb{R}$ *(coefficients in* $\mathbb{R}$*) and of order less or equal of* n *is a vector space.*

Example 1.4 (Set of continuous functions) *The set of continuous functions* $\mathbf{f}\colon \mathbb{R}^n \to \mathbb{R}$*, where* $(\mathbf{f} + \mathbf{g})(\mathbf{x}) = \mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x})$ *and* $(a\mathbf{f})(\mathbf{x}) = a\mathbf{f}(\mathbf{x})$ *is a vector space.*

If S is a subset of V ($S \subset V$) which is closed respect the operations of sum of vectors (if $\mathbf{u}, \mathbf{v} \in S$, then $\mathbf{u} + \mathbf{v} \in S$), multiplication by an scalar (if $\mathbf{u} \in S$ and a scalar then $a\mathbf{u} \in S$), and the zero vector is in S, we say that S is a *vector subspace* of V. Some examples/non-examples of **vector subspaces V** are:

Example 1.5 *A line or a plane containing the origin is a vector subspace.*

Example 1.6 *A line or a plane non containing the origin is not a vector subspace.*

Example 1.7 *A quadrant is not a vector subspace (fails to be close under scalar multiplication).*

Example 1.8 *A circle is not a vector subspace (fails to be close under scalar or vector multiplication and does not contain the zero).*

1.2 Linear combinations and independence

If we have a set of vectors $\{\boldsymbol{v_1}, ..., \boldsymbol{v_k}\}$, a **linear combination** of these vectors is an expression of the form $\boldsymbol{v} = \sum_i a_i \boldsymbol{v_i}$ for some scalars a_i.

If we have a set of vectors belonging to a vector space, $\{\boldsymbol{v_1}, ..., \boldsymbol{v_k}\}$, the set of all linear combinations of these vectors is a vector subspace. This subspace is called:

$$span\{\boldsymbol{v_1}, ..., \boldsymbol{v_k}\} = \{\sum_{j=1}^{k} \beta_j \boldsymbol{v_j};\ with\ \ \beta_j \in \mathbb{R}\} \tag{1.2.1}$$

One vector $\boldsymbol{w}$ is **linearly independent** of a set of vectors $\{\boldsymbol{v_1}, ..., \boldsymbol{v_k}\}$ when $\boldsymbol{w}$ cannot be expressed as linear combination of the vectors $\{\boldsymbol{v_1}, ..., \boldsymbol{v_k}\}$, or in other words, when $\boldsymbol{w} \notin\ span\{\boldsymbol{v_1}, ..., \boldsymbol{v_k}\}$. A set of vectors $\{\boldsymbol{v_1}, ..., \boldsymbol{v_k}\}$ are linearly independent when the only linear combination that produces the vector $\mathbf{0}$ is the one with all coefficients equal to zero. Summarizing, let us assume that $\sum_{j=1}^{k} a_j \boldsymbol{v_j} = 0$; if some $a_j \neq 0$ then the vectors are linearly dependent (l.d.), while if all a_j=0 then the vectors are linearly independent (l.i.).

1.3 Bases and dimension

There are many examples of vector spaces: $\mathbb{R}^n$, $\mathbb{C}^n$, or $P_n(\mathbb{R})$, the set of polynomials of n degree with real coefficients. These spaces are examples of *finite dimensional* vector spaces. This means that there is a set of linearly independent vectors of V, $\{\boldsymbol{u_i}\}_{i=1,...,n}$ (a **base** of V) such that all other vectors of V can be expressed as $\boldsymbol{v} = \sum_{i=1}^{n} v_i \boldsymbol{u_i}$. A vector space has in general an infinite number of possible bases, but the number of elements in each of those basis is always the same. We say that this number n the **dimension** of V, $\dim(V) = n$. Moreover, we can use these coefficients to represent $\boldsymbol{v}$ as a column vector:

$$\boldsymbol{v} = [v_1, v_2, ..., v_n]^\top = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix} \tag{1.3.1}$$

The same definitions apply to vector subspaces.

Some vector spaces have *infinite* dimensions, for instance, $P_\infty(\mathbb{R})$, the set of polynomials of an arbitrary order, $C[0, 1]$, the set of continuous functions defined on the interval $[0, 1]$, or $L^2[0, 1]$, the set of square integrable functions f for which $\int_0^1 |f|^2 d\mu < \infty$. In this course, we will deal with the finite dimensional case only. The infinite dimensional case is studied in *functional analysis*, and is important, for instance, when dealing with stochastic processes.

1.4 Scalar product, orthogonality, and vector norms

The **scalar product** is an operation that takes two vectors $\boldsymbol{u}, \boldsymbol{v} \in V$ and returns a scalar; i.e., $\langle \cdot, \cdot \rangle \colon V \times V \to \mathbb{R}$. For this operation to qualify as a scalar product, it must fulfill the following properties:

- commutative: $\langle \boldsymbol{u}, \boldsymbol{v} \rangle = \langle \boldsymbol{v}, \boldsymbol{u} \rangle$,
- distributive: $\langle \boldsymbol{u}, \boldsymbol{v} + \boldsymbol{w} \rangle = \langle \boldsymbol{u}, \boldsymbol{v} \rangle + \langle \boldsymbol{u}, \boldsymbol{w} \rangle$,
- linearity in any argument: $\langle a\boldsymbol{u}, \boldsymbol{v} \rangle = \langle \boldsymbol{u}, a\boldsymbol{v} \rangle = a\langle \boldsymbol{u}, \boldsymbol{v} \rangle$,
- positive definiteness: if $\boldsymbol{u} \neq 0$, then $\langle \boldsymbol{u}, \boldsymbol{u} \rangle > 0$.

The most commonly used definition of the scalar product for two column vectors $\boldsymbol{u}$ and $\boldsymbol{v}$ is:

$$\langle \boldsymbol{u}, \boldsymbol{v} \rangle = \boldsymbol{u}^\top \boldsymbol{v}. \tag{1.4.1}$$

Note, however, that we can define other scalar products. For instance, if $\boldsymbol{S}$ is a **symmetric positive definite matrix** (see section 2.6), we can define a scalar product as:

$$\langle \boldsymbol{u}, \boldsymbol{v} \rangle = \boldsymbol{u}^\top \boldsymbol{S} \boldsymbol{v}. \tag{1.4.2}$$

Another important example is the scalar product of two square matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ defined as (see section 3.3 for the definition of *trace*):

$$\langle \boldsymbol{A}, \boldsymbol{B} \rangle = tr(\boldsymbol{A}^\top \boldsymbol{B}). \tag{1.4.3}$$

A very important property of the scalar product is the **Cauchy-Schwarz inequality**:

$$|\langle \boldsymbol{u}, \boldsymbol{v}\rangle|^2 \leq \ \langle \boldsymbol{u}, \boldsymbol{u}\rangle \ \langle \boldsymbol{v}, \boldsymbol{v}\rangle. \tag{1.4.4}$$

We have equality only when $\boldsymbol{u} = a\boldsymbol{v}$ for some scalar a.

A simple proof of this important fact is the following: consider an arbitrary scalar a and two non-zero vectors $\boldsymbol{u}, \boldsymbol{v}$. Assume first that there is no scalar a for which $a\boldsymbol{u} = \boldsymbol{v}$. Then, the positive-definiteness property of the scalar product implies that for any a:

$$0 < \langle a\boldsymbol{u} - \boldsymbol{v}, a\boldsymbol{u} - \boldsymbol{v}\rangle = \langle \boldsymbol{u}, \boldsymbol{u}\rangle a^2 - 2\langle \boldsymbol{u}, \boldsymbol{v}\rangle a + \langle \boldsymbol{v}, \boldsymbol{v}\rangle. \tag{1.4.5}$$

But this second-degree polynomial on a is always non-negative for all a only when:

$$4(\langle \boldsymbol{u}, \boldsymbol{v}\rangle)^2 - 4\langle \boldsymbol{u}, \boldsymbol{u}\rangle\langle \boldsymbol{v}, \boldsymbol{v}\rangle < 0, \tag{1.4.6}$$

and from this, we arrive at the strict Cauchy-Schwarz inequality.

Assume now that there is a scalar a^* for which $a^*\boldsymbol{u} = \boldsymbol{v}$. In this case, we have

$$0 \leq \langle a\boldsymbol{u} - \boldsymbol{v}, a\boldsymbol{u} - \boldsymbol{v}\rangle = \langle \boldsymbol{u}, \boldsymbol{u}\rangle a^2 - 2\langle \boldsymbol{u}, \boldsymbol{v}\rangle a + \langle \boldsymbol{v}, \boldsymbol{v}\rangle. \tag{1.4.7}$$

with equality for $a = a^*$. This means that the second-degree polynomial in a has a double root at a^*, which is only possible if:

$$4(\langle \boldsymbol{u}, \boldsymbol{v}\rangle)^2 - 4\langle \boldsymbol{u}, \boldsymbol{u}\rangle\langle \boldsymbol{v}, \boldsymbol{v}\rangle = 0, \tag{1.4.8}$$

and from this, we arrive at the equality case of Cauchy-Schwarz inequality when $\boldsymbol{v}$ is colinear with $\boldsymbol{u}$.

We can define the angle α between two non-zero vectors $\boldsymbol{u}, \boldsymbol{v}$ as:

$$cos(\alpha) = \frac{\langle \boldsymbol{u}, \boldsymbol{v}\rangle}{\sqrt{\langle \boldsymbol{u}, \boldsymbol{u}\rangle \ \langle \boldsymbol{v}, \boldsymbol{v}\rangle}}. \tag{1.4.9}$$

Two non-zero vectors are **orthogonal** when its scalar product is zero:

$$\langle \boldsymbol{u}, \boldsymbol{v}\rangle = 0. \tag{1.4.10}$$

Thus, the angle between two orthogonal vectors is $\alpha = 90°$.

A *norm* is a function $\|\cdot\|$: $\mathbb{R}^n \to \mathbb{R}$, such that for each vector $\mathbf{v}\in\mathbb{R}^n$, the following conditions are satisfied:

- $\|\cdot\|$ is non-negative, $\|\mathbf{v}\| \geq 0$
- $\|\cdot\|$ is definite, $\|\mathbf{v}\| = 0$ iif $\mathbf{v} = 0$,

- $\|\cdot\|$ is homogeneous $\|(a\mathbf{v})\| = |a|\|\mathbf{v}\|$ with a scalar,
- $\|\cdot\|$ satisfies the triangle inequality (subadditivity property) $\|\mathbf{u}+\mathbf{v}\| \leq \|\mathbf{u}\| + \|\mathbf{v}\|$.

The **Euclidean norm** or also called **L2-norm** represents the length of a vector $\boldsymbol{v}$, and is defined as the non-negative number:

$$\|\boldsymbol{v}\| = +\sqrt{\langle \boldsymbol{v}, \boldsymbol{v}\rangle} \tag{1.4.11}$$

Other well-known norms are represented by a subscript $\|\cdot\|_p$, where $p\in\mathbb{R}_+$ ($\mathbb{R}_+$ means any real number $\geq$0). Examples of useful norms are:

- $\|\boldsymbol{v}\|_0$ (zero-norm or L0-norm) defined as the number of non-zero coordinates of vector $\boldsymbol{v}$ (or also as the Hamming distance of the vector from zero),
- $\|\boldsymbol{v}\|_1$ (Taxicab norm or Manhattan norm or sum-absolute norm) defined as $\|\boldsymbol{v}\|_1 = \sum_{i=1}^{n} |v_i|$,
- $\|\boldsymbol{v}\|_2$ (Euclidean norm or L2-norm) defined as $\|\boldsymbol{v}\|_2 = \sqrt{\sum_{i=1}^{n} |v_i|^2} = \sqrt{\boldsymbol{v}^\top \boldsymbol{v}}$,
- $\|\boldsymbol{v}\|_p$ (Lp-norm) defined as $\|\boldsymbol{v}\|_p = (\sum_{i=1}^{n} |v_i|^p)^{1/p}$,
- $\|\boldsymbol{v}\|_\infty$ (maximum norm or infinity norm or Chebyshev norm) defined as $\|\boldsymbol{v}\|_\infty = max_{i=1\ldots n}\{|v_i|\}$.

Example 1.9 (vector norms) *Let us assume vector* $\boldsymbol{v}$*=[1,0,0,4,2,-3]. Then,* $\|\boldsymbol{v}\|_0 = 4$*,* $\|\boldsymbol{v}\|_1 = 10$*,* $\|\boldsymbol{v}\|_2=\sqrt{30} = 5.477$*,* $\|\boldsymbol{v}\|_{3.2} = 4.558$ *and* $\|\boldsymbol{v}\|_\infty = 4$

The Cauchy-Schwarz inequality for the Euclidean norm reads:

$$|\boldsymbol{u}^\top \boldsymbol{v}| \leq \|\boldsymbol{u}\| \; \|\boldsymbol{v}\| \tag{1.4.12}$$

Let us define the **unit circle** or **unit ball** as the set of all vectors $\boldsymbol{v}$ of norm 1 ($\|\boldsymbol{v}\|_p = 1$). Then **plot, as an exercise**, the unit circle for norms $\|\boldsymbol{v}\|_0$, $\|\boldsymbol{v}\|_1$, $\|\boldsymbol{v}\|_2$, $\|\boldsymbol{v}\|_\infty$, and $\|\boldsymbol{v}\|_p$ for any p.

1.5 Linear maps and matrices

A **linear map** L is an application L: $V \rightarrow W$, where V (domain) and W (codomain) are vector spaces, that satisfies $\forall \boldsymbol{u}, \boldsymbol{v}\in V$, and $\forall a, b\in\mathbb{R}$:

$$L(\boldsymbol{u}+\boldsymbol{v}) = L(\boldsymbol{u}) + L(\boldsymbol{v}) \;\; and \;\; L(a\boldsymbol{u}) = aL(\boldsymbol{u}), \tag{1.5.1}$$

or (identical definition)

$$L(a\boldsymbol{u} + b\boldsymbol{v}) = aL(\boldsymbol{u}) + bL(\boldsymbol{v}). \tag{1.5.2}$$

For example; L(x$_1$,x$_2$)=(x$_1$+x$_2$,x$_1$,x$_2$+1) is not a linear map, as it fails in the condition $L(a\boldsymbol{u}) = aL(\boldsymbol{u})$. A linear map always transforms $\mathbf{0}$ to $\mathbf{0}$, i.e., $L(\mathbf{0}) = \mathbf{0}$.

Assume that $\{\boldsymbol{v_i}\}_{i=1,...,n}$ is a base of a n dimensional space V, while $\{\boldsymbol{w_i}\}_{i=1,...,m}$ is a base of a m dimensional space W. The image of the $\boldsymbol{u_i}$ base vector by the linear map L, i.e. $L(\boldsymbol{u_i})$, is a vector of W, that we can express in the base $\{\boldsymbol{w_i}\}_{i=1,...,m}$:

$$L(\boldsymbol{u_i}) = a_{1i}\boldsymbol{w_1} + a_{2i}\boldsymbol{w_2} + ... + a_{mi}\boldsymbol{w_m} \tag{1.5.3}$$

An $m{\times}n$ matrix A is an arrangement of these numbers a_{ij} into an $m{\times}n$ array $A = [a_{ij}]$. For an arbitrary vector $\boldsymbol{v}$, expressed as a column vector in the base $\{\boldsymbol{v_i}\}_{i=1,...,n}$, the product $A\boldsymbol{v}$ gives a result the vector $\boldsymbol{w}$, which is the image of the vector $\boldsymbol{v}$ by the linear map L expressed in the base $\{\boldsymbol{w_i}\}_{i=1,...,m}$.

This is a bit confusing at first. Let's think on the following example: $\{1, x, x^2\}$ is a possible base of $P_2(\mathbb{R})$. Using this base, we can represent the polynomial $p(x) = a + bx + cx^2$ by the $\mathbb{R}^3$ vector $p = [a, b, c]^\top$. Let us define the linear map $\frac{d}{dx} : P_2(\mathbb{R}) \rightarrow P_2(\mathbb{R})$ that assigns to a polynomial $p(x)$ its derivative (another polynomial). If we represent the polynomials in the image and domain sets by the same base $\{1, x, x^2\}$, we can thus represent this linear map by the matrix:

$$D = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 2 \\ 0 & 0 & 0 \end{bmatrix}. \tag{1.5.4}$$

Usually, we will use matrices to represent linear maps. Very often, we express a matrix A as an arrangement of its columns (or its rows) considered vectors. For instance if $A = [a_{ij}]$ and let us define the n column vectors $\boldsymbol{a_i} = [a_{i1}, ..., a_{im}]^\top$ for $i = 1, ..., n$. We can write the matrix as:

$$A = [\boldsymbol{a_1}, ..., \boldsymbol{a_n}] \tag{1.5.5}$$

Similarly, we can write A using its row vectors $\boldsymbol{r_i}^\top = [a_{1i}, ..., a_{ni}]$:

$$A = \begin{bmatrix} — & \boldsymbol{r_1}^\top & — \\ & \vdots & \\ — & \boldsymbol{r_m}^\top & — \end{bmatrix}. \tag{1.5.6}$$

If we have a matrix A and a column vector $\boldsymbol{v} = [v_1, ..., v_n]^\top$, we can express its matrix times vector product as

$$\boldsymbol{w} = A\boldsymbol{v} = [\boldsymbol{a_1}, ..., \boldsymbol{a_n}] \begin{bmatrix} v_1 \\ \vdots \\ v_n \end{bmatrix} = \sum_{i=1,..,n} v_i\boldsymbol{a_i}, \tag{1.5.7}$$

which is saying that the image vector $\boldsymbol{w}$ is a linear combination of the column vectors $\boldsymbol{a_i}$.

1.6 Rank of a matrix

The image f of all vectors of a subspace S of V by means of a matrix A is also a subspace T of W; $f{:}S{\subset}V \rightarrow T{\subset}W$.

In the special case when the subspace S is V itself, the generated subspace is span$\{\boldsymbol{a_1}, ..., \boldsymbol{a_n}\}$. The dimension of this subspace is **the rank** of the matrix $\boldsymbol{A}$, rank($\boldsymbol{A}$), and it is the number of linearly independent columns.

Very surprisingly, this number is also the number of linearly independent *rows* of the matrix $\boldsymbol{A}$, meaning that $\boldsymbol{A}$ *and* $\boldsymbol{A}^\top$ *have the same rank*.

In terms of rows and columns of a matrix $\boldsymbol{A}{\in}\mathbb{R}^{m\times n}$ and matrix $\boldsymbol{A}^\top{\in}\mathbb{R}^{n\times m}$, we can state the actions of an $m{\times}n$ matrix remembering that matrix $\boldsymbol{A}$ can be expressed as column vectors $\{\boldsymbol{a_1}, ..., \boldsymbol{a_n}\}$ or matrix $\boldsymbol{A}$ can be expressed as row vectors $\{\boldsymbol{r_1}^\top, ..., \boldsymbol{r_m}^\top\}$. In the same way matrix $\boldsymbol{A}^\top$ can be expressed as column vectors $\{\boldsymbol{r_1}, ..., \boldsymbol{r_m}\}$ or matrix $\boldsymbol{A}^\top$ can be expressed as row vectors $\{\boldsymbol{a_1}^\top, ..., \boldsymbol{a_n}^\top\}$.

For any matrix, we can associate several subspaces: the null space (NS), the column space (CS), the row space (RS), and the left null space (LNS).

Let us define the Range($\boldsymbol{A}$) or Im($\boldsymbol{A}$)=Image($\boldsymbol{A}$) as the image of the linear transformation $f{:}\mathbb{R}^n{\subset}V \rightarrow \mathbb{R}^m{\subset}W$:

$$Im(\boldsymbol{A}) = Range(\boldsymbol{A}) = \{\mathbf{y}{\in}\mathbb{R}^m | \boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} \; for \; some \; \mathbf{x}{\in}\mathbb{R}^n\} \tag{1.6.1}$$

and we can observe, Range($\boldsymbol{A}$)= Im($\boldsymbol{A}$) is a subspace of $\mathbb{R}^m$. The Im($\boldsymbol{A}^\top$) is defined in a similar way and is a subspace of $\mathbb{R}^n$.

$$Im(\boldsymbol{A}^\top) = Range(\boldsymbol{A}^\top) = \{\mathbf{x}{\in}\mathbb{R}^n | \boldsymbol{x} = \boldsymbol{A}^\top\boldsymbol{y} \; for \; some \; \mathbf{y}{\in}\mathbb{R}^m\} \tag{1.6.2}$$

and, Range($\boldsymbol{A}^\top$)= Im($\boldsymbol{A}^\top$) is a subspace of $\mathbb{R}^n$.

Define the **column space** of $\boldsymbol{A}$, CS($\boldsymbol{A}$), as the linear combination of its columns: CS($\boldsymbol{A}$)=$\{\boldsymbol{w}{\in}\mathbb{R}^m, \boldsymbol{w}{=}c_1\boldsymbol{a_1} + \cdots + c_n\boldsymbol{a_n}\}$, where $\boldsymbol{a_i}{\in}\mathbb{R}^m$. The CS($\boldsymbol{A}$) is then a subspace of $\mathbb{R}^m$. The CS($\boldsymbol{A}$) is the Range($\boldsymbol{A}$) or Im($\boldsymbol{A}$) of the linear transformation $f{:}\mathbb{R}^n{\subset}V \rightarrow \mathbb{R}^m{\subset}W$.

In a similar way, we define the **row space** of $\boldsymbol{A}$, RS($\boldsymbol{A}$), as the linear combination of its rows: RS($\boldsymbol{A}$)= $\{\boldsymbol{v}{\in}\mathbb{R}^n, \boldsymbol{w}{=}d_1\boldsymbol{r_1} + \cdots + d_n\boldsymbol{r_m}\}$, where $\boldsymbol{r_i}{\in}\mathbb{R}^n$. The RS($\boldsymbol{A}$) is then a subspace of $\mathbb{R}^n$, and then it is the Range($\boldsymbol{A}^\top$) or Im($\boldsymbol{A}^\top$).

We know that dim(RS($\boldsymbol{A}$))= rank($\boldsymbol{A}$) and that dim(CS($\boldsymbol{A}$))= rank($\boldsymbol{A}$), which makes dim(RS($\boldsymbol{A}$))=dim(CS($\boldsymbol{A}$))= rank($\boldsymbol{A}$). Since the columns of $\boldsymbol{A}$ are the rows of $\boldsymbol{A}^\top$, finding a basis for CS($\boldsymbol{A}$) is equivalent to finding a basis for RS($\boldsymbol{A}^\top$).

The null space NS($\boldsymbol{A}$) or Ker(f) is defined as:

$$NS(\boldsymbol{A}) = \{\mathbf{x}{\in}\mathbb{R}^n | \boldsymbol{A}\boldsymbol{x} = 0\} \tag{1.6.3}$$

and we can observe that NS($\boldsymbol{A}$) is a subspace of $\mathbb{R}^n$. The NS($\boldsymbol{A}^\top$) (also called the Left Null Space of $\boldsymbol{A}$, LNS($\boldsymbol{A}$) or the CoKer(f)) is defined in a similar way and is a subspace of $\mathbb{R}^m$.

$$LNS(\boldsymbol{A}) = NS(\boldsymbol{A}^\top) = \{\mathbf{y}{\in}\mathbb{R}^m | \boldsymbol{A}^\top \boldsymbol{y} = 0\} = \{\mathbf{y}{\in}\mathbb{R}^m | \boldsymbol{y}^\top \boldsymbol{A} = 0\} \tag{1.6.4}$$

If $\boldsymbol{A}$=[$\boldsymbol{a_1}, ..., \boldsymbol{a_n}$] is a column partitioning, and rank($\boldsymbol{A}$)=span{$\boldsymbol{a_1}, ..., \boldsymbol{a_n}$}. Now, since the rank($\boldsymbol{A}$) is the dimension of the image; rank($\boldsymbol{A}$)= dim(Range($\boldsymbol{A}$))= dim(Im($\boldsymbol{A}$))=dim(CS($\boldsymbol{A}$)), and we know that rank($\boldsymbol{A}$)=rank($\boldsymbol{A}^\top$). We say that the matrix $\mathbf{A}{\in}\mathbb{R}^{m\times n}$ is *rank deficient* if rank($\boldsymbol{A}$)<min{m,n}, and therefore:

$$dim(RS(\boldsymbol{A})) + dim(NS(\boldsymbol{A})) = rank(\boldsymbol{A}) + dim(NS(\boldsymbol{A})) = n \tag{1.6.5}$$

We can state that RS($\boldsymbol{A}$)= Im($\boldsymbol{A}^\top$) $\subset\mathbb{R}^n \perp$ NS($\boldsymbol{A}$)$\subset\mathbb{R}^n$. On the other hand, in terms of the CS and LNS:

$$CS(\boldsymbol{A}) = Im(\boldsymbol{A}){\subset}\mathbb{R}^m \perp LNS(\boldsymbol{A}) = NS(\boldsymbol{A}^\top){\subset}\mathbb{R}^m \tag{1.6.6}$$

and

$$dim(CS(\boldsymbol{A})) + dim(LNS(\boldsymbol{A})) = rank(\boldsymbol{A}) + dim(LNS(\boldsymbol{A})) = m \tag{1.6.7}$$

You can see this in the following way: $\boldsymbol{A}^\top\boldsymbol{y}$=0 (or $\boldsymbol{y}^\top\boldsymbol{A}$=0), so rows of $\boldsymbol{A}^\top$ multiplied by vectors $\boldsymbol{y}$ in the null space are equal to 0 (or vectors $\boldsymbol{y}$ of the left null space multiplied by columns of $\boldsymbol{A}$ are equal to 0), so they are orthogonal.

Example 1.10 (Row space, column space, null space and left null space) *Let's see an example of a matrix $\boldsymbol{A}{\in}\mathbb{R}^{m\times n}$, with m=3 and n=4:*

$$\boldsymbol{A} = \begin{bmatrix} 2 & 1 & 1 & 1 \\ 3 & 0 & 0 & 2 \\ 1 & 3 & 6 & 0 \end{bmatrix}$$

The rank of this matrix is r=3. It has 3 row vectors; $\boldsymbol{r}_1^\top = [2,1,2,1]$, $\boldsymbol{r}_2^\top = [3,0,0,2]$ and $\boldsymbol{r}_3^\top = [1,3,6,0]$. They form a basis of the row space that is a subspace of $\mathbb{R}^n = \mathbb{R}^4$. The null space has 1 vector with basis $\boldsymbol{r}_{ns} = [-2/3, 4/9, -1/9, 1]$. Now, vectors $\boldsymbol{r}_1$, $\boldsymbol{r}_1$, $\boldsymbol{r}_1$ and $\boldsymbol{r}_{ns}$ form a basis of the space $\mathbb{R}^n = \mathbb{R}^4$.

Matrix $\boldsymbol{A}$ has 4 column vectors; $\boldsymbol{a}_1^\top = [2,3,1]$, $\boldsymbol{a}_2^\top = [1,0,3]$, $\boldsymbol{a}_3^\top = [1,0,6]$ and $\boldsymbol{a}_4^\top = [1,2,0]$. vectors $\boldsymbol{a}_2$ and $\boldsymbol{a}_3$ are linear dependent, thus, vectors $\boldsymbol{a}_1$, $\boldsymbol{a}_2$ and $\boldsymbol{a}_4$ form a basis of the column space that is a subspace of $\mathbb{R}^m = \mathbb{R}^3$, and the left null space only contains vector $\boldsymbol{a}_{lns} = [0,0,0]$.

1.7 Applications

Let us see some applications where these concepts appear.

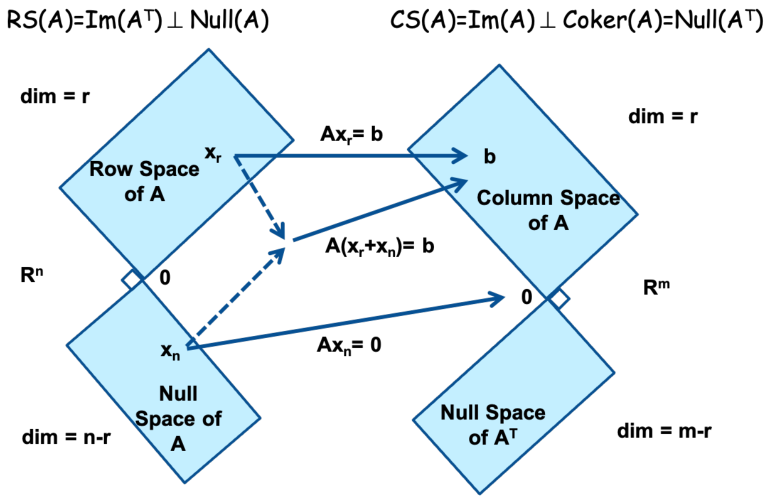


Figure 1: Fundamental theorem of linear algebra.

1.7.1 Linear equations

Let us consider the Figure 1, where we have plotted the orthogonality of the subspaces and we consider the system of equations:

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b} \tag{1.7.1}$$

where $\boldsymbol{A}\in\mathbb{R}^{m\times n}$, $\boldsymbol{x}\in\mathbb{R}^{n}$ and $\boldsymbol{b}\in\mathbb{R}^{m}$.

We can observe that the action of matrix $\boldsymbol{A}$ over a vector $\boldsymbol{x_r}$ in the RS is to transform it in a vector $\boldsymbol{b}$ in the CS.

On the other hand, we can observe that the action of matrix $\boldsymbol{A}$ over a vector $\boldsymbol{x_n}$ in the NullSpace is to transform it in vector $\mathbf{0}$.

An interesting property is that a vector $\boldsymbol{x_p} = \boldsymbol{x_r} + \boldsymbol{x}_{ns}$ that is the sum of a vector in the RS and a vector of the NullSpace goes to the CS, since $\boldsymbol{A}\boldsymbol{x_p}$= $\boldsymbol{A}(\boldsymbol{x_r} + \boldsymbol{x}_{ns}) = \boldsymbol{A}\boldsymbol{x_r} + \boldsymbol{A}\boldsymbol{x}_{ns} = \boldsymbol{b} + \mathbf{0} = \boldsymbol{b}$.

The conclusion of these facts are that the **particular solution of $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$** is $\boldsymbol{x_r}$, the **homogeneous solution of $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$** is $\boldsymbol{x}_{ns}$, and the **general solution of $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$** is $\boldsymbol{x_p} = \boldsymbol{x_r} + \boldsymbol{x}_{ns}$.

1.7.2 Least squares equations

Let us consider the Figure 2, where we have plotted the orthogonality of the subspaces and we consider the system of equations:

$$\boldsymbol{Ax} = \boldsymbol{b}$$

where $\boldsymbol{A} \in \mathbb{R}^{m \times n}$, $\boldsymbol{x} \in \mathbb{R}^n$ and $\boldsymbol{b} \in \mathbb{R}^m$. The objective is to find a vector $\boldsymbol{x}$ that satisfies the equation $\boldsymbol{Ax} = \boldsymbol{b}$. Let us consider three cases:

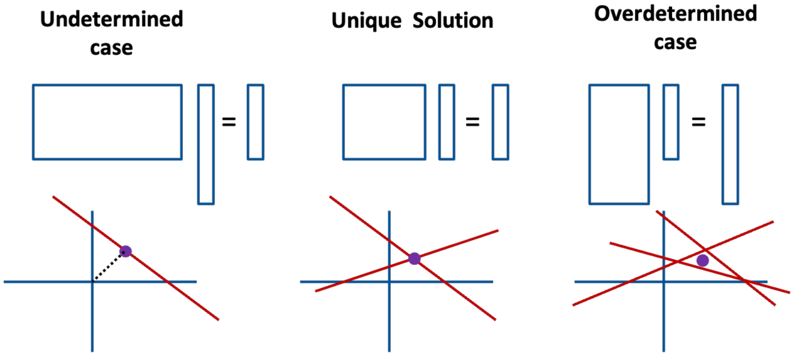


Figure 2: Least squares equations.

- **Undetermined case:** This is the case in which there are more variables than equations, meaning $m < n$. In this case, there exists infinity solutions, since $\hat{\boldsymbol{x}} = \{\boldsymbol{x}\text{: } \boldsymbol{Ax}=\boldsymbol{b}\} = \{\boldsymbol{x_r} + \boldsymbol{x_{ns}}\}$ with $\boldsymbol{x_r} \in RS(A)$ and $\boldsymbol{x_{ns}} \in Null(A)$. This is because rank($\boldsymbol{A}$)=$r$=$m$, so the Null($\boldsymbol{A}^\top$) only contains the $\mathbf{0}$ vector and Null($\boldsymbol{A}$) has n-m=n-r vectors (assuming that $\boldsymbol{A}$ is full row rank).

  The best that can be done is to find the vector $\boldsymbol{x}$ in the RS which when transformed by matrix $\boldsymbol{A}$ is closest to CS, it is to say:

$$\begin{array}{ll} \text{minimize} & \|\boldsymbol{x}\|_2^2 \\ \text{subject to} & \boldsymbol{Ax} = \boldsymbol{b} \\ \text{variable} & \boldsymbol{x} \end{array} \tag{1.7.2}$$

  This vector is given (we will prove it in TOML-MIRI when we will study non-linear optimization) by the **right pseudo-inverse**:

$$\boldsymbol{A}^\dagger = \boldsymbol{A}^\top(\boldsymbol{A}\boldsymbol{A}^\top)^{-1} \tag{1.7.3}$$

  We call it right pseudo-inverse because $\boldsymbol{A}\boldsymbol{A}^\dagger = \boldsymbol{I}$ ($\boldsymbol{A}^\dagger$ is on the right). We will come back to a derivation of the left pseudo-inverse when we study the singular value decomposition (SVD).

Example 1.11 (Undertermined case) *Let us consider the following linear system:*

$$\boldsymbol{Ax} = \boldsymbol{b} \rightarrow \begin{bmatrix} 2 & 1 & 0 \\ 1 & 3 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 3 \\ 1 \end{bmatrix}$$

A null space vector is $\boldsymbol{x_{ns}} = [1, -2, 5]$. *The solution* $\boldsymbol{x}_r$ *is given by* $\boldsymbol{x}_r = \boldsymbol{A}^\top(\boldsymbol{AA}^\top)^{-1}\boldsymbol{b}$*:*

$$\boldsymbol{x}_r = \boldsymbol{A}^\top(\boldsymbol{AA}^\top)^{-1}\boldsymbol{b} = \begin{bmatrix} 2 & 1 \\ 1 & 3 \\ 0 & 1 \end{bmatrix} (\begin{bmatrix} 2 & 1 & 0 \\ 1 & 3 & 1 \end{bmatrix} \begin{bmatrix} 2 & 1 \\ 1 & 3 \\ 0 & 1 \end{bmatrix})^{-1} \begin{bmatrix} 3 \\ 1 \end{bmatrix}$$

and:

$$\boldsymbol{x}_r = \begin{bmatrix} 2 & 1 \\ 1 & 3 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 11/30 & -1/6 \\ -1/6 & 1/6 \end{bmatrix} \begin{bmatrix} 3 \\ 1 \end{bmatrix} = 1/30 \begin{bmatrix} 46 \\ -2 \\ -10 \end{bmatrix}$$

and a solution of this linear system of equations is in the form of $\boldsymbol{x} = \boldsymbol{x}_r + c\boldsymbol{x}_{ns}$, *with c any real constant.*

- **Unique solution:** This is the case in which there are the same number of variables as equations, meaning $m = n$. In this case, there are several possibilities:

  – If matrix $\boldsymbol{A}$ is non-singular (invertible), then there exists a unique solution:

  $$\hat{\boldsymbol{x}} = \boldsymbol{A}^{-1}\boldsymbol{b} \tag{1.7.4}$$

  – If matrix $\boldsymbol{A}$ is singular (non-invertible), then rank($\boldsymbol{A}$)=$r < (m{=}n)$, and there exists infinity number of solutions given by:

  $$\hat{\boldsymbol{x}} = \boldsymbol{A}^{\dagger}\boldsymbol{b} + \boldsymbol{x_{ns}} \tag{1.7.5}$$

  with $\boldsymbol{x_{ns}} \in NS(A)$, and $\boldsymbol{A}^\dagger$ the pseudinverse of $\boldsymbol{A}$.

- **Overdetermined case:** This is the case in which there are fewer variables than equations, meaning $m > n$. In this case, there exists no solution.

  The best that can be done is to make the error $\boldsymbol{e}$=$\boldsymbol{Ax} - \boldsymbol{b}$ as small as possible. Since $\boldsymbol{Ax}$ can never leave the CS, we have to find a vector $\boldsymbol{x}$ such that $\boldsymbol{Ax}$ is closest to $\boldsymbol{b}$, or in other words this point is the projection $\boldsymbol{p}$= $\boldsymbol{A\hat{x}}$ of $\boldsymbol{b}$ in the CS. In this way $\boldsymbol{e}$=$\boldsymbol{b}$-$\boldsymbol{p}$ is the smaller length if:

  $$\begin{array}{ll} \text{minimize} & \|\boldsymbol{Ax} - \boldsymbol{b}\|_2^2 \\ \text{variable} & \boldsymbol{x} \end{array} \tag{1.7.6}$$

This vector is given (we will prove it in TOML-MIRI) by the **left pseudo-inverse**

$$\boldsymbol{A}^{\dagger} = (\boldsymbol{A}^{\top}\boldsymbol{A})^{-1}\boldsymbol{A}^{\top} \tag{1.7.7}$$

We call it left pseudo-inverse because $\boldsymbol{A}^{\dagger}\boldsymbol{A}= \boldsymbol{I}$ ($\boldsymbol{A}^{\dagger}$ is on the left). However, we can derive it from the interpretation of Figure 1. We have to note that $\boldsymbol{e}$=$\boldsymbol{b}$-$\boldsymbol{p}$ is perpendicular to the CS, so $\boldsymbol{e}$ is in the left null space. Then:

$$\boldsymbol{A}^{\top}\boldsymbol{e} = \boldsymbol{A}^{\top}(\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}) = 0 \tag{1.7.8}$$

From here, we obtain:

$$\boldsymbol{A}^{\top}\boldsymbol{A}\boldsymbol{x} = \boldsymbol{A}^{\top}\boldsymbol{b} \tag{1.7.9}$$

And finally:

$$\hat{\boldsymbol{x}} = (\boldsymbol{A}^{\top}\boldsymbol{A})^{-1}\boldsymbol{A}^{\top}\boldsymbol{b} = \boldsymbol{A}^{\dagger}\boldsymbol{b} \tag{1.7.10}$$

We will come back to a derivation of the left pseudo-inverse when we study the singular value decomposition (SVD).

Example 1.12 (Overdetermined case) *Let us consider the following linear system:*

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b} \rightarrow \begin{bmatrix} 2 & 1 \\ 1 & 3 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 3 \\ 1 \\ 4 \end{bmatrix}$$

The solution $\boldsymbol{x}_r$ is given by $\boldsymbol{x}$= $(\boldsymbol{A}^{\top}\boldsymbol{A})^{-1}\boldsymbol{A}^{\top}\boldsymbol{b}$:

$$\boldsymbol{x} = (\boldsymbol{A}^{\top}\boldsymbol{A})^{-1}\boldsymbol{A}^{\top}\boldsymbol{b} = (\begin{bmatrix} 2 & 1 & 0 \\ 1 & 3 & 1 \end{bmatrix} \begin{bmatrix} 2 & 1 \\ 1 & 3 \\ 0 & 1 \end{bmatrix})^{-1} \begin{bmatrix} 2 & 1 & 0 \\ 1 & 3 & 1 \end{bmatrix} \begin{bmatrix} 3 \\ 1 \\ 4 \end{bmatrix}$$

and:

$$\boldsymbol{x} = \begin{bmatrix} 11/30 & -1/6 \\ -1/6 & 1/6 \end{bmatrix} \begin{bmatrix} 2 & 1 & 0 \\ 1 & 3 & 1 \end{bmatrix} \begin{bmatrix} 3 \\ 1 \\ 4 \end{bmatrix} = 1/30 \begin{bmatrix} 27 \\ 15 \end{bmatrix}$$

and a solution of this linear system of equations is in the form of $\boldsymbol{x} = 1/30[27, 15] = [0.9, 0.5]$.

2 Eigenvectors and eigenvalues

An eigenvector of a linear transformation is a nonzero vector that changes at most by a scalar factor when that linear transformation is applied to it. The corresponding eigenvalue λ is the factor by which the eigenvector is scaled. The eigenvector points in a direction in which the vector is scaled by the transformation and the eigenvalue is the factor by which the vector is scaled. A main application is the decomposition of a matrix by using eigenvectors and eigenvalues (called de eigenvector decomposition, EVD). Other applications appear when solving differential equations, dimensionality reduction (e.g. principal component analysis, PCA), denoising (e.g. eigenfaces), data compression, spectral graph theory, signal reconstruction, etc. The EVD (applied to squared matrices) is connected to another important decomposition called singular value decomposition (SVD) when the matrices are not-squared.

2.0.1 Linear transformations

If T is a linear transformation on a vector space over itself, i.e., $T{:}V \to V$, an eigenvector is a vector that satisfies:

$$T(\boldsymbol{v}) = \lambda \boldsymbol{v} \tag{2.0.1}$$

Assuming that the linear transformation (as seen in previous lectures) can be related to a matrix. Let, then, $A{\in}\mathbb{R}^{n\times n}$ be a square real matrix (although the matrix A has real components, in this section is better to think that vectors can have complex coefficients and that scalars are also in general complex numbers). A non zero vector $\boldsymbol{v}$ is an **eigenvector**, and the scalar λ is an **eigenvalue** of the matrix A when:

$$\boldsymbol{A}\boldsymbol{v} = \lambda \boldsymbol{v} \tag{2.0.2}$$

Eigenvalues must fulfill the condition: $\boldsymbol{A}\boldsymbol{v} - \lambda\boldsymbol{v} = (\boldsymbol{A} - \lambda\boldsymbol{I})\boldsymbol{v} = \mathbf{0}$. As $\boldsymbol{v}$ is non-zero, this is only possible if $\text{rank}(\boldsymbol{A}) < n$, which is equivalent to the condition $\det(\boldsymbol{A} - \lambda\boldsymbol{I}) = 0$. This determinant is in general a polynomial on λ of degree n, meaning that has n complex roots (if we count the multiplicity of roots). For very small matrices we can find the roots of this polynomial to compute eigenvalues. For larger matrices, there are more computationally efficient methods.

Once we know the eigenvalues, we can find the associated eigenvectors by solving the undetermined system of linear equations $(\boldsymbol{A} - \lambda\boldsymbol{I})\boldsymbol{v} = \mathbf{0}$. We can set, for instance, the condition $\|\boldsymbol{v}\| = 1$ to find unique solutions (up to the sign).

In the case of eigenvalues of multiplicity larger than 1, we can have several linearly independent associated eigenvectors. The dimension of the generated subspace must be less or equal to the multiplicity of the root. When this dimension is strictly lower than the multiplicity of λ, we say that the matrix is **defective**. You can find detailed discussions on this in any text on linear algebra. We will be interested mainly in symmetric matrices, which are never defective.

2.1 Diagonalization and eigendecomposition

Let A be a square $n{\times}n$ matrix with n linearly independent eigenvectors v_i (where i=1, . . . , n). Then A can be factorized as:

$$\mathbf{A} = \mathbf{V}\boldsymbol{\Lambda}\mathbf{V}^{-1} \tag{2.1.1}$$

where V is a squared $n{\times}n$ matrix with column vectors $\mathbf{v_i}$ (eigenvectors), and $\boldsymbol{\Lambda}$ is a squared $n{\times}n$ diagonal matrix whose diagonal elements Λ_{ii}=λ_i (are the eigenvalues). This is easy to see since if $\mathbf{v}$ i an eigenvector:

$$\begin{aligned} \mathbf{Av} &= \lambda\mathbf{v} \\ \mathbf{AV} &= \mathbf{V}\boldsymbol{\Lambda} \\ \mathbf{A} &= \mathbf{V}\boldsymbol{\Lambda}\mathbf{V}^{-1} \end{aligned} \tag{2.1.2}$$

See the following property. The linearly independent eigenvectors associated with the eigenvalues different of 0, form the range(A) or Image(A), and they are the basis of the column space CS(A). The linearly independent eigenvectors associated with eigenvalues equal to 0, form the basis of the null space of A.

2.2 Some important types of matrices

- An square $n{\times}n$ matrix Λ with 0 off-diagonal elements is a **diagonal matrix**. The product $\Lambda\boldsymbol{v}$ produces a stretching (or directional scaling) of the different components of the vector $\boldsymbol{v}$ according to the corresponding values of the diagonal. If some components of the diagonal of Λ are zero, the product collapses to zero the corresponding components of $\boldsymbol{v}$. The inverse of Λ is obtained by simply inverting the diagonal elements (if any of those elements is zero, then the matrix is not invertible).

- A square matrix Q with columns that are orthonormal vectors (i.e. orthogonal and with norm 1) is called an **orthonormal matrix** (or very often simply **orthogonal** matrix, as we will assume the norm 1 condition). The product $Q\boldsymbol{v}$ produces a rotation, a reflection, or a combination of both operations (called roto-reflection), on the vector $\boldsymbol{v}$. Orthonormal matrices are always invertible and $Q^{-1} = Q^\top$.

- A square matrix $\boldsymbol{A}$ that fulfills $\boldsymbol{A}$=$\boldsymbol{A}^\top$ is a **symmetric matrix**. Symmetric matrices satisfy the following properties: i) sum (or difference) of symmetric matrices is symmetric, ii) if $\boldsymbol{A}$ and $\boldsymbol{B}$ are symmetric, then $\boldsymbol{AB}$ is symmetric only if $\boldsymbol{AB}$=$\boldsymbol{BA}$; iii) if $\boldsymbol{A}^{-1}$ exists, then it is symmetric if and only if $\boldsymbol{A}$ is symmetric;

- If $\boldsymbol{v}$ is a column vector of V, the square $n{\times}n$ matrix $\boldsymbol{vv}^\top$ is a **matrix** (do not confuse with the scalar product $\boldsymbol{v}^\top\boldsymbol{v}$ which is a number) of rank 1.

- We define a **projection** as a linear operator $P{:}V \to V$ such that $\boldsymbol{P}^2{=}\boldsymbol{P}$. If the vector space V is finite-dimensional, a square matrix P is called *projection matrix* if $\boldsymbol{P}^2{=}\boldsymbol{P}$. Moreover, if $\boldsymbol{P}$ is real, and $\boldsymbol{P}^2{=}\boldsymbol{P}{=}\boldsymbol{P}^\top$ then $\boldsymbol{P}$ is called a **orthogonal projection matrix**. For the general case of a non-unitary vector $\boldsymbol{v}$, the projection matrix is defined as:

$$\boldsymbol{P} = \boldsymbol{v} < \boldsymbol{v}, \boldsymbol{v} >^{-1} \boldsymbol{v}^\top \tag{2.2.1}$$

so the projection of a vector $\boldsymbol{t}$ on the vector $\boldsymbol{v}$, would be given by:

$$\hat{\boldsymbol{t}} = \boldsymbol{P}\boldsymbol{t} = \boldsymbol{v} < \boldsymbol{v}, \boldsymbol{v} >^{-1} \boldsymbol{v}^\top \boldsymbol{t} \tag{2.2.2}$$

Example 2.1 (Projection matrix for a vector) *We want to project vector $\boldsymbol{t}{=}\ [1, 2]$ over vector $\boldsymbol{v}{=}\ [2, 1]$. We first obtain the projection matrix $\boldsymbol{P}$:*

$$\boldsymbol{P} = \frac{1}{\|v\|_2^2}\boldsymbol{v}\boldsymbol{v}^\top = \frac{1}{5}\begin{bmatrix}2\\1\end{bmatrix}\begin{bmatrix}2 & 1\end{bmatrix} = \frac{1}{5}\begin{bmatrix}4 & 2\\2 & 1\end{bmatrix}$$

Then, now, the projection of vector $\mathbf{t}$ *on the vector* $\mathbf{v}$*, would be given by:*

$$\hat{\boldsymbol{t}} = \boldsymbol{P}\boldsymbol{t} = \frac{1}{5}\begin{bmatrix}4 & 2\\2 & 1\end{bmatrix}\begin{bmatrix}1\\2\end{bmatrix} = \frac{1}{5}\begin{bmatrix}6\\4\end{bmatrix} = \begin{bmatrix}1.2\\0.8\end{bmatrix}$$

and the projected vector will be $\hat{\boldsymbol{t}} = [1.2, 0.8]$. We can observe that the matrix $\boldsymbol{P}$ satisfies property $\boldsymbol{P}^2{=}\boldsymbol{P}{=}\boldsymbol{P}^\top$:

$$\boldsymbol{P}^2 = \frac{1}{5}\begin{bmatrix}4 & 2\\2 & 1\end{bmatrix}\frac{1}{5}\begin{bmatrix}4 & 2\\2 & 1\end{bmatrix} = \frac{1}{25}\begin{bmatrix}20 & 10\\10 & 5\end{bmatrix} = \frac{1}{5}\begin{bmatrix}4 & 2\\2 & 1\end{bmatrix}$$

Finally, observe that since we project a vector over a vector, matrix $\boldsymbol{P}$ has rank $r = 1$.

In the special case in which $\boldsymbol{v}$ is unitary, $< \boldsymbol{v}, \boldsymbol{v} >{=}1$, and $\boldsymbol{P}{=}\ \boldsymbol{v}\boldsymbol{v}^\top$. The projection of vector $\boldsymbol{t}$ on the unitary vector $\boldsymbol{v}$ will then be $\hat{\boldsymbol{t}} = \boldsymbol{P}\boldsymbol{t} = \boldsymbol{v}\boldsymbol{v}^\top\boldsymbol{t}$.

In the case that the we want to project a vector $\boldsymbol{t}$ on a subspace generated by the matrix $\boldsymbol{A}$, then the projection matrix will be given by:

$$\boldsymbol{P} = \boldsymbol{A}(\boldsymbol{A}^\top\boldsymbol{A})^{-1}\boldsymbol{A}^\top \tag{2.2.3}$$

and the projection of the vector $\boldsymbol{t}$ on the subspace generated by the matrix $\boldsymbol{A}$ will be given by:

$$\hat{\boldsymbol{t}} = \boldsymbol{A}(\boldsymbol{A}^\top\boldsymbol{A})^{-1}\boldsymbol{A}^\top\boldsymbol{t} \tag{2.2.4}$$

Example 2.2 (Projection matrix for a matrix) *We want to project vector $\boldsymbol{t}{=}\ [1, 2]$ over the space generated by matrix:*

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

The projection matrix $\boldsymbol{P}$ *will be given* $\boldsymbol{P} = \boldsymbol{A}(\boldsymbol{A}^\top \boldsymbol{A})^{-1}\boldsymbol{A}^\top$*:*

$$\boldsymbol{P} = \begin{bmatrix} 2 & 1 \\ 1 & 3 \\ 0 & 1 \end{bmatrix} (\begin{bmatrix} 2 & 1 & 0 \\ 1 & 3 & 1 \end{bmatrix} \begin{bmatrix} 2 & 1 \\ 1 & 3 \\ 0 & 1 \end{bmatrix})^{-1} \begin{bmatrix} 2 & 1 & 0 \\ 1 & 3 & 1 \end{bmatrix} = \frac{1}{30} \begin{bmatrix} 29 & 2 & -5 \\ 2 & 26 & 10 \\ -5 & 10 & 5 \end{bmatrix}$$

and we can observe that matrix $\boldsymbol{P}$ *has rank* $r = 2$ *(the same as matrix* $\boldsymbol{A}$*).*

2.3 An $n \times m$ matrix of rank r maps a sphere of dimension n into an ellipsoid of dimension r

A basic fact of linear algebra is the following: Assume that we have an $n \times m$ matrix A with rank(A)=r. Assume that we compute the products $y = Ax$, where x is an n-dimensional vector that lies in a sphere of radius 1 in the space R^n. Then the locus (set of points) of all the generated vector y lies in an ellipsoid of dimension r embedded in the space R^m.

As an example, assume a 2×2 matrix A of rank 2. If we compute $y = Ax$ for $x = (cos(\theta), sin(\theta))^\top$ with $\theta \in [0, 2\pi)$, the vector y will lie in an ellipsoid centered in the origin in R^2. If rank(A)=1, the ellipsoid will collapse one of its dimensions, resulting in a segment that crosses the origin.

2.4 Matrix factorization

In this course, we deal with two important matrix factorizations:

- $S = Q\Lambda Q^\top$, for symmetric matrices.
- $A = U\Sigma V^\top$, Singular Value Decomposition (SVD) for general matrices.

We are interested in the first factorization as covariance matrices are symmetric, and we are interested in the second factorization as it allows us to approximate clouds of points in high-dimensional spaces by clouds of points in lower dimensional spaces.

2.5 Diagonalization of symmetric matrices

Assume that $\boldsymbol{S}$ is an $n \times n$ **symmetric matrix**, i.e. $\boldsymbol{S} = \boldsymbol{S}^\top$, i.e., $a_{ij} = a_{ji}\ \forall i, j$. Symmetric matrices have the following properties: i) $\boldsymbol{S}_1 + \boldsymbol{S}_2$ (sum) is symmetric if $\boldsymbol{S}_1$, $\boldsymbol{S}_2$ are symmetric, ii) $\boldsymbol{S}_1\boldsymbol{S}_2$ (product) is not necessarily symmetric even if $\boldsymbol{S}_1$, $\boldsymbol{S}_2$ are symmetric, iii) if $\boldsymbol{S}^{-1}$ exists, is symmetric if and only if $\boldsymbol{S}_1$ is symmetric.

The **spectral theorem** tells us when a linear operator or matrix can be diagonalized. In the case of $\boldsymbol{S}$ (symmetric matrix), the finite-dimensional spectral theorem says that any symmetric matrix $\boldsymbol{S}$ whose entries are real can be diagonalized by an orthogonal matrix (so, the eigenvectors are orthonormal).

- $\boldsymbol{S}$ has n real eigenvalues, λ_i, (counting possible multiplicities)
- The n associated eigenvectors, $\boldsymbol{q_i}$ are orthonormal.
- These matrices are not defective (defective means that rank($\boldsymbol{A}$)=r<n, then rank($\boldsymbol{S}$)=n) if they are positive definite.

Let us prove it for the case in which the matrix has non-repeated eigenvalues. This result can be extended for the repeated eigenvalues case by using continuity arguments.

2.5.1 Eigenvectors are orthogonal

Assume that $\boldsymbol{S}\boldsymbol{v} = \lambda\boldsymbol{v}$ and $\boldsymbol{S}\boldsymbol{u} = \mu\boldsymbol{u}$ different eigenvalues λ and μ. We have:

$$\boldsymbol{v}^\top\boldsymbol{S}\boldsymbol{u} = \mu\boldsymbol{v}^\top\boldsymbol{u} \tag{2.5.1}$$

and

$$\boldsymbol{u}^\top\boldsymbol{S}\boldsymbol{v} = \lambda\boldsymbol{u}^\top\boldsymbol{v} \tag{2.5.2}$$

but $\boldsymbol{u}^\top\boldsymbol{S}\boldsymbol{v} = \boldsymbol{v}^\top\boldsymbol{S}^\top\boldsymbol{u} = \boldsymbol{v}^\top\boldsymbol{S}\boldsymbol{u}$ as $\boldsymbol{S}$ is symmetric, and $\boldsymbol{u}^\top\boldsymbol{v} = \lambda\boldsymbol{v}^\top\boldsymbol{u}$, which implies $\lambda\boldsymbol{u}^\top\boldsymbol{v} = \mu\boldsymbol{u}^\top\boldsymbol{v}$, and from this we get $\boldsymbol{u}^\top\boldsymbol{v} = 0$ (eigenvectors are orthogonal).

2.5.2 Eigenvalues are real

Assume that λ is a complex eigenvalue. As $\boldsymbol{S}$ is real, λ^* (i.e. its complex conjugate) must also be an eigenvalue:

$$\boldsymbol{S}\boldsymbol{v} = \lambda\boldsymbol{v} \tag{2.5.3}$$

and:

$$\boldsymbol{S}\boldsymbol{v}^* = \lambda^*\boldsymbol{v}^* \tag{2.5.4}$$

Then we have:

$$\boldsymbol{v}^{*T}\boldsymbol{S}\boldsymbol{v} = \lambda\boldsymbol{v}^{*T}\boldsymbol{v} = \lambda \tag{2.5.5}$$

since eigenvectors $\boldsymbol{v}$ are orthonormal ($\boldsymbol{v^*v} = 1$), and:

$$\boldsymbol{v}^T S\boldsymbol{v}^* = \lambda^*\boldsymbol{v}^T\boldsymbol{v}^* = \lambda^* \tag{2.5.6}$$

But $\lambda^* = \boldsymbol{v}^T\boldsymbol{S}\boldsymbol{v}^* = (\boldsymbol{v}^T\boldsymbol{S}\boldsymbol{v}^*)^\top = \boldsymbol{v}^{*T}\boldsymbol{S}\boldsymbol{v} = \lambda$, meaning $\lambda = \lambda^*$.

As a consequence, we can write $\boldsymbol{S}$ as: $\boldsymbol{S}$=$\boldsymbol{Q\Lambda Q}^\top$, where Q is an $n{\times}n$ matrix with columns the eigenvectors of S, $\boldsymbol{Q}$=[$\boldsymbol{q_1}$,..., $\boldsymbol{q_n}$], and Λ is a diagonal matrix with diagonal elements $\Lambda_{i,i} = \lambda_i$.

An alternative way of expressing this is by the formula: $\boldsymbol{S} = \sum_i \lambda_i \boldsymbol{q_i}\boldsymbol{q_i}^t$. Recall that the terms $\boldsymbol{q_i}\boldsymbol{q_i}^t$ are *rank* 1 matrices.

If *rank*($\boldsymbol{S}$)=n, the eigenvalues λ_i must be different from 0 (if not, the correspondent eigenvector $\boldsymbol{q_i}$ would belong to the null space and rank(S)<n). In this case, S has an inverse, which is also a symmetric matrix, and: $\boldsymbol{S}^{-1}$=$\boldsymbol{Q\Lambda}^{-1}\boldsymbol{Q}^\top$ = $\sum_i \frac{1}{\lambda_i}\boldsymbol{q_i}\boldsymbol{q_i}^\top$.

Example 2.3 (Symmetric matrices) *Let us assume the following symmetric matrix* $\boldsymbol{A}$*:*

$$\boldsymbol{A} = \begin{bmatrix} 2 & 1 & 0 & 1 \\ 1 & 3 & 4 & 5 \\ 0 & 4 & 1 & 4 \\ 1 & 5 & 4 & 4 \end{bmatrix}$$

Then the eigendecomposition of matrix $\boldsymbol{A}$ *will* $\boldsymbol{A} = \boldsymbol{Q\Lambda Q}^\top$*, with:*

$$\boldsymbol{Q} = \begin{bmatrix} -0.128 & -0.978 & -0.155 & -0.051 \\ -0.596 & 0.021 & 0.553 & -0.582 \\ -0.463 & 0.204 & -0.812 & -0.29 \\ -0.643 & 0.028 & 0.103 & 0.758 \end{bmatrix}$$

and

$$\boldsymbol{\Lambda} = \begin{bmatrix} 11.712 & 0.0 & 0.0 & 0.0 \\ 0.0 & 1.95 & 0.0 & 0.0 \\ 0.0 & 0.0 & -2.228 & 0.0 \\ 0.0 & 0.0 & 0.0 & -1.434 \end{bmatrix}$$

and as we can observe, all eigenvalues are real, and the eigenvectors are othornormal; $\boldsymbol{q}_i^\top\boldsymbol{q}_i = 1$ *and* $\boldsymbol{q}_i^\top\boldsymbol{q}_j = 0$ *with* $i{\neq}j$*, e.g.* $\boldsymbol{q}_0^\top\boldsymbol{q}_0$= [−0.128, −0.596, −0.463, −0.643]$^\top$ [−0.128, −0.596, −0.463, −0.643]= 1.0 *and* $\boldsymbol{q}_0^\top\boldsymbol{q}_2$= [−0.128, −0.596, −0.463, −0.643]$^\top$ [−0.155, 0.553, −0.812, 0.103]= 0.0.

Observe also that the matrix $\boldsymbol{Q}$ *is a rotation matrix; for example* $\boldsymbol{Q}\boldsymbol{q}_i$ = [0, ..., 1, ..., 0]*, a unitary vector with a* 1 *at position* i*,* 0*'s in the remainder.*

2.6 Positive definite matrices

When all eigenvalues $\boldsymbol{S}$ are positive, we say that the symmetric matrix is **positive definite**. In this case, $\boldsymbol{S}^{-1}$ is also a symmetric positive definite matrix. These matrices somehow play the role of positive numbers in the matrix world.

For example, the variance-covariance matrix $\mathbf{\Sigma} = \mathbb{E}[(X_1, .., X_n)(X_1, ..., X_n)^\top]$ of multivariate Gaussian distributions is a positive definite matrix. The inverse of the covariance matrix, called **precision matrix** is also positive definite.

- An $n{\times}n$ symmetric real matrix $\mathbf{A}$ is said to be **positive definite** if

  $$\mathbf{x}^\top \mathbf{A}\mathbf{x}{>}0 \tag{2.6.1}$$

  for all non-zero vectors $\mathbf{x}{\in}\mathbb{R}^n$ and it is said **negative definite** if

  $$\mathbf{x}^\top \mathbf{A}\mathbf{x}{<}0. \tag{2.6.2}$$

- An $n{\times}n$ symmetric real matrix $\mathbf{A}$ is said to be **positive semi-definite** if

  $$\mathbf{x}^\top \mathbf{A}\mathbf{x}{\geq}0 \tag{2.6.3}$$

  $\mathbf{x}^\top \mathbf{A}\mathbf{x}{\geq}0$ for all non-zero vectors $\mathbf{x}{\in}\mathbb{R}^n$ and it is said **negative semi-definite** if

  $$\mathbf{x}^\top \mathbf{A}\mathbf{x}{\leq}0 \tag{2.6.4}$$

- An $n{\times}n$ symmetric real matrix which is neither positive semi-definite nor negative semi-definite is called indefinite.

- A matrix $\mathbf{A}$ is positive (negative) definite if and only if all of its eigenvalues are >0 (<0).

- A matrix $\mathbf{A}$ is positive (negative) semi-definite if and only if all of its eigenvalues are ≥ 0 (≤ 0).

You can find several interesting properties such as: i) if $\mathbf{A}$ and $\mathbf{B}$ are positive definite matrices, then the sum $\mathbf{A} + \mathbf{B}$ is a positive definite matrix; or ii) if $\mathbf{A}$ is a positive definite matrix, then the inverse $\mathbf{A}^{-1}$ is a positive definite matrix.

Finally, an interesting result is the following: let be a $m{\times}n$ $\mathbf{A}$ matrix. The matrix $\mathbf{S}{=}\mathbf{A}^\top\mathbf{A}$ is positive definite (and then symmetric), and therefore $\mathbf{S}$ has orthonormal eigenvectors and positive eigenvalues.

Example 2.4 (Positive definite matrices) *Let us consider matrices* $\boldsymbol{A}$*,* $\boldsymbol{B}$*,* $\boldsymbol{C}$ *and* $\boldsymbol{D}$*. Check positive definiteness.*

$$\boldsymbol{A} = \begin{bmatrix} -1 & 1 & -1 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}; \; \boldsymbol{B} = \begin{bmatrix} -3 & 0 & -1 \\ 0 & -5 & -2 \\ -1 & -2 & -3 \end{bmatrix}$$

$$\boldsymbol{C} = \begin{bmatrix} 3 & 1 & -1 \\ 0 & 4 & -2 \\ -1 & -2 & 3 \end{bmatrix}; \; \boldsymbol{D} = \begin{bmatrix} 3 & 1 & -1 \\ 0 & 4 & -2 \\ 3 & 4 & -3 \end{bmatrix}$$

Let us obtain the eigenvalues. The eigenvalues of matrix $\boldsymbol{A}$ *are* λ=[$-$1.879, 0.347, 1.532]*. The matrix is neither positive or negative definite. The eigenvalues of matrix* $\boldsymbol{B}$ *are* λ=[$-$1.319, $-$3.358, $-$6.323] *are all negative, and then the matrix is negative definite. The eigenvalues of matrix* $\boldsymbol{C}$ *are* λ=[1.097, 3.194, 5.709] *are all positive, and then the matrix is positive definite. The eigenvalues of matrix* $\boldsymbol{D}$ *are* λ=[0.0, 0.586, 3.414] *are all positive or equal to zero, and then the matrix is positive semidefinite. Moreover, since there are one eigenvalue equal to zero, the rank of this matrix is 2. On the other hand, the rank of matrices* $\boldsymbol{A}$, $\boldsymbol{B}$, $\boldsymbol{C}$ *is 3.*

3 Useful properties of matrices

3.1 Geometric interpretation for symmetric matrices: Rotation/Reflection, Stretching, Rotation/Reflection^{-1}

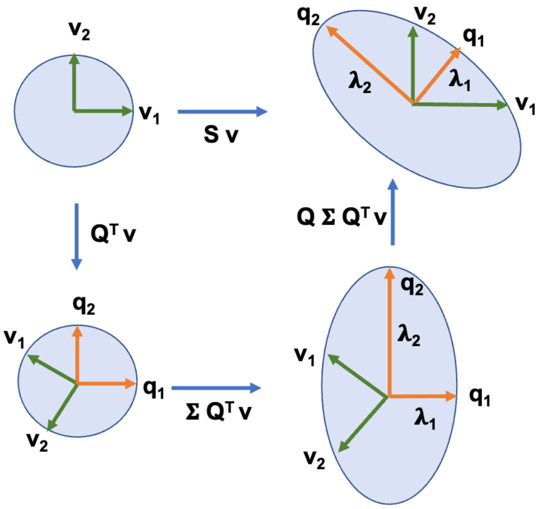


Figure 3: Symmetric matrix applied to a disc of radius 1.

We know that matrix multiplication can be interpreted as linear map composition, meaning that the geometric interpretation of $S\boldsymbol{v} = \boldsymbol{Q}\Lambda\boldsymbol{Q}^\top\boldsymbol{v}$ for an arbitrary vector $\boldsymbol{v}$ is:

- Rotate the coordinate system to align it with the set of vectors $\boldsymbol{q_i}$ which form the columns of the matrix $\boldsymbol{Q}$. The vector $\boldsymbol{v}$ in this new coordinate system has the expression $\boldsymbol{Q}^\top\boldsymbol{v}$ (note that the vectors $\boldsymbol{q_i}$ expressed in the new coordinate system have the expression $\boldsymbol{Q}^\top \boldsymbol{q_i} = [0, ..., 1, ...0]$ as we expect).

- Stretch each component of the resulting vector $\boldsymbol{Q}^\top \boldsymbol{v}$ according with the diagonal elements of the matrix Λ, obtaining the vector $\Lambda \boldsymbol{Q}^\top \boldsymbol{v}$. This stretching causes in general a change of direction of the vector, but if $\boldsymbol{v}$ is aligned with a vector $\boldsymbol{q_i}$, it does not change its direction.
- Apply the inverse rotation to the coordinate system. If we express the resulting $\Lambda \boldsymbol{Q}^\top \boldsymbol{v}$ in this new coordinate system we obtain $\boldsymbol{Q}\Lambda \boldsymbol{Q}^\top \boldsymbol{v}$.

For instance, a positive definite matrix S will map an n-dimensional sphere of radius 1 to an n-dimensional ellipsoid with axis given by its eigenvectors $\boldsymbol{q_i}$, and axis lengths given by its eigenvalues $\boldsymbol{\lambda_i}$, (think on this, Figure 3).

3.2 Derivatives with vectors and matrices

First, some matrix manipulations typically appear when working with differentiation.

- $(\mathbf{AB})^{-1} = \mathbf{B}^{-1}\mathbf{A}^{-1}$
- $(\mathbf{AB})^\top = \mathbf{B}^\top \mathbf{A}^\top$
- $(\mathbf{a}^\top \mathbf{A} \mathbf{b})^\top = \mathbf{b}^\top \mathbf{A}^\top \mathbf{a}$
- $\mathbf{a}^\top \mathbf{b} = \mathbf{b}^\top \mathbf{a}$
- $(\mathbf{A} + \mathbf{B})\mathbf{C} = \mathbf{AC} + \mathbf{BC}$
- $\mathbf{AB} \neq \mathbf{BA}$
- $\mathbf{a}^\top \mathbf{b} = \mathbf{b}^\top \mathbf{a}$
- $(\mathbf{a} + \mathbf{b})^\top \mathbf{C} = \mathbf{a}^\top \mathbf{C} + \mathbf{b}^\top \mathbf{C}$

The **Hadamard product** (also known as the element-wise product, entry-wise product, or Schur product) returns a matrix of the multiplied corresponding elements. It is defined with the symbol $\odot$ (sometimes also with symbol $\circ$):

$$(\boldsymbol{A} \circ \boldsymbol{B})_{ij} = (\boldsymbol{A} \odot \boldsymbol{B})_{ij} = (A)_{ij}(\boldsymbol{B})_{ij}. \tag{3.2.1}$$

Example 3.1 (Hadamard product of matrices) *Consider matrices* $\boldsymbol{A}$ *and* $\boldsymbol{B}$.

$$\boldsymbol{A} = \begin{bmatrix} 2 & 1 & -2 \\ 1 & 0 & 2 \\ 0 & 1 & 3 \end{bmatrix}; \quad \boldsymbol{B} = \begin{bmatrix} -3 & 0 & -1 \\ 2 & -5 & -2 \\ -1 & 2 & 4 \end{bmatrix}$$

Its Hadamard product is given by:

$$\boldsymbol{A} \odot \boldsymbol{B} = \begin{bmatrix} 2 & 1 & -2 \\ 1 & 0 & 2 \\ 0 & 1 & 3 \end{bmatrix} \odot \begin{bmatrix} -3 & 0 & -1 \\ 2 & -5 & -2 \\ -1 & 2 & 4 \end{bmatrix} = \begin{bmatrix} -6 & 0 & 2 \\ 2 & 0 & -4 \\ 0 & 2 & 12 \end{bmatrix}$$

Let us assume vector $\mathbf{x}$ and we want to obtain vector derivatives over the function f($\mathbf{x}$) of the form $df(\mathbf{x})/d\mathbf{x}$. We use the denominator layout (meaning that $\mathbf{f}^\top$ and $\mathbf{x}$)

- $\mathrm{f}(\mathbf{x}) = \mathbf{x}^\top \mathbf{a} \longrightarrow df(\mathbf{x})/d\mathbf{x} = \mathbf{a}$
- $\mathrm{f}(\mathbf{x}) = \mathbf{x}^\top \mathbf{A} \longrightarrow df(\mathbf{x})/d\mathbf{x} = \mathbf{A}$
- $\mathrm{f}(\mathbf{x}) = \mathbf{A}\mathbf{x} \longrightarrow df(\mathbf{x})/d\mathbf{x} = \mathbf{A}^\top$
- $\mathrm{f}(\mathbf{x}) = \mathbf{x}^\top \mathbf{x} \longrightarrow df(\mathbf{x})/d\mathbf{x} = 2\mathbf{x}$
- $\mathrm{f}(\mathbf{x}) = \mathbf{x}^\top \mathbf{A}\mathbf{x} \longrightarrow df(\mathbf{x})/d\mathbf{x} = 2\mathbf{A}\mathbf{x}$ if $\mathbf{A}$ is symmetric
- $\mathrm{f}(\mathbf{x}) = \mathbf{x}^\top \mathbf{A}\mathbf{x} \longrightarrow df(\mathbf{x})/d\mathbf{x} = (\mathbf{A} + \mathbf{A}^\top)\mathbf{x}$

3.3 The trace operator

For a square matrix $n{\times}n$ A we define the trace as the sum of the elements of its diagonal:

$$\operatorname{tr}(\mathbf{A}) = \sum_{k=1}^{n} a_{kk} = a_{11} + a_{22} + \cdots + a_{nn} \tag{3.3.1}$$

The trace of the $n{\times}n$ I identity matrix is the dimension of the space, namely n: $\operatorname{tr}(\mathbf{I}_n) = n$. The following relationships are satisfied:

$$\begin{aligned} \operatorname{tr}(\mathbf{A} + \mathbf{B}) &= \operatorname{tr}(\mathbf{A}) + \operatorname{tr}(\mathbf{B}) \\ \operatorname{tr}(c\mathbf{A}) &= c \operatorname{tr}(\mathbf{A}) \end{aligned} \tag{3.3.2}$$

for all square matrices A and B, and all scalars c. Moreover:

$$\operatorname{tr}(\mathbf{A}) = \operatorname{tr}\left(\mathbf{A}^\mathsf{T}\right) \tag{3.3.3}$$

3.3.1 The trace of a matrix is the sum of its eigenvalues counting multiplicities

As we know, the eigenvalues λ_k are the solutions of the equation:

$$det(\boldsymbol{A} - \lambda I) = (-1)^n \lambda^n + (-1)^{n-1} tr(\boldsymbol{A}) \lambda^{n-1} + ... = 0 \tag{3.3.4}$$

The eigenvalues λ_k are the roots of the polynomial in λ, meaning that we have:

$$(-1)^n(\lambda - \lambda_1)(\lambda - \lambda_2)... = 0 \tag{3.3.5}$$

and from this we obtain:

$$\sum_k \lambda_k = tr(\boldsymbol{A}) \tag{3.3.6}$$

Example 3.2 (Symmetric matrices) *Let us assume the following symmetric matrix* $\boldsymbol{A}$*:*

$$\boldsymbol{A} = \begin{bmatrix} 2 & 1 & 0 & 1 \\ 1 & 3 & 4 & 5 \\ 0 & 4 & 1 & 4 \\ 1 & 5 & 4 & 4 \end{bmatrix}$$

The eigenvalues are λ=[11.712, 2.228, 1.95, 1.434]. *We can observe that* $\sum_i \lambda_i$= $11.712 + 1.95 + (-2.228) + (-1.434)$= 10.0 *and that* $tr(\boldsymbol{A}) = 2 + 3 + 1 + 4 = 10$

3.3.2 The trace operator is cyclic

Let A and B be general non-square matrices of sizes $n{\times}m$ and $m{\times}n$. The diagonal elements of $\boldsymbol{P} = \boldsymbol{AB}$ can be found as:

$$p_{kk} = \sum_{i=1,...,m} a_{ki}b_{ik} \tag{3.3.7}$$

while that for $\boldsymbol{Q} = \boldsymbol{BA}$ we have:

$$q_{ii} = \sum_{k=1,...,n} a_{ik}b_{ki} \tag{3.3.8}$$

We have then

$$tr(\boldsymbol{AB}) = \sum_{k=1,..,n} \sum_{i=1,...,m} a_{ki}b_{ik} = \sum_{i=1,..,m} \sum_{k=1,...,n} a_{ki}b_{ik} = tr(\boldsymbol{BA}) \tag{3.3.9}$$

If we have now three arbitrary matrices of the right sizes to produce a square matrix in its product $\boldsymbol{BAC}$ we have:

$$tr(\boldsymbol{BAC}) = tr((\boldsymbol{BA})\boldsymbol{C}) = tr(\boldsymbol{C}(\boldsymbol{BA})) = tr(\boldsymbol{CBA}) \tag{3.3.10}$$

Moreover for real column vectors $\mathbf{a} \in \mathbb{R}^n$ and $\mathbf{b} \in \mathbb{R}^n$, the trace of the outer product is equivalent to the inner product:

$$\operatorname{tr}\left(\mathbf{b}\mathbf{a}^\mathsf{T}\right) = \mathbf{a}^\mathsf{T}\mathbf{b} \tag{3.3.11}$$

3.3.3 Derivatives of a trace

Let $\boldsymbol{A}{\in}\mathbb{R}^{n\times m}$ and $\boldsymbol{X}{\in}\mathbb{R}^{m\times n}$ matrices. Then:

$$\frac{d}{d\boldsymbol{X}}tr(\boldsymbol{AX}) = \frac{d}{d\boldsymbol{X}}tr(\boldsymbol{XA}) = \boldsymbol{A}^\top \tag{3.3.12}$$

Moreover, if $\boldsymbol{A}, \boldsymbol{X}{\in}\mathbb{R}^{n\times m}$:

$$\frac{d}{d\boldsymbol{X}}tr(\boldsymbol{AX}^\top) = \frac{d}{d\boldsymbol{X}}tr(\boldsymbol{X}^\top\boldsymbol{A}) = \boldsymbol{A} \tag{3.3.13}$$

Then, assuming correct matrices sizes $\boldsymbol{A}$, $\boldsymbol{B}$, $\boldsymbol{X}$:

$$\frac{d}{d\boldsymbol{X}}tr(\boldsymbol{AXB}) = \frac{d}{d\boldsymbol{X}}tr(\boldsymbol{BAX}) = \boldsymbol{(BA)}^\top \tag{3.3.14}$$

On the other hand, for deriving $(\boldsymbol{X}^\top\boldsymbol{AX})$ with respect to $\boldsymbol{X}$, we first fix one of the $\boldsymbol{X}$ and then the other (e.g. fix one $\boldsymbol{X}$ and substitute the other $\boldsymbol{X}$ by $\boldsymbol{Y}$ and derive with respect $\boldsymbol{Y}$, and then repeat exchanging the order):

$$\frac{d}{d\boldsymbol{X}}tr(\boldsymbol{X}^\top\boldsymbol{AX}) = \frac{d}{d\boldsymbol{Y}}tr(\boldsymbol{Y}^\top\boldsymbol{AX}) + \frac{d}{d\boldsymbol{Y}}tr(\boldsymbol{XAY}) = \boldsymbol{(A + A^\top)X} \tag{3.3.15}$$

Using these rules, we can obtain the derivative of more complex trace expressions.

3.4 Quadratic forms, sub-level sets, paraboloids and ellipsoids

A quadratic function has the form of

$$f(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^\top\boldsymbol{Px} + \boldsymbol{b}^\top\boldsymbol{x} + c \tag{3.4.1}$$

where $\boldsymbol{P}$ is a $n{\times}n$ symmetric matrix, $\boldsymbol{b}$ and $\boldsymbol{x}$ are n-dim vectors, and c is a real number. Define now a α sub-level set C_α of a function $f{:}\mathbb{R}^n \to \mathbb{R}$, as

$$C_\alpha = \{\boldsymbol{x}{\in}dom\{f\}|f(\boldsymbol{x}) \leq \alpha\} \tag{3.4.2}$$

Let us now remember the equation of an n-dim ellipsoid as

$$\frac{x_1^2}{a_1^2} + \frac{x_2^2}{a_2^2} + \cdots + \frac{x_n^2}{a_n^2} = 1$$

Thus, we can see that the α sub-level set of a quadratic form is an ellipsoid. In fact,

$$\epsilon = \{\boldsymbol{x}|(\boldsymbol{x} - \boldsymbol{x_c})^\top\boldsymbol{P^{-1}}(\boldsymbol{x} - \boldsymbol{x_c}) \leq 1\} \tag{3.4.3}$$

where $\boldsymbol{P}$ is symmetric and positive definite, and the vector $\boldsymbol{x_c}$ is the center of the ellipsoid. The matrix $\boldsymbol{P}$ defines how far the ellipsoid extends in every direction from $\boldsymbol{x_c}$ (directions given by eigenvectors of $\boldsymbol{P}$), and the length of the semi-axes of ϵ are given by $\sqrt{\lambda_i}$ (with λ_i the eigenvalues of $\boldsymbol{P}$.

If we consider now a new dimension x_{n+1}, and make

$$x_{n+1} = \frac{x_1^2}{a_1^2} + \frac{x_2^2}{a_2^2} + \cdots + \frac{x_n^2}{a_n^2} \tag{3.4.4}$$

we obtain the equation of a paraboloid.

3.5 Multivariate Gaussian distribution

Recall that we say that $\boldsymbol{X}$ follows a *multivariate gaussian distribution* of parameters $\boldsymbol{\mu}$ and Σ, where $\boldsymbol{\mu}\in\mathbb{R}^n$ and Σ is a *positive definite matrix*, if the joint probability density function of $\boldsymbol{X}$ is of the form:

$$f_{\boldsymbol{X}}(\boldsymbol{x}) = \frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}} e^{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^\top \Sigma^{-1}(\boldsymbol{x}-\boldsymbol{\mu})} \tag{3.5.1}$$

Many times, we will express the *multivariate gaussian distribution* as: $p\{\boldsymbol{x}|\boldsymbol{\mu},\boldsymbol{\Sigma}\} = N\{\boldsymbol{x}|\boldsymbol{\mu},\boldsymbol{\Sigma}\}$, or $\boldsymbol{X} \sim N\{\boldsymbol{\mu},\boldsymbol{\Sigma}\}$.

3.5.1 The quadratic form $\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^\top \Sigma^{-1}(\boldsymbol{x}-\boldsymbol{\mu})$

If S is a definite positive matrix, then the graph of the quadratic form

$$z = \frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^\top \Sigma^{-1}(\boldsymbol{x}-\boldsymbol{\mu}) \tag{3.5.2}$$

is a $n+1$ dimensional paraboloid that takes always values of z which are non-negative and the only point at which $z=0$ is $\boldsymbol{\mu}$.

Example 3.3 (Precission matrix Σ^{-1}) *For instance, let* $\Sigma = \begin{bmatrix} \frac{3}{2} & -\frac{1}{2} \\ -\frac{1}{2} & \frac{3}{2} \end{bmatrix}$. *To diagonalize Σ, we find the eigenvalues and orthonormal eigenvectors:*

$$\Sigma = \begin{bmatrix} \frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} \\ \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix} \begin{bmatrix} \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \\ -\frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \end{bmatrix}$$

Meaning that:

$$\Sigma^{-1} = \begin{bmatrix} \frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} \\ \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & \frac{1}{2} \end{bmatrix} \begin{bmatrix} \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \\ -\frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \end{bmatrix} = \begin{bmatrix} \frac{3}{4} & \frac{1}{4} \\ \frac{1}{4} & \frac{3}{4} \end{bmatrix}.$$

Figure 4 we plot the 3-d parabole $\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^\top \Sigma^{-1}(\boldsymbol{x}-\boldsymbol{\mu})$ for $\boldsymbol{\mu} = [5,5]^\top$:

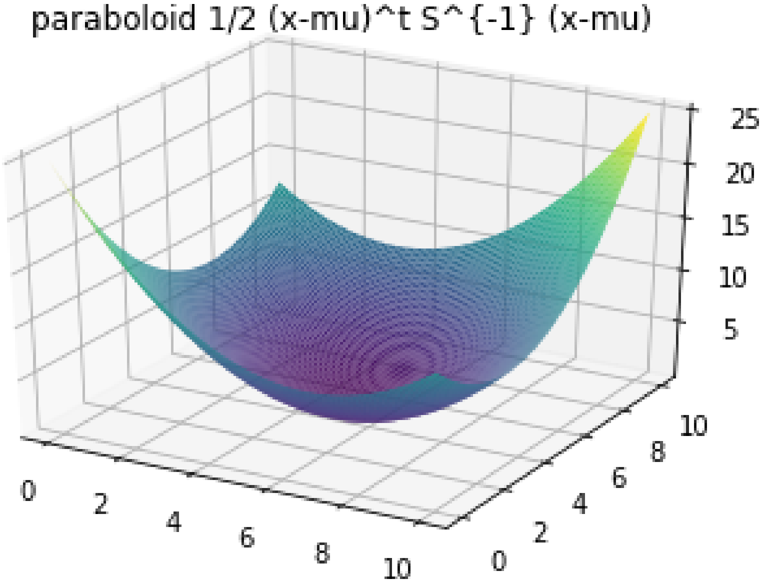


Figure 4: Paraboloid of a quadratic form given in the example.

3.5.2 Isocontour lines (or α-level sets)

We define the **isocontour lines** (or α-level sets) as the surface that represents points of a constant value within a volume of space:

$$(\boldsymbol{x}-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu}) = c^2 \tag{3.5.3}$$

The equation of an ellipsoid centered at the origin and of semi-axis given by a_i oriented according to the orthonormal vectors $\boldsymbol{q_i}$ is

$$\boldsymbol{x}^\top \boldsymbol{Q} \begin{bmatrix} \frac{1}{a_1^2} & 0 & \dots & 0 \\ 0 & \frac{1}{a_2^2} & \dots & 0 \\ 0 & 0 & \dots & 0 \\ \dots & \dots & \dots & \frac{1}{a_n^2} \end{bmatrix} \boldsymbol{Q}^\top \boldsymbol{x} = 1 \tag{3.5.4}$$

If we plot the geometrical locus of the points that fulfill the equation:

$$(\boldsymbol{x}-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu}) = c^2$$

we would obtain an n dimensional ellipsoid, centered at the point $\boldsymbol{\mu}$, with axis aligned with the (orthonormal) eigenvectors $\boldsymbol{q_i}$ of the matrix Σ, and with semi-axis length in the axis pointed by $\boldsymbol{q_i}$ equal to $c\sqrt{\lambda_i}$, where λ_i is the eigenvalue associated with $\boldsymbol{q_i}$.

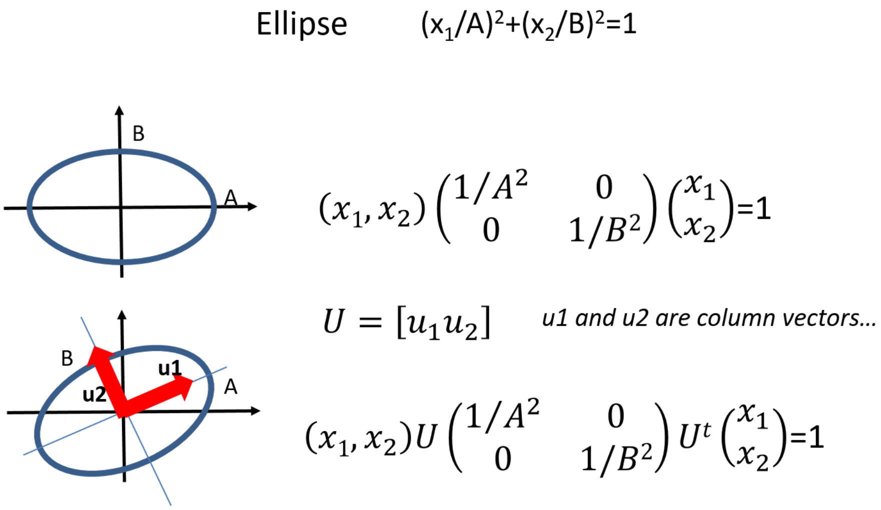

Ellipse $(x_1/A)^2+(x_2/B)^2=1$
B
A
$(x_1, x_2)\begin{pmatrix} 1/A^2 & 0 \\ 0 & 1/B^2 \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix}=1$
$U = [u_1 u_2]$
u1 and u2 are column vectors…
B
u1
u2
A
$(x_1, x_2)U\begin{pmatrix} 1/A^2 & 0 \\ 0 & 1/B^2 \end{pmatrix}U^t\begin{pmatrix} x_1 \\ x_2 \end{pmatrix}=1$

Figure 5: Ellipse equations in terms of a quadratic form.

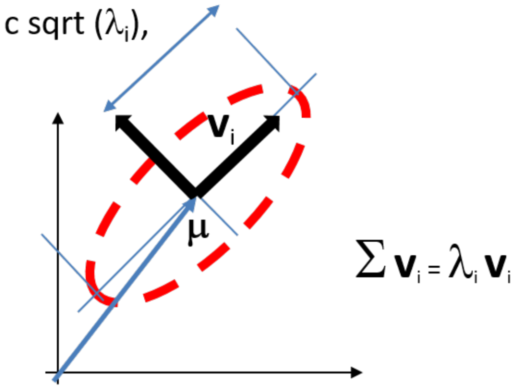


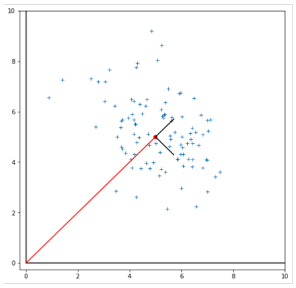
Figure 6: Ellipse equations in terms of eigenvalue-eigenvectors.

3.5.3 Isotropic Gaussian distribution

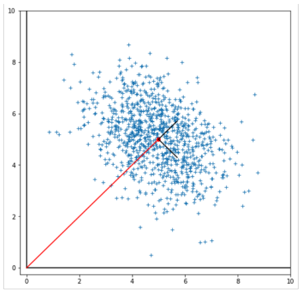
The special case of having $\boldsymbol{\mu} = 0$ and $\boldsymbol{\Sigma}$=$\boldsymbol{I}$, we say that we have the *multivariate standard normal distribution*, and if $\boldsymbol{\Sigma}$= $\sigma^2\boldsymbol{I}$, the distribution is called a *isotropic Gaussian distribution*, meaning that instead of ellipsoids, we will obtain hyperspheres.

3.5.4 Datasets generated from independent sampling of a multivariate Gaussian distribution

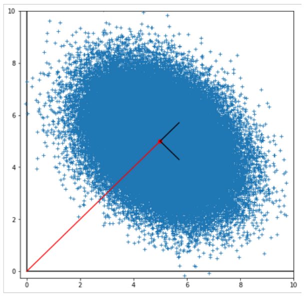
If we perform a number of independent sampling of a multivariate Gaussian distribution, we will obtain clouds of points following the previously described ellipsoids:



N=100 points



N=1000 points



N=100000 points

3.5.5 Expressing $\frac{1}{2n}\boldsymbol{x}^\top\boldsymbol{\Sigma}^{-1}\boldsymbol{x}$ as the trace of the product of two matrices

Assume that $\boldsymbol{x}$ is a vector with 0 mean (otherwise, we would use $\boldsymbol{x}-\boldsymbol{m}$ instead). Scalars are special cases of square matrices, meaning that

$$\frac{1}{2n}\boldsymbol{x}^\top\boldsymbol{\Sigma}^{-1}\boldsymbol{x} = tr(\frac{1}{2n}\boldsymbol{x}^\top\boldsymbol{\Sigma}^{-1}\boldsymbol{x}) = tr(\frac{1}{2n}\boldsymbol{x}\boldsymbol{x}^\top\boldsymbol{\Sigma}^{-1}) = \frac{1}{2}tr(S_n\boldsymbol{\Sigma}^{-1}) \quad (3.5.5)$$

where $\boldsymbol{S}_n = \frac{1}{n}\boldsymbol{x}\boldsymbol{x}^\top$ is the sample covariance-variance matrix.

3.5.6 The term $|\boldsymbol{\Sigma}|^{1/2}$

The determinant of a matrix is the product of its eigenvalues, meaning that $|\boldsymbol{\Sigma}|^{1/2} = (\prod_i \lambda_i)^{1/2}$.

4 The Singular Value Decomposition (SVD)

The diagonalization of a symmetric matrix is an extremely important result, that tells us that a symmetric matrix has as a ”core” a diagonal matrix with real diagonal elements. A similar result can be generalized for some other square matrices, but it cannot be applied to general (possibly non-square) matrices.

There is however another factorization that can be applied to *general* matrices, even for non-square matrices which is known as Singular Value Decomposition

(SVD), that has a lot of applications, including dimensionality reduction applications (principal component analysis, PCA), obtaining the effective rank of a matrix (closest rank approximation of a matrix, e.g. Eckart-Young theorem), calculate the generalized inverse of a matrix (pseudoinverse), or in linear least squares problems.

4.1 The SVD

As we have seen if $\boldsymbol{S}$ is a symmetric $n{\times}n$ matrix, we can find a set of orthonormal vectors $\boldsymbol{u}_k$, which are left and right eigenvectors of $\boldsymbol{S}$ associated with the real eigenvalues λ_k, meaning that they fulfill the equations:

$$\boldsymbol{S}\boldsymbol{u}_i = \lambda_i \boldsymbol{u}_i, \quad i \in \{1,..,n\} \tag{4.1.1}$$

The SVD generalizes these equalities for a general $m{\times}n$ (i.e. m rows and n columns) matrix $\boldsymbol{A}{\in}\mathbb{R}^{m\times n}$ of rank $r{\leq}min(m,n)$. The idea is to find **orthonormal** matrices $\boldsymbol{U}{\in}\mathbb{R}^{m\times m}$ and $\boldsymbol{V}{\in}\mathbb{R}^{n\times n}$ and diagonal matrix $\boldsymbol{\Sigma}{\in}\mathbb{R}^{m\times n}$, such as $\boldsymbol{A} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top$ (Figure 7). The elements of the diagonal matrix $\boldsymbol{\Sigma}_{ii} = \sigma_i$ are real positive numbers called singular values and which we will order as a nonincreasing order $\sigma_1 \geq \sigma_2 \geq ... \geq \sigma_r \geq 0$.

$$\boldsymbol{\Sigma} = \left[\begin{array}{c|c} \boldsymbol{\Sigma}_r & 0 \\ \hline 0 & 0 \end{array}\right] \tag{4.1.2}$$

with $\boldsymbol{\Sigma}_r$=diag(σ_1, $\ldots$, σ_r). For example, let us see the structure of matrix $\boldsymbol{\Sigma}$ for rank 1 and 2 in a 5×5 matrix:

$$\left[\begin{array}{c|cccc} \sigma_1 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{array}\right] \quad ; \quad \left[\begin{array}{cc|ccc} \sigma_1 & 0 & 0 & 0 & 0 \\ 0 & \sigma_2 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{array}\right] \tag{4.1.3}$$

We can observe the similitude with the EVD (eigenvector decomposition) if put matrix $\boldsymbol{V}$ on the left and see that it is satisfied the following expression:

$$\boldsymbol{A}\boldsymbol{v}_i = \sigma_i \boldsymbol{u}_i, \quad i \in \{1,..,n\} \tag{4.1.4}$$

Surprisingly, *we will see that this decomposition is always possible.* The values σ_k are called *singular values* of $\boldsymbol{A}$, while the columns of $\boldsymbol{U}$ and $\boldsymbol{V}$ are called the **left** and **right singular vectors** of $\boldsymbol{A}$.

Remember that orthonormality of singular vectors mean that

$$\boldsymbol{U}\boldsymbol{U}^\top = \boldsymbol{U}^\top\boldsymbol{U} = \boldsymbol{I_m} \tag{4.1.5}$$

and

$$\boldsymbol{V}\boldsymbol{V}^\top = \boldsymbol{V}^\top\boldsymbol{V} = \boldsymbol{I_n} \tag{4.1.6}$$

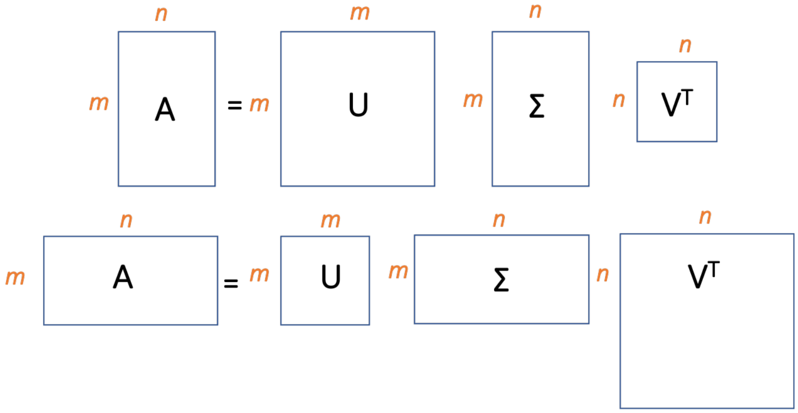


Figure 7: SVD for a $m \times n$ $\boldsymbol{A}$ matrix.

4.2 Relation between EVD (eigenvalue decomposition) and SVD (singular value decomposition)

Let us have a non-symmetric real matrix $\boldsymbol{A} \in \mathbb{R}^{m \times n}$, and assume a SVD as $\boldsymbol{A} = \boldsymbol{U\Sigma V}^\top$. We can see that the following relationships hold:

- Case $\boldsymbol{A}^\top \boldsymbol{A}$: in this case we obtain a $n \times n$ square matrix and:

  $$\boldsymbol{A}^\top \boldsymbol{A} = (\boldsymbol{U\Sigma V}^\top)^\top \boldsymbol{U\Sigma V}^\top = \boldsymbol{V\Sigma}^\top \boldsymbol{U}^\top \boldsymbol{U\Sigma V}^\top = \boldsymbol{V\Sigma}^\top \boldsymbol{\Sigma V}^\top \quad (4.2.1)$$

  which says that $\boldsymbol{A}^\top \boldsymbol{A}$ has as eigenvectors the columns of $\boldsymbol{V}$ (right singular vectors).

- Case $\boldsymbol{A}\boldsymbol{A}^\top$: in this case we obtain a $m \times m$ square matrix and:

  $$\boldsymbol{A}\boldsymbol{A}^\top = \boldsymbol{U\Sigma V}^\top (\boldsymbol{U\Sigma V}^\top)^\top = \boldsymbol{U\Sigma V}^\top \boldsymbol{V}^\top \boldsymbol{\Sigma}^\top \boldsymbol{U}^\top = \boldsymbol{U\Sigma\Sigma}^\top \boldsymbol{U}^\top \quad (4.2.2)$$

  which says that $\boldsymbol{A}\boldsymbol{A}^\top$ has as eigenvectors the columns of $\boldsymbol{U}$ (left singular vectors).

Finally, we can observe that the eigenvalues of $\boldsymbol{A}^\top \boldsymbol{A}$ and $\boldsymbol{A}\boldsymbol{A}^\top$ are the squares of the singular values of $\boldsymbol{A}$: $\sigma_i^2 {=} \lambda_i$ or $\sigma_i {=} \sqrt{\lambda_i}$.

4.3 Selecting the orthonormal basis in the SVD

Let us see how do we select the orthonormal basis $\{\boldsymbol{u_1}, \dots, \boldsymbol{u_m}\}$ for the range and the orthonormal basis $\{\boldsymbol{v_1}, \dots, \boldsymbol{v_n}\}$ for the domain (remember that we have a function $f{:}\mathbb{R}^n \rightarrow \mathbb{R}^m$ associated to matrix $\boldsymbol{A} \in \mathbb{R}^{m \times n}$). Let us assume that rank($\boldsymbol{A}$)=r; then, we will have:

- The left eigenvectors $\boldsymbol{u_1},\dots,\boldsymbol{u_r}$ with $\boldsymbol{u_i}\in\mathbb{R}^m$ are the orthonormal basis of the Image($\boldsymbol{A}$)= CS($\boldsymbol{A}$);

- The left eigenvectors $\boldsymbol{u_{r+1}},\dots,\boldsymbol{u_m}$ with $\boldsymbol{u_i}\in\mathbb{R}^m$ are the orthonormal basis of the Left Null Space LNS($\boldsymbol{A}$)= NS($\boldsymbol{A}^\top$);

$$\boldsymbol{U} = [\boldsymbol{U}_r|\tilde{\boldsymbol{U}}_{m-r}] = [\underbrace{\boldsymbol{u}_1,\dots,\boldsymbol{u}_r}_{\boldsymbol{U}_r},\underbrace{\boldsymbol{u}_{r+1},\dots,\boldsymbol{u}_m}_{\tilde{\boldsymbol{U}}_r}] \tag{4.3.1}$$

- The right eigenvectors $\boldsymbol{v_1},\dots,\boldsymbol{v_r}$ with $\boldsymbol{v_i}\in\mathbb{R}^n$ are the orthonormal basis of the row space RS($\boldsymbol{A}$);

- The right eigenvectors $\boldsymbol{v_{r+1}},\dots,\boldsymbol{v_n}$ with $\boldsymbol{v_i}\in\mathbb{R}^n$ are the orthonormal basis of the null space NS($\boldsymbol{A}$);

$$\boldsymbol{V} = [\boldsymbol{V}_r|\tilde{\boldsymbol{V}}_{n-r}] = [\underbrace{\boldsymbol{v}_1,\dots,\boldsymbol{v}_r}_{\boldsymbol{V}_r},\underbrace{\boldsymbol{v}_{r+1},\dots,\boldsymbol{v}_n}_{\tilde{\boldsymbol{V}}_r}] \tag{4.3.2}$$

We first assume symmetric matrix $\boldsymbol{A}^\top\boldsymbol{A}\in\mathbb{R}^{n\times n}$, and its eigenvalue decomposition (obtain its eigenvalues and eigenvectors since it is a squared $n\times n$ matrix): $\boldsymbol{A}^\top\boldsymbol{A} = \boldsymbol{VDV}^\top$, where $\boldsymbol{D}$ is a diagonal matrix with eigenvalues λ_i arranged in nonincreasing order ($\lambda_1 \geq \lambda_2 \geq \cdots \geq 0$), and the columns of $\boldsymbol{V}$ (eigenvectors of $\boldsymbol{A}^\top\boldsymbol{A}$) are the orthonormal vectors $\{\boldsymbol{v_1},\dots,\boldsymbol{v_n}\}$. We can observe that vectors $\{\boldsymbol{v_1},\dots,\boldsymbol{v_n}\}$ form a basis of $\mathbb{R}^n$. Moreover, since our matrix $\boldsymbol{A}$ has rank($\boldsymbol{A}$)=r, the right singular vectors associated with non-zero singular values (there are r of them) form a basis of RS($\boldsymbol{A}$), while the n-r right singular vectors associated with the zero n-r singular values form a basis of the NS($\boldsymbol{A}$).

Remember that the $Im(\boldsymbol{A}) = Range(\boldsymbol{A}) = \{\mathbf{y}\in\mathbb{R}^m|\boldsymbol{y} = \boldsymbol{Ax}\ for\ some\ \mathbf{x}\in\mathbb{R}^n\}$, and then:

$$\boldsymbol{Av_i}\cdot\boldsymbol{Av_j} = (\boldsymbol{Av_i})^\top\cdot\boldsymbol{Av_j} = \boldsymbol{v_i}^\top(\boldsymbol{A}^\top\boldsymbol{Av_j}) = \boldsymbol{v_i}^\top(\lambda_j\boldsymbol{v_j}) = \lambda_j\boldsymbol{v_i}^\top\boldsymbol{v_j} = 0 \tag{4.3.3}$$

since $\boldsymbol{v_i}$ and $\boldsymbol{v_j}$ are orthonormal. Then $\boldsymbol{Av_i}$ and $\boldsymbol{Av_j}$ also are orthogonal (not necessarily orthonormal). Then the eigenvectors of $\boldsymbol{A}^\top\boldsymbol{A}$ and their images under $\boldsymbol{A}$ form a basis for the range or image of $\boldsymbol{A}$ (or the column space CS(A)).

We must check now that the vectors $\boldsymbol{u}_k$ defined as $A\boldsymbol{v}_k = \sigma_k\boldsymbol{u}_k$ are also orthonormal:

$$\boldsymbol{u}_i^\top\boldsymbol{u}_j = \frac{1}{\sigma_i\sigma_j}\boldsymbol{v}_i^\top A^\top A\boldsymbol{v}_j = \frac{\sigma_j^2}{\sigma_i\sigma_j}\boldsymbol{v}_i^\top\boldsymbol{v}_j = 0,\ \ i\neq j \tag{4.3.4}$$

In order to complete the orthonormal bases of the range, we have to normalize, thus:

$$\boldsymbol{u_i} = \frac{\boldsymbol{Av_i}}{|\boldsymbol{Av_i}|} = \frac{\boldsymbol{Av_i}}{\sqrt{\lambda_i}} = \frac{\boldsymbol{Av_i}}{\sigma_i};\quad 1\leq i\leq r \tag{4.3.5}$$

and defining $\sigma_i^2 = \lambda_i$, we obtain $\boldsymbol{Av_i} = \sigma_i \boldsymbol{u_i}$, with $1 \le i \le r$, which shows that $\boldsymbol{AV} = \boldsymbol{U\Sigma}$ taking $\boldsymbol{u_i}$ the columns of $\boldsymbol{U}$ and $\boldsymbol{v_i}$ the columns of $\boldsymbol{V}$ and $\boldsymbol{\Sigma}_{ii} = \sigma_i$.

Finally, we can express the matrix $\boldsymbol{A}$ as:

$$\boldsymbol{A} = [\boldsymbol{U}_r, \tilde{\boldsymbol{U}}_r] \left[\begin{array}{c|c} \boldsymbol{\Sigma}_r & 0 \\ \hline 0 & 0 \end{array}\right] \left[\begin{array}{c} \boldsymbol{V}_r \\ \hline \tilde{\boldsymbol{V}}_r \end{array}\right] \tag{4.3.6}$$

The dimensions of the matrices are $\boldsymbol{U}_r \in \mathbb{R}^{m\times r}$, $\tilde{\boldsymbol{U}}_r \in \mathbb{R}^{m\times(m-r)}$, $\boldsymbol{V}_r \in \mathbb{R}^{n\times r}$, and $\tilde{\boldsymbol{V}}_r \in \mathbb{R}^{n\times(n-r)}$. Observe that the following conditions are satisfied:

$$\begin{array}{llll} \boldsymbol{U}_r^\top \boldsymbol{U}_r = \boldsymbol{I}_r & \tilde{\boldsymbol{U}}_r^\top \tilde{\boldsymbol{U}}_r = \boldsymbol{I}_{m-r} & \boldsymbol{U}_r^\top \tilde{\boldsymbol{U}}_r = \boldsymbol{0} & \boldsymbol{U}_r^\top \boldsymbol{U}_r + \tilde{\boldsymbol{U}}_r^\top \tilde{\boldsymbol{U}}_r = \boldsymbol{I}_m \\ \boldsymbol{V}_r^\top \boldsymbol{V}_r = \boldsymbol{I}_r & \tilde{\boldsymbol{V}}_r^\top \tilde{\boldsymbol{V}}_r = \boldsymbol{I}_{n-r} & \boldsymbol{V}_r^\top \tilde{\boldsymbol{V}}_r = \boldsymbol{0} & \boldsymbol{V}_r^\top \boldsymbol{V}_r + \tilde{\boldsymbol{V}}_r^\top \tilde{\boldsymbol{V}}_r = \boldsymbol{I}_n \end{array} \tag{4.3.7}$$

Example 4.1 (SVD example) *Let's calculate the SVD for matrix $\boldsymbol{A} \in \mathbb{R}^{3\times 2}$:*

$$\boldsymbol{A} = \begin{bmatrix} 2 & 1 \\ 3 & 0 \\ 1 & 3 \end{bmatrix} = \begin{bmatrix} -0.531 & -0.113 & -0.839 \\ -0.595 & -0.654 & 0.466 \\ -0.602 & 0.748 & 0.279 \end{bmatrix} \begin{bmatrix} 4.169 & 0 \\ 0 & 2.572 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} -0.560 & 0.828 \\ 0.828 & 0.56 \end{bmatrix}$$

where we have used the full matrix decomposition; $\boldsymbol{A} = \boldsymbol{U\Sigma V}^\top$. If we consider the SVD with rank r=2, we can use the economy SVD $\boldsymbol{A} = \boldsymbol{U}_r \boldsymbol{\Sigma}_r \boldsymbol{V}_r^\top$:

$$\boldsymbol{A} = \begin{bmatrix} 2 & 1 \\ 3 & 0 \\ 1 & 3 \end{bmatrix} = \begin{bmatrix} -0.531 & -0.113 \\ -0.595 & -0.654 \\ -0.602 & 0.748 \end{bmatrix} \begin{bmatrix} 4.169 & 0 \\ 0 & 2.572 \end{bmatrix} \begin{bmatrix} -0.560 & 0.828 \\ 0.828 & 0.56 \end{bmatrix}$$

It can be easily seen that $\boldsymbol{u}_i^\top \boldsymbol{u}_i = 1$ and $\boldsymbol{u}_i^\top \boldsymbol{u}_j = 0$ with $i \neq j$, e.g. $\boldsymbol{u}_0^\top \boldsymbol{u}_0 =$ $[-0.531, -0.595, -0.602]^\top [-0.531, -0.595, -0.602] = 1.0$ and $\boldsymbol{u}_0^\top \boldsymbol{u}_1 =$ [−0.531, −0.595, −0.602]$^\top$ [−0.113, −0.654, 0.748]= 0.0.

4.4 Geometric interpretation

Remember that we have a function f:$\mathbb{R}^n \to \mathbb{R}^m$ associated to matrix $\boldsymbol{A} \in \mathbb{R}^{m\times n}$. What happens if we apply matrix $\boldsymbol{A}$ to a unit sphere $\mathbb{R}^n$ (we assume $m \le n$) ?

Let us assume that $\boldsymbol{x} \in \mathbb{R}^n$ are the vectors in the unit sphere. Remembering that when we apply an orthonormal matrix to a vector we rotate the vector, then:

$$\boldsymbol{Ax} = \boldsymbol{U\Sigma V}^\top \boldsymbol{x} \tag{4.4.1}$$

which means that we first apply a rotation $\boldsymbol{V}^\top$ to vectors $\boldsymbol{x}$ (still in $\mathbb{R}^n$), then we stretch or shrink in each direction (since we multiply vectors by singular values larger or than 0), producing an ellipsoid in $\mathbb{R}^m$, and finally we rotate again using $\boldsymbol{U}$ in $\mathbb{R}^m$.

In other words, we produce a linear mapping in which a sphere in $\mathbb{R}^n$ is converted to an ellipsoid in $\mathbb{R}^m$ (make some figures showing the geometric interpretation).

4.5 SVD, pseudo-inverse and projection matrices

Let us consider a linear system of equations:

$$\boldsymbol{A}x = y \tag{4.5.1}$$

where $\boldsymbol{A}\in\mathbb{R}^{m\times n}$, $\boldsymbol{x}\in\mathbb{R}^{n}$ and $\boldsymbol{y}\in\mathbb{R}^{m}$.

If $m{=}n$ we have a **two-sided inverse** of matrix $\boldsymbol{A}$, i.e., $\boldsymbol{A}\boldsymbol{A}^{-1} = \boldsymbol{A}^{-1}\boldsymbol{A} = \boldsymbol{I}$, that is what we call *inverse* of $\boldsymbol{A}$ if the nullspace($\boldsymbol{A}$) and nullspace($\boldsymbol{A}^\top$) only contain the zero vector.

Let us consider the case in which the rank($\boldsymbol{A}$)=r=n, and then the nullspace($\boldsymbol{A}$) only contains the zero vector. In this case $m>n$ (**overdetermined case**) in which case we can not produce an inverse. However, $\boldsymbol{A}^\top\boldsymbol{A}$ has inverse since it is a symmetric $n\times n$ matrix. From here and the SVD we can deduce what is called the **left pseudo-inverse**:

$$\boldsymbol{A}^{-1}_{left} = \boldsymbol{A}^{\dagger} = (\boldsymbol{A}^\top\boldsymbol{A})^{-1}\boldsymbol{A}^\top \tag{4.5.2}$$

Deduction:

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{y} \rightarrow \boldsymbol{A}^\top\boldsymbol{A}\boldsymbol{x} = \boldsymbol{A}^\top\boldsymbol{y} \tag{4.5.3}$$

We can observe that: $\boldsymbol{A}^\top\boldsymbol{A} = \boldsymbol{V}\boldsymbol{\Sigma}^\top\boldsymbol{\Sigma}\boldsymbol{V}^\top$, and:

$$\boldsymbol{V}\boldsymbol{\Sigma}^\top\boldsymbol{\Sigma}\boldsymbol{V}^\top\boldsymbol{x} = \boldsymbol{A}^\top\boldsymbol{y} \rightarrow \boldsymbol{x} = \boldsymbol{V}(\boldsymbol{\Sigma}^\top\boldsymbol{\Sigma})^{-1}\boldsymbol{V}^\top\boldsymbol{A}^\top\boldsymbol{y} \tag{4.5.4}$$

since $(\boldsymbol{A}^\top\boldsymbol{A})^{-1} = \boldsymbol{V}(\boldsymbol{\Sigma}^\top\boldsymbol{\Sigma})^{-1}\boldsymbol{V}^\top$, we conclude that:

$$\boldsymbol{x} = (\boldsymbol{A}^\top\boldsymbol{A})^{-1}\boldsymbol{A}^\top\boldsymbol{y} = \boldsymbol{A}^{-1}_{left}\boldsymbol{y} \tag{4.5.5}$$

Note that $\boldsymbol{A}^{-1}_{left}\boldsymbol{A} = \boldsymbol{I_n}$ and that $\boldsymbol{A}\boldsymbol{A}^{-1}_{left}$ only is $\boldsymbol{I}_n$ if $n = m$. As a statement, a non-symmetric matrix can not have a two-sided since $\boldsymbol{A}$ or $\boldsymbol{A}^\top$ has null-space different of the zero vector.

In the same way, let us consider the case in which the rank($\boldsymbol{A}$)=r=m, and then the nullspace($\boldsymbol{A}^\top$) only contains the zero vector. In this case $m<n$ (**underdetermined case**) in which case we can not produce an inverse. However, $\boldsymbol{A}\boldsymbol{A}^\top$ has inverse since it is a symmetric $n\times n$ matrix. From here and the SVD we can deduce what is called the **right pseudo-inverse**:

$$\boldsymbol{A}^{-1}_{right} = \boldsymbol{A}^{\dagger} = \boldsymbol{A}^\top(\boldsymbol{A}\boldsymbol{A}^\top)^{-1} \tag{4.5.6}$$

Note that $\boldsymbol{A}\boldsymbol{A}^{-1}_{right} = \boldsymbol{I_m}$ and that $\boldsymbol{A}^{-1}_{right}\boldsymbol{A}$ only is $\boldsymbol{I}_m$ if $n = m$. As a statement, a non-symmetric matrix can not have a two-sided since $\boldsymbol{A}$ or $\boldsymbol{A}^\top$ has null-space different of the zero vector.

Finally, If $\boldsymbol{A}$ is full column rank ($r = n$), and $\boldsymbol{A}^{-1}_{left} = (\boldsymbol{A}^\top\boldsymbol{A})^{-1}\boldsymbol{A}^\top$, then matrix

$$\boldsymbol{P} = \boldsymbol{A}\boldsymbol{A}^{-1}_{left} = \boldsymbol{A}(\boldsymbol{A}^\top\boldsymbol{A})^{-1}\boldsymbol{A}^\top \tag{4.5.7}$$

projects $\mathbb{R}^m$ into the column space of $\boldsymbol{A}$.

In a similar way, If $\boldsymbol{A}$ is full row rank ($r = m$), and $\boldsymbol{A}_{right}^{-1} = \boldsymbol{A}^\top(\boldsymbol{A}^\top\boldsymbol{A})^{-1}$, then matrix

$$\boldsymbol{P} = \boldsymbol{A}_{right}^{-1}\boldsymbol{A} = \boldsymbol{A}^\top(\boldsymbol{A}^\top\boldsymbol{A})^{-1}\boldsymbol{A} \tag{4.5.8}$$

projects $\mathbb{R}^n$ into the row space of $\boldsymbol{A}$.

What is the pseudo-inverse in terms of the SVD? The pseudo-inverse is a matrix that satisfies that $\boldsymbol{x} = \boldsymbol{A}\boldsymbol{A}^\dagger\boldsymbol{x}$. In other words: $\boldsymbol{U\Sigma V}^\top\boldsymbol{A}^\dagger = \boldsymbol{I}$, and remembering that matrices $\boldsymbol{U}$ and $\boldsymbol{V}$ are othornormal (and easy to invert), then $\boldsymbol{A}^\dagger = \boldsymbol{U}^\top\boldsymbol{\Sigma}^\dagger\boldsymbol{V}$. The best approximation of $\boldsymbol{\Sigma}^\dagger$ is a matrix with diagonal values for $i = 1, \ldots, r$ for the first r rows (or columns), with r the rank of $\boldsymbol{A}$ and the rest of diagonal values are zero.

Example 4.2 (Pseudo-inverse using the SVD) *Remembering example 1.12, we obtained the pseudoinverse of matrix* $\boldsymbol{A}$

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

as $\boldsymbol{A}^\dagger = (\boldsymbol{A}^\top\boldsymbol{A})^{-1}\boldsymbol{A}^\top$*:*

$$\boldsymbol{A}^\dagger = (\begin{bmatrix} 2 & 1 & 0 \\ 1 & 3 & 1 \end{bmatrix} \begin{bmatrix} 2 & 1 \\ 1 & 3 \\ 0 & 1 \end{bmatrix})^{-1} \begin{bmatrix} 2 & 1 & 0 \\ 1 & 3 & 1 \end{bmatrix} = \begin{bmatrix} 0.567 & -0.133 & -0.167 \\ -0.167 & 0.333 & 0.167 \end{bmatrix}$$

then, if we want to solve $\boldsymbol{Ax} = \boldsymbol{b}$*, with* $\boldsymbol{b} = [3, 1, 4]$*, we obtain* $\boldsymbol{x}$=$\boldsymbol{A}^\dagger\boldsymbol{b}$= $[0.9, 0.5]$. *Using the SVD,* $\boldsymbol{A}^\dagger = \boldsymbol{U}^\top\boldsymbol{\Sigma}^\dagger\boldsymbol{V}$*. Since (see example 4.1)*

$$\boldsymbol{U} = \begin{bmatrix} -0.499 & 0.847 & 0.183 \\ -0.834 & -0.413 & -0.365 \\ -0.234 & -0.335 & 0.913 \end{bmatrix}; \boldsymbol{\Sigma} = \begin{bmatrix} 3.719 & 0 \\ 0 & 1.473 \\ 0 & 0 \end{bmatrix}; \boldsymbol{V} \begin{bmatrix} -0.493 & -0.87 \\ 0.87 & -0.493 \end{bmatrix}$$

and considering the inverse of the $\boldsymbol{\Sigma}$ *matrix (*$\boldsymbol{\Sigma}^\dagger$*) as:*

$$\boldsymbol{\Sigma}^\dagger = \begin{bmatrix} 0.268 & 0 \\ 0 & 0.679 \\ 0 & 0 \end{bmatrix}$$

Then, now:

$$\boldsymbol{A}^\dagger = \boldsymbol{U}^\top\boldsymbol{\Sigma}^\dagger\boldsymbol{V} = \begin{bmatrix} 0.567 & -0.133 & -0.167 \\ -0.167 & 0.333 & 0.167 \end{bmatrix}$$

4.6 Economy, compact and truncated SVD

Remember that matrix Σ is a diagonal matrix with r singular values in positions Σ_{ii}, with $i = 1, \ldots, r$ and zeros otherwise:

$$\Sigma = \begin{bmatrix} \sigma_1 & 0 & 0 & ... & 0 & 0 \\ 0 & \sigma_2 & 0 & ... & 0 & 0 \\ ... & ... & ... & ... & ... & ... \\ 0 & 0 & ... & \sigma_r & 0 & ... \\ 0 & 0 & ... & ... & 0 & ... \\ 0 & 0 & ... & ... & ... & 0 \end{bmatrix} \tag{4.6.1}$$

while $V = [\boldsymbol{v}_1, ..., \boldsymbol{v}_n]$ and $U = [\boldsymbol{u}_1, ..., \boldsymbol{u}_m]$.

Let us assume that $k = min(m, n)$. Then, $\boldsymbol{A}$ can be written as:

$$\begin{aligned} \boldsymbol{A} &= \boldsymbol{U\Sigma V}^\top \\ &= [\sigma_1\boldsymbol{u}_1, ..., \sigma_k\boldsymbol{u}_k, \boldsymbol{0}, ..., \boldsymbol{0}][\boldsymbol{v}_1, ..., \boldsymbol{v}_n]^\top \\ &= \sum_{j=1,..k} \sigma_j \boldsymbol{u}_j \boldsymbol{v}_j^\top + \sum_{j=k+1,..n} \boldsymbol{0}\ \boldsymbol{v}_j^\top \\ &= \sum_{j=1,..k} \sigma_j \boldsymbol{u}_j \boldsymbol{v}_j^\top \end{aligned} \tag{4.6.2}$$

Defining $\boldsymbol{\Sigma}_j = diag(\sigma_1, ..., \sigma_k)$, $U_k = [\boldsymbol{u}_1, ..., \boldsymbol{u}_k]$, and $V_k = [\boldsymbol{v}_1, ..., \boldsymbol{v}_k]$, we obtain:

$$\boldsymbol{A} = \boldsymbol{U}_k \boldsymbol{\Sigma}_k \boldsymbol{V}_k^\top \tag{4.6.3}$$

which is known as the *economy SVD.* On the other hand, if the rank r of $\boldsymbol{A}$ is $r<min(m, n)$, then there only are r singular values different of zero, and then

$$\boldsymbol{A} = \boldsymbol{U}_r \boldsymbol{\Sigma}_r \boldsymbol{V}_r^\top \tag{4.6.4}$$

which is known as the *compact SVD.* This fact is useful when obtaining matrices $\boldsymbol{U}$ and $\boldsymbol{V}$ since only u_r and v_r vectors have to be calculated.

Finally, in low-rank aproximation, only t singular values (the t highest ones) are considered, and:

$$\boldsymbol{A} = \boldsymbol{U}_t \boldsymbol{\Sigma}_t \boldsymbol{V}_t^\top \tag{4.6.5}$$

which is known as the *truncated SVD.*

4.7 The SVD as the sum of r matrices of rank 1

Let us take the economy SVD expression: $\boldsymbol{A}{=}\boldsymbol{U}_r\boldsymbol{\Sigma}_r\boldsymbol{V}_r^\top$. and express matrix $\boldsymbol{A}$ as the sum of r matrices of rank 1:

$$\boldsymbol{A} = \boldsymbol{U}_r \boldsymbol{\Sigma}_r \boldsymbol{V}_r^\top = \sum_{k=1}^{r} \sigma_k \boldsymbol{u}_k \boldsymbol{v}_k^\top \tag{4.7.1}$$

This will be a key fact to find the best low-rank approximation (in the following sections) of a matrix by using the SVD.

4.8 Matrix norms

Remember that norms assign a real number (a length) to an element of vector space. The four (non-negativity, positive definiteness, absolute homogeneity and subadditivity or triangle inequality) defining properties of any norm applied to matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ are (assume correct dimensions of the matrices):

- $\|\boldsymbol{A}\| \geq 0$ (positive-valued),
- $\|\boldsymbol{A}\| = 0$ only if $\boldsymbol{A}=0_{mn}$ (definite),
- $\|\lambda \boldsymbol{A}\| = |\lambda| \; \|\boldsymbol{A}\|$ (absolutely homogeneous),
- $\|\boldsymbol{A}+\boldsymbol{B}\| \leq \|\boldsymbol{A}\| + \|\boldsymbol{B}\|$ (triangle inequality).

For matrix norms we introduce the additional condition:

- $\|\boldsymbol{AB}\| \leq \|\boldsymbol{A}\| \; \|\boldsymbol{B}\|$ (sub-multiplicative).

4.8.1 Matrix norms induced by vectors p-norms

One way of defining a matrix norm, is using vector norms. The matrix norm measures how much a vector (assuming a vector p-norm) can increase in size when it is multiplied by $\boldsymbol{A}$. Observe that in the definition we use a vector $\boldsymbol{x}$ that is a unitary vector (i.e., with norm $\|\cdot\| = 1$). From this we can define:

- $l{-}1$ **norm**: maximum absolute column sum of $\boldsymbol{A}$, i.e., sum all columns and take the highest one:

$$\|\boldsymbol{A}\|_1 = max\frac{\|\boldsymbol{A}x\|_1}{\|x\|_1} = max_{\|x\|_1=1}\|\boldsymbol{A}x\|_1 = \max_{1\leq j\leq n}\sum_{i=1}^{m}|a_{ij}| \tag{4.8.1}$$

- $l{-}2$ **norm**: maximum singular value of $\boldsymbol{A}$:

$$\|\boldsymbol{A}\|_2 = max\frac{\|\boldsymbol{A}x\|_2}{\|x\|_2} = max_{\|x\|_2=1}\|\boldsymbol{A}x\|_2 = \sqrt{\lambda_{max}(\boldsymbol{A}^\top\boldsymbol{A})} = \sigma_{max}(\boldsymbol{A}) \tag{4.8.2}$$

  This can be seen because $\boldsymbol{A}=\boldsymbol{U\Sigma V}^\top$, and the vector with maximum length is $\boldsymbol{v}_1$. Then $\|\boldsymbol{A}\boldsymbol{v}_1\|_2 = \|\sigma_1\boldsymbol{u}_1\|_2 = \sigma_1\|\boldsymbol{u}_1\|_2$, taking into account that $\|\boldsymbol{v}_1\|_2$=1 and $\|\boldsymbol{u}_1\|_2$=1, which yields the result;

- $l{-}\infty$ **norm**: maximum absolute row sum of $\boldsymbol{A}$, i.e., sum all rows and take the highest one:

$$\|\boldsymbol{A}\|_\infty = max\frac{\|\boldsymbol{A}x\|_\infty}{\|x\|_\infty} = max_{\|x\|_\infty=1}\|\boldsymbol{A}x\|_\infty = \max_{1\leq i\leq m}\sum_{j=1}^{n}|a_{ij}| \quad (4.8.3)$$

You can find geometrically the value of a matrix norm for a given matrix $\boldsymbol{A}$ geometrically by:

- Plotting the unit sphere for the matrix
- Finding the image under the transformation $\boldsymbol{y} = \boldsymbol{A} = \boldsymbol{x}$
- Finding the maximum of $\|\boldsymbol{y}\|$

That means that an induced matrix norm $\|\boldsymbol{A}\|$ is how much a matrix can stretch a vector to a maximum. If norm of a matrix is say number d; it means it can stretch a vector x by d maximum.

Example 4.3 (Matrix norms) *Let us take as example the matrix:*

$$\boldsymbol{A} = \begin{bmatrix} 1 & 2 \\ 0 & 2 \end{bmatrix}$$

and plot figures for $\|\boldsymbol{A}\|_1$, $\|\boldsymbol{A}\|_2$ and $\|\boldsymbol{A}\|_{infty}$ (Figure 8). Observe that $\sigma_1 = 2.9208$ and $\sigma_2 = 0.6847$. Thus $\|\boldsymbol{A}\|_1 = 4$, $\|\boldsymbol{A}\|_2 = 2.9208$ and $\|\boldsymbol{A}\|_\infty = 3$. Observe, also, how vectors (1,0) transforms to (1,0) and (0,1) transforms in (2,2), that precisely is y_{max} for $\|\boldsymbol{A}\|_1$ and $\|\boldsymbol{A}\|_2$. However, y_{max}=(3,2) corresponds to point (1,1) for $\|\boldsymbol{A}\|_\infty$. Applying the norm definition to y_{max}, you should obtain the same norm than applying the matrix p-norm definitions.

4.8.2 Schatten norms: Matrix norms that can be expressed in terms of singular values

Other possible matrix norms are the Schatten norms, which are defined in terms of the singular values, and which in some cases can be expressed in terms of the trace operator.

The **spectral norm** is the operator norm induced by the vector 2-norm. Then, this norm coincides with the induced vector $p = 2$ norm:

$$\|\boldsymbol{A}\|_2 = max\frac{\|\boldsymbol{A}x\|_2}{\|x\|_2} = \sigma_{max}(\boldsymbol{A}) \quad (4.8.4)$$

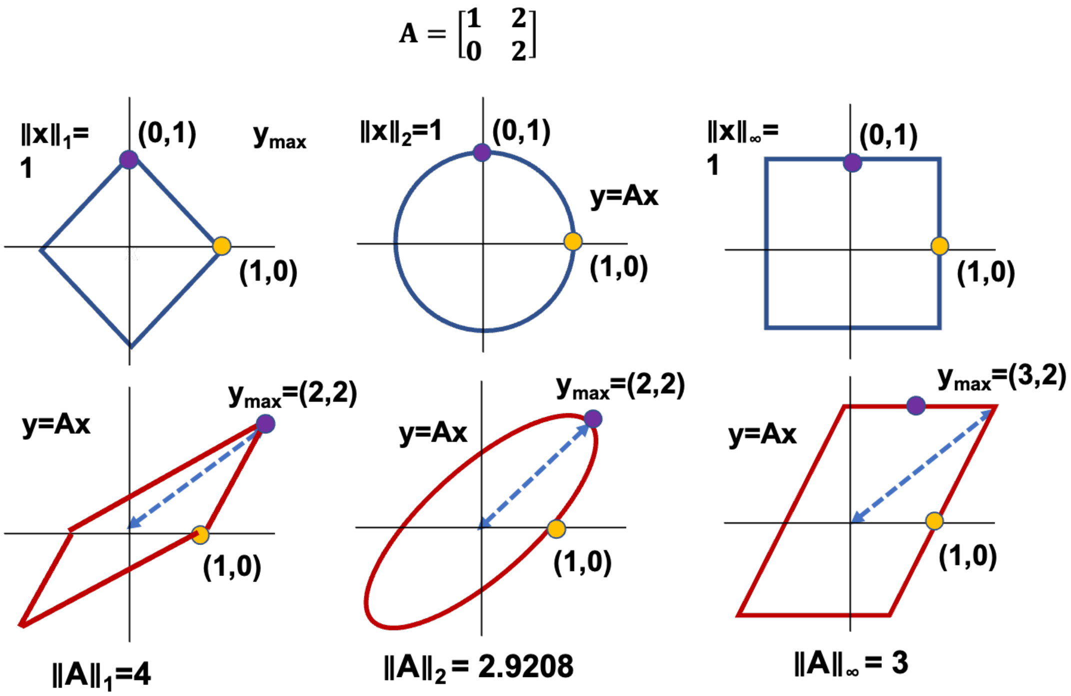


Figure 8: Induced matrix norms representation.

The **Frobenius norm** can be defined in several ways (in terms of sum of all absolute coefficient values of the matrix, in terms of the trace of $(\boldsymbol{A}\boldsymbol{A}^\top)$ or trace of $(\boldsymbol{A}^\top\boldsymbol{A})$, and in terms of the sum of singular values):

$$\|\boldsymbol{A}\|_F = \sqrt{\sum_k \sum_i |a_{ik}|^2} = \sqrt{tr(\boldsymbol{A}^\top\boldsymbol{A})} = \sqrt{tr(\boldsymbol{A}\boldsymbol{A}^\top)} = \sqrt{\sum_k \sigma_k^2} \quad (4.8.5)$$

Frobenius norm is often easier to compute than induced norms, and has the useful property of being invariant under rotations (and unitary operations in general), meaning that if $\boldsymbol{U}$ is a rotation (unitary matrix), then $\|\boldsymbol{A}\|_F = \|\boldsymbol{A}\boldsymbol{U}\|_F = \|\boldsymbol{U}\boldsymbol{A}\|_F$. From here, we get the connection between the expression "summatory of all matrix a_{ij} coefficients" and the summatory of square singular values, since $\boldsymbol{A}=\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top$.

This norm is used in many applications, such as regularization in machine learning/optimization problems when using matrices. An example is in obtaining the Laplacian matrix coefficients in graph signal processing (GSP) from the data measurements (we will see this application in some days).

For the semidefinite matrix $\boldsymbol{A}^\top\boldsymbol{A}$ we can define an square root as a matrix $\boldsymbol{B}$

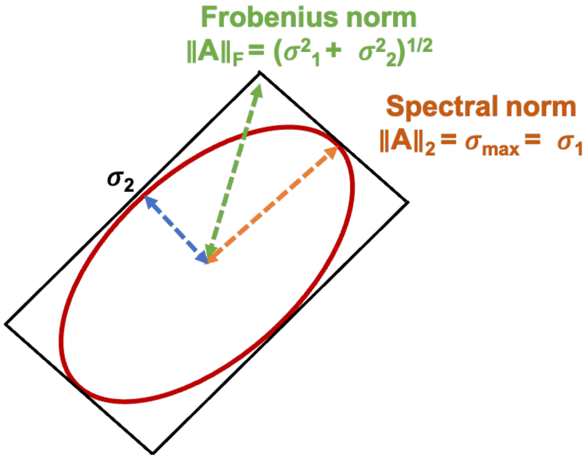

Frobenius norm
$\|A\|_F = (\sigma^2{}_1 + \sigma^2{}_2)^{1/2}$
Spectral norm
$\|A\|_2 = \sigma_{max} = \sigma_1$
σ_2

Figure 9: Spectral and Frobenius norms representation (2-D). The nuclear norm is the sum of $\sigma_1+\sigma_2$ and thus is the perimeter of the paralelogram (sum of orange and blue arrows).

for which $\boldsymbol{B}^2 = \boldsymbol{A}^\top \boldsymbol{A}$. We define the **nuclear (or Ky-Fan) norm or trace norm** as:

$$\|\boldsymbol{A}\|_N = \sum_k \sigma_k = tr(\sqrt{(\boldsymbol{A}^\top \boldsymbol{A})}) \tag{4.8.6}$$

these norms usually appear in infinite dimensional spaces, and also it is often used in mathematical optimization to search for low-rank matrices (measures the "amount of rank-1 matrices" needed to construct $\boldsymbol{A}$). It has applications in deep learning (chooses the best weights in gradient descent when there is more weights than samples), and also appears in other applications such as compressive sensing (express a vector in a "compressed" way with many zero entries).

Example 4.4 (Schatten norms) *Let us assume matrix* $\boldsymbol{A}$

$$\boldsymbol{A} = \begin{bmatrix} 2 & 1 & 0 & 1 \\ 1 & 3 & 4 & 2 \\ 0 & 1 & 3 & 3 \end{bmatrix}$$

Their singular values are $\sigma_1 = 6.909$, $\sigma_2 = 2.238$ *and* $\sigma_3 = 1.501$. *The* **spectral norm** *is* $\|\boldsymbol{A}\|_2 = \sigma_{max}(\boldsymbol{A}) = \sigma_1 = 6.909$. *The* **Frobenius norm** *is* $\|\boldsymbol{A}\|_F = \sqrt{\sum_k \sigma_k^2} = 7.416$, *and the* **nuclear norm** *is* $\|\boldsymbol{A}\|_N = \sum_k \sigma_k = 10.648$.

4.9 Condition number of a matrix

The condition number of a matrix $\boldsymbol{A}\in\mathbb{R}^{n\times n}$ characterizes the sensitivity of the solution of a linear system $\boldsymbol{Ax} = \boldsymbol{b}$ to small changes in $\boldsymbol{A}$ and $\boldsymbol{b}$. Let us take derivatives at both sides of the linear system:

$$\boldsymbol{A}d\boldsymbol{x} + (d\boldsymbol{A})\boldsymbol{x} = d\boldsymbol{b} \quad \Rightarrow \quad d\boldsymbol{x} = \boldsymbol{A}^{-1}(d\boldsymbol{b} - (d\boldsymbol{A})\boldsymbol{x}) \tag{4.9.1}$$

Now, taking the Euclidean norm at both sides:

$$\|d\boldsymbol{x}\| \leq \|\boldsymbol{A}^{-1}\| \, \|(d\boldsymbol{b} - (d\boldsymbol{A})\boldsymbol{x})\| \leq \|\boldsymbol{A}^{-1}\| \, (\|d\boldsymbol{b}\| + \|d\boldsymbol{A}\| \, \|\boldsymbol{x}\|) \tag{4.9.2}$$

Now, we can use the inequality $\|\boldsymbol{b}\| = \|\boldsymbol{Ax}\| \leq \|\boldsymbol{A}\| \, \|\boldsymbol{x}\|$, we get:

$$\frac{\|d(\boldsymbol{x})\|}{\|\boldsymbol{x}\|} \leq \kappa(\boldsymbol{A}) \, (\frac{\|d\boldsymbol{A}\|}{\|\boldsymbol{A}\|} + \frac{\|d\boldsymbol{b}\|}{\|\boldsymbol{b}\|}) \tag{4.9.3}$$

We define the **condition number of matrix A** as:

$$\kappa(\boldsymbol{A}) = \|\boldsymbol{A}\|_2 \|\boldsymbol{A}^{-1}\|_2 = \frac{\sigma_{max}}{\sigma_{min}} \tag{4.9.4}$$

where σ_{max} and σ_{max} are the maximum and minimum singular values of matrix $\boldsymbol{A}$. Large condition number $\kappa(\boldsymbol{A})$ results in a highly sensitive system, that is, small changes in $\boldsymbol{A}$ or $\boldsymbol{b}$ may result in very large changes in the solution $\boldsymbol{x}$. On the other hand, a large condition number $\kappa(\boldsymbol{A})$ implies that $\sigma_{max} >> \sigma_{min}$, and then the matrix $\boldsymbol{A}$ is almost singular (is not invertible).

Example 4.5 (Condition number) *Let us assume matrix $\boldsymbol{A}_0$ (it is a singular matrix) and an approximated matrix $\boldsymbol{A}$.*

$$\boldsymbol{A}_0 = \begin{bmatrix} 2 & 4 \\ 1 & 2 \end{bmatrix}; \quad \boldsymbol{A} = \begin{bmatrix} 2.0002 & 3.9999 \\ 0.9996 & 2.0002 \end{bmatrix}$$

The SVD of matrix $\boldsymbol{A}$ is:

$$\boldsymbol{A} = \begin{bmatrix} 2.0002 & 3.9999 \\ 0.9996 & 2.0002 \end{bmatrix} = \begin{bmatrix} -0.894 & -0.447 \\ -0.447 & 0.894 \end{bmatrix} \begin{bmatrix} 4.999 & 0 \\ 0 & 0.00049 \end{bmatrix} \begin{bmatrix} -0.447 & -0.894 \\ -0.894 & 0.447 \end{bmatrix}$$

The condition number is $\kappa(\boldsymbol{A}) = \sigma_{max}/\sigma_{min} = 4.999/0.00049 = 1000.00$. Let us calculate $\boldsymbol{A}^\dagger$:

$$\boldsymbol{A}^\dagger = \begin{bmatrix} 800.08 & -1599.96 \\ -399.84 & 800.08 \end{bmatrix} =$$

Let us solve the linear system $\boldsymbol{Ax} = \boldsymbol{b}$ with several $\boldsymbol{b}$'s:

$$\begin{bmatrix} 2.0002 & 3.9999 \\ 0.9996 & 2.0002 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}$$

For example $b = [2, 1]$ results in $x = [0.2, 0.4]$, while a small change in component b_1 such as $b = [2.05, 1]$ results in $x = [40.204, -19.592]$, or a small change in component b_2 such as $b = [2, 1.02]$ results in $x = [-31.79, 16.402]$.

4.10 Eckart-Young approximation (low-rank approximation)

Let $\boldsymbol{A}{\in}\mathbb{R}^{m\times n}$ be a matrix with n columns and m rows with $m{\geq}n$ (thus it is full-rank when rank $r{=}n$). Suppose that $\boldsymbol{A} = \boldsymbol{A}\boldsymbol{\Sigma}\boldsymbol{V}^\top$ is the SVD, with $\boldsymbol{U}$ and $\boldsymbol{V}$ are orthonormal matrices, and $\boldsymbol{\Sigma}$ is an $m{\times}n$ diagonal matrix with entries $(\sigma_1, \sigma_2, \cdots, \sigma_n)$ such that $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n \geq 0$.

The Eckart-Young Th. says that the best rank k approximation (a matrix $\boldsymbol{B}$ with rank $k{\leq}n$) to matrix $\boldsymbol{A}$ in the spectral norm $\|\boldsymbol{A}\|_2$ is given by the truncated SVD:

$$\boldsymbol{B} = \boldsymbol{A}_k = \sum_{i=1}^{k} \sigma_i \boldsymbol{u}_i \boldsymbol{v}_i^t \tag{4.10.1}$$

This result can be extended to the Frobenius and the Nuclear norms.

We can prove it for the special cases $n{=}m{=}2$ and $k{=}1$. For the general case, the proof follows the same reasoning.

Let $\boldsymbol{A} = \sigma_1 \boldsymbol{u}_1 \boldsymbol{v}_1^\top + \sigma_2 \boldsymbol{u}_2 \boldsymbol{v}_2^\top$, and $\boldsymbol{A}_1 = \sigma_1 \boldsymbol{u}_1 \boldsymbol{v}_1^\top$. We can easily see that $\|\boldsymbol{A} - \boldsymbol{A}_1\| = \sigma_2$.

Let $\boldsymbol{B}$ an arbitrary rank-1 2×2 general matrix. We can thus express $\boldsymbol{B}$ as: $\boldsymbol{B} = \rho_1 \boldsymbol{x}_1 \boldsymbol{y}_1^\top$. Let $\boldsymbol{w}$ be an element of $Ker(\boldsymbol{B})$ of length 1, which in our case would be an orthonormal vector to $\boldsymbol{y}$. We can express $\boldsymbol{w}$ in terms of $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$ as $\boldsymbol{w} = \gamma_1 \boldsymbol{v}_1 + \gamma_2 \boldsymbol{v}_2$, with $\gamma_1^2 + \gamma_2^2 = 1$, with $\boldsymbol{B}\boldsymbol{w} = 0$.

Using the definition of the spectral norm (which coincides with the induced norm with $p = 2$) we have:

$$\|\boldsymbol{A} - \boldsymbol{B}\|_2^2 \geq \|(\boldsymbol{A} - \boldsymbol{B})\boldsymbol{w}\|_2^2 = \|\boldsymbol{A}\boldsymbol{w}\|_2^2 = \sigma_1^2\gamma_1^2 + \sigma_1^2\gamma_2^2 \geq \sigma_2^2 \tag{4.10.2}$$

meaning that:

$$\|\boldsymbol{A} - \boldsymbol{B}\|_2^2 \geq \|\boldsymbol{A} - \boldsymbol{A}_1\|_2^2 \tag{4.10.3}$$

Example 4.6 (Low rank approximation example) *Let us consider matrix $\boldsymbol{A}$. Find the best rank-1, rank-2 and rank-3 approximations.*

$$\boldsymbol{A} = \begin{bmatrix} 2 & 1 & 0 & 1 \\ 1 & 3 & 4 & 2 \\ 0 & 1 & 3 & 3 \end{bmatrix}$$

First, we obtain the SVD of matrix $\boldsymbol{A} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top$:

$$\boldsymbol{U} = \begin{bmatrix} -0.188 & -0.881 & 0.434 \\ -0.78 & -0.134 & -0.611 \\ -0.596 & 0.454 & 0.662 \end{bmatrix}$$

$$\boldsymbol{V} = \begin{bmatrix} -0.167 & -0.452 & -0.711 & -0.512 \\ -0.847 & -0.371 & 0.368 & 0.094 \\ 0.172 & -0.49 & -0.304 & 0.799 \\ -0.474 & 0.646 & -0.517 & 0.302 \end{bmatrix}$$

and the singular values are:

$$\boldsymbol{\Sigma} = \begin{bmatrix} 6.909 & 0 & 0 & 0 \\ 0 & 2.238 & 0 & 0 \\ 0 & 0 & 1.501 & 0 \end{bmatrix}$$

The best rank-1 approximation is given by $\boldsymbol{A}_1 = \sigma \boldsymbol{u}_1 \boldsymbol{v}_1^\top$

$$\boldsymbol{A}_1 = \sigma_1 \boldsymbol{u}_1 \boldsymbol{v}_1^\top = 6.909 \begin{bmatrix} -0.188 \\ -0.78 \\ -0.596 \end{bmatrix} \begin{bmatrix} -0.167 & -0.452 & -0.711 & -0.512 \end{bmatrix}$$

$$= \begin{bmatrix} 0.217 & 0.588 & 0.923 & 0.665 \\ 0.902 & 2.439 & 3.832 & 2.761 \\ 0.69 & 1.864 & 2.929 & 2.11 \end{bmatrix}$$

In the same way, we can obtain rank-2 ($\boldsymbol{A}_2$) and rank-3 ($\boldsymbol{A}_3$) approximations:

$$\boldsymbol{A}_2 = \sigma_1 \boldsymbol{u}_1 \boldsymbol{v}_1^\top + \sigma_2 \boldsymbol{u}_2 \boldsymbol{v}_2^\top = \begin{bmatrix} 1.888 & 1.319 & 0.198 & 0.479 \\ 1.157 & 2.551 & 3.721 & 2.732 \\ -0.171 & 1.487 & 3.302 & 2.206 \end{bmatrix}$$

$$\boldsymbol{A}_3 = \sigma_1 \boldsymbol{u}_1 \boldsymbol{v}_1^\top + \sigma_2 \boldsymbol{u}_2 \boldsymbol{v}_2^\top + \sigma_3 \boldsymbol{u}_3 \boldsymbol{v}_3^\top = \begin{bmatrix} 2 & 1 & 0 & 1 \\ 1 & 3 & 4 & 2 \\ 0 & 1 & 3 & 3 \end{bmatrix}$$

5 Principal component analysis (PCA)

Large datasets are common in many data science applications. In order to interpret such datasets, it is useful to drastically reduce their dimensionality in an interpretable way, such that most of the information in the data is preserved. One of the oldest and most widely used technique is **principal component analysis (PCA)**, which reduces the dimensionality of a dataset by solving an eigenvalue/eigenvector problem, while preserving as much ”variability” - i.e., statistical information - as possible.

Principal component analysis is basically used as an exploratory tool for data analysis, although there exist several adaptations to other applications such as functional PCA (continuous variables), robust PCA (to avoid sensitiveness to the presence of outliers), etc.

Thus, PCA is a dimensionality reduction method that is typically used to reduce the dimensionality of large data sets. The reduced dimensional representation retains the information conveyed by the large dimensional representation.

PCA is an orthogonal linear transformation that transforms the data into a new coordinate system such that the largest variance by some scalar projection of the data is placed in the first coordinate (called the first principal component), the second largest variance in the second coordinate, etc.

5.1 Interpretation 1: maximizing directions with maximum variability

In other words: **the principal components** of a set of data $\boldsymbol{X}\in\mathbb{R}^{m\times n}$ (assuming $m\leq n$) provide a sequence of best linear approximations to that data, of all ranks $q\leq n$. In other words, given $\boldsymbol{x}_1,\ldots,\boldsymbol{x}_m$ measurements, we want to find a q-rank linear model for representing them.

$$\begin{array}{lcl} y_1 & = & a_{11}\boldsymbol{x}_1 + a_{12}\boldsymbol{x}_2 + \cdots + a_{1n}\boldsymbol{x}_n \\ y_2 & = & a_{21}\boldsymbol{x}_1 + a_{22}\boldsymbol{x}_2 + \cdots + a_{2n}\boldsymbol{x}_n \\ \ldots & & \\ y_q & = & a_{q1}\boldsymbol{x}_1 + a_{q2}\boldsymbol{x}_2 + \cdots + a_{qn}\boldsymbol{x}_n \end{array} \tag{5.1.1}$$

The new axes represent the *directions* with **maximum variability** and provide simpler more concise description of the covariance structure. Let us assume in general that we have matrices $\boldsymbol{C}$ and $\boldsymbol{X}$. Let us assume that $\boldsymbol{X}$ is a random variable with mean $\boldsymbol{\mu_X}$ and covariance matrix $\boldsymbol{\Sigma_X}$ and let's take linear combinations $\boldsymbol{Y}$=$\boldsymbol{CX}$. We know the following properties (seen in probability sections):

$$\boldsymbol{\mu_Y} = E[\boldsymbol{Y}] = E[\boldsymbol{CX}] = \boldsymbol{C}E[\boldsymbol{X}] = \boldsymbol{C\mu_X} \tag{5.1.2}$$

$$\boldsymbol{\Sigma_Y} = Cov[\boldsymbol{Y}] = Cov[\boldsymbol{CX}] = \boldsymbol{C\Sigma_X C}^\top \tag{5.1.3}$$

Then, the PCA are those *uncorrelated* linear combinations $y_1,\ldots,y_q$ whose variances are as large as possible, meaning to maximize Var(y_1)=$\boldsymbol{a}_1^\top\boldsymbol{\Sigma_X}\boldsymbol{a}_1$. We should take care with the length of vectors $\boldsymbol{a}$'s, since multiplying a_1 by any constant will increase the variance. Then, we have to restrict to vectors $\boldsymbol{a}$'s whose lengths are unitary: $\boldsymbol{a}_1^\top\boldsymbol{a}_1$=1. Then the algorithm has to solve the following:

Fist component: $y_1 = \boldsymbol{a}_1^\top\boldsymbol{X}$ that maximizes Var($\boldsymbol{a}_1^\top\boldsymbol{X}$) subject to $\boldsymbol{a}_1^\top\boldsymbol{a}_1$=1.

Second component: $y_2 = \boldsymbol{a}_2^\top\boldsymbol{X}$ that maximizes Var($\boldsymbol{a}_2^\top\boldsymbol{X}$) subject to $\boldsymbol{a}_2^\top\boldsymbol{a}_2$=1 and Cov($\boldsymbol{a}_2^\top\boldsymbol{X}$,$\boldsymbol{a}_1^\top\boldsymbol{X}$)=0.

q-th component: $y_q = \boldsymbol{a}_q^\top\boldsymbol{X}$ that maximizes Var($\boldsymbol{a}_q^\top\boldsymbol{X}$) subject to $\boldsymbol{a}_q^\top\boldsymbol{a}_q$=1 and Cov($\boldsymbol{a}_q^\top\boldsymbol{X}$,$\boldsymbol{a}_j^\top\boldsymbol{X}$)=0 for $j < q$.

Result: Let $\boldsymbol{\Sigma_X}$ be the covariance matrix associated to $\boldsymbol{X}$ with random variable $\boldsymbol{X}$, and have $\boldsymbol{\Sigma_X}$ the eigenvalue-eigenvector pairs (e_1,λ_1),. . . , (e_n,λ_n) with $\lambda_1\geq\lambda_2\geq\ldots\geq\lambda_n\geq 0$. Then, the $i-th$ principal component is given by:

$$y_i = \boldsymbol{e}_i^\top\boldsymbol{X} \tag{5.1.4}$$

with Var(y_i)= $\boldsymbol{e}_i^\top\boldsymbol{\Sigma_X}\boldsymbol{e}_i$=$\lambda_i$ and Covar(y_i,y_j)= $\boldsymbol{e}_i^\top\boldsymbol{\Sigma_X}\boldsymbol{e}_j$=0. That means that the first component is the one with the largest eigenvector.

5.2 Interpretation 2: projecting in a subspace

Let us assume that we have m data measurements, each of dimension n. We arrange the data in a matrix $\boldsymbol{X}{\in}\mathbb{R}^{m\times n}$. Let us assume that we want to project this data in 1-D subspace, whose direction is defined by vector u_1. Without loss of generality, we assume that this vector has length equal to 1 ($\boldsymbol{u}_1^\top\boldsymbol{u}_1{=}1$) since we are interested in the direction and not in the length.

We now project each data point $\boldsymbol{x}_i$, $i{=}1,\dots,m$ in the 1-D subspace defined by $\boldsymbol{u}_1$. This projection amounts to $\boldsymbol{u}_1^\top\boldsymbol{x}_i$, the projected mean is given by $\boldsymbol{u}_1^\top\bar{\boldsymbol{x}}$, with $\bar{\boldsymbol{x}}$ the sample mean:

$$\bar{\boldsymbol{x}} = \frac{1}{m}\sum_{i=1}^{m}\boldsymbol{x}_i \tag{5.2.1}$$

and the projected variance $\boldsymbol{\sigma}_{u_1}^2$ is given by:

$$\boldsymbol{\sigma}_{u_1}^2 = \frac{1}{m}\sum_{i=1}^{m}(\boldsymbol{u}_1^\top\boldsymbol{x}_i - \boldsymbol{u}_1^\top\bar{\boldsymbol{x}})^2{=}\boldsymbol{u}_1^\top\boldsymbol{\Sigma}\boldsymbol{u}_1 \tag{5.2.2}$$

with $\boldsymbol{\Sigma} = \boldsymbol{X}\boldsymbol{X}^\top$ the covariance matrix.

If we want to maximize the projected variance $\boldsymbol{u}_1^\top\boldsymbol{\Sigma}\boldsymbol{u}_1$ we have to consider the constraint $\boldsymbol{u}_1^\top\boldsymbol{u}_1{=}1$, and add a Lagrange multiplier (we will see the meaning of Lagrange multipliers in TOML, non-linear optimization). Then we multiply the constraint by a scalar λ_1 and add it to the variance objective function:

$$\boldsymbol{u}_1^\top\boldsymbol{\Sigma}\boldsymbol{u}_1 + \lambda_1(1 - \boldsymbol{u}_1^\top\boldsymbol{u}_1) \tag{5.2.3}$$

To maximize this expression, we obtain the derivatives with respect to $\boldsymbol{u}_1$ equal to 0, and obtain the following expression we obtain:

$$\boldsymbol{\Sigma}\boldsymbol{u}_1 = \lambda_1\boldsymbol{u}_1 \tag{5.2.4}$$

which says that $\boldsymbol{u}_1$ is an eigenvector with eigenvalue λ_1 and they maximize the variance of the projected data on subspace defined by $\boldsymbol{u}_1$. We can extend easily this idea to higher dimensional projected spaces (more principal components).

5.3 Connection PCA-SVD

PCA and SVD are closely related approaches and can be both applied to decompose any rectangular matrices. Let us assume our data matrix $\boldsymbol{X}{\in}\mathbb{R}^{m\times n}$, and consider the covariance matrix $\boldsymbol{S}{=}\boldsymbol{X}^\top\boldsymbol{X}/(n-1)$. Then:

$$\boldsymbol{S} = \frac{\boldsymbol{X}^\top\boldsymbol{X}}{(n-1)} = \frac{\boldsymbol{V}\boldsymbol{\Sigma}\boldsymbol{U}^\top\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top}{(n-1)} = \frac{\boldsymbol{V}\boldsymbol{\Sigma}^2\boldsymbol{V}^\top}{(n-1)} = \frac{\boldsymbol{V}\boldsymbol{\Sigma}^2\boldsymbol{V}^{-1}}{(n-1)} \tag{5.3.1}$$

since $\boldsymbol{V}$ is a unitary matrix and $\boldsymbol{V}^\top = \boldsymbol{V}^{-1}$, and we know that $\boldsymbol{\Lambda} = \boldsymbol{\Sigma}^2/(n-1)$, meaning that we can perform PCA using SVD or viceversa. In general, when performing PCA is computationally easier to use SVD than EVD, due to the economic/truncated SVD representations.

5.4 Amount of total variance explained

Once we have obtained the principal components, the proportion of total variance explained by the $i-th$ principal component is given by:

$$\frac{\lambda_i}{\sum_{i=1}^{n} \lambda_i} \tag{5.4.1}$$

For example, if the first two-three principal components explain 80-90% of the variability, then it could be worth replacing the n features by these principal components.

5.5 Eigenfaces

We are going to use the SVD/PCA in a denoising application using a set of images. The application is called **eigenfaces**. Let us assume that we have a set of images of people where one person images are taken from different angles, e.g. different bright/lighting conditions or different poses. For example 40 people with 50 images of each person, forming a set of K images. Each image has $p{\times}q$ pixels that are vectorized. Consider that we have $\boldsymbol{L}_1, \ldots, \boldsymbol{L}_M$ faces, where each $\boldsymbol{L}_i$ is a vector representing a face in $\mathbb{R}^N$, with $N{=}p{\times}q$ (a vectorized representation of an image of $p{\times}q$ pixels).

The average face is given by $\boldsymbol{\Psi}{=}1/M \sum_{i=1}^{M} \boldsymbol{L}_i$, and now we obtain the difference of each face with respect the average face: $\boldsymbol{x}_i{=}\boldsymbol{L}_i - \boldsymbol{\Psi}$ with $i{=}1, \ldots, M$. We then organize our database of faces in a matrix $\boldsymbol{X}{=}[\boldsymbol{x}_1, \ldots, \boldsymbol{x}_M]{\in}\mathbb{R}^{N\times M}$. we obtain the SVD of $\boldsymbol{X}$ as:

$$\boldsymbol{X} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top \tag{5.5.1}$$

Where $\boldsymbol{U}{\in}\mathbb{R}^{N\times N}$ and $\boldsymbol{V}{\in}\mathbb{R}^{M\times M}$ are the left and right singular vector matrices, and $\boldsymbol{\Sigma}{\in}\mathbb{R}^{N\times M}$ is the diagonal singular value matrix, with singular values $\sigma_1{\geq}\sigma_2{\geq}\ldots{\geq}\sigma_N{\geq}0$.

Then, applying the Eckart–Young theorem, the best r-rank approximation of matrix $\mathbf{X}$ can be obtained taking the singular vectors related to the r-largest singular values, $\mathbf{X}_r{=}\mathbf{U}_r\boldsymbol{\Sigma}_r\mathbf{V}_r^\top$, with $\mathbf{U}_r{\in}\mathbb{R}^{\mathrm{N}\times r}$, $\mathbf{V}_r{\in}\mathbb{R}^{\mathrm{M}\times r}$ and $\boldsymbol{\Sigma}_r{\in}\mathbb{R}^{r\times r}$ expressed in their truncated form. Afterwards, any new image taken can be projected onto the subspace generated by the left-singular vectors $\mathbf{U}_r$. The idea behind this operation lies in projecting the image into a subspace generated by the most important latent patterns of face images encoded in the database.

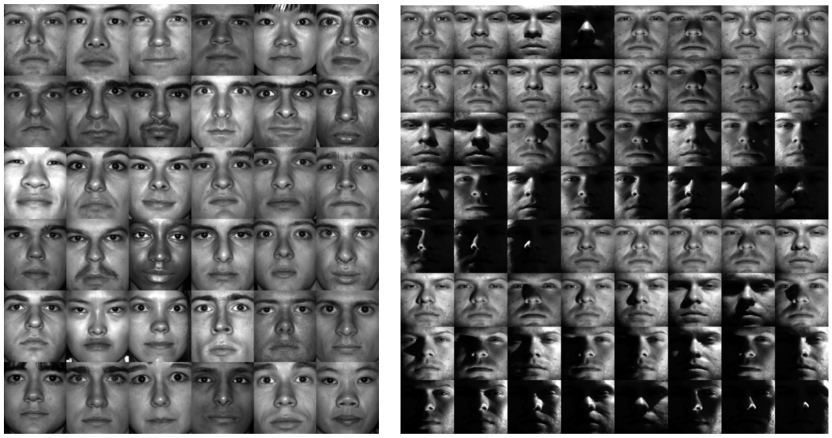


Figure 10: Eigenfaces database (taken from Brunton book "Data driven Science & Engineering). (Left:) several faces from different people; (Right:) faces from the same person.

Suppose now that we have a new face image. The aim is to denoise the image, encoded in the vector $\mathbf{x}_c{\in}\mathbb{R}^{\mathrm{N}}$, by projecting it onto the subspace generated by $\mathbf{U}_r$, and then perform a signal reconstruction. First, we find $\hat{\mathbf{x}}_c{=}\mathbf{x}_c{-}\boldsymbol{\Psi}$, the difference between the daily in-situ calibrated LCS data with the new image and the average of faces in the database. The new estimated vector will be given by: $\tilde{\mathbf{x}}_c = \boldsymbol{\Psi} + \mathbf{U}_r\mathbf{U}_r^\top\hat{\mathbf{x}}_c$

A key parameter is the best r-rank approximation of matrix $\mathbf{X}$, or in other words what is the **optimal hard threshold** r to denoise and reconstruct the images.

We assume that matrix $\mathbf{X}$ is the sum of a true value and some *noise*:

$$\mathbf{X} = \mathbf{X}_{true} + \mathbf{X}_{nosie} \tag{5.5.2}$$

where entries in $\mathbf{X}_{nosie}$ are identically and independently distributed with Gaussian random variables of zero mean and variance γ. If the **magnitude of γ is known**, then:

- If $\boldsymbol{X}{\in}\mathbb{R}^{n\times n}$ (square), then

$$r = \frac{4}{\sqrt{3}}\sqrt{n}\gamma \tag{5.5.3}$$

- If If $\boldsymbol{X}{\in}\mathbb{R}^{m\times n}$ (non-square) and $m{<<}n$, then the fraction $4/\sqrt{3}$ is substituted by a function $\lambda(\beta)$, with $\beta{=}m/n$ (you can find the expression in papers or in the book of S. Brunton "Data driven Science & Engineering").

$$r = \lambda(\beta)\sqrt{n}\gamma \tag{5.5.4}$$

$$r = \lambda(\beta)=(2(\beta+1) + \frac{8\beta}{(\beta+1)+(\beta^2+14\beta+1)^{1/2}})^{1/2} \tag{5.5.5}$$

If $n<<m$, then $\beta=n/m$, and of $\beta=1$ the expression reduces to the previous one.

If the **magnitude of γ is unknown**, and $\boldsymbol{X}\in\mathbb{R}^{m\times n}$ (non-square) then the optimal threshold is given by:

$$\tilde{\sigma} = w(\beta)\sigma_{med} \tag{5.5.6}$$

where σ_{med} is the median of the singular values, while $w(\beta)$ is obtained as:

$$w(\beta) \approx 0.56\beta^3 - 0.95\beta^2 + 1.82\beta + 1.43 \tag{5.5.7}$$

where β=m/n. Finally, r corresponds to the number of singular values that are greater than the threshold $\tilde{\sigma}$.

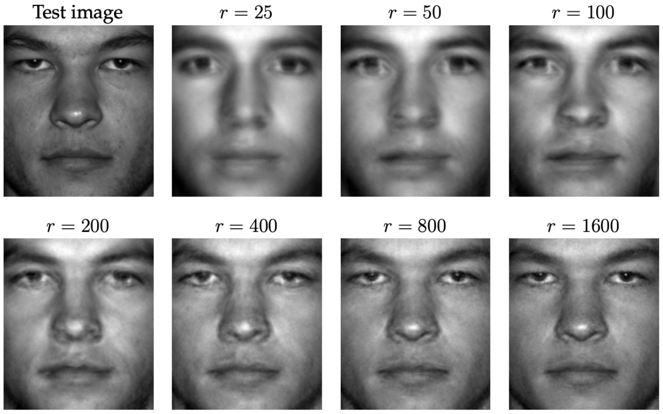


Figure 11: Eigenfaces database (taken from Brunton book "Data driven Science & Engineering"). The approximation improves for $r\geq$400.

6 Fourier transform and its applications

The idea of Fourier series and Fourier transforms is to decompose functions into their basic components. Fourier transforms have many applications such as noise filtering, spectral derivatives, transforming partial differential equations,

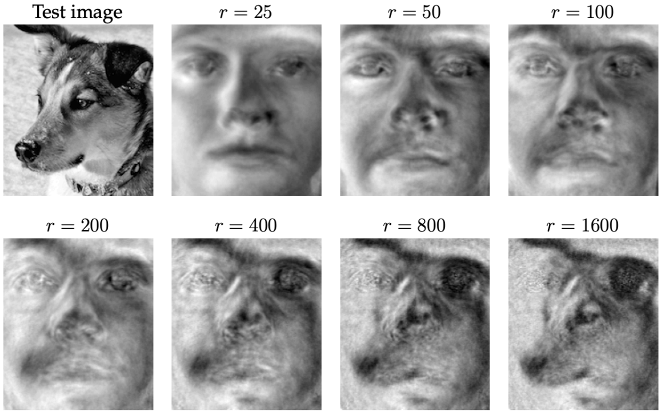


Figure 12: Eigenfaces database (taken from Brunton book "Data driven Science & Engineering"). Since faces have mouth, eyes, cheeks, and a lot of features, and there are more than 1600 faces representing a lot of situations, the approximation works pretty well for a dog.

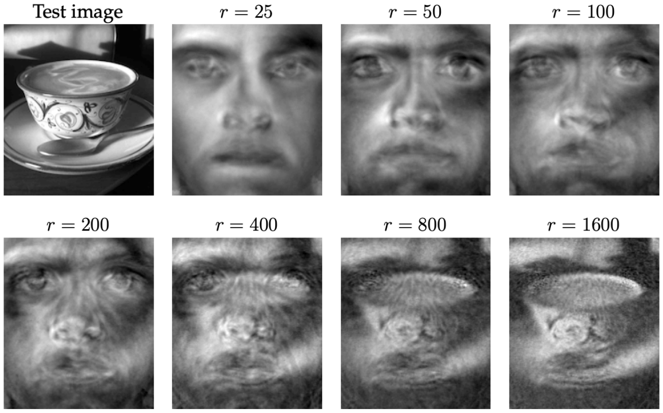


Figure 13: Eigenfaces database (taken from Brunton book "Data driven Science & Engineering"). Approximation for a cappuccino, works well because the 1600 faces also represent non-localized spatial features.

image processing, and a way of express vector data in generic or universal bases, in contrast with vector data expressed in tailored bases, in which we were able to compress data using the SVD (reduction of dimensionality).

Sparsity consists in expressing a signal with a vector in which many components are zero, Although the fast Fourier transform is a technology that allows a signal to be reconstructed from its sparse coefficients, it is not the only way to do so, e.g. Fourier is the basis of JPEG or MPEG. The Fourier modes are generic or universal bases, in the sense that nearly all natural images or audio signals are sparse in these bases. It is also possible to compress signals using the SVD, resulting in a tailored basis.

6.1 Fourier series

As in finite-dimensional vector spaces, the inner product may be used to project a function into an new coordinate system defined by a basis of orthogonal functions. A Fourier series representation of a function f is precisely a projection of this function onto the orthogonal set of sine and cosine functions with integer period on the domain [a, b].

An important result is that if $f(x)$ is periodic and piecewise smooth, then it can be written as a Fourier series. For example, if $f(x)$ is L-periodic in [0,L), then:

$$f(x) = \frac{a_0}{2} + \sum_{k=1}^{\infty}(a_k cos(\frac{2\pi kx}{L}) + b_k sin(\frac{2\pi kx}{L})) \tag{6.1.1}$$

$$a_k = \frac{2}{L}\int_0^L f(x)cos(\frac{2\pi kx}{L})dx \ = \ \frac{< f(x), cos(\frac{2\pi kx}{L}) >}{\|cos(\frac{2\pi kx}{L})\|^2} \tag{6.1.2}$$

$$b_k = \frac{2}{L}\int_0^L f(x)sin(\frac{2\pi kx}{L})dx \ = \ \frac{< f(x), sin(\frac{2\pi kx}{L}) >}{\|sin(\frac{2\pi kx}{L})\|^2} \tag{6.1.3}$$

These coefficients may be viewed as the coordinates obtained by projecting the function onto the orthogonal cosine and sine basis $\{cos(kx),\, sin(kx)\}_{k=0}^{\infty}$.

Since we can write the **Fourier series** in complex form using the facts that $e^{ikx} = cos(\frac{2\pi kx}{L}) + i\ sin(\frac{2\pi kx}{L})$ and $c_k = a_k + i\ b_k$:

$$f(x) = \sum_{k=-\infty}^{\infty} c_k e^{ikx} = \sum_{k=-\infty}^{\infty} c_k \psi_k(x) \tag{6.1.4}$$

Thus, a Fourier series is just a change of coordinates of a function $f(x)$ into an infinite-dimensional orthogonal function space spanned by sines and cosines.

6.2 Fourier transform

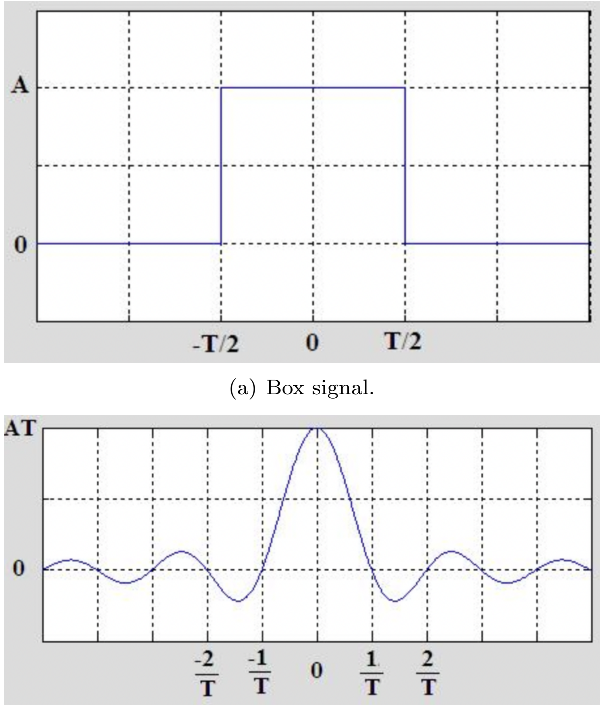
The Fourier series is defined for periodic functions, so that outside the domain of definition, the function repeats itself forever. The **Fourier transform integral** is essentially the limit of a Fourier series as the length of the domain goes to infinity, which allows us to define a function defined on $(-\infty,\, \infty)$.

Thus, we represent the set of frequencies as w_k=$k\pi/L$, and taking the limit $L \to \infty$, such as $k/L \to f$, w=$2\pi f$, Δw=$2\pi/L$, and $\Delta w \to 0$, we will arrive to the classical **Fourier transforms** formulas (not necessary to proof here the pass from Fourier series to Fourier transform, see books for that proof if interested):

$$F(w) = \int_{-\infty}^{\infty} f(x)\; e^{-iwx}\, dx \tag{6.2.1}$$

$$f(x) = \int_{-\infty}^{\infty} F(w)\; e^{iwx}\, dw \tag{6.2.2}$$

Example 6.1 (Fourier Transform example) *For example, assume a box signal, Figure 14, f(x)=A if* x∈$[-T/2, T/2]$ *and 0 otherwise, has Fourier Transform F(f)=ATsinc(fT)= ATsin(*$\pi fT/(\pi fT)$*), where the sinc(f)=sin(*πf*)/(*πf*).*



(a) Box signal.

(b) Sinc signal

Figure 14: The box signal f(x)=A if x∈$[-T/2, T/2]$ and 0 otherwise, has Fourier Transform F(f)=ATsinc(fT)= ATsin(πfT)/(πfT)).

6.2.1 Some basic properties of the FT

Let us assume that we have functions f(x), g(x), and their Fourier Transform F(w), G(w). The following basic properties hold:

- **Linearity:** the Fourier transform of sum of two or more functions that are multiplied by a constant is the sum of the Fourier transforms of the functions

$$(af(x) + bg(x)) \Longrightarrow (aF(w) + bG(w)) \tag{6.2.3}$$

- **Scaling:** if we stretch a function by the factor in the time domain then squeeze the Fourier transform by the same factor in the frequency domain:

$$f(ax) \Longrightarrow (1/|a|)F(w/a) \tag{6.2.4}$$

- **Derivative:** differentiating function with respect to time yields to the constant multiple of the initial function:

$$df(x)/dx \Longrightarrow (jw)F(w) \tag{6.2.5}$$

- **Convolution:** the Fourier transform of a convolution of two functions is the point-wise product of their respective Fourier transforms:

$$f(x)*g(x) \Longrightarrow F(w)G(w) \tag{6.2.6}$$

  Note: the convolution of two functions is a mathematical operation that says how the shape of a function is changed by the other function, a convolution in continuous time is expressed as

$$f(x)*g(x) = \int_{-\infty}^{\infty} f(y)g(x-y)\,dy = \int_{-\infty}^{\infty} f(x-y)g(y)\,dy \tag{6.2.7}$$

- **Time shift:** a linear displacement in time corresponds to a linear phase factor in the frequency domain:

$$f(x-x') \Longrightarrow F(w)e^{-jwx'} \tag{6.2.8}$$

- **Frequency shift:** frequency is shifted according to the co-ordinates:

$$f(x)e^{jw'x} \Longrightarrow F(w-w') \tag{6.2.9}$$

6.2.2 Applications

There are many applications in engineering in which Fourier Transforms are used. Among them, we can list: i) solving of partial differential equations such as the heat equation d^2f(x,t)$/dx^2$= df(x,t)/dt or the wave equation d^2f(x,t)$/dx^2$= d^2f(x,t)$/dt^2$; ii) spectral analysis of time-series (e.g. to find the response of the LTI (linear time invariant) systems); iii) filtering (lowpass, bandpass or highpass); iv) etc, etc, etc. We will see some examples after seeing the discrete version of the FT.

6.3 Discrete Fourier Transform and Fast Fourier Transform

We have seen how to obtain the Fourier transform when the signal f(x) is a continuous function. However, in general, we discretize the analog/continuous signals, e.g., using an ADC that samples the signal to a specific sampling rate or spacing Δx, having a vector of data $[f_0, f_1, \ldots, f_{N-1}]$. Thus, it is necessary, when dealing with vectors of data, to approximate the Fourier transform for dealing with discrete vectors. This is called the **discrete Fourier transform, DFT**. Remember that the domain of the function was called x and then the data is sampled at point x_i and the evaluation of the function (samples) are then called f_i. However, for simplicity, from now, we will express our vectors of sampled data $[f_0, f_1, \ldots, f_{N-1}]$ as $[x_0, x_1, \ldots, x_{N-1}]$, to be coherent with the variables expressed during the course (meaning that vector x is the sample data and not the domain of the function).

Let us define the complex number $w_N = e^{j\frac{2\pi}{N}} = cos(\frac{2\pi}{N}) + j\ sin(\frac{2\pi}{N})$, i.e. is the first N-th root of -1. The **Fourier matrix $\boldsymbol{F_N}$** is defined as:

$$\boldsymbol{F_N} = \frac{1}{\sqrt{N}} \begin{bmatrix} 1 & 1 & 1 & ... & 1 \\ 1 & w_N^{-1} & w_N^{-2} & ... & w_N^{-(N-1)} \\ 1 & w_N^{-2} & w_N^{-4} & ... & w_N^{-2(N-1)} \\ 1 & w_N^{-2} & w_N^{-6} & ... & w_N^{-3(N-1)} \\ ... & ... & ... & ... & ... \\ 1 & w_N^{-(N-1)} & w_N^{-2(N-1)} & ... & w_N^{-(N-1)(N-1)} \end{bmatrix} \tag{6.3.1}$$

Sometimes the factor $\frac{1}{\sqrt{N}}$ is omitted in the definition.

This matrix is *unitary*, meaning that $\boldsymbol{F_N}^H\boldsymbol{F_N}$=$\boldsymbol{F_N}\boldsymbol{F_N}^H$=$\boldsymbol{I}$. The matrix $\boldsymbol{A}^H$ is called *hermitian* if it is a complex square matrix that is equal to its own conjugate transpose, i.e., $a_{ij} = \bar{a}_{ji}$ (remember complex conjugate; if a_{ij}=3+j4, then $\bar{a}_{ji}$=3-j4) or $\boldsymbol{A}$=$\boldsymbol{A}^H$ (special case is when the component is a real number in which $\boldsymbol{A}$=$\boldsymbol{A}^\top$). In other words, hermitian matrices can be understood as the complex extension of real symmetric matrices.

Remember that as we are dealing with complex vectors and matrices, the scalar product between two vectors $\boldsymbol{x}$ and $\boldsymbol{y}$ (e.g. two columns of $\boldsymbol{F_N}$) is defined as $\boldsymbol{x}^H\boldsymbol{y}$. *Do not forget to use the conjugate transpose instead of simply transpose!.*

For instance, if $N = 4$ we have:

$$\boldsymbol{F_4} = \frac{1}{2} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix} \tag{6.3.2}$$

The Fourier matrix is one of the most important matrices in applied mathemat-

ics and engineering.

6.3.1 The Discrete Fourier Transform (DFT)

The *discrete Fourier transform* (DFT), $\hat{\boldsymbol{x}}$, is the operation of multiplying a vector of data $\boldsymbol{x}$ by the matrix $\boldsymbol{F_N}$:

$$\hat{\boldsymbol{x}} = \boldsymbol{F_N}\boldsymbol{x} = \frac{1}{\sqrt{N}} \begin{bmatrix} 1 & 1 & 1 & ... & 1 \\ 1 & w_N^{-1} & w_N^{-2} & ... & w_N^{-(N-1)} \\ 1 & w_N^{-2} & w_N^{-4} & ... & w_N^{-2(N-1)} \\ 1 & w_N^{-2} & w_N^{-6} & ... & w_N^{-3(N-1)} \\ ... & ... & ... & ... & ... \\ 1 & w_N^{-(N-1)} & w_N^{-2(N-1)} & ... & w_N^{-(N-1)(N-1)} \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ \cdots \\ x_{N-1} \end{bmatrix} \tag{6.3.3}$$

This operation is the same as:

$$\hat{x}_k = \sum_{i=0}^{N-1} x_i e^{-j2\pi ik/N} \quad \forall k = 0, \ldots, N-1 \tag{6.3.4}$$

meaning that:

$$\boldsymbol{x}{=}[x_0, x_1, \ldots, x_{N-1}] \overset{DFT}{\Longrightarrow} \hat{\boldsymbol{x}}{=}[\hat{x}_0, \hat{x}_1, \ldots, \hat{x}_{N-1}] \tag{6.3.5}$$

The result of this multiplication $\hat{\boldsymbol{x}}$ is called the *transformed vector*. For commodity, we will use the term *signal* for the vector $\boldsymbol{x}$ and *Fourier transform* for the vector $\hat{\boldsymbol{x}}$. Recall that both vectors have dimension N. We can observe that the DFT is a linear operator (a matrix) that maps data points $\boldsymbol{x}$ in the frequency domain $\hat{\boldsymbol{x}}$. It is to say, we see the DFT operation corresponds to finding the components of vector $\boldsymbol{x}$ expressed in the unitary base created by the columns of $\boldsymbol{F_N}$ (the so called *Fourier base*).

The *Inverse DFT* is defined as:

$$\boldsymbol{x} = \boldsymbol{F_N^H}\hat{\boldsymbol{x}} \tag{6.3.6}$$

and this operation is the same as:

$$x_i = \frac{1}{N}\sum_{k=0}^{N-1} \hat{x}_k e^{j2\pi ki/N} \quad \forall i = 0, \ldots, N-1 \tag{6.3.7}$$

meaning that:

$$\hat{\boldsymbol{x}}{=}[\hat{x}_0, \hat{x}_1, \ldots, \hat{x}_{N-1}] \overset{IDFT}{\Longrightarrow} \boldsymbol{x}{=}[x_0, x_1, \ldots, x_{N-1}] \tag{6.3.8}$$

For a 2D signal X (e.g. a picture represented by a matrix with M rows and N columns) we define the DFT as a two-step process: First find the DFT of the columns, and then find the DFT of the columns of the resulting matrix (the order can be changed):

$$\hat{\boldsymbol{X}} = \boldsymbol{F_M^H} \boldsymbol{X} \boldsymbol{F_N} \tag{6.3.9}$$

6.3.2 The Fast Fourier Transform (FFT)

In general, if $\boldsymbol{A}$ is a $N \times N$ matrix, and $\boldsymbol{x}$ is a vector with N components, the operation $\boldsymbol{Ax}$ requires N^2 multiplications.

In the special case of $\boldsymbol{F_N x}$, the FFT algorithm gives a method for computing this product in only $\frac{1}{2} N log_2 N$ multiplications. *This is a huge improvement.* Some people (e.g. G. Strang) considers that the FFT is the most important numerical algorithm of the XXth century.

6.3.3 The complex conjugate of the columns of the Fourier matrix are the eigenvectors of the circulant matrices

Another extraordinary property of the columns of the Fourier matrix is that the complex conjugate of its columns are the eigenvectors of circulant matrices.

A circulant matrix has the following form:

$$\boldsymbol{C} = \begin{bmatrix} h_0 & h_{N-1} & h_{N-2} & ... & h_1 \\ h_1 & h_0 & h_{N-1} & ... & h_2 \\ ... & ... & ... & ... & ... \\ h_{N-1} & h_{N-2} & h_{N-3} & ... & h_0 \end{bmatrix} \tag{6.3.10}$$

Circulant matrices are especially important as they can be used to represent a very rich and important class of linear system, the linear systems that are invariant to time shifts.

The eigenvector $(1, w^i, w^{2i}, ..., w^{(N-1)i})^\top$ has associated the eigenvalue $\lambda_i = h_0 + h_1 w^{-i} + h_2 w^{-2i} + ... + h_{N-1} w^{(N-1)i}$. Defining $\boldsymbol{\Lambda} = diag(\lambda_0, ..., \lambda_{N-1})$, we obtain the decomposition:

$$\boldsymbol{C} = \boldsymbol{F_N^H} \boldsymbol{\Lambda} \boldsymbol{F_N} \tag{6.3.11}$$

The output signal for an input $\boldsymbol{x}$ can be obtained very efficiently by applying the FFT algorithm.

Example 6.2 (Filtering noise (denoising) in a signal) *A* **first example** *es filtering noise (denoising) in a signal. For example consider figure 15 where we have a signal f(t)= sin(2πf_1t) + sin(2πf_2t) for f_1=*50 *and f_2=*120 *(black curve). We add some Gaussian noise distributed with zero mean and σ^2 variance, i.e., N(0,σ^2), (red curve). If we obtain the FT, we can observe the two*

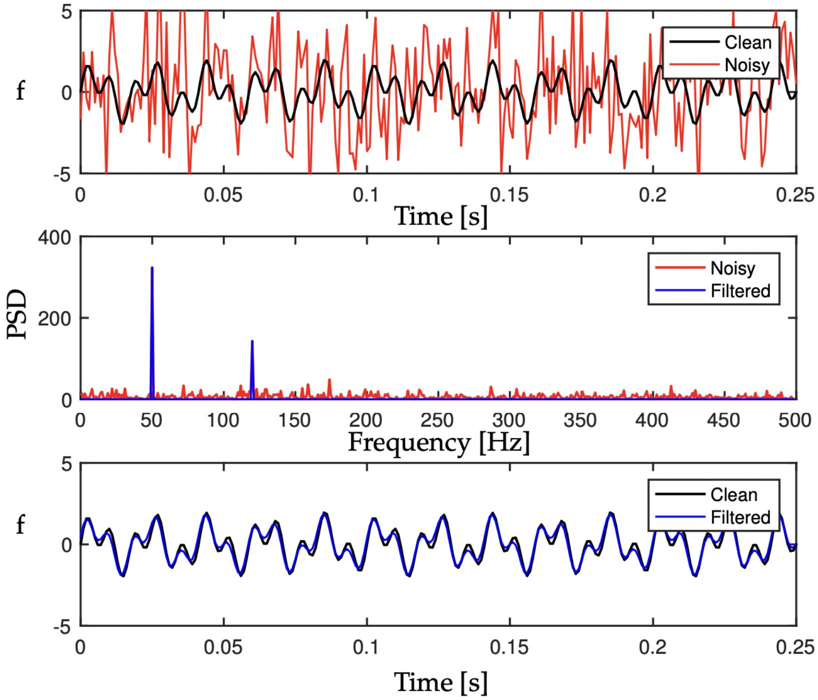


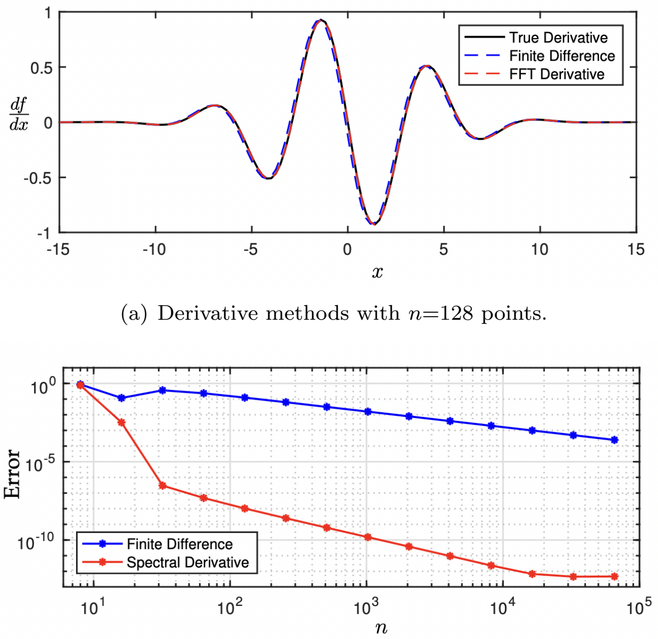
Figure 15: Denoising a signal (Figure taken from S.L. Brunton book, "Data driven science & engineering").

peaks centered at f_1=50 and f_2=120. We can then filter the signal keeping frequencies lower than f_2=120, and then removing high frequency components. We can observe in the third figure the original (without noise) signal and the denoised filtered signal.

*An example of a filter is to pass the signal f(t) by a linear system h(t). The output is the convolution y(t)=f(t)*h(t) $\longrightarrow$ Y(W)=F(W)H(w). If H(w) is a filter allowing to pass frequencies f_1 and f_2 and not allowing the rest of frequencies, we are denoising (filtering noise) our signal.*

Example 6.3 (Spectral derivative) *A* **second example** *is the spectral derivative. We know that the FT of the derivative of a function in continuous time is F(df(x)/dx)=jwF(f(x)). If we discretize, then we can substitute $jw \rightarrow j\kappa$, with $\kappa = 2\pi k/n$, assuming n components. For example, let us assume the function:*

$$f(x) = cos(x)e^{-x^2/25} \Longrightarrow df(x)/dx = -sin(x)e^{-x^2/25} - \frac{2}{25}xf(x) \quad (6.3.12)$$



(a) Derivative methods with n=128 points.

(b) Error as a function of n

Figure 16: Taking the derivative of a signal (Figure taken from S.L. Brunton book, "Data driven science & engineering").

One way of obtaining the derivative is to use finite differences:

$$\frac{df}{dx}(x_k) = \frac{f(x_{k+1}) - f(x_k)}{\Delta x} \tag{6.3.13}$$

for some Δx. So, if we want to obtain the derivative of our signal, we have two options: i) use finite differences, and ii) take our signal, discretize with n samples, obtain the FFT of the signal, multiple by $j\kappa$, and obtain the IFFT. We can observe in Figure 16.a) the result of the true derivative (black), the finite difference method (blue) and the FFT (red). Both, the finite difference method (blue) and the FFT (red) are very close to the true derivative, but we can observe in Figure 16.b) the error in both methods, in which is seen how the FFT performs much better than the finite Euler method.

6.4 Most natural signals are sparse in the Fourier base

Let us assume a vector of data $\boldsymbol{x}\in\mathbb{R}^n$, a vector $\boldsymbol{x}$ is called **sparse** if it has a large number of components equal to zero (or a small number of nonzero components). A vector $\boldsymbol{x}$ is called **dense** if it has a large number of components non equal to zero. A vector $\boldsymbol{x}$ is called **k-sparse** if it has $n-k$ components equal to zero, and k components equal to zero.

A fundamental property in science and engineering is that many natural signals (pictures, video, audio, music, physical magnitudes such as heat, pressure, etc) are sparse when expressed in the Fourier base. This is consequence of a fact of nature that many of these signals are *smooth*.

On the contrary, if we generate a random vector, for instance with i.i.d components sampled from a Gaussian distribution, we would have that with very high probability the components of the vector expressed in Fourier base will be not sparse. We call this sequence *white noise*.

7 Graph signal processing and its applications

7.1 Introduction

The framework of graph signal processing (GSP) was conceived in the last decade with the ambition of generalizing the tools from classical digital signal processing to the case in which the signal is defined over an irregular structure modeled by a graph.

Let us take the example (taken from Ljubisa Stankovic *et al.* survey paper, "Understanding the basis of graph signal processing via an intuitive example-driven approach", arXiv, May 2019) where a set of temperature sensors are deployed over a large region. We are interested, Figure 17.a), of finding the local neighborhood of nodes, in suchc a way that we will find a graph that represents the network, Figure 17.b).

We have been able to connect (we will see later how), for example, node 20 with nodes 19, 22, 23, and node 29 with nodes 27, 28, 51, 59, and so on.

If $x(n)$ is the temperature value of node n, then we can now draw the temperature using bars, Figure 18.a) or even with color in the vertices, Figure 18.b).

The objective is that now we can consider that the signal in a given node n is related to the node itself and its neighborhood:

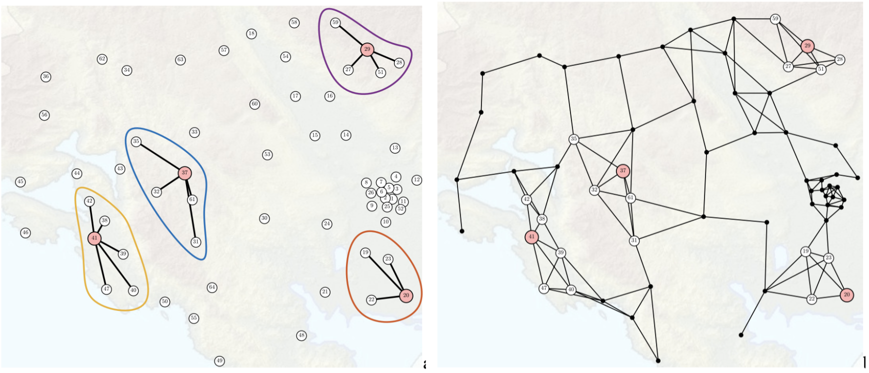
$$y(n) = x(n) + \sum_{m \in N(n)} x(n) \tag{7.1.1}$$

with N(n) the neighborhood of node n. For example, for node n=20:

$$y(20) = x(20) + x(19) + x(22) + x(23) \tag{7.1.2}$$

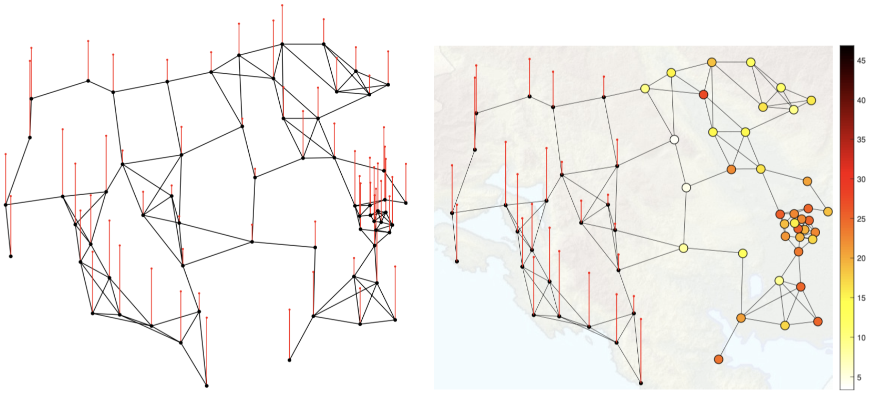
For convenience, we will write this expression as:

$$\boldsymbol{y} = \boldsymbol{x} + \boldsymbol{A}\boldsymbol{x} \tag{7.1.3}$$



(a) Local neighborhood of single nodes. (b) Local neighborhood for all sensors.

Figure 17: Multisensor IoT example, where nodes measure temperature (Figure taken from L. Stankovic et al ”Understanding the Basis of Graph Signal Processing via an Intuitive Example-Driven Approach” paper). Graph representation.



(a) Bars indicate temperature values. (b) Temperature values as bars (left) and vertex color (right)

Figure 18: Multisensor IoT example, where nodes measure temperature (Figure taken from L. Stankovic et al ”Understanding the Basis of Graph Signal Processing via an Intuitive Example-Driven Approach” paper). Colour representation of the temperature field.

where matrix $\boldsymbol{A}$ is the adjacency matrix. There are several ways of creating a graph. The key is in discovering which are the relationships among the nodes.

Example 7.1 (Subgraph of node 29**)** *The subgraph represented by node* 29 *with neighbors 27, 28, 51, 59 in Figure 18.b) would be represented by matrix:*

$$A = \begin{bmatrix} 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 \end{bmatrix} \tag{7.1.4}$$

The objective of GSP is to give a framework able to operate over such a graph making use of these relationships.

7.2 Adjacency matrix, the weighted matrix and the Laplacian matrix

The adjacency matrix is not the only way of representing the graph. Other ways is using a weighted matrix and the Laplacian matrix.

The weighted matrix $\boldsymbol{W}$ has coefficients $w_{ij}>0$ if node i is connected to node j, and zero otherwise. The idea behind the weighted matrix is that the cost of the edges between connected nodes is not equal to one, thus considering that are nodes better connected than others. In this case:

$$y(n) = x(n) + \sum_{m \in N(n)} w_{mn} x(n) \longrightarrow \boldsymbol{y} = \boldsymbol{x} + \boldsymbol{W}\boldsymbol{x} \tag{7.2.1}$$

We can observe that the operator $\boldsymbol{A}$ is a special case of the operator $\boldsymbol{W}$, in which all weights are considered of the same value.

Finally, we can use a third operator called the Laplacian matrix $\boldsymbol{L}$, which is obtained as:

$$\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{W} \tag{7.2.2}$$

where $\boldsymbol{D}$ is the degree matrix and it has coefficients in the diagonal $d_{ii} = \sum_{j \neq i} w_{ij}$ (sum of row except the value at the diagonal) and the rest are 0.

Example 7.2 (Laplacian matrix) *We want to obtain the laplacian matrix $\boldsymbol{L}$, from the weight matrix $\boldsymbol{W}$:*

$$\boldsymbol{W} = \begin{bmatrix} 0 & .6 & .3 & 0 \\ .6 & 0 & .1 & .4 \\ .3 & .1 & 0 & .2 \\ 0 & .4 & .2 & 0 \end{bmatrix} \tag{7.2.3}$$

We first obtain the diagonal matrix $\boldsymbol{D}$ as sum of rows $d_{ii} = \sum_{j \neq i} w_{ij}$:

$$\boldsymbol{D} = \begin{bmatrix} .9 & 0 & 0 & 0 \\ 0 & 1.1 & 0 & 0 \\ 0 & 0 & .6 & 0 \\ 0 & 0 & 0 & .6 \end{bmatrix} \tag{7.2.4}$$

Then, the Laplacian will be:

$$\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{W} = \begin{bmatrix} .6 & -.6 & -.3 & 0 \\ -.6 & 1.1 & -.1 & -.4 \\ -.3 & -.1 & .6 & -.2 \\ 0 & -.4 & -.2 & 0.6 \end{bmatrix} \tag{7.2.5}$$

7.3 Creating the graph

There are several ways of creating a graph. The key is in discovering which are the relationships among the nodes:

- **Physically knowledge of the weights:** there is an intrinsic knowledge of what are the weights, for example, circuits in electronic systems, social networks, etc;
- **Geometry of the vertex:** use Euclidean distances. In this case it is built a decreasing function of the distance:

  $$w_{ij} = e^{d_{ij}^2/\alpha} \quad or \quad w_{ij} = e^{d_{ij}/\alpha} \tag{7.3.1}$$

- **Obtain the weighted matrix or the Laplacian matrix from the data measured matrix $\boldsymbol{X}$.** An example is to use a non-linear optimization model:

  $$\begin{aligned} \underset{\mathbf{L},\mathbf{Y}}{\text{minimize}} \quad & \underbrace{\|\mathbf{X}-\mathbf{Y}\|_F^2}_{data\ \ fidelity} + \underbrace{\underbrace{\alpha\ tr(\mathbf{Y}^\top \mathbf{L}\mathbf{Y})}_{smoothness} + \beta\|\mathbf{L}\|_F^2}_{sparsity} \\ \text{subject to} \quad & tr(\mathbf{L}) = n, \\ & L_{ij} = L_{ji} \leq 0, \qquad i \neq j, \\ & \mathbf{L}\cdot\mathbf{1} = \mathbf{0}. \end{aligned} \tag{7.3.2}$$

  We will study convex non-linear optimization in TOML-MIRI course, but we can say that this optimization problem is not convex (it has not global minimum). But it can be solved using in an alternating procedure, in which we first fix $\boldsymbol{Y} = \boldsymbol{X}$, so we find $\boldsymbol{L}$ from our data measurements $\boldsymbol{X}$. We can play the sparsity (more L_{ij}=0, meaning more nodes disconnected) of our solution using the β parameter. Finally when $\boldsymbol{L}$ is found, we can solved again fixing the $\boldsymbol{L}$ found and finding $\boldsymbol{Y}$ that will be a filtered version of our data $\boldsymbol{X}$.

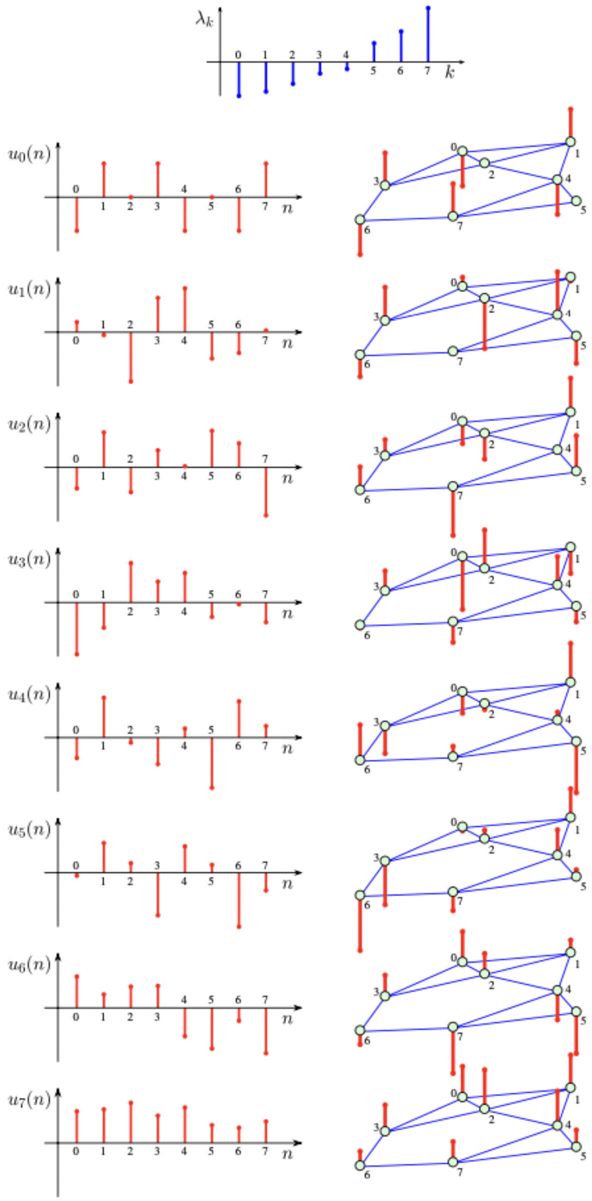


Figure 19: Eigenvalues and eigenvectors of an N=8 node network (Figure taken from L. Stankovic et al "Understanding the Basis of Graph Signal Processing via an Intuitive Example-Driven Approach" paper).

7.4 Spectral graph theory

Since adjacency matrix $\boldsymbol{A}$ is squared and symmetric ($\boldsymbol{A}$=$\boldsymbol{A}^\top$), is i) diagonizable, ii) its eigenvalues are real ($\boldsymbol{\Lambda}$=$diag(\lambda_0, \dots, \lambda_{N-1})$ is a diagonal matrix), and iii) its eigenvectors are orthogonal ($\|u_k\|_2^2 = 1$):

$$\boldsymbol{A} = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^{-1} = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^\top \tag{7.4.1}$$

7.5 The adjacency matrix and the graph signal shift

Let us consider N samples of a signal expressed as a vector $\boldsymbol{x}$=$[x_0, x_1, \dots, x_{N-1}]$. A signal shift on a graph can be defined as the movement of the signal sample, x_n, from its original vertex, n, along all walks of length one that start at vertex n. If we define $\boldsymbol{x}^{(1)}$ as the signal shifted, then:

$$\boldsymbol{x}^{(1)} = \boldsymbol{A}\boldsymbol{x} \tag{7.5.1}$$

In case we go on shifting the signal:

$$\boldsymbol{x}^{(2)} = \boldsymbol{A}\boldsymbol{x}^{(1)} = \boldsymbol{A}^2\boldsymbol{x} \tag{7.5.2}$$

and thus, m shiftings result in:

$$\boldsymbol{x}^{(m)} = \boldsymbol{A}\boldsymbol{x}^{(m-1)} = \boldsymbol{A}^2\boldsymbol{x}^{(m-2)} = \cdots = \boldsymbol{A}^{m-1}\boldsymbol{x}^{(1)} = \boldsymbol{A}^m\boldsymbol{x} \tag{7.5.3}$$

You can recall that instead of the adjacency matrix $\boldsymbol{A}$ we can use the Laplacian matrix $\boldsymbol{L}$, and consider the adjacency matrix a special case of the Laplacian matrix, and thus:

$$\boldsymbol{x}^{(m)} = \boldsymbol{L}^m\boldsymbol{x} \tag{7.5.4}$$

7.6 The graph discrete Fourier transform

The **graph discrete Fourier transform** (GDFT) of a signal, $\boldsymbol{x}$, is defined as:

$$\boldsymbol{X} = \boldsymbol{U}^{-1}\boldsymbol{x} \tag{7.6.1}$$

where $\boldsymbol{X}$ denotes a vector of the GDFT coefficients, and $\boldsymbol{U}$ is a matrix whose columns represent the eigenvectors of the adjacency matrix, $\boldsymbol{A}$ or the Laplacian matrix $\boldsymbol{L}$. Let us assume that we take the adjacency matrix (same considerations but different behavior if we use the Laplacian matrix), then vector $\boldsymbol{X}$ has k=0,1, . . . , N-1 coefficients. Since matrix $\boldsymbol{A}$ is symmetric ($\boldsymbol{A}^\top$ =$\boldsymbol{A}$), we have that $\boldsymbol{U}^{-1}$ =$\boldsymbol{U}^\top$, and:

$$\boldsymbol{X} = \boldsymbol{U}^\top\boldsymbol{x} \tag{7.6.2}$$

The **inverse graph discrete Fourier transform** (IGDFT) of a signal, $\boldsymbol{x}$, is thus defined as:

$$\boldsymbol{x} = \boldsymbol{U}\boldsymbol{X} \tag{7.6.3}$$

In case of having a circular graph, the GDFT reduces to the classical DFT. Given that we have defined a GDFT/IGDFT, we can define filters, convolutions, spectral analysis, signal reconstruction, denoising, and different operations performed in classical signal processing.

7.7 Signal reconstruction in a IoT network

We consider the problem of having a subset of nodes (vertices of the graph) with samples and we would like to estimate the signal of the graph in the other vertices so that the resulting signal is smooth. Let is call $\mathcal{M}$ the set of nodes in the graph with observed data, and $\mathcal{U}$ the set of nodes in the graph with observed data (missing data). Thus the objective is to find the unobserved data from the observed neighboring nodes. This can be seen as a signal reconstruction problem that can be solved using methods from various fields. Let us consider the following methods; Laplacian interpolation and GSP low-pass based graph signal reconstruction. Laplacian interpolation is a graph-based semi-supervised learning algorithm whose goal is regression with graph regularization assuming smoothness with respect to the Laplacian matrix. This method regresses a function $f{:}\mathcal{V}{\to}\mathbb{R}$ over the graph G, assuming partial information, it is to say, information for M nodes. GSP low-pass based graph signal reconstruction is a graph signal processing reconstruction method that considers subsampling low-pass graph signals, thus assuming a sparse Fourier coefficient vector.

7.7.1 Laplacian interpolation

Also known as graph interpolated regularization by Belkin *et al.*, this method minimizes the quadratic form of the Laplacian matrix with respect to the graph signal $\mathbf{x}$, which is a measure of signal smoothness, given that the observed measurements $\{x_m{:}\,\forall m{\in}\mathcal{M}\}$ remain unchanged. This reconstruction results in a linear combination of the observations weighted by the Laplacian matrix entries L_{ij}.

$$\begin{aligned} &\underset{\mathbf{y}}{Minimize} && \mathbf{y}^\top \mathbf{L}\mathbf{y} \\ &\text{s.t.} && y_m = x_m, \qquad \forall m{\in}\mathcal{M} \end{aligned} \tag{7.7.1}$$

7.7.2 Graph Signal Processing (GSP) low-pass reconstruction

This technique recovers a set of unobserved nodes $\{x_u{:}\,\forall u{\in}\mathcal{U}\}$ given that the graph discrete Fourier transform of the complete signal is sparse and of low-pass nature, meaning that it has K nonzero components corresponding to the lowest frequencies (smallest eigenvalues λ_i of the Laplacian matrix). Given that the Laplacian matrix admits the eigendecomposition $\mathbf{L}$=$\mathbf{U}\mathbf{\Lambda}\mathbf{U}^\top$, the graph discrete Fourier transform (GDFT) of a graph signal $\mathbf{x}$ can be computed as:

$$\mathbf{X} = \mathbf{U}^{-1}\mathbf{x} \tag{7.7.2}$$

Now, a K-sparse GDFT coefficient vector of the following form is to be recovered:

$$\mathbf{X} = (X(0), \ldots, X(K-1), 0, \ldots, 0)^\top \tag{7.7.3}$$

For this purpose a subset of measurements $\mathcal{M}$ are used to recover the sparse coefficient vector by solving the following system:

$$\mathbf{x}_{\mathcal{M}} = \mathbf{U}_{\mathcal{M}K}\mathbf{X}_K \tag{7.7.4}$$

Since the system is overdetermined, the solution of the above system in the least squares sense is given by $\mathbf{X}_K$= $\mathbf{U}^\dagger_{\mathcal{M}K}\mathbf{x}_{\mathcal{M}}$, where $\mathbf{U}^\dagger_{\mathcal{M}K}$= $(\mathbf{U}^\top_{\mathcal{M}K}\mathbf{U}_{\mathcal{M}K})^{-1}\mathbf{U}^\top_{\mathcal{M}K}$ is the matrix pseudo-inverse of $\mathbf{U}_{\mathcal{M}K}$; the nonzero coefficients are obtained, and after appending the corresponding zero coefficients, the inverse graph discrete Fourier transform (IGDFT) $\mathbf{x}$=$\mathbf{U}\mathbf{X}$ is computed to obtain the complete set of measurements $\mathbf{x}$ at all vertices.

Example 7.3 (GSP low pass reconstruction) *Let us assume a network with $N = 8$ nodes, in which we only have $M = 4$ measurements, e.g. x(2), x(4), x(5) and x(7), and we want to reconstruct the signal with $K = 2$ coefficients.*

Then, we have to find X(0) and X(1) that satisfies:

$$\begin{bmatrix} x(4) \\ x(5) \\ x(7) \end{bmatrix} = \begin{bmatrix} u_0(4) & u_1(4) \\ u_0(5) & u_1(5) \\ u_0(7) & u_1(7) \end{bmatrix} \begin{bmatrix} X(0) \\ X(1) \end{bmatrix} \tag{7.7.5}$$

For finding $X(O)$ and $X(1)$, we solve the overdetermined system, and when we have these GDFT coefficients, we can obtain the original ones applying the IGDFT $\mathbf{x}$=$\mathbf{U}\mathbf{X}$, *with* $\mathbf{X}$=$[X(0), X(1), 0, 0, 0, 0, 0, 0]$.