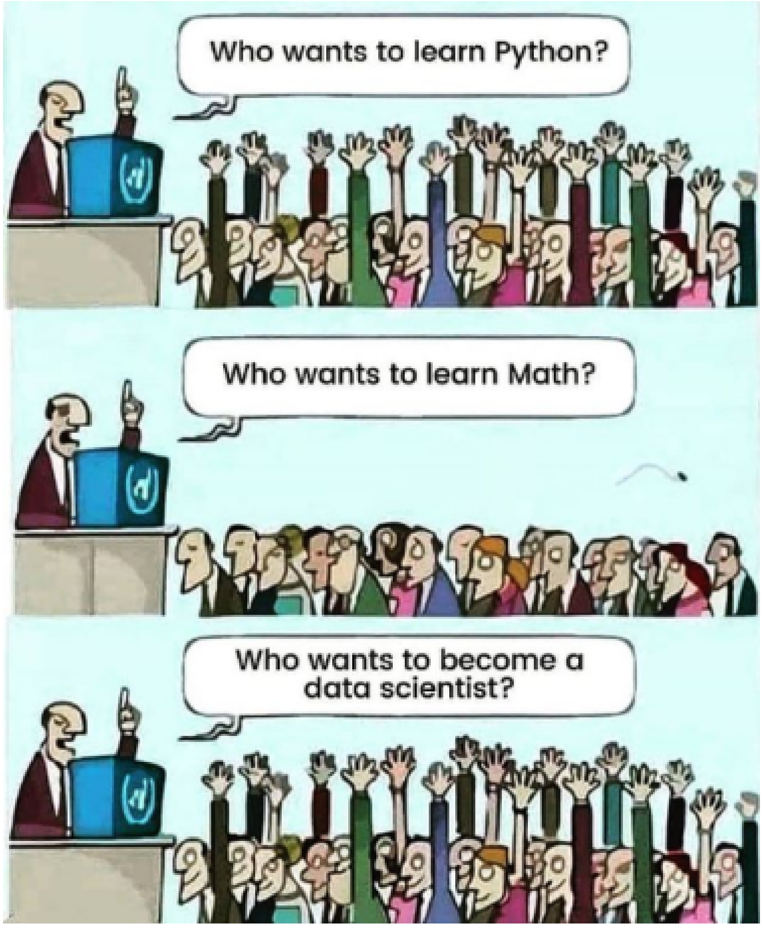
Mathematical and statistical foundations of Data Science

Peter Arndt

HHU Düsseldorf
Winter 2021/22

Contents

.



Who wants to learn Python?
Who wants to learn Math?
Who wants to become a data scientist?

1 Linear Algebra

Linear algebra is one of the basic foundations of (pure and) practical mathematics. We will see concrete applications that are relevant in machine learning, but maybe the more important role of linear algebra is as a basic language used in many fields, e.g. to formulate the notion of multivariate Gaussian distribution in probability theory, for linear models in statistics, for explaining the notion of derivative and many other things.

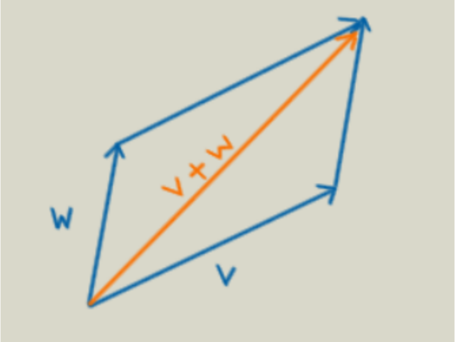
A great resource, especially for linear algebra, but also for other parts of the lecture course, is the free book

Deisenroth, Faisal, Ong: Mathematics for Machine Learning.

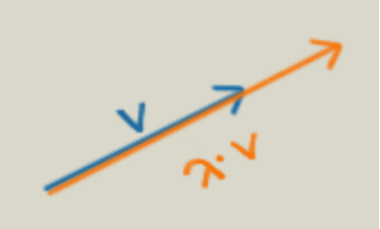
(the above is a clickable link, as are most things written in blue)

1.1 Vector spaces

Linear algebra studies vector spaces and linear maps between them. Intuitively, a vector space is an environment for geometry. If you fix a point (usually called the "origin") and picture points in space by arrows from the origin - "vectors" - then you can add two vectors and stretch a vector by some factor:



Addition of two vectors:
$v + w \;=\; w + v$.



Stretching a vector
by a factor λ.

The notion of vector space is an axiomatic description of these two operations.

To define the notion of vector space, one first has to choose a "field", i.e. a domain where you can calculate with operations $+$ and $\cdot$ as we know it from usual real numbers. Intuitively, a vector space is an environment for geometry and the elements of the field are the possible coordinates.

We only use the field $\mathbb{R}$ (except for one small appearance of the complex numbers $\mathbb{C}$). Most of the time it does not matter which field we choose, and we just call it K.

Definition 1.1.1. *A K-*vector space *is a set V (the set of vectors) together with an element $0 \in V$ (the "origin", or the "zero vector") and operations*

$$+ : V \times V \to V, \quad (v, w) \mapsto v + w \ \textit{(addition)},$$

and

$$\cdot : K \times V \to V, \quad (\lambda, v) \mapsto \lambda \cdot v \ \textit{(scalar multiplication)},$$

satisfying the following statements for all $\lambda, \mu \in K$ and $u, v, w \in V$:

1. *(laws of addition)*
   a) $(u + v) + w = u + (v + w)$
   b) $u + v = v + u$
   c) $u + 0 = u$
   d) $\exists u' \colon u + u' = 0$
2. *(distributive laws):*
   a) $(\lambda + \mu) \cdot v = \lambda \cdot v + \mu \cdot v$
   b) $\lambda \cdot (v + w) = \lambda \cdot v + \lambda \cdot w$
3. $\lambda \cdot (\mu \cdot v) = (\lambda \cdot \mu) \cdot v$
4. $1 \cdot v = v$

You can verify these axioms directly for the graphical operations of vector addition and stretching.

Example 1.1.2. Our main example (and in this first part almost the only one) will be the $\mathbb{R}$-vector space $\mathbb{R}^n$, the column vectors with n entries, with the usual operations.

$$\begin{pmatrix} a_1 \\ \vdots \\ a_n \end{pmatrix} + \begin{pmatrix} b_1 \\ \vdots \\ b_n \end{pmatrix} := \begin{pmatrix} a_1 + b_1 \\ \vdots \\ a_n + b_n \end{pmatrix} \qquad \lambda \cdot \begin{pmatrix} a_1 \\ \vdots \\ a_n \end{pmatrix} := \begin{pmatrix} \lambda \cdot a_1 \\ \vdots \\ \lambda \cdot a_n \end{pmatrix}$$

Of course $\mathbb{R}^3$ is our standard model for 3-dimensional space. Looking at a particle with its position and impulse places you in $\mathbb{R}^6$, looking at two particles in $\mathbb{R}^{12}$ and so on. Measuring medical data, a snapshot of the state of a patient at a given time may include height, weight, body temperature, concentration of many chemicals in their blood etc. and can quickly place you in $\mathbb{R}^{50}$. In natural language processing one uses so-called word vector embeddings and represents and e.g. represents an English word as a vector in $\mathbb{R}^{300}$, codifying its occurence probabilities in English sentences. Time series - i.e. repeated measurements of always the same variable at 10.000 consecutive moments can be represented as vectors in $\mathbb{R}^{10.000}$.

Example 1.1.3. The set $\mathbb{R}^{n \times m}$ of matrices with n rows and m columns is an $\mathbb{R}$-vector space. Addition works by adding corresponding places:

$$\begin{pmatrix} a_{11} & a_{12} & \dots & a_{1m} \\ a_{21} & a_{22} & & a_{2m} \\ \vdots & & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nm} \end{pmatrix} + \begin{pmatrix} b_{11} & b_{12} & \dots & b_{1m} \\ b_{21} & b_{22} & & b_{2m} \\ \vdots & & \ddots & \vdots \\ b_{n1} & b_{n2} & \dots & b_{nm} \end{pmatrix}$$

$$= \begin{pmatrix} a_{11} + b_{11} & a_{12} + b_{12} & \dots & a_{1m} + b_{1m} \\ a_{21} + b_{21} & a_{22} + b_{22} & & a_{2m} + b_{2m} \\ \vdots & & \ddots & \vdots \\ a_{n1} + b_{n1} & a_{n2} + b_{n1} & \dots & a_{nm} + b_{nm} \end{pmatrix}$$

Similarly with scalar multiplication:

$$\lambda \cdot \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1m} \\ a_{21} & a_{22} & & a_{2m} \\ \vdots & & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nm} \end{pmatrix} = \begin{pmatrix} \lambda \cdot a_{11} & \lambda \cdot a_{12} & \dots & \lambda \cdot a_{1m} \\ \lambda \cdot a_{21} & \lambda \cdot a_{22} & & \lambda \cdot a_{2m} \\ \vdots & & \ddots & \vdots \\ \lambda \cdot a_{n1} & \lambda \cdot a_{n2} & \dots & \lambda \cdot a_{nm} \end{pmatrix}$$

For example a grey scale image of 100×100 pixels can be represented as a 100×100-matrix in $\mathbb{R}^{100\times 100}$, a color image is usually given by a red, a green and a blue layer and can be given as a vector in $\mathbb{R}^{3\times 100\times 100}$.

Example 1.1.4. For a set X, the set of all real valued functions $\{f\colon X \to \mathbb{R}\}$ is a vector space with "pointwise" addition and scalar multiplication. This means that given two functions f, g, we can define a new function $f + g$ by declaring that the value of the new function $f + g$ at the point $x \in X$ is f the sum of the values $f(x)$ and $g(x)$. More formally:

$$\begin{array}{rcl} \mathrm{Map}(X,K) \times \mathrm{Map}(X,K) & \longrightarrow & \mathrm{Map}(X,K) \\ (f,g) & \mapsto & \left(\begin{array}{rcl} f+g\colon X & \to & K \\ x & \mapsto & f(x)+g(x) \end{array} \right) \end{array}$$

Similarly scalar multiplication is defined pointwise:

$$\begin{array}{rcl} K \times \mathrm{Map}(X,K) & \longrightarrow & \mathrm{Map}(X,K) \\ (\lambda,f) & \mapsto & \left(\begin{array}{rcl} \lambda\cdot f\colon X & \to & K \\ x & \mapsto & \lambda\cdot f(x) \end{array} \right) \end{array}$$

One can also write K^X for $\mathrm{Map}(X,K)$.

Example: If $X = [0,1]$, $f(x) = 3\sin(x) + \cos(x)$ and $g(x) = 2\sin(x) + e^x$, then $(f+g)(x) = 5\sin(x) + \cos(x) + e^x$.

Often one considers subsets of the set of all functions which are closed under addition and scalar multiplication, e.g. the sets of continuous or polynomial functions, or the set of all $\mathbb{R}$-valued random variables on a probability space – see later.

We can iterate additions and scalar multiplications: Given a vector space V and $\lambda_1, \ldots, \lambda_n \in K$ and $v_1, \ldots, v_n \in V$ we can form the new vector $\lambda_1 \cdot v_1 + \ldots + \lambda_n \cdot v_n \in V$. Such a vector is called a *linear combination* of the vectors $v_1, \ldots, v_n$.

Definition 1.1.5. *Let V be a K-vector space. A subset $U \subseteq V$ is called a* sub-vector space *(or just* subspace*) if it is non-empty and closed under addition and scalar multiplication, i.e. if for all $\lambda \in K$, $s, t, \in U$ we have $\lambda v \in U$ and $s + t \in U$.*

Equivalently we can demand that U be closed under all linear combinations, or, still equivalently, that U be closed under linear combinations of two vectors.

Examples 1.1.6. (i) The following is a subvector space:

$$\left\{ \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix} \in \mathbb{R}^3 \;\middle|\; a_3 = 0 \right\} \subseteq \mathbb{R}^3$$

Indeed, adding two vectors whose third component is zero results in another vector whose third component ist zero, and so does multiplying a scalar with a vector whose third component is zero.

(ii) The vector space $\{f\colon \mathbb{R} \to \mathbb{R}\}$ of all functions from the reals to themselves has the subvector space of all polynomial functions: Adding and scalar multiplying two polynomial functions results in a polynomial function again. For example, if $p(x) = 2x^2 + 4x + 1$ and $q(x) = 4x^7 + 3x^2 + 16$ then $(p+q)(x) = 4x^7 + 5x^2 + 4x + 17$.

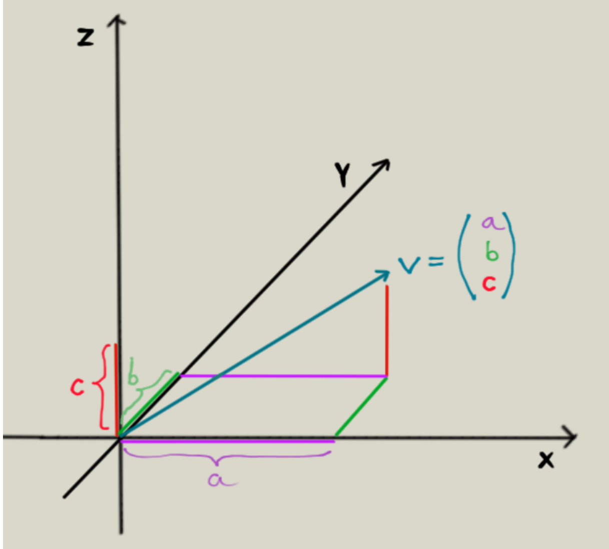
(iii) Further subvector spaces of $\{f\colon \mathbb{R} \to \mathbb{R}\}$ are the sets of all continuous functions and all smooth (i.e. infinitely often differentiable) functions:

(iv) For a probability space $(\Omega, \mathcal{B}, P)$, the set of all $\mathbb{R}$-valued random variables is a subvector space of $\mathbb{R}^\Omega$ – this is precisely the subset of all measurable functions. If you don't remember these notions, don't think about them right now: we will cover them later in the course.

Subvector spaces of finite dimensional vector spaces can be pictured as planes through the origin. For an algebraic intuition: Subvector spaces are exactly the solution sets of homogeneous systems of linear equations.

Fact 1.1.7. For every subset X of a vector space V, there exists the smallest subvector space containing that set. One calls it the subvector space *spanned by* X*, the span of* X, in symbols $\operatorname{span}(X)$. One has $\operatorname{span}(X) = \{\sum_{i=1}^{n} \lambda_i x_i \mid n \in \mathbb{N}, \lambda_i \in K, x_i \in X\}$.

What makes vector spaces easy to work with is that one can choose a basis. Geometrically, a basis is a set of vectors, such that each point can be reached in a unique way following stretched versions of those vectors. The stretching factors are the coordinates of the point:



Coordinates of a point

The idea of a basis of giving unique coordinates for every point is formalized as follows:

Definition 1.1.8. *Let V be a K-vector space.*

(i) *A subset $S \subseteq V$ is called a* generating system, *if for every $v \in V$ there exist $\lambda_1, \ldots, \lambda_n \in K$ and $e_1, \ldots, e_n \in S$ such that $v = \lambda_1 e_1 + \ldots + \lambda_n e_n$.*

(ii) *A subset $S \subseteq V$ is called* linearly independent *if, whenever $v = \lambda_1 e_1 + \ldots + \lambda_n e_n$, for some $e_1, \ldots, e_n \in S$, then these $\lambda_1, \ldots, \lambda_n \in K$ and $e_1, \ldots, e_n \in V$ are unique with that property. Equivalently: If $\lambda_1 e_1 + \ldots + \lambda_n e_n = 0$ then $\lambda_1 = \ldots = \lambda_n = 0$.*

(iii) *A subset $\{e_1, \ldots, e_n\} \subseteq V$ is called* a basis *if it is both linearly independent and a generating system.*

Remark 1.1.9. For better readability, here are the definitions of generating systems and linearly independent systems, in the case of *finite* sets $\{e_1, \ldots, e_n\} \subseteq V$. Let V be a K-vector space.

(i) A subset $\{e_1, \ldots, e_n,\} \subseteq V$ is called a *generating system*, if for every $v \in V$ there exist $\lambda_1, \ldots, \lambda_n \in K$ such that $v = \lambda_1 e_1 + \ldots + \lambda_n e_n$.

(ii) A subset $\{e_1, \ldots, e_n\} \subseteq V$ is called *linearly independent* if, whenever $v = \lambda_1 e_1 + \ldots + \lambda_n e_n$, then these $\lambda_1, \ldots, \lambda_n \in K$ are unique with that property. Equivalently: If $\lambda_1 e_1 + \ldots + \lambda_n e_n = 0$ then $\lambda_1 = \ldots = \lambda_n = 0$.

(iii) A subset $\{e_1, \ldots, e_n\} \subseteq V$ is called *a basis* if it is both linearly independent and a generating system.

Facts 1.1.10. (i) Every vector space has a basis.

(ii) Every linearly independent set can be completed to form a basis. That means: Given a linearly independent set, one can add further vectors such that the set becomes a generating system, while still staying linearly independent.

(iii) Every generating set can be reduced to form a basis. That means: Given a generating set, one can take away vectors such that the set becomes linearly independent, while still staying a generating system.

(iv) A basis is the same thing as a maximal linearly independent set (i.e. putting in any further vector it stops being linearly independent).

(v) A basis is the same thing as a minimal generating set (i.e. if one takes away any vector it stops being a generating set).

(vi) Any two bases of the same vector space have the same number of elements

Definition 1.1.11. *The number of elements of a basis of a vector space is called the* dimension *of that vector space. Notation:* $\dim V$.

Example 1.1.12. In $\mathbb{R}^n$ one has the vectors

$$e_1 := \begin{pmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}, e_2 := \begin{pmatrix} 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \ldots, e_n := \begin{pmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 1 \end{pmatrix}$$

The set of vectors $\{e_1, \ldots, e_n\}$ forms a basis: the vector with coefficients $\lambda_1, \lambda_2, \ldots, \lambda_n$ is $\Sigma_{i=1}^n \lambda_i e_i$, which shows that this is a generating set. Linear independence can be seen with the help of a system of linear equations. This basis is called the *standard basis* of $\mathbb{R}^n$ and the vector e_i is called the i-th standard basis vector.

Observation 1.1.13. The dimension of a subvector space ist always $\leq$ the dimension of the ambient vector space. In particular, in $\mathbb{R}^n$ there cannot exist a subset of more than n linearly independent vectors: otherwise we could complete these vectors to a basis which would then consist of more than n vectors. And in particular a subset of n linearly independent vectors in $\mathbb{R}^n$ is a basis, because if it was not, then we could complete it to a basis, which would then have more than n elements – in contradiction to the fact that any two bases have the same number of elements.

Example 1.1.14. Is the set $\left\{\begin{pmatrix}1\\1\end{pmatrix}, \begin{pmatrix}2\\1\end{pmatrix}\right\}$ linearly independent? For testing that, we assume that some linear combination of these two vectors is zero:

$$\lambda \cdot \begin{pmatrix}1\\1\end{pmatrix} + \mu \cdot \begin{pmatrix}2\\1\end{pmatrix} = \begin{pmatrix}\lambda + 2\mu\\ \lambda + \mu\end{pmatrix} = \begin{pmatrix}0\\0\end{pmatrix}$$

Since two column vectors are equal iff their coefficients conincide in all places, this gives us two equations:

$$\begin{array}{rcl}\lambda + 2\mu & = & 0\\ \lambda + \mu & = & 0\end{array}$$

If we subtract the lower equation from the upper equation, we get

$$\mu = (\lambda + 2\mu) - (\lambda + \mu) = 0 - 0 = 0$$

So we know $\mu = 0$ ein, and with the first equation we get $\lambda = 0$. This was one criterion for linear independence.

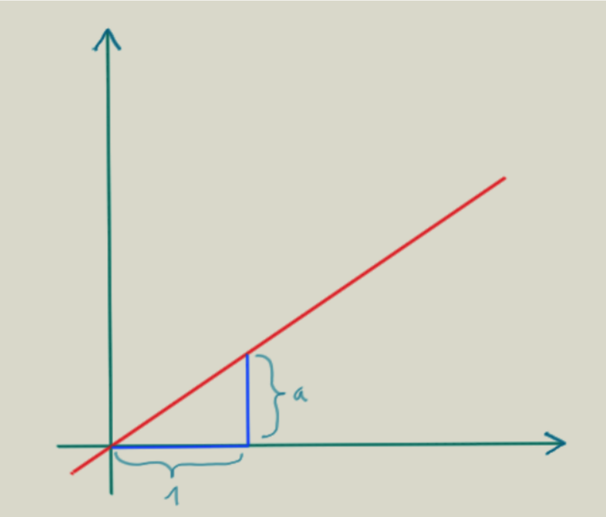
In the 2-dimensional vector space $\mathbb{R}^2$ there cannot be more than two linearly independent vectors. Thus this must be a basis of $\mathbb{R}^2$, because otherwise we could complete it to a basis which would then consist of more than two vectors.

1.2 Linear maps and matrices

Definition 1.2.1. *A map $f\colon V \to W$ between K-vector spaces is* linear *if for all $\lambda, \mu \in K$, $v, w \in V$ we have $f(v + w) = f(v) + f(w)$ and $f(\lambda \cdot v) = \lambda f(v)$. Equivalently: $f(\lambda v + \mu w) = \lambda f(v) + \mu f(w)$.*

Linear maps more generally preserve linear combinations: $f(\sum_{i=1}^n \lambda_i v_i) = \sum_{i=1}^n \lambda_i f(v_i)$.

Example 1.2.2. Linear functions $\mathbb{R} \to \mathbb{R}$ are exactly the ones whose graph looks like a straight line through the origin. In formulas, they are the ones of the form $f(x) = ax$ for some $a \in \mathbb{R}$ – the number a is the slope of the line.



Linear function $f(x) = ax$

Likewise, linear functions $\mathbb{R}^2 \to \mathbb{R}$ are those whose graph looks like a plane through the origin.

⚠ Linear functions $\mathbb{R}^m \to \mathbb{R}^n$ are precisely those of the form $f(x) = Ax$ for an $n \times m$-matrix A – more about that below. In the machine learning literature the term "linear function" is often used for functions of the form $f(x) = ax + c$, or $f(x) = Ax + b$, with an additional constant c, resp. an aditional vector b and a matrix A. Such maps are *not* linear in the sense of Def. 1.2.1! In mathematics, and also in this lecture course, such maps are called *affine*. This can be a source of confusion; it is good to keep it in mind.

Examples 1.2.3. (i) The map

$$\begin{array}{rcl} r\colon \mathbb{R}^2 & \longrightarrow & \mathbb{R}^2 \\ \begin{pmatrix} a \\ b \end{pmatrix} & \mapsto & \begin{pmatrix} -b \\ a \end{pmatrix} \end{array}$$

is linear: We have

$$\begin{aligned} r\left(\lambda\begin{pmatrix}a\\b\end{pmatrix}+\mu\begin{pmatrix}c\\d\end{pmatrix}\right) &= r\left(\begin{pmatrix}\lambda a+\mu c\\\lambda b+\mu d\end{pmatrix}\right)=\begin{pmatrix}-\lambda b-\mu d\\\lambda a+\mu c\end{pmatrix}\\ &= \lambda\begin{pmatrix}-b\\a\end{pmatrix}+\mu\begin{pmatrix}-d\\c\end{pmatrix}\\ &= \lambda\cdot r\left(\begin{pmatrix}a\\b\end{pmatrix}\right)+\mu\cdot r\left(\begin{pmatrix}c\\d\end{pmatrix}\right)\end{aligned}$$

So the requirements for a map to be linear are satisfied.

This map describes a counterclockwise rotation of 90 degrees - I leave it to you to visualize that this is indeed the case.

(ii) The map

$$\begin{aligned} s\colon \mathbb{R}^3 &\longrightarrow \mathbb{R}\\ (a,b,c) &\mapsto a+b+c\end{aligned}$$

summing the entries of a column vector is linear:

$$\begin{aligned} s(\lambda(a,b,c)+\mu(e,f,g)) &= s((\lambda a+\mu e,\lambda b+\mu f,\lambda c+\mu g))\\ &= \lambda a+\mu e+\lambda b+\mu f+\lambda c+\mu g\\ &= \lambda(a+b+c)+\mu(e+f+g)\\ &= \lambda\cdot s((a,b,c))+\mu\cdot s((e,f,g))\end{aligned}$$

(iii) The map

$$\begin{aligned} q\colon \mathbb{R}^2 &\longrightarrow \mathbb{R}\\ (a,b) &\mapsto a^2+b\end{aligned}$$

is not linear: We have

$$\begin{aligned} q((1,0)+(1,0)) &= q((2,0))=2^2=4\neq 2=1+1\\ &= q((1,0))+q((1,0))\end{aligned}$$

(iv) The map

$$\begin{aligned} k\colon \mathbb{R} &\longrightarrow \mathbb{R}\\ a &\mapsto 2a+3\end{aligned}$$

is not linear: We have $k(0)=3\neq 0$, but a linear map would satisfy $k(0)=k(0\cdot 0)=0\cdot k(0)=0$.

(v) The infinitely often differentiable functions form a subvector space of $\mathrm{Map}(\mathbb{R}, \mathbb{R})$, denoted by $C^\infty(\mathbb{R}, \mathbb{R})$. Mapping a function to its derivative is a linear map from that vector space to itself $C^\infty(\mathbb{R}, \mathbb{R}) \to C^\infty(\mathbb{R}, \mathbb{R})$, $f \mapsto f'$. To see this, remember that we have the differentiation rules $(\lambda f)' = \lambda \cdot f'$ and $(f+g)' = f' + g'$.

(vi) The continuous functions from the unit interval $[0,1]$ to the real numbers $\mathbb{R}$ form a subvector space of $\mathrm{Map}([0,1], \mathbb{R})$ denoted by $C^0(\mathbb{R}, \mathbb{R})$. Integrating is a linear map

$$C^0([0,1], \mathbb{R}) \to \mathbb{R}, f \mapsto \int_0^1 f(x)\mathrm{d}x,$$

because of the integration rules $\int_0^1 \lambda f(x)\mathrm{d}x = \lambda \int_0^1 f(x)\mathrm{d}x$ und $\int_0^1 (f(x) + g(x))\mathrm{d}x = \int_0^1 f(x)\mathrm{d}x + \int_0^1 g(x)\mathrm{d}x$

The last two examples are fundamental for many applications of linear algebra, e.g. for solving differential equations.

Remember that a map is called “injective”, or 1-1, if each value in the range of the map occurs at most one time. A map is called “surjective” or “onto” if every element in the range occurse as a value. It is called “bijective” if it is both injective and surjective. In this case there is an inverse map.

Definition 1.2.4. *A linear map $f\colon V \to W$ is called*

(i) isomorphism $:\Leftrightarrow$ *f is bijective.*

(ii) endomorphism $:\Leftrightarrow V = W$.

(iii) automorphism $:\Leftrightarrow$ *f is bijective und $V = W$.*

To mark that a linear map is an isomorphism, one sometimes writes a tilde on top of the arrow: $f : V \xrightarrow{\sim} W$.

Fact 1.2.5. The inverse map of a bijective linear map is also linear.

Examples 1.2.6. (i) In example 1.2.3 the map from (i), the rotation by 90 degrees, is an automorphism – you can just rotate back to undo the effect of that map!

(ii) The map from example (ii) is surjective but not injective, hence no isomorphism: $(a, b, c) \mapsto a + b + c$ ist surjective, since every $x \in \mathbb{R}$ occurs as value of the map, e.g. as the value of $(0, x, 0)$. The different vectors $(1, 0, 0)$ and $(0, 1, 0)$ have the same image 1, and so the map is not injective.

(iii) The polynomials of degree $\leq n$ form a subvector space $\mathbb{R}[x]^{\leq n} := \{p(x) \mid \deg p(x) \leq n\} \subseteq \mathrm{Map}([0, 1], \mathbb{R})$ of the vector space of all maps from $[0, 1]$ to $\mathbb{R}$. The map

$$\begin{array}{rcl} \mathbb{R}[x]^{\leq n} & \longrightarrow & \mathbb{R}^{n+1} \\ a_0 + a_1 x + \ldots + a_n x^n & \mapsto & (a_0, \ldots, a_n) \end{array}$$

is an isomorphism with inverse map

$$\begin{array}{rcl} \mathbb{R}^{n+1} & \longrightarrow & \mathbb{R}[x]^{\leq n} \\ (a_0, \ldots, a_n) & \mapsto & a_0 + a_1 x + \ldots + a_n x^n \end{array}$$

The last example illustrates that an isomorphism between two different vector spaces can be seen as an encoding of the elements of one vector space in terms of the other.

Observation 1.2.7. Given a vector space V and a basis $\{e_1, \ldots, e_n\}$ of V, one gets an isomorphism $V \to \mathbb{R}^n$ defined by

$$v = \lambda_1 e_1 + \ldots + \lambda_n e_n \quad \mapsto \quad \begin{pmatrix} \lambda_1 \\ \vdots \\ \lambda_n \end{pmatrix}$$

(this definition works because there exist unique such λ_i for every v, by the definition of basis!).

Its inverse is given by

$$\begin{pmatrix} \lambda_1 \\ \vdots \\ \lambda_n \end{pmatrix} \quad \mapsto \quad \lambda_1 e_1 + \ldots + \lambda_n e_n$$

Therefore any n-dimensional vector space is isomorphic to $\mathbb{R}^n$. Note that we need to choose an order of the basis vectors to build the isomorphism!

1.2.1 The matrix associated to a linear map

Observation 1.2.8. (i) Given a linear map $f\colon V \to W$ and a basis $B = \{e_1, \ldots, e_n\}$ of V, it is enough to know the values $f(e_1), \ldots, f(e_n)$ to calculate $f(v)$ for all $v \in V$: Indeed, we can always express a vector v as a linear combination of the e_i, as $v = \lambda_1 e_1 + \ldots + \lambda_n e_n$, and by linearity we have

$$f(v) = f(\lambda_1 e_1 + \ldots + \lambda_n e_n) = \lambda_1 f(e_1) + \ldots + \lambda_n f(e_n)$$

(ii) If we additionally have a basis $B'\{c_1, \ldots, c_m\}$ of W (thus $m = dim\ W$), we can further write

$$f(e_j) = \mu_{1j} c_1 + \ldots + \mu_{mj} c_m$$

Thus f is described completely by the matrix

$$\begin{pmatrix} \mu_{11} & \mu_{12} & \cdots & \mu_{1n} \\ \mu_{21} & \ddots & & \mu_{2n} \\ \vdots & & \ddots & \vdots \\ \mu_{m1} & \mu_{m2} & \cdots & \mu_{mn} \end{pmatrix}$$

A compact notation for the above matrix is (μ_{ij}). In the coefficient μ_{ij} the index i is for the row and the index j is for the column.

Notation: We denote the matrix associated to a linear map $f\colon V \to W$, a basis B of V and a basis B' of W by ${}_{B'}M(f)_B$.

Example 1.2.9. Let's look again at the counterclockwise rotation of 90 degrees:

$$\begin{array}{cccc} r\colon \mathbb{R}^2 & \longrightarrow & \mathbb{R}^2 \\ \begin{pmatrix} a \\ b \end{pmatrix} & \mapsto & \begin{pmatrix} -b \\ a \end{pmatrix} \end{array}$$

We choose as basis for domain and codomain each time the standard basis of $\mathbb{R}^2$. We have

$$r\left(\begin{pmatrix} 1 \\ 0 \end{pmatrix}\right) = \begin{pmatrix} 0 \\ 1 \end{pmatrix} = 0 \cdot \begin{pmatrix} 1 \\ 0 \end{pmatrix} + 1 \cdot \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

and

$$r\left(\begin{pmatrix}0\\1\end{pmatrix}\right)=\begin{pmatrix}-1\\0\end{pmatrix}=(-1)\cdot\begin{pmatrix}1\\0\end{pmatrix}+0\cdot\begin{pmatrix}0\\1\end{pmatrix}$$

Therefore

$$_SM_S(r)=\begin{pmatrix}0&-1\\1&0\end{pmatrix}.$$

Example 1.2.10. Consider the map

$$\begin{aligned} f\colon \mathbb{R}^2 &\longrightarrow \mathbb{R}^3\\ \begin{pmatrix}a\\b\end{pmatrix} &\mapsto \begin{pmatrix}a-b\\2a\\-a+3b\end{pmatrix}\end{aligned}$$

I leave it to you to convince yourself that this map is indeed linear. First we consider the standard bases $S:=\left(\begin{pmatrix}1\\0\end{pmatrix},\begin{pmatrix}0\\1\end{pmatrix}\right)$ of $\mathbb{R}^2$ and

$$S':=\left(\begin{pmatrix}1\\0\\0\end{pmatrix},\begin{pmatrix}0\\1\\0\end{pmatrix},\begin{pmatrix}0\\0\\1\end{pmatrix}\right)\text{ of }\mathbb{R}^3$$

We have

$$f\left(\begin{pmatrix}1\\0\end{pmatrix}\right)=\begin{pmatrix}1\\2\\-1\end{pmatrix}=1\begin{pmatrix}1\\0\\0\end{pmatrix}+2\begin{pmatrix}0\\1\\0\end{pmatrix}+(-1)\begin{pmatrix}0\\0\\1\end{pmatrix}$$

and

$$f\left(\begin{pmatrix}0\\1\end{pmatrix}\right)=\begin{pmatrix}-1\\0\\3\end{pmatrix}=(-1)\begin{pmatrix}1\\0\\0\end{pmatrix}+0\begin{pmatrix}0\\1\\0\end{pmatrix}+3\begin{pmatrix}0\\0\\1\end{pmatrix},$$

and therefore

$$_{S'}M_S(f)=\begin{pmatrix}1&-1\\2&0\\-1&3\end{pmatrix}$$

You see: The columns of the matrix are the images of the basis vectors – but this is only the case because we chose the standard basis as the basis for the domain!

Let's still keep the standard basis S for $\mathbb{R}^2$ but now choose the following basis for $\mathbb{R}^3$:

$$C := \left(\begin{pmatrix}1\\1\\1\end{pmatrix}, \begin{pmatrix}0\\1\\2\end{pmatrix}, \begin{pmatrix}3\\1\\0\end{pmatrix}\right)$$

Now we need to find a_{11}, a_{21}, a_{31} such that

$$f\left(\begin{pmatrix}1\\0\end{pmatrix}\right) = \begin{pmatrix}1\\2\\-1\end{pmatrix} = a_{11}\begin{pmatrix}1\\1\\1\end{pmatrix} + a_{21}\begin{pmatrix}0\\1\\2\end{pmatrix} + a_{31}\begin{pmatrix}3\\1\\0\end{pmatrix}$$

Put differently: We search for solutions a_{11}, a_{21}, a_{31} of the system of linear equations

$$\begin{array}{rcl} 1\cdot a_{11} + 0\cdot a_{21} + 3\cdot a_{31} & = & 1 \\ 1\cdot a_{11} + 1\cdot a_{21} + 1\cdot a_{31} & = & 2 \\ 1\cdot a_{11} + 2\cdot a_{21} + 0\cdot a_{31} & = & -1 \end{array}$$

With Gauß elimination we get:

$$\left(\begin{array}{ccc|c} 1 & 0 & 3 & 1 \\ 1 & 1 & 1 & 2 \\ 1 & 2 & 0 & -1 \end{array}\right) \underset{A(-1,1,2)}{\overset{A(-1,1,3)}{\rightsquigarrow}} \left(\begin{array}{ccc|c} 1 & 0 & 3 & 1 \\ 0 & 1 & -2 & 1 \\ 0 & 2 & -3 & -2 \end{array}\right) \overset{A(-2,2,3)}{\rightsquigarrow} \left(\begin{array}{ccc|c} 1 & 0 & 3 & 1 \\ 0 & 1 & -2 & 1 \\ 0 & 0 & 1 & -4 \end{array}\right)$$

Therefore we have $a_{31} = -4, a_{21} = 1 + 2\cdot(-4) = -7$ und $a_{11} = 1 - 3\cdot(-4) = 13$, i.e.

$$f\left(\begin{pmatrix}1\\0\end{pmatrix}\right) = \begin{pmatrix}1\\2\\-1\end{pmatrix} = 13\begin{pmatrix}1\\1\\1\end{pmatrix} + (-7)\begin{pmatrix}0\\1\\2\end{pmatrix} + (-4)\begin{pmatrix}3\\1\\0\end{pmatrix}$$

Likewise we need to find a_{12}, a_{22}, a_{32} such that

$$f\left(\begin{pmatrix}0\\1\end{pmatrix}\right) = \begin{pmatrix}-1\\0\\3\end{pmatrix} = a_{12}\begin{pmatrix}1\\1\\1\end{pmatrix} + a_{22}\begin{pmatrix}0\\1\\2\end{pmatrix} + a_{32}\begin{pmatrix}3\\1\\0\end{pmatrix}$$

Put differently: We look for solutions a_{12}, a_{22}, a_{32} of the system of linear equations

$$\begin{array}{rcl} 1\cdot a_{12} + 0\cdot a_{22} + 3\cdot a_{32} & = & -1 \\ 1\cdot a_{12} + 1\cdot a_{22} + 1\cdot a_{32} & = & 0 \\ 1\cdot a_{12} + 2\cdot a_{22} + 0\cdot a_{32} & = & 3 \end{array}$$

With *precisely the same steps* for Gauß elimination we get:

$$\left(\begin{array}{ccc|c} 1 & 0 & 3 & -1 \\ 1 & 1 & 1 & 0 \\ 1 & 2 & 0 & 3 \end{array}\right) \underset{A(-1,1,2)}{\overset{A(-1,1,3)}{\rightsquigarrow}} \left(\begin{array}{ccc|c} 1 & 0 & 3 & -1 \\ 0 & 1 & -2 & 1 \\ 0 & 2 & -3 & 4 \end{array}\right) \overset{A(-2,2,3)}{\rightsquigarrow} \left(\begin{array}{ccc|c} 1 & 0 & 3 & -1 \\ 0 & 1 & -2 & 1 \\ 0 & 0 & 1 & 2 \end{array}\right)$$

We could just have carried along the extra column $(-1, 0, 3)$ in the previous system of linear equations.

We get $a_{12} = -7, a_{22} = 5, a_{32} = 2$, and so

$$f\left(\begin{pmatrix} 0 \\ 1 \end{pmatrix}\right) = \begin{pmatrix} -1 \\ 0 \\ 3 \end{pmatrix} = (-7)\begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} + 5\begin{pmatrix} 0 \\ 1 \\ 2 \end{pmatrix} + 2\begin{pmatrix} 3 \\ 1 \\ 0 \end{pmatrix}$$

Thus we have

$$_C M_S(f) = \begin{pmatrix} 13 & -7 \\ -7 & 5 \\ -4 & 2 \end{pmatrix}$$

As you see, the matrix is different from before. It really depends on the choice of basis!

Example 1.2.11. Consider the vector space of all functions $[0, 2\pi] \to \mathbb{R}$ of the form $x \mapsto a \cdot \sin(x) + b \cdot \cos(x)$ for $a, b \in \mathbb{R}$. Let us call the set of all such functions V. This is a subvector space of $\operatorname{Map}([0, 2\pi], \mathbb{R})$, because:

1. The constant map with value 0 is of this form (just take $a = b = 0$)

2. If $f(x) = a \cdot \sin(x) + b \cdot \cos(x)$ and $g(x) = c \cdot \sin(x) + d \cdot \cos(x)$, then $(f + g)(x) = (a + c) \cdot \sin(x) + (b + d) \cdot \cos(x)$, so $f + g$ is again of this form, hence the set is closed under sums

3. If $f(x) = a \cdot \sin(x) + b \cdot \cos(x)$ and $\lambda \in \mathbb{R}$, then $(\lambda f)(x) = (\lambda \cdot a) \cdot \sin(x) + (\lambda \cdot b) \cdot \cos(x)$ is again of this form, so the set is closed under scalar products.

Clearly the functions sin and cos are a generating system of the set we are considering. They also are linearly independent: If a linear combination of these functions results in the constant function with value 0, i.e. if we have $a \cdot \sin(x) + b \cdot \cos(x) = 0$ for all $x \in [0, 2\pi]$, then in particular for $x = 0$, because of $\sin(0) = 0$ and $\cos(0) = 1$, we obtain the equation $b = a \cdot 0 + b \cdot 1 = a \cdot sin(0) + b \cdot \cos(0) = 0$. For $x = \frac{\pi}{2}$ we obtain, because of $\sin(\frac{\pi}{2}) = 1$

and $\cos(\frac{\pi}{2}) = 0$, the equation $a = a \cdot 1 + b \cdot 0 = a \cdot sin(\frac{\pi}{2}) + b \cdot \cos(\frac{\pi}{2}) = 0$. Thus both coefficients a and b must be $= 0$, so sin and cos are linearly indeendent, and thus form a basis.

As we noticed in example 1.2.3.(v), taking the derivative is a linear map because of the rules $(f+g)' = f' + g'$ and $(\lambda \cdot f)' = \lambda \cdot f'$.

The derivatives of sine and cosine are $\sin'(x) = \cos(x) = 0 \cdot \sin(x) + 1 \cdot \cos(x)$ and $\cos'(x) = -\sin(x) = (-1) \cdot \sin(x) + 0 \cdot \cos(x)$.

This implies, first, that the derivative of a function from V lies in V again. Furthermore it implies that the matrix of the map $d\colon V \to V$, $f \mapsto f'$ with respect to the basis $B := (\sin, \cos)$ is the following:

$$_BM_B(d) = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}$$

We have seen that the numbers in the matrix are all you need in order to calculate the linear map for every vector. We also have seen that it is especially easy to set up the matrix of a map $f\colon \mathbb{R}^m \to \mathbb{R}^n$ with respect to the standard bases of $\mathbb{R}^m$ and $\mathbb{R}^n$: The first column is the image of the first standard basis vector, the second is the image of the second standard basis vector and so on.

The columns of the matrix $_SM(f)_S$ are the images of the standard basis vectors of $\mathbb{R}^m$.

Vice versa, one can interpret any matrix as the matrix of the linear map sending the standard basis vectors to the columns. It is easy to calculate, using the matrix, what that map does to a general vector $(\lambda_1, \ldots, \lambda_m)$: If one interprets the matrix

$$A := \begin{pmatrix} a_{11} & a_{12} & \ldots & a_{1m} \\ a_{21} & \ddots & & a_{2m} \\ \vdots & & \ddots & \vdots \\ a_{n1} & a_{n2} & \ldots & a_{nm} \end{pmatrix}$$

as $A =_S M_S(f)$ for a linear map $f \colon \mathbb{R}^m \to \mathbb{R}^n$ then one obtains

$$\begin{aligned} f\left(\begin{pmatrix} \lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_m \end{pmatrix}\right) &= f\left(\lambda_1 \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} + \lambda_2 \begin{pmatrix} 0 \\ 1 \\ \vdots \\ 0 \end{pmatrix} + \ldots + \lambda_m \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{pmatrix}\right) \\ &= \lambda_1 f\left(\begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}\right) + \lambda_2 f\left(\begin{pmatrix} 0 \\ 1 \\ \vdots \\ 0 \end{pmatrix}\right) + \ldots + \lambda_m f\left(\begin{pmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{pmatrix}\right) \\ &= \lambda_1 \begin{pmatrix} a_{11} \\ a_{21} \\ \vdots \\ a_{n1} \end{pmatrix} + \lambda_2 \begin{pmatrix} a_{12} \\ a_{22} \\ \vdots \\ a_{n2} \end{pmatrix} + \ldots + \lambda_m \begin{pmatrix} a_{1m} \\ a_{2m} \\ \vdots \\ a_{nm} \end{pmatrix} \\ &= \begin{pmatrix} \lambda_1 a_{11} + \lambda_2 a_{12} + \ldots + \lambda_m a_{1m} \\ \lambda_1 a_{21} + \lambda_2 a_{22} + \ldots + \lambda_m a_{2m} \\ \vdots \\ \lambda_1 a_{n1} + \lambda_2 a_{n2} + \ldots + \lambda_m a_{nm} \end{pmatrix} \end{aligned}$$

This motivates the following definition of the product of a matrix with a vector:

Definition 1.2.12. *The multiplication of a matrix with m columns from the left with a vector with m rows is defined by:*

$$\begin{pmatrix} a_{11} & a_{12} & \ldots & a_{1m} \\ a_{21} & \ddots & & a_{2m} \\ \vdots & & \ddots & \vdots \\ a_{n1} & a_{n2} & \ldots & a_{nm} \end{pmatrix} \cdot \begin{pmatrix} \lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_m \end{pmatrix} := \begin{pmatrix} \lambda_1 a_{11} + \lambda_2 a_{12} + \ldots + \lambda_m a_{1m} \\ \lambda_1 a_{21} + \lambda_2 a_{22} + \ldots + \lambda_m a_{2m} \\ \vdots \\ \lambda_1 a_{n1} + \lambda_2 a_{n2} + \ldots + \lambda_m a_{nm} \end{pmatrix}$$

Example 1.2.13.

$$\begin{pmatrix} 1 & 0 & 3 & -1 \\ 0 & 0 & -1 & 1 \\ 0 & 5 & -3 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 \\ 2 \\ 1 \\ -1 \end{pmatrix} = \begin{pmatrix} 1\cdot 1 + 0\cdot 2 + 3\cdot 1 + (-1)\cdot(-1) \\ 0\cdot 1 + 0\cdot 2 + (-1)\cdot 1 + 1\cdot(-1) \\ 0\cdot 1 + 5\cdot 2 + (-3)\cdot 1 + 1\cdot(-1) \end{pmatrix} = \begin{pmatrix} 5 \\ -2 \\ 6 \end{pmatrix}$$

1.2.2 Composition of functions and matrix multiplication

Given linear maps $f\colon K^m \to K^n$ und $g\colon K^n \to K^r$, one can compose them and obtains another linear map $g \circ f\colon K^m \to K^r$. To each of the three maps $f, g, g \circ f$ we can associate its matrix with respect to the standard basis. It is easy to calculate ${}_SM_S(g \circ f)$ from the matrices ${}_SM_S(g)$ and ${}_SM_S(f)$: Let

$${}_SM_S(f) = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1m} \\ a_{21} & \ddots & & a_{2m} \\ \vdots & & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nm} \end{pmatrix} \text{ and } {}_SM_S(g) = \begin{pmatrix} b_{11} & b_{12} & \dots & b_{1n} \\ b_{21} & \ddots & & b_{2n} \\ \vdots & & \ddots & \vdots \\ b_{r1} & b_{r2} & \dots & b_{rn} \end{pmatrix}$$

The columns of the matrix ${}_SM_S(g \circ f)$, that we are looking for, are the images of the standard basis vectors $(e_1, \dots, e_m)$ under the map $g \circ f$. Thus in the j-th column we have $g(f(e_j))$. The vector $f(e_j)$ is the j-th column of ${}_SM_S(f)$, i.e. the vector with entries $a_{1j}, a_{2j}, \dots, a_{nj}$. To determine the j-th column of ${}_SM_S(g \circ f)$, we need to calculate the image of this vector uner the map g. But this is exactly the Produkt of the matrix ${}_SM_S(g)$ with this column vector, by the calculation before Def. 1.2.12:

$$\begin{pmatrix} b_{11} & b_{12} & \dots & b_{1n} \\ b_{21} & \ddots & & b_{2n} \\ \vdots & & \ddots & \vdots \\ b_{r1} & b_{r2} & \dots & b_{rn} \end{pmatrix} \cdot \begin{pmatrix} a_{1j} \\ a_{2j} \\ \vdots \\ a_{nj} \end{pmatrix} := \begin{pmatrix} a_{1j}b_{11} + a_{2j}b_{12} + \dots + a_{nj}b_{1n} \\ a_{1j}b_{21} + a_{2j}b_{22} + \dots + a_{nj}b_{2n} \\ \vdots \\ a_{1j}b_{r1} + a_{2j}b_{r2} + \dots + a_{nj}b_{rn} \end{pmatrix} = \begin{pmatrix} \sum_{k=1}^n a_{kj}b_{1k} \\ \sum_{k=1}^n a_{kj}b_{2k} \\ \vdots \\ \sum_{k=1}^n a_{kj}b_{rk} \end{pmatrix}$$

Like this we can calculate all m columns of the matrix ${}_SM_S(g \circ f)$. Thus the entry in the i-th row and j-th column of ${}_SM_S(g \circ f)$ is $\sum_{k=1}^n a_{kj}b_{ik}$.

Definition 1.2.14. *The* product *of the matrices $B = (b_{ik}) \in \mathrm{Mat}_{r\times n}(K)$ and $A = (a_{kj}) \in \mathrm{Mat}_{n\times m}(K)$ ist defined to be the matrix $(c_{ij}) \in \mathrm{Mat}_{r\times m}(K)$ with $c_{ij} := \sum_{k=1}^n a_{kj}b_{ik}$.*

One writes for this matrix: $B \cdot A$ or, omitting the point, just BA.

Note that the product $B \cdot A$ of matrices is only defined if A has as many rows as B has columns.

Examples 1.2.15.

$$\begin{pmatrix} 2 & 1 & 0 \\ 3 & 1 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 \\ 2 & 1 \\ 3 & 1 \end{pmatrix} = \begin{pmatrix} 2\cdot 1 + 1\cdot 2 + 0\cdot 3 & 2\cdot 0 + 1\cdot 1 + 0\cdot 1 \\ 3\cdot 1 + 1\cdot 2 + 1\cdot 3 & 3\cdot 0 + 1\cdot 1 + 1\cdot 1 \end{pmatrix} = \begin{pmatrix} 4 & 1 \\ 8 & 2 \end{pmatrix}$$

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

$$\begin{pmatrix} 2 & 1 \\ 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 2 \\ 0 & 3 \end{pmatrix} = \begin{pmatrix} 2\cdot 1 + 1\cdot 0 & 2\cdot 2 + 1\cdot 3 \\ 0\cdot 1 + 1\cdot 0 & 0\cdot 2 + 1\cdot 3 \end{pmatrix} + \begin{pmatrix} 2 & 7 \\ 0 & 3 \end{pmatrix}$$

As explained above, matrix multiplication is defined in such a way that the following becomes true:

Proposition 1.2.16. *Let $f\colon K^m \to K^n$ und $g\colon K^n \to K^r$ be linear maps. Then*

$${}_SM_S(g \circ f) =_S M_S(g) \cdot_S M_S(f)$$

More generally, taking into account different bases, we have

Proposition 1.2.17. *Let $f\colon V \to W$ und $g\colon W \to U$ be linear maps, and let B, C and D be bases of V, W and U respectively. Then*

$${}_DM_C(g) \cdot_C M_B(f) =_D M_B(g \circ f)$$

The proof can be extracted from the considerations before Def. 1.2.14.

This also tells us the following: Given a linear map $f\colon V \to W$ and bases B, B' of V and C, C' of W, the matrices ${}_CM_B(f)$ and ${}_{C'}M_{B'}(f)$ are related by the following equation:

$${}_CM_B(f) =_C M_B(\mathrm{id}_W \circ f \circ \mathrm{id}_V) =_C M_{C'}(\mathrm{id}_W) \cdot_{C'} M_{B'}(f) \cdot_{B'} M_B(\mathrm{id}_V)$$

Here $\mathrm{id}_V\colon V \to V$ denotes the identity map of V, which sends every vector to itself, and similarly id_W.

The matrices ${}_{B'}M_B(\mathrm{id}_V)$ are also called *base change matrices*.

Example 1.2.18. Given a basis $B = \{b_1, \ldots, b_n\}$ of $\mathbb{R}^n$, the base change matrix ${}_S M_B(id)$ from B to the standard basis is the matrix whose columns are the vectors $b_1, \ldots, b_n$: Indeed, by the recipe for setting up a matrix, one has to take the basis vectors of the basis B, apply the identity map, i.e. leave them unchanged, express them as linear combinations of the standard basis and record the coefficients in the columns. The result will exactly be the vectors b_i themselves.

The other base change matrix ${}_B M_S(id)$, with S and B switched is the inverse of the above one – see below.

1.2.3 Invertible matrices

If $f\colon V \to W$ is an isomorphism, then there is an inverse map f^{-1} satisfying $f \circ f^{-1} = \mathrm{id}_W$ and $f^{-1} \circ f = \mathrm{id}_V$. For bases B and C of V and W we then have ${}_C M_B(f) \cdot_B M_C(f^{-1}) =_C M_C(f \circ f^{-1}) =_C M_C(\mathrm{id}_W)$, and similar for the composition in the other order. As one can see immediately, the matrix of the identity map with respect to the same basis for the domain and the target is always the so-called *unit matrix*, henceforth denoted by I, which has 1s on the diagonal and 0s otherwise:

$${}_C M_C(\mathrm{id}) = \begin{pmatrix} 1 & 0 & \ldots & 0 \\ 0 & 1 & & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \ldots & 1 \end{pmatrix} =: I$$

Definition 1.2.19. *A matrix A is called* invertible *if there exists a matrix B with $A \cdot B = I = B \cdot A$. Such a B is then necessarily unique, and is called the* inverse *of A. Notation: A^{-1}.*

Only a square matrix can be invertible: The image of a basis under an isomorphism is again a basis, thus source and target of an isomorphism must have the same dimensions, and therefore the corresponding matrix must have as many rows as it has columns.

In particular, base change matrices are invertible and satisfy ${}_C M_B(id) \cdot_B M_C(id) =_C M_C(id) = I$ and ${}_B M_C(id) \cdot_C M_B(id) =_B M_B(id) = I$, which shows that ${}_C M_B(id)$ is the inverse of ${}_B M_C(id)$.

One can calculate inverses using the so-called elementary row and column operations – see below.

1.2.4 Row operations and elementary matrices

You probably know how to solve a system of linear equations, by bringing it into row echelon form. This procedure is called the Gauss algorithm. Here is an example:

Example 1.2.20. The system of linear equations

$$\begin{array}{rrrrrcl} & x_2 & -x_3 & +x_4 & +x_5 & = & 0 \\ 2x_1 & -x_2 & +x_3 & +x_4 & -x_5 & = & 0 \\ 4x_1 & +x_2 & -x_3 & +x_4 & +x_5 & = & 1 \\ 2x_1 & +x_2 & -x_3 & +x_4 & +2x_5 & = & 0 \end{array}$$

corresponds to the matrix

$$\left(\begin{array}{ccccc|c} 0 & 1 & -1 & 1 & 1 & 0 \\ 2 & -1 & 1 & 1 & -1 & 0 \\ 4 & 1 & -1 & 1 & 1 & 1 \\ 2 & 1 & -1 & 1 & 2 & 0 \end{array}\right)$$

We swap the first two rows and obtain

$$\left(\begin{array}{ccccc|c} 2 & -1 & 1 & 1 & -1 & 0 \\ 0 & 1 & -1 & 1 & 1 & 0 \\ 4 & 1 & -1 & 1 & 1 & 1 \\ 2 & 1 & -1 & 1 & 2 & 0 \end{array}\right)$$

Now we add (-2) times the first row to the third row and (-1) times the first row to the fourth row. We obtain

$$\left(\begin{array}{ccccc|c} 2 & -1 & 1 & 1 & -1 & 0 \\ 0 & 1 & -1 & 1 & 1 & 0 \\ 0 & 3 & -3 & -1 & 3 & 1 \\ 0 & 2 & -2 & 0 & 3 & 0 \end{array}\right)$$

From now on the first row (and automatically also column) remains unchanged. We add (-3) times the second row to the third row and (-2) times the second row to the third row:

$$\left(\begin{array}{ccccc|c} 2 & -1 & 1 & 1 & -1 & 0 \\ 0 & 1 & -1 & 1 & 1 & 0 \\ 0 & 0 & 0 & -4 & 0 & 1 \\ 0 & 0 & 0 & -2 & 1 & 0 \end{array}\right)$$

From now on the second row (and automatically also column) remains unchanged. Finally we add $-\frac{1}{2}$ times the third row to the fourth row and obtain a matrix in row echelon form:

$$\left(\begin{array}{ccccc|c} 2 & -1 & 1 & 1 & -1 & 0 \\ 0 & 1 & -1 & 1 & 1 & 0 \\ 0 & 0 & 0 & -4 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & -\frac{1}{2} \end{array}\right)$$

Now we can go backwards and read off the values of x_1, x_2, x_3, x_4, x_5: In the fourth row we see that $x_5 = 1 \cdot x_5 = -\frac{1}{2}$. In the third row we see that $-4x_4 = 1$, i.e. that $x_4 = -\frac{1}{4}$.

With this the second row becomes

$$0 = x_2 + (-1)x_3 + x_4 + x_5 = x_2 + (-1)x_3 - \frac{1}{4} - \frac{1}{2},$$

and we obtain $x_2 = x_3 + \frac{3}{4}$, where one can choose x_3 freely.

Finally, the first row becomes

$$\begin{aligned} 0 &= 2x_1 + (-1)x_2 + x_3 + x_4 - x_5 \\ &= 2x_1 - (x_3 + \tfrac{3}{4}) + x_3 - \tfrac{1}{4} + \tfrac{1}{2} \\ &= 2x_1 - \tfrac{1}{2} \end{aligned}$$

So $x_1 = \frac{1}{4}$.

One can understand this procedure a bit more systematically. What we are doing here is performing so-called elementary row operations on matrices.

A solution $x_1, \ldots, x_m$ of the system of linear equations

$$\begin{array}{ccccc} a_{11}x_1 + a_{12}x_2 + & \ldots & +a_{1m}x_m & = & b_1 \\ a_{21}x_1 + a_{22}x_2 + & \ldots & +a_{2m}x_m & = & b_2 \\ \vdots & & \vdots & & \vdots \\ a_{n1}x_1 + a_{n2}x_2 + & \ldots & +a_{nm}x_m & = & b_n \end{array}$$

is exactly the same thing as a vector $x = (x_1, \ldots, x_m)$ satisfying

$$\begin{pmatrix} a_{11} & a_{12} & \ldots & a_{1m} \\ a_{21} & a_{22} & & a_{2m} \\ \vdots & & \vdots & \vdots \\ a_{n1} & a_{n2} & \ldots & a_{nm} \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_{m-1} \\ x_m \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix}$$

So if we call the left matrix A and the right vector b, then we are looking for a vector x satisfying $A \cdot x = b$.

Definition 1.2.21. *There are the following three* elementary row operations *that we can perform on a matrix:*

$S(i,j)$ *switch the ith and the jth row*

$M(\lambda,i)$ *multiply the ith row by $\lambda \in \mathbb{R} \setminus \{0\}$*

$A(\lambda,i,j)$ *add λ times the ith row to the jth row*

Fact 1.2.22. These operations correspond to multiplication from the left with so-called *elementary matrices*:

$$S(i,j) = \begin{pmatrix}
1 & 0 & 0 & 0 & & & \dots & & & & & 0 \\
0 & 1 & & & & & & & & & & \\
0 & & \ddots & & & & & & & & & \\
0 & & & 1 & & & & & & & & \\
 & & & & 0 & \dots & \dots & 0 & 1 & & & \\
 & & & & \vdots & 1 & 0 & \vdots & 0 & & & \\
\vdots & & & & \vdots & 0 & \ddots & 0 & \vdots & & & \vdots \\
 & & & & 0 & \dots & 0 & 1 & \vdots & & & \\
 & & & & 1 & 0 & \dots & \dots & 0 & & & \\
 & & & & & & & & & 1 & & \\
 & & & & & & & & & & \ddots & \\
0 & & & & & & \dots & & & & & 1
\end{pmatrix}$$

$$M(\lambda,i) = \begin{pmatrix}
1 & 0 & & & \dots & & & 0 \\
0 & 1 & & & & & & \\
 & & \ddots & & & & & \\
 & & & 1 & & & & \\
\vdots & & & & \lambda & & & \vdots \\
 & & & & & 1 & & \\
 & & & & & & \ddots & \\
0 & & & & \dots & & & 1
\end{pmatrix}$$

$$A(\lambda,i,j) = \begin{pmatrix} 1 & 0 & & & \cdots & & & 0 \\ 0 & 1 & & & & & & \\ & & \ddots & & & & & \\ & & & 1 & & \lambda & & \\ \vdots & & & & \ddots & & & \vdots \\ & & & 0 & & 1 & & \\ & & & & & & \ddots & \\ 0 & & & & \cdots & & & 1 \end{pmatrix}$$

You should verify this for yourselves!

Observation 1.2.23. Elementary matrices are invertible:

$$V(i,j)^{-1} = V(i,j), \quad M(\lambda,i)^{-1} = M(\frac{1}{\lambda},i), \quad A(\lambda,i,j)^{-1} = A(-\lambda,i,j)$$

You can easily check that the given matrices really are the inverses as claimed. Alternatively you can also think about which row operation one needs to perform, to undo the effect of a previous row operation:

If you switched rows i and j, then switching them again will return the matrix to its previous state. If you multiplied row i with the factor λ, then multiplying it with λ^{-1} will undo that effect. And if you added λ times the ith row to the j-th row, then adding $-\lambda$ times the ith row to the j-th row will undo that effect.

The Gauss algorithm does the following: Given a system of linear equations $Ax = b$, one multiplies from the left with elementary matrices $E_1, \ldots, E_k$ until one has a row echelon form: $E_k \cdot \ldots \cdot E_1 \cdot Ax = E_k \cdot \ldots \cdot E_1 \cdot b$. Since all the E_i are invertible, x is a solution of $Ax = b$, if and only if it is a solution of this latter equation.

1.2.5 Matrix inversion using row operations

To compute the inverse of an *invertible* matrix A one can do the following:

1. Use elementary row operations to transform A into the unit matrix.
2. Simultaneously use the same row operations on the unit matrix. The result of ths will be the inverse of A.

Why does this work? If the row operations correspond to multiplication with elementary matrices $E_1, \dots, E_k$, then after finishing one obtains the equation

$$E_k \cdot \ldots \cdot E_1 \cdot A = I_n.$$

Therefore $E_k \cdot \ldots \cdot E_1 = E_k \cdot \ldots \cdot E_1 \cdot I_n$ is the inverse of A. But $E_k \cdot \ldots \cdot E_1 \cdot I_n$ is exactly the matrix that one obtains, if one performs the row operations corresponding to $E_1, \dots, E_k$ on the unit matrix.

In practice one writes A and the unit matrix I side by side performs the row operations smultaneously on both matrices:

Example 1.2.24. We invert the matrix

$$A := \begin{pmatrix} 1 & 0 & 1 \\ 3 & 1 & 0 \\ 1 & 2 & 1 \end{pmatrix}$$

$$\left(\begin{array}{ccc|ccc} 1 & 0 & 1 & 1 & 0 & 0 \\ 3 & 1 & 0 & 0 & 1 & 0 \\ 1 & 2 & 1 & 0 & 0 & 1 \end{array}\right) \underset{A(-1,1,3)}{\overset{A(-3,1,2)}{\rightsquigarrow}} \left(\begin{array}{ccc|ccc} 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & -3 & -3 & 1 & 0 \\ 0 & 2 & 0 & -1 & 0 & 1 \end{array}\right)$$

$$\overset{A(-2,2,3)}{\rightsquigarrow} \left(\begin{array}{ccc|ccc} 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & -3 & -3 & 1 & 0 \\ 0 & 0 & 6 & 5 & -2 & 1 \end{array}\right) \overset{M(\frac{1}{6},3)}{\rightsquigarrow} \left(\begin{array}{ccc|ccc} 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & -3 & -3 & 1 & 0 \\ 0 & 0 & 1 & \frac{5}{6} & -\frac{1}{3} & \frac{1}{6} \end{array}\right)$$

$$\overset{A(3,3,2)}{\rightsquigarrow} \left(\begin{array}{ccc|ccc} 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & -\frac{1}{2} & 0 & \frac{1}{2} \\ 0 & 0 & 1 & \frac{5}{6} & -\frac{1}{3} & \frac{1}{6} \end{array}\right) \overset{A(-1,3,1)}{\rightsquigarrow} \left(\begin{array}{ccc|ccc} 1 & 0 & 0 & \frac{1}{6} & \frac{1}{3} & -\frac{1}{6} \\ 0 & 1 & 0 & -\frac{1}{2} & 0 & \frac{1}{2} \\ 0 & 0 & 1 & \frac{5}{6} & -\frac{1}{3} & \frac{1}{6} \end{array}\right)$$

Thus we have

$$A^{-1} = \begin{pmatrix} \frac{1}{6} & \frac{1}{3} & -\frac{1}{6} \\ -\frac{1}{2} & 0 & \frac{1}{2} \\ \frac{5}{6} & -\frac{1}{3} & \frac{1}{6} \end{pmatrix}$$

1.2.6 Some matrix operations

Two more operations on matrices:

Definition 1.2.25. *The* transpose *of $A = (a_{ij})$, denoted by A^T, is defined by $(A^T) := (a_{ji})$, i.e. by swapping rows and columns.*

Examples 1.2.26.

$$\begin{pmatrix} 13 & -33 \\ -7 & 19 \\ -4 & 10 \end{pmatrix}^T = \begin{pmatrix} 13 & -7 & -4 \\ -33 & 19 & 10 \end{pmatrix}$$

$$\begin{pmatrix} 1 & -2 \\ 0 & 0 \end{pmatrix}^T = \begin{pmatrix} 1 & 0 \\ -2 & 0 \end{pmatrix}$$

$$\begin{pmatrix} a_1 & a_2 & \dots & a_n \end{pmatrix}^T = \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{pmatrix}$$

Definition 1.2.27. *The* trace *of an $n \times n$-matrix $A = (a_{ij})$ is the sum of the diagonal entries:* $\mathrm{tr}(A) := \sum_{i=1}^{n} a_{ii}$.

We have already discussed the operations of matrix multiplication and sum of matrices (when we saw the example of the vector space of matrices). Here is a list of laws governing their interactions:

Fact 1.2.28.

- $A \cdot (B \cdot C) = (A \cdot B) \cdot C$
- $A \cdot (B + C) = A \cdot B + A \cdot C$
- $A \cdot I = A = I \cdot A$
- $(A^{-1})^{-1} = A$
- $(A \cdot B)^{-1} = B^{-1} \cdot A^{-1}$
- $(A^T)^T = A$
- $(A + B)^T = A^T + B^T$
- $(A \cdot B)^T = B^T \cdot A^T$
- $\mathrm{tr}(ABC) = \mathrm{tr}(CAB) = \mathrm{tr}(BCA)$ (but in general not $\mathrm{tr}(ABC) = \mathrm{tr}(BAC)$!)
- in particular: $\mathrm{tr}(AB) = \mathrm{tr}(IAB) = \mathrm{tr}(BIA) = \mathrm{tr}(BA)$
- also in particular: $\mathrm{tr}(B^{-1}AB) = \mathrm{tr}(A)$

Most of these laws are easy to verify – ask me if you try and fail. Only the laws about traces are a bit tricky.

You may not, at this point, be convinced that the trace is a thing you want to know about a matrix. Why sum up the entries on the diagonal, and not, for example, the entries in the first column? The last point gives one reason: Base change of a matrix does not change the trace of that matrix. Therefore the trace is an intrinsic property of a linear map, independent of the choice of basis to represent that map by a matrix. For a diagonalizable matrix, the trace is the sum of its eigenvalues – see later.

1.2.7 Kernel, image, rank and dimension formula

Every linear map $f\colon V \to W$ determines a subvector space of its domain, its *kernel*

$$\ker f := \{v \in V \mid f(v) = 0\}$$

and a subvector space of its target, its *image*:

$$\operatorname{Im} f := \{w \in W \mid \exists v \in V\colon f(v) = w\} = \{f(v) \mid v \in V\}$$

Proposition 1.2.29. *Kernel and image of a linear map $f\colon V \to W$ are subvector spaces.*

Proof. Kernel: Clearly $0 \in \ker f$. If $v, v' \in Ker\ f$, and $\lambda \in K$ then $f(\lambda \cdot v) = \lambda \cdot f(v) = \lambda \cdot 0 = 0$ and $f(v + v') = f(v) + f(v') = 0 + 0 = 0$. Therefore both $v + v'$ and $\lambda \cdot v$ are in the kernel of f.

Image: Clearly $0 \in \operatorname{Im} f$. If $w, w' \in Im\ f$, and $\lambda \in K$ then choose $v, v' \in V$ such that $f(v) = w$ and $f(v') = w'$. Then, since f is linear, we have $w + w' = f(v) + f(v') = f(v + v')$ and $\lambda w = \lambda f(v) = f(\lambda v)$. Therefore both $w + w'$ and λw are again in the image of f. □

Being subvector spaces, kernel and image have bases and dimensions. The dimension formula relates these dimensions:

Theorem 1.2.30 (Dimension formula)**.** *For a linear map $f\colon V \to W$ we have*

$$\dim V = \dim \ker f + \dim \operatorname{Im} f$$

Proof. Choose a basis $\{b_1, \ldots, b_k\}$ of $\ker f$, then complete it to a basis $\{b_1, \ldots, b_k, b_{k+1}, \ldots, b_n\}$ of all of V. Thus the dimension of V is n and the dimension of $\ker f$ is k, and we have to show that $\dim \operatorname{Im} f = n - k$.

Any vector in $\operatorname{Im} f$ is of the form

$$\begin{aligned} f(v) &= f(\lambda_1 b_1 + \ldots + \lambda_k b_k + \lambda_{k+1} b_{k+1} + \ldots + \lambda_n b_n) \\ &= \lambda_1 f(b_1) + \ldots + \lambda_k f(b_k) + \lambda_{k+1} f(b_{k+1}) + \ldots + \lambda_n f(b_n) \\ &= \lambda_1 0 + \ldots + \lambda_k 0 + \lambda_{k+1} f(b_{k+1}) + \ldots + \lambda_n f(b_n) \\ &= \lambda_{k+1} f(b_{k+1}) + \ldots + \lambda_n f(b_n) \end{aligned}$$

Hence $\{f(b_{k+1}), \ldots, f(b_n)\}$ spans $\operatorname{Im} f$.

If some linear combination of these vectors is zero, then

$$0 = \lambda_{k+1} f(b_{k+1}) + \ldots + \lambda_n f(b_n) = f(\lambda_{k+1} b_{k+1} + \ldots + \lambda_n b_n),$$

so $v := \lambda_{k+1} b_{k+1} + \ldots + \lambda_n b_n \in \ker f$. But vectors from $\ker f$ have a unique expression as linear combination of the basis $\{b_1, \ldots, b_n\}$ of V, *only using the vectors* $b_1, \ldots, b_k$ (since these form a basis of $\ker f$) and thus the other coefficients $\lambda_{k+1}, \ldots, \lambda_n$ must be zero. □

Intuitively, f has to map all of V somewhere. Some of the vectors of V are no longer visible in the image because they are mapped to zero, the others span the image. Maybe the following reformulation reflects this intuition better: *dim Im f = dim V − dim Ker f*.

Corollary 1.2.31. *For an endomorphism $f\colon V \to V$ the following are equivalent:*

(i) f is injective

(ii) f is surjective

(iii) f is bijective

Proof. A linear map is injective if and only if its kernel is $\{0\}$ (because if for $v \neq v'$ we have $f(v) = f(v')$, then $f(v - v') = f(v) - f(v') = 0$, but $v - v' \neq 0$, so we have a non-zero element in the kernel). By the dimension formula $\dim \ker f = 0$ if and only if $\dim \operatorname{Im} f = n$. The latter means precisely that f is surjective, because the only n-dimensional subspace of V is V itself. □

As always statements about linear maps can be spelled out in terms of matrices.

Definition 1.2.32. *The* rank *of a matrix is the dimension of the vector space spanned by its columns.*

Since the columns are the images of the basis vectors, the rank is precisely the dimension of the image of the linear map given by the matrix (e.g. with respect to the standard basis).

Computing the rank of a matrix

For a matrix in row echelon form, the rank is precisely the number of non-zero rows: To generate the vector space spanned by the columns, it is enough to take those columns where in the echelon form one goes one step down (the so-called pivot columns). Furthermore these column vectors are clearly linearly independent, due to the echelon form.

For a general matrix, one can compute the rank by bringing it into row echelon form: The elementary row operations correspond to multiplication with elementary matrices, i.e. to composition with isomorphisms. But an isomorphism does not change the dimension of the image.

Fact 1.2.33. $\operatorname{rank}(A) = \operatorname{rank}(A^T)$

1.2.8 Determinants

Determinants are usually defined as the unique map $\det\colon (\mathbb{R}^n)^n \to \mathbb{R}$ enjoying the following properties:

(i) det is multilinear, i.e. for any $v_1, \dots, v_{i-1}, v_{i+1}, \dots, v_r \in \mathbb{R}^n$ the map

$$\mathbb{R}^n \to \mathbb{R}, \quad v \mapsto \det(v_1, \dots, v_{i-1}, v, v_{i+1}, \dots, v_r)$$

is linear.

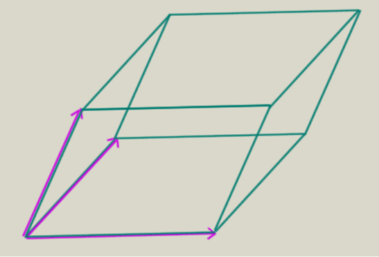
(ii) det is alternating, i.e. if two vectors are equal, the value is zero:

$$\forall v_1, \dots, v_n \in V, 1 \le i < j \le n\colon\ v_i = v_j \Rightarrow f(v_1, \dots, v_n) = 0$$

(iii) The value on the standard basis is 1, i.e. $\det(e_1, \dots, e_n) = 1$.

One can show that there is only one function having these three properties.

Determinants are best understood by considering how the volume of a parallelotope depends on the vectors spanning it:

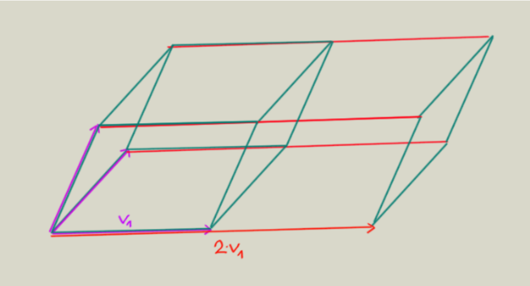


Parallelotope in $\mathbb{R}^3$

By “volume” here we understand a measure of size that is apropriate for the dimension of the ambient vector space – in $\mathbb{R}^1$ we mean length, in $\mathbb{R}^2$ we mean area, in $\mathbb{R}^3$ volume in the everyday sense and so on.

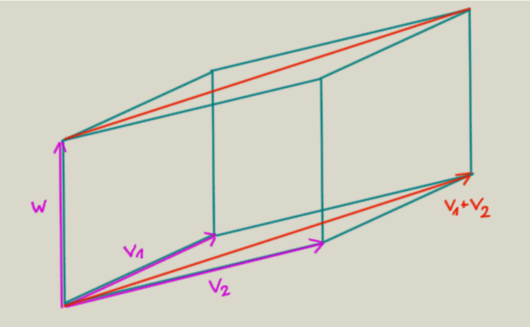
The function that associates to n vectors in $\mathbb{R}^n$ the volume of the parallelotope spanned by them, has the following properties:

Scalar products: Stretching one of the spanning vectors by a *positive* real number λ, multiplies the volume of the parallelotope also by λ.



Doubling one vector doubles the volume.

Addition: Given two vectors $v_1, v_2 in\mathbb{R}^n$, and $n-1$ other vectors, we can span polytopes P_1, P_2 and P_{sum} with these other vectors and v_1, v_2 and $v_1 + v_2$. Then the volume of P_{sum} is the sum of the volumes of P_1 and P_2. The drawing shows the 2-dimensional case:



Adding vectors results in adding volumes

Here P_1 is the area spanned by v_1 and w, P_2 is the area spanned by v_2 and w. Comparing these areas with the one spanned by $v_1 + v_2$ and w, there is a flat triangle missing below, but a triangle on top makes exactly up for that missing area.

"Alternation": If two of the n vectors are equal, then the spanned polytope is "flat", i.e. does not fill all the dimensions of the ambient vector space, and therefore has volume 0.

Signs: To understand the role of signs, think of what happens if one adds a vector pointing in the opposite direction, i.e. stretched by a negative factor: The volume decreases. So multilinearity also makes some sense for scalar products with negative numbers. However, this means that the determinant can also be negative. The *absolute value of the determinant* is the volume of the parallelotope. The determinant is sometimes also called the *oriented volume* of the parallelotope, because the direction (orientation) of the vectors is taken into account.

Usually the determinant is seen as an invariant of linear maps or matrices. Since the columns of a matrix are the images of the basis vectors under the corresponding linear map, and since the determinant of the standard basis is 1, the determinant computes the factor by which the volume of the unit (hyper)cube is changed under the linear map.

This gives a way to think about the following theorem:

Theorem 1.2.34. *A matrix A is invertible if and only if* $\det(A) \neq 0$.

Indeed, a linear map is surjective, if and only if the image of the basis spans the target space. For an endomorphism this is the case if and only if the volume of the corresponding parallelotope is non-zero. Further, by Cor. 1.2.31 for an endomorphism being surjective is equivalent to being an isomorphism, i.e. to corresponding to an invertible matrix.

1.2.9 Computing determinants

First some general laws for determinants:

Theorem 1.2.35. *For matrices A, B we have:*

(i) $\det(A \cdot B) = \det(A) \cdot \det(B)$

(ii) $\det(A^{-1}) = \det(A)^{-1}$

(iii) $\det(A^T) = \det(A)$

In particular it follows that base change does not affect the determinant of a matrix: $\det(B^{-1}AB) = \det(B^{-1})\det(A)\det(B) = \det(B)^{-1}\det(A)\det(B) = \det(A)$ (for the last step: this is a calculation of real numbers, so we can just switch the order of the factors).

Directly from the definition one can derive ways of computing determinants. The first is a recursive algorithm called Laplace expansion:

Theorem 1.2.36 (Laplace expansion). *Let $A = \left(a_{ij}\right)$ be an $(n \times n)$-matrix. For $k, l \in \{1, \dots, n\}$ let $A_{\langle k,l \rangle}$ denote the $(n-1) \times (n-1)$-matrix, arising from A by deleting the k-th row and the l-th column.*

Then we have for every $i \in \{1, \dots, n\}$ the so-called Laplace expansion along the i-th row:

$$\det A = \sum_{j=1}^{n} (-1)^{i+j} a_{ij} \det(A_{\langle i,j \rangle})$$

and for every $j \in \{1, \dots, n\}$ the so-called Laplace expansion along the j-th column:

$$\det A = \sum_{i=1}^{n} (-1)^{i+j} a_{ij} \det(A_{\langle i,j \rangle})$$

Example 1.2.37. We calculate the determinant of

$$A := \begin{pmatrix} 1 & 2 & 0 & -1 \\ 0 & 1 & 0 & 2 \\ 1 & -1 & 1 & -1 \\ 7 & 2 & 1 & 3 \end{pmatrix}$$

For having to calculate less, it is advantageous to expand by a row or column with many zeros. We take the third column.

$$\begin{aligned} \det(A) &= (-1)^{1+3} \cdot 0 \cdot \det \begin{pmatrix} 0 & 1 & 2 \\ 1 & -1 & -1 \\ 7 & 2 & 3 \end{pmatrix} + (-1)^{2+3} \cdot 0 \cdot \det \begin{pmatrix} 1 & 2 & -1 \\ 1 & -1 & -1 \\ 7 & 2 & 3 \end{pmatrix} \\ &\quad + (-1)^{3+3} \cdot 1 \cdot \det \begin{pmatrix} 1 & 2 & -1 \\ 0 & 1 & 2 \\ 7 & 2 & 3 \end{pmatrix} + (-1)^{4+3} \cdot 1 \cdot \det \begin{pmatrix} 1 & 2 & -1 \\ 0 & 1 & 2 \\ 1 & -1 & -1 \end{pmatrix} \end{aligned}$$

For clarity I wrote the factors which are 1, and the summands which have a factor 0 in front, but of course one can drop them. We now go on doing Laplace expansion on the remaining determinants of smaller matrices.

$$\begin{aligned}
\ldots \;=\;& (-1)^{1+1}\cdot 1\cdot\det\begin{pmatrix}1&2\\2&3\end{pmatrix} && +(-1)^{2+1}\cdot 0\cdot\det\begin{pmatrix}2&-1\\2&3\end{pmatrix}\\
& && +(-1)^{3+1}\cdot 7\det\begin{pmatrix}2&-1\\1&2\end{pmatrix}\\
& -(-1)^{1+1}\cdot 1\cdot\det\begin{pmatrix}1&2\\-1&-1\end{pmatrix} && -(-1)^{2+1}\cdot 0\cdot\det\begin{pmatrix}2&-1\\-1&-1\end{pmatrix}\\
& && -(-1)^{3+1}\cdot 1\cdot\det\begin{pmatrix}2&-1\\1&2\end{pmatrix}\\
=\;& 1\cdot(1\cdot 3-2\cdot 2)+7\cdot(2\cdot 2-1\cdot(-1)) && -1\cdot(1\cdot(-1)-(-1)\cdot 2)\\
& && -1\cdot(2\cdot 2-1(-1))\\
=\;& -1+35-1-5=28
\end{aligned}$$

Proposition 1.2.38. *Let D be an $n\times n$-matrix of the form*

$$D=\left(\begin{array}{c|c}A&C\\\hline \mathbf{0}&B\end{array}\right)$$

with an $r\times r$-matrix A, an $s\times s$-matrix B an $s\times r$-matrix C and $\mathbf{0}$ the $s\times r$-matrix consisting only of zeros – of course we then have $r+s=n$. A matrix of this form is called block matrix, and $A, B, C, \mathbf{0}$ are called the blocks of the matrix.

Then we have:

$$\det(D)=\det(A)\cdot\det(B)$$

Of course B can itself again be a block matrix, in which case one can apply this recursively. In particular for an upper triangular $n\times n$matrix $A=(a_{ij})$ we have $\det(A)=a_{11}\cdot\ldots\cdot a_{nn}$ – the determinant is then the product of the diagonal entries.

Example 1.2.39. Two of the three types of elementary matrices are upper triangular matrices. This gives:

$$\det(A(\lambda,i,j))=1 \qquad\text{and}\qquad \det(M(\lambda,i))=\lambda$$

For the third type one gets $\det(V(i,j))=-1$, because swapping two vectors in a matrix switches the sign of the determinant, an $V(i,j)$ arises from the unit matrix by swapping two columns.

Since row operations arise by multiplication with elementary matrices, this tells us together with Thm. 1.2.35 what are the effects of the usual row operations on the determinant:

(i) After an operation of type $V(i,j)$ the determinant changes by a factor of (-1).

(ii) After an operation of type $M(\lambda,i)$ the determinant changes by a factor of λ.

(iii) After an operation of type $A(\lambda,i,j)$ the determinant does not change.

This suggests another strategy for computing determinants: Bring the given matrix to row echelon form, recording on the way how the applied row operations alter the determinant, and then calculate the determinant of the resulting upper triangular matrix.

Example 1.2.40.

$$\begin{aligned}
&\det\begin{pmatrix} 0 & 1 & 1 & 1 \\ 2 & -1 & 1 & -1 \\ 4 & 1 & 1 & 1 \\ 2 & 1 & 1 & 0 \end{pmatrix} \overset{(1)}{=} -\det\begin{pmatrix} 2 & -1 & 1 & -1 \\ 0 & 1 & 1 & 1 \\ 4 & 1 & 1 & 1 \\ 2 & 1 & 1 & 0 \end{pmatrix} \\
&\overset{(2)}{=} -\det\begin{pmatrix} 2 & -1 & 1 & -1 \\ 0 & 1 & 1 & 1 \\ 0 & 3 & -1 & 3 \\ 0 & 2 & 0 & 1 \end{pmatrix} \overset{(3)}{=} -\det\begin{pmatrix} 2 & -1 & 1 & -1 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & -4 & 0 \\ 0 & 0 & -2 & -1 \end{pmatrix} \\
&\overset{(4)}{=} 4\cdot\det\begin{pmatrix} 2 & -1 & 1 & -1 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & -2 & -1 \end{pmatrix} \overset{(5)}{=} 4\det\begin{pmatrix} 2 & -1 & 1 & -1 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix} \\
&\overset{(6)}{=} 4\cdot 2\cdot 1\cdot 1\cdot(-1) = -8
\end{aligned}$$

Comments on the single steps:

(1) we use the row operation $V(1,2)$; the determinant changes by (-1) – we put another factor (-1) in front, so that altogether nothing changes.

(2) we use the row operationen $A(-2,1,3)$ und $A(-1,1,4)$; the determinant does not change.

(3) we use the row operationen $A(-3,2,3)$ und $A(-2,2,4)$; the determinant does not change.

(4) we use the row operation $M(-\frac{1}{4},3)$. The determinant of the new matrix is $-\frac{1}{4}$ times that of the old one. To undo that effect, we put another factor -4 in front, so that altogether nothing changes. The two minus signs cancel each other.

(5) we use the row operation $A(2,3,4)$; the determinant does not change.

(6) we calculate the determinant of an upper triangular matrix

1.2.10 Eigenvalues and eigenvectors

Let V be a vector space and $A\colon V \to V$ a linear map.

Question 1.2.41. Are there any $\lambda \in \mathbb{R}$, $v \in V$ such that $Av = \lambda v$? If yes, what are these λ and v?

Why would we want to know this? For many reasons, some of which will be explained below. Eigenvectors have meaning in physics, for example for calculating resonance patterns, see here for the maths. If one has a basis of eigenvectors for a linear map, one can base change and will get a diagonal matrix – which is very easy to compute with. This is for example how one solves linear differential equations. In probability theory, the equilibrium states of so-called Markov chains are examples of eigenvectors. Application of this abound in Machine Learning (and the Google page rank algorithm is also an example of this)! Still in this chapter we will see "Principal Component Analysis", a technique for what is called "dimensionality reduction" or "feature extraction" in data science.

Definition 1.2.42. *(i) $\lambda \in \mathbb{R}$ is called eigenvalue of A if there exist $v \neq 0 \in V$ such that $Av = \lambda v$.*

(ii) $v \in V$ is called eigenvector for the eigenvalue λ if $Av = \lambda v$.

Example 1.2.43.

$$\begin{pmatrix} 5 & 4 \\ 2 & 3 \end{pmatrix} \begin{pmatrix} 2 \\ 1 \end{pmatrix} = \begin{pmatrix} 2 \cdot 5 + 1 \cdot 4 \\ 2 \cdot 2 + 1 \cdot 3 \end{pmatrix} = \begin{pmatrix} 14 \\ 7 \end{pmatrix} = 7 \cdot \begin{pmatrix} 2 \\ 1 \end{pmatrix}$$

Hence $\lambda = 7$ is an eigenvalue of $A = \begin{pmatrix} 5 & 4 \\ 2 & 3 \end{pmatrix}$ and $v = \begin{pmatrix} 2 \\ 1 \end{pmatrix}$ is an eigenvector for the eigenvalue 7.

Proposition 1.2.44. *Let $f\colon V \to V$ be a linear map. The eigenvectors for a fixed eigenvalue λ of f form a subvector space.*

Proof. Clearly 0 is an eigenvector vor the eigenvalue λ since $A0 = 0 = \lambda 0$. We have to show that the sum of two eigenvectors v, v' for the eigenvalue λ is an eigenvector for λ again. Indeed, $f(v + v') = f(v) + f(v') = \lambda v + \lambda v' = \lambda(v + v')$. Also for the scalarproducts of an eigenvectors v with a scala μ we have $f(\mu v) = \mu f(v) = \mu \lambda v = \lambda \mu v$, so μv is an eigenvector again. □

How to find the eigenvalues of a matrix
By subtracting Av from the equation $Av = \lambda v$ one obtains the following equivalence

$$
\begin{aligned}
& Av = \lambda v = \begin{pmatrix} \lambda & & 0 \\ & \ddots & \\ 0 & & \lambda \end{pmatrix} v \\
\iff \quad & \left(\begin{pmatrix} \lambda & & 0 \\ & \ddots & \\ 0 & & \lambda \end{pmatrix} - A \right) \cdot v = \begin{pmatrix} \lambda & & 0 \\ & \ddots & \\ 0 & & \lambda \end{pmatrix} v - Av = 0.
\end{aligned}
$$

So λ is an eigenvalue precisely if $A - \lambda \cdot I$ has a nontrivial kernel, i.e. if

$$
\det \left(\begin{pmatrix} \lambda & & 0 \\ & \ddots & \\ 0 & & \lambda \end{pmatrix} - A \right) = 0.
$$

This equation can be solved for λ, since there is no v anymore.

Example 1.2.45. Finding the eigenvalues of $\begin{pmatrix} 5 & 4 \\ 2 & 3 \end{pmatrix}$:

$$
\begin{aligned}
\det \left(\begin{pmatrix} \lambda & 0 \\ 0 & \lambda \end{pmatrix} - \begin{pmatrix} 5 & 4 \\ 2 & 3 \end{pmatrix} \right) &= \det \begin{pmatrix} \lambda - 5 & -4 \\ -2 & \lambda - 3 \end{pmatrix} \\
&= (\lambda - 5)(\lambda - 3) - 2 \cdot 4 \\
&= \lambda^2 - 8\lambda + 7 \\
&\overset{!}{=} 0
\end{aligned}
$$

Solving this equation gives us the following eigenvalues

$$
\lambda_{1/2} = 4 \pm \sqrt{16 - 7} = 4 \pm \sqrt{9} = \begin{cases} 7 \\ 1 \end{cases} .
$$

When you calculate $\det(\lambda \cdot I - A)$ for an $n \times n$-matrix A, you get a polynomial of degree n in the variable λ. As it is customary, one rather chooses to denote the variable of a polynomial by x in the following definition:

Definition 1.2.46. *The characteristic polynomial of a matrix A is defined as*

$$
\chi_A(x) := \det(x \cdot I - A)
$$

Thus the roots of the characteristic polynomial χ_A are precisely the eigenvalues of A.

How to find eigenvectors
Now solve $Av = \lambda v$ for v (with λ that we have found). This is equivalent to finding the vs such that

$$\left(A - \begin{pmatrix} \lambda & & 0 \\ & \ddots & \\ 0 & & \lambda \end{pmatrix} \right) \cdot v = 0.$$

This is asking for a solution to a linear system of equations, so we know how to do it. In particular it follows that the set of eigenvectors for a fixed eigenvalue λ is a subvector space: It is the kernel of the linear operator $A - \lambda I$. One can ask for the dimension and for a basis of this subvector space.

Example 1.2.47.

$$\left(\begin{pmatrix} 5 & 4 \\ 2 & 3 \end{pmatrix} - \begin{pmatrix} 7 & 0 \\ 0 & 7 \end{pmatrix} \right) v = \begin{pmatrix} -2 & 4 \\ 2 & -4 \end{pmatrix} v = 0$$

I.e. solve

$$\begin{aligned} -2v_1 + 4v_2 &= 0 \\ 2v_16 + (-4)v_2 &= 0 \end{aligned}$$

A possible solution is $v_1 = 2$ and $v_2 = 1$. Hence $v = \begin{pmatrix} 2 \\ 1 \end{pmatrix}$ is an eigenvector for the eigenvalue $\lambda = 7$. The matrix describing the linear system has rank 1, and it is a linear system in 2 variables. By the dimension formula, its kernel, which is the space of solutions, has dimension $2 - 1 = 1$. Thus the vector we found forms a basis and all other eigenvectors are scalar multiples of this one.

Not every linear map has eigenvectors! For example the rotation by 90 degrees, clearly does not map any vector into the same direction.

Definition 1.2.48. *(i) A matrix $A = (a_{ij})$ is called a* diagonal matrix *if the only non-zero entries are on the diagonal, i.e. if $a_{ij} = 0$ whenever $i \neq j$.*

(ii) A matrix A is called diagonalizable *if there is an invertible matrix T such that $T^{-1}AT$ is a diagonal matrix.*

Theorem 1.2.49. *The following are equivalent for an $n \times n$-matrix A:*

(i) A is diagonalizable

(ii) There is a basis of $\mathbb{R}^n$ consisting of eigenvectors of A

(iii) The sum of the dimensions of the eigenspaces is n

A general application of eigenvalues and eigenvectors

Suppose we find a basis of eigenvectors $(u_1, \ldots, u_n) =: U$ for the eigenvalues $\lambda_1, \ldots, \lambda_n$. Then

$$A = U^{-1} \begin{pmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_n \end{pmatrix} U.$$

Diagonal matrices are easier to handle than general matrices,for example it is easier to compute their powers. This we can use to compute powers of general diagonalizable matrices:

$$A^k = \underbrace{U^{-1} \begin{pmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_n \end{pmatrix} U \cdots U^{-1} \begin{pmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_n \end{pmatrix} U}_{\text{k-times}} = U^{-1} \begin{pmatrix} \lambda_1^k & & \\ & \ddots & \\ & & \lambda_n^k \end{pmatrix} U.$$

This can be concretely applied in many situations, e.g. for solving differential equations by computing matrix exponentials, or for classifying the states of a Markov chain (see later in the course).

1.3 Geometry in $\mathbb{R}^n$

Definition 1.3.1. *The length of a vector $a = \begin{pmatrix} a_1 \\ \vdots \\ a_n \end{pmatrix} \in \mathbb{R}^n$ is defined as*

$$\left\| \begin{pmatrix} a_1 \\ \vdots \\ a_n \end{pmatrix} \right\| := \sqrt{a_1^2 + \ldots + a_n^2}.$$

(This definition comes from Pythagoras' theorem.)

Definition 1.3.2. *The dot product or standard scalar product is the map*

$$V \times V \longrightarrow \mathbb{R}\,,\ (a,b) \mapsto a^T \cdot b.$$

One writes $\langle a,b\rangle := a^T b$.

Note 1.3.3. $\left\| \begin{pmatrix} a_1 \\ \vdots \\ a_n \end{pmatrix} \right\|^2 = a^T \cdot a = a^T I a.$

Note 1.3.4. We note the following properties of the scalar product:

(i) $\langle a, \lambda b + \mu c\rangle = \lambda\langle a,b\rangle + \mu\langle a,c\rangle$

(ii) $\langle \lambda b + \mu c, d\rangle = \lambda\langle b,d\rangle + \mu\langle c,d\rangle$

(iii) $\langle a,b\rangle = \langle b,a\rangle$

(iv) $\langle a,a\rangle = 0$ if and only if $a = 0$

Theorem 1.3.5. $a^T b = \|a\| \cdot \|b\| \cdot \cos(\varphi)$

Remark 1.3.6. In particular, two vectors $a, b \in \mathbb{R}^n$ are orthogonal iff $a^T b = 0$.

Definition 1.3.7. *A square matrix A is called orthogonal if $A^T = A^{-1}$.*

Proposition 1.3.8. *A matrix is orthogonal if and only if its column vectors all have length* 1 *and are mutually orthogonal.*

Proof. Let $A = (a_1, \ldots, a_n)$ be an orthogonal matrix with column vectors $a_1, \ldots, a_n$. Then

$$\begin{pmatrix} \langle a_1,a_1\rangle & \cdots & \langle a_1,a_n\rangle \\ \vdots & \ddots & \vdots \\ \langle a_n,a_1\rangle & \cdots & \langle a_n,a_n\rangle \end{pmatrix} = A^T A = \begin{pmatrix} 1 & & 0 \\ & \ddots & \\ 0 & & 1 \end{pmatrix}.$$

The 1s on the diagonal tell us that we have length 1 vectors and the zeros elsewhere tell us about orthogonality.

Vice versa if the columns are mutually orthogonal vectors of length 1 then

$$A^T A = \begin{pmatrix} \langle a_1,a_1\rangle & \cdots & \langle a_1,a_n\rangle \\ \vdots & \ddots & \vdots \\ \langle a_n,a_1\rangle & \cdots & \langle a_n,a_n\rangle \end{pmatrix} = \begin{pmatrix} 1 & & 0 \\ & \ddots & \\ 0 & & 1 \end{pmatrix}.$$

So then A is an orthogonal matrix. □

Proposition 1.3.9. *Orthogonal matrices correspond precisely to those linear maps which preserve lengths and angles.*

Proof. Clearly a map preserving lengths and angles sends the standard basis to a set of orthogonal vectors of length 1. Thus such a map is given by an orthogonal matrix.

On the other hand, let A be an orthogonal matrix.

A preserves lengths: $\langle Av, Av\rangle = (Av)^T \cdot (Av) = v^T \underbrace{A^T A}_{=I} v = v^T v = \langle v, v\rangle$

A preserves angles: Let φ be the angle between Av and Av', ψ the angle between v and v'. Then

$$\cos(\varphi) = \frac{\langle Av, Av'\rangle}{\|Av\| \, \|Av'\|} = \frac{v^T A^T Av'}{\|v\| \, \|v'\|} = \frac{v^T v'}{\|v\| \, \|v'\|} = \cos(\psi).$$

□

1.4 Metrics, norms and inner products

We have seen that the scalar product, or dot product, can be used to measure the length of a vector by inserting it into both arguments (and taking the square root). This measures the length in the ususal sense of geometry as we know it. Sometimes, however, one wants to judge the importance of a vector in a different way, than by its usual geometric length.

The length of a vector is the distance between that vector and the origin. More generally, one may ask what is the distance between two points $a, b \in \mathbb{R}^n$. The usual geometric answer is that it is the length of the straight line segment between the two points, or equivalently, the length of $a - b$. Depending on the concrete situation, however, other measures of distance, or of "similarity", can make more sense.

A standard set of requirements on such a notion of distance are given in the notions of *pseudometric* and *metric*.

Definition 1.4.1. *A* pseudometric *on a set X is a function $d \colon X \times X \to \mathbb{R}_{\geq 0}$ to the non-negative real numbers, such that for all $x, y, z \in X$*

(i) $d(x, x) = 0$

(ii) $d(x, y) = d(y, x)$ *(symmetry)*

(iii) $d(x,z) \leq d(x,y) + d(y,z)$ *(triangle inequality)*

It is called a metric *if additionally one has*

(i) $d(x,y) = 0 \quad \Rightarrow \quad x = y$

There is a popular Python Machine learning library called `sklearn`, which contains a whole package around metrics `sklearn.metrics` – for examples of the above notion of metric see in particular the submodule `sklearn.metrics.pairwise-metrics`.

Examples 1.4.2. 1. The *taxicab metric* or *Manhattan metric* on $\mathbb{R}^2$ is defined as $d(x,y) := |x_1 - y_1| + |x_2 - y_2|$, i.e. as the sum of differences between their coordinates. It is the length of the shortest path between x and y if one is only allowed to move parallel to the axes.

2. According to the *river jungle metric* on $\mathbb{R}^2$, the distance between two points is the length of the shortest path between the two points, if one is only allowed to move vertically, or on the x-axis.

3. The angular distance, defined by $d_{\cos}(x,y) := 1 - \frac{1}{\pi}\cos^{-1}\frac{x^T y}{\|x\|\|y\|}$, is a pseudometric on $\mathbb{R}^n_{\geq 0} \setminus \{0\}$. Vectors with a small angle in between are considered close to each other, orthogonal vectors are considered far. This is not a metric, because different vectors pointing into the same direction have angular distance 0.

   The angular distance is used in natural language processing as a measure of similarity between documents: Given a vocabulary of n words, a document is assigned a vector in $\mathbb{R}^n$, whose i-th entry is the number of times that the i-th word occurs in the document. Longer documents give rise to larger vectors in the usual sense, but angles only depend on the ratio of the occurring words, not their quantity. Thus a small angular distance between two documents can be taken as an indicator that they talk about similar topics.

On vector spaces one often prefers metrics which behave the same in all regions of the space. One way to make this precise is the notion of *norm*:

Definition 1.4.3. *A norm on a vector space V is a function $\| - \| \colon V \to \mathbb{R}_{\geq 0}$ such that for all $v, w \in V$ and $\lambda \in \mathbb{R}$:*

(i) $\|\lambda v\| = |\lambda| \cdot \|\|$

(ii) $\|v+w\| \leq \|v\| + \|w\|$ *(triangle inequality)*

(iii) $\|v\| = 0 \;\Rightarrow\; v = 0$

Examples 1.4.4. 1. For a natural number p, the ℓ_p-norm on $\mathbb{R}^n$ is given, for $x = (x_1 \; \dots \; x_n)^T$ by $\|x\|_p := \sqrt[p]{\sum_{i=1}^n |x_i|^p}$. In particular the ℓ_2-norm is the usual norm of a vector. For p other than 1 or 2 it is somewhat tricky to verify that this is actually a norm.

2. The ℓ_∞-norm, or *maximum norm*, on $\mathbb{R}^n$ is given, for $x = (x_1 \; \dots \; x_n)^T$ by $\|x\|_\infty := \max\{|x_i|\}$

From a norm one can define a metric by setting $d(v,w) := \|v-w\|$. The metric coming from a norm satisfies $d(x-z, y-z) = \|x-z-(y-z)\| = \|x-z-y+z\| = \|x-y\| = d(x,y)$ – i.e. it does not change, if one shifts both vectors by the same vector.

Example 1.4.5. 1. The taxicab metric is the metric associated to the ℓ_1-norm.

2. The river jungle metric is not associated to any norm: shifting two vectors further away from the x-axis (e.g. upwards if they both are above the x-axis) increases their distance.

The nicest norms come from so-called inner products:

Definition 1.4.6. *An inner product on a vector space is a map $\langle -,- \rangle \colon V \to \mathbb{R}$ such that for all $a,b,c,d \in V$ and $\lambda, \mu in \mathbb{R}$:*

(i) $\langle a, \lambda b + \mu c \rangle = \lambda \langle a,b \rangle + \mu \langle a,c \rangle$

(ii) $\langle \lambda b + \mu c, d \rangle = \lambda \langle b,d \rangle + \mu \langle c,d \rangle$

(iii) $\langle a,b \rangle = \langle b,a \rangle$

(iv) $\langle a,a \rangle = 0$ *if and only if* $a = 0$

The norm associated to an inner product $\langle -,- \rangle$ is defined by $\|v\| := \sqrt{\langle v,v, \rangle}$.

Note that the standard scalar product satisfies all the properties of an inner product. A map $V \to \mathbb{R}$ satisfying properties (i) and (ii) is called a bilinear form. If it additionally satisfies property (iii), it is called *symmetric*. And if it satisfies property (iv), it is called non-degenerate, or *positive* *definite*. Thus an inner product is a symmetric, positive definite bilinear form.

From bilinearity it follows that an inner product is completely determined by the values it takes on pairs of basis vectors: For an inner product $\langle -, - \rangle$ (or actually any bilinear map) and for any basis $\{b_1, \ldots, b_n\}$ it is enough to know the values $\langle b_i, b_j \rangle$ in order to reconstruct the values for any pair of vectors.

These values can be taken as the coefficients of a matrix $A = (\langle b_i, b_j \rangle)$ – so this matrix completely describes the bilinear form. Clearly that the bilinear form is symmetric means that this matrix is symmetric.

If we write vectors as linear combinations of the basis vectors, $v = \lambda_1 b_1 + \ldots + \lambda_n b_n$ and $w = \mu_1 b_1 + \ldots + \mu_n b_n$, then one can see that $\langle v, w \rangle = v^T A w$.

Example 1.4.7. If we take the standard scalar product and the standard basis then clearly the matrix $A = (\langle e_i, e_j \rangle)$ is the unit matrix – the same would be true with any orthonormal basis.

Indeed, the standard scalar product can be seen as applying the general recipe to the unit matrix: $\langle a, b \rangle = a^T b = a^T I b$.

One can see general inner products as what one gets by replacing I in the standard scalar product by other matrices.

Definition 1.4.8.

- *A map of the type* $V \times V \longrightarrow \mathbb{R}$ *,* $(v, v') \mapsto v^T A v'$ *for some matrix* A *is called* bilinear form.
- *A map of the type* $V \longrightarrow \mathbb{R}$ *,* $v \mapsto v^T A v$ *for some matrix* A *is called* quadratic form

Examples 1.4.9. (a) the Lorentz form is the symmetric bilinear form given by the matrix

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix}$$

It is not positive definite.

(b) the Hesse matrix of a smooth function $f\colon \mathbb{R}^n \to \mathbb{R}$ at a point p, given by $(df/dx_i dx_j(p))_{ij}$ is a symmetric bilinear form. If it is positive definite and the gradient is zero, then one has a local minimum at p.

(c) the covariance matrix of a set of random variables (see later) is a symmetric bilinear form, whose value on a pair of linear combinations of these random variables is their covariance.

    The *inverse of the covariance matrix* gives rise to a bilinear form, and hence a metric, which one can see this as a kind of adjusted distance between the data points. A very nice visual explanation is given in this forum post

If one wants to express the inner product $\langle -, - \rangle$ given by a matrix A with respect to the standard basis, with respect to a new basis, one has to write the basis vectors of the new basis into a matrix U and form $U^T AU$: The matrix U transforms a vector of coefficients for the new basis into one of coefficients for the standard basis, and for those, the matrix A describes the given linear form:

Example 1.4.10. Consider the quadratic form given (in the standard basis) by the identity matrix (i.e. the usual squared length of a vector).

If we want to use the basis $\{\begin{pmatrix} 2 \\ 1 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix}\}$, i.e. if we we interpret a vector (a, b) as $a \begin{pmatrix} 2 \\ 1 \end{pmatrix} + b \begin{pmatrix} 0 \\ 1 \end{pmatrix}$, then we have to calculate

$$(a, b) \begin{pmatrix} 2 & 1 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 2 & 0 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} a \\ b \end{pmatrix} = (a, b) \begin{pmatrix} 5 & 1 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} a \\ b \end{pmatrix}$$

So the base changed matrix of the quadratic form is $\begin{pmatrix} 5 & 1 \\ 1 & 1 \end{pmatrix}$.

1.5 Diagonalization of symmetric matrices

Proposition 1.5.1. *Let A be a symmetric matrix. Then eigenvectors for different eigenvalues of A are orthogonal.*

Proof. Let λ, μ be two different eigenvalues of A, v eigenvector for λ and w eigenvector for μ. Then

$$\lambda\langle v, w\rangle = \langle \lambda v, w\rangle = \langle Av, w\rangle = v^T A^T w = v^T Aw = \langle v, Aw\rangle = \langle v, \mu w\rangle = \mu\langle v, w\rangle.$$

If the eigenvectors v, w are not othogonal, hence $\langle v, w\rangle \neq 0$ it would follow that $\lambda = \mu$. However, this is a contradiction to our assumption. □

In the next proof we will use fact that a matrix with real entries can also be seen as a matrix with complex entries, encoding a quadratic form or a linear map of $\mathbb{C}$-vector spaces.

For $\mathbb{C}$-matrices we have the operation $A \mapsto \bar{A}$ which conjugates every entry of a matrix. On $\mathbb{C}^n$ one has the standard complex scalar product $(v, w) \mapsto \langle v, w \rangle = \bar{v}^T \cdot w$. Note that in this way $\langle v, v \rangle$ is precisely the length of v seen as a vector in $\mathbb{R}^{2n}$. Also note that it satisfies $\langle \lambda v, w \rangle = \bar{\lambda} \langle v, w \rangle$.

Proposition 1.5.2. *Let A be a symmetric matrix. Then A has a real eigenvalue.*

Proof. The characteristic polynomial of A, $p_A(x) = det(A - x \cdot I)$ has a complex eigenvalue λ. Let v be an eigenvector for this λ. Then we have

$$\begin{aligned}\lambda\langle v, v \rangle = \langle v, \lambda v \rangle = \langle v, Av \rangle = \bar{v}^T A v &\overset{*}{=} \bar{v}^T \bar{A}^T v \\ &= (\bar{Av})^T v = \langle Av, v \rangle = \langle \lambda v, v \rangle = \bar{\lambda} \langle v, v \rangle\end{aligned}$$

Since by definition of eigenvector $v \neq 0$, we have $\langle v, v \rangle > 0$. Thus we can divide the above equation by $\langle v, v \rangle$, which yields $\lambda = \bar{\lambda}$. This means that λ is a real number. □

Theorem 1.5.3 (Principal axis theorem)**.** *Every quadratic form can be diagonalized by an orthogonal basis: For every real symmetric matrix A there is an orthogonal matrix U such that $U^T A U$ is a diagonal matrix.*

Proof. By Prop. 1.5.2 A has a real eigenvalue λ. Pick an eigenvector v, make it into an eigenvector of length 1 by taking $b_1 := \frac{v}{\|v\|}$, and complete it to an *orthonormal* basis $\{b_1, \ldots, b_n\}$. These vectors are the columns of the base change matrix for this basis, and hence the base change matrix $B = (b_1 | \ldots | b_n)$ is an orthogonal matrix.

The base change of A looks as follows:

$$BAB^{-1} = \left(\begin{array}{c|c} \lambda & \\ 0 & \\ \vdots & * \\ 0 & \end{array}\right)$$

This is because the first basis vector b_1 is an eigenvector, and hence mapped to λb_1, with no contributions from the other vectors. Now B is orthogonal, so $B = B^T$, and hence the base changed matrix is symmetric again:

$$(BAB^{-1})^T = (BAB^T)^T = (B^T)^T (BA)^T = BA^T B^T = BAB^{-1}$$

Hence it really looks as follows:

$$BAB^{-1} = \left(\begin{array}{c|ccc} \lambda & 0 & \cdots & 0 \\ \hline 0 & & & \\ \vdots & & D & \\ 0 & & & \end{array}\right)$$

Here D is a symmetric matrix too, and we can repeat the above process until we have a diagonal matrix. □

Remark 1.5.4. If you know about generalized eigenspaces, then there is also another smooth proof:

Remember that a *generalized eigenvector* for the eigenvalue λ is a vector v such that $(A - \lambda I)^k v = 0$ for some $k \in \mathbb{N}$. For every matrix A there is a basis of generalized eigenvectors of A (the base change with respect to this basis brings A into Jordan normal form).

Let now A be a symmetric matrix. To show that it is diagonalizable, we have to show that every generalized eigenvector is an actual eigenvector (which means that the Jordan normal form has no 1s next to the diagonal).

Suppose A is not diagonalizable. Then there are v and λ such that $(A - \lambda I)^2 v = 0$, but $(A - \lambda I)v \neq 0$. From this we obtain the following contradiction:

$$\begin{aligned} 0 &= \langle v, (A - \lambda I)^2 v\rangle = v^T(A - \lambda I)(A - \lambda I)v = v^T(A^T - \lambda I^T)(A - \lambda I)v \\ &= v^T(A - \lambda I)^T(A - \lambda I)v = \langle (A - \lambda I)v, (A - \lambda I)v\rangle > 0 \end{aligned}$$

Here we used that transposition respects matrix sums and hence $(A - \lambda I)^T = A^T - \lambda I^T = A - \lambda I$.

1.6 Princial Component Analysis

Principal Component Analysis (PCA) is a technique for dimensionality reduction. Suppose we are given a collection of data, in the form of k points

$$x^{(1)} = \begin{pmatrix} x_1^{(1)} \\ \vdots \\ x_n^{(1)} \end{pmatrix}, \ldots, \quad x^{(k)} = \begin{pmatrix} x_1^{(k)} \\ \vdots \\ x_n^{(k)} \end{pmatrix} \in \mathbb{R}^n.$$

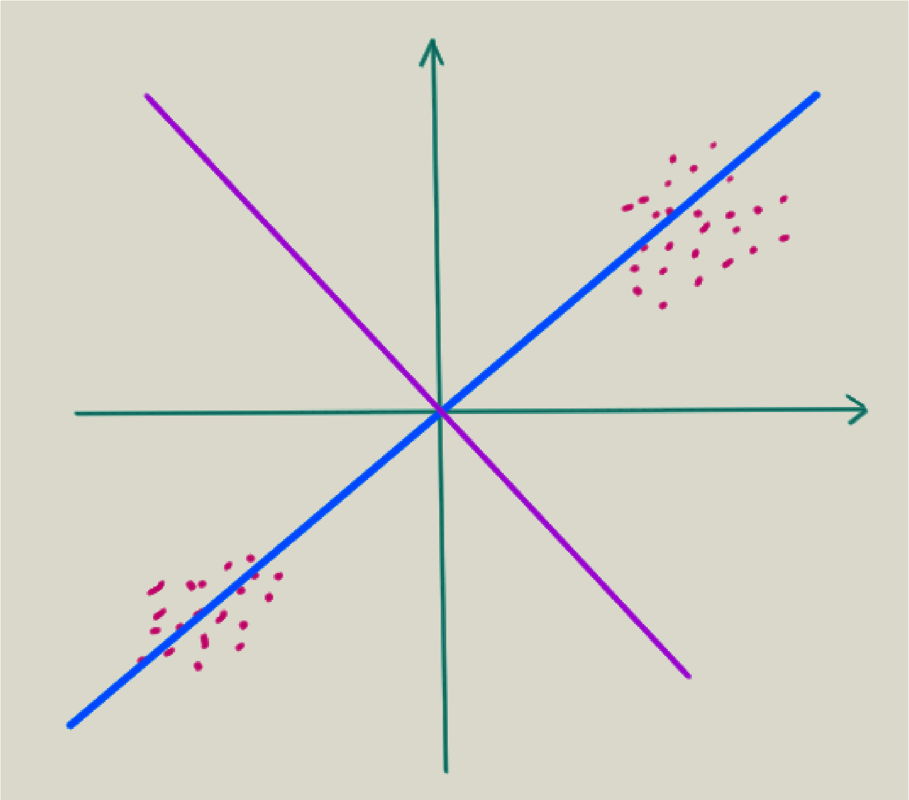
You can imagine that each vector $x^{(i)}$ corresponds to one experiment, or object of study – e.g. a patient in a clinical trial – and each entry $x_j^{(i)}$ corresponds to one measurement taken from that experiment or object of study – e.g. blood pressure, oxygen level, etc.

Then we can ask whether there is a lower dimensional subspace of $\mathbb{R}^n$ to which we can project the points, such that we can still read off most of the relevant information.

There are several possible reasons for which one might want to pass to a lower dimensional subspace, for example

1. Projecting the data down to dimensions 2 or 3 for visualization.
2. Lowering the dimension to make computations feasible
3. Guessing a single missing value for a vector
4. Finding out which linear combinations of the measurements are relevant for the question at hand (feature extraction) – more on this below.

In the following picture a data collection is illustrated as a point cloud, forming two clusters. Imagine that you want to want to project down to a subvector space in a way that still allows you to distinguish these two clusters. Then the subvector space given by the blue line clearly is better suited for this (since the projections of the two different clusters still are far apart) than the projection to the purple subvector space (since there we would only see one mixed cloud):



Principal axes for a point cloud

Example 1.6.1 (Feature extraction)**.** To illustrate the point of "feature extraction" mentioned in the above list, imagine that we run a medical trial and measure calory consumption and daily sport activity of patients. There could appear two clusters, one of people having a heart disease, one of healthy people. Now the principal axis, i.e. the line the projection to which would best distinguish the two clusters, could be in the direction of "calory consumption"-"sports hours": A high index here means eating much and moving little; a dangerous lifestyle for your cardiovascular system.

The orthogonal axis would then be "calory consumption"+"sports hours". Moving along this axis does not change your risk of a heart disease; e.g. eating more while also moving more is ok.

This (crudely) illustrates how the principal axis of a data cloud can tell you what are the relevant linear combinations of your measured data that can best tell you what you want to predict.

The question is now, how to find the optimal subspace for projecting. We start by recalling/introducing some basic notions from statistics:

Definition 1.6.2. *Consider vectors $x^{(1)}, \dots, x^{(k)} \in \mathbb{R}^n$.*

1. *The* mean value *of the r-th coordinate is* $\bar{x}_r := \frac{1}{k}\sum_{i=1}^{k} x_r^{(i)}$
2. *The* empirical covariance *of the r-th and the s-th coordinates is* $\mathrm{Cov}(r,s) := \frac{1}{k-1}\sum_{i=1}^{k}(x_r^{(i)} - \bar{x}_r)(x_s^{(i)} - \bar{x}_s)$[1]
3. *The empirical covariance of r-th coordinate with itself is also called the* empirical variance *of the r-th coordinate and is denoted by* $\mathrm{Var}(r) := \mathrm{Cov}(r,r) = \frac{1}{k-1}\sum_{i=1}^{k}(x_r^{(i)} - \bar{x}_r)^2$
4. *The $n \times n$-matrix $C = (c_{ij})$ with $c_{ij} = \mathrm{Cov}(i,j)$ is called the* empirical covariance matrix, *or sometimes simply the* covariance matrix.

The covariance, roughly, measures whether if the r-th value goes up, the s-th value tends to go up (positive covariance) or down (negative covariance). The variance measures how far the values are spread around their mean value.

As our goal is to find a lower dimensional vector space in which the data lies, the data should better be centered around the origin, because any vector space has to contain the origin. We can ensure that by subtracting from every vector the mean vector $(\bar{x}_1, \dots, \bar{x}_n)$.

The role of the covariance matrix is that of a bilinear form. It is obviously symmetric, and one can also show that it is positive definite, so it is in fact an inner product.

The number obtained by applying this inner product to two vectors $a = \begin{pmatrix} a_1 \\ \vdots \\ a_n \end{pmatrix}, b = \begin{pmatrix} b_1 \\ \vdots \\ b_n \end{pmatrix}$ has the following interpretation:

Using the vector a, from each vector $x^{(i)}$ we can extract a number by using the entries a_i as coefficients in a linear combination of the entries of $x^{(i)}$, namely $x_a^{(i)} := a^T x^{(i)} = \sum_{r=1}^{n} a_r x_r^{(i)}$. This sequence of numbers $x_a^{(1)}, \dots, x_a^{(k)}$ has a mean value $\bar{x}_a := \frac{1}{k}\sum_{i=1}^{k} x_a^{(i)}$.

Likewise, from b we have numbers $x_b^{(i)}$ and a mean value $\bar{x}_b$.

Now we can form the empirical covariance of these two sequences of numbers: $\mathrm{Cov}(a,b) := \sum_{i=1}^{k}(x_a^{(i)} - \bar{x}_a)(x_b^{(i)} - \bar{x}_b)$

[1] Maybe you know this definition with the factor $\frac{1}{k}$ instead of $\frac{1}{k-1}$ – for large k the factor $\frac{1}{k}$ is also ok...

Fact 1.6.3. $\text{Cov}\colon \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$, $(a,b) \mapsto \text{Cov}(a,b)$ is a bilinear form whose matrix is exactly the covariance matrix C. To see this, one simply has to verify the equation $\text{Cov}(a,b) := \sum_{i=1}^{k}(x_a^{(i)} - \bar{x}_a)(x_b^{(i)} - \bar{x}_b) = a^T C b$

Now since the covariance matrix is symmetric, from the Principal axis theorem Thm. 1.5.3 we know that there exists an orthogonal basis of eigenvectors for it. Performing the base change with respect to this basis results in a diagonal matrix – it is still the covariance matrix, but now of different features.

The different coordinates in the new coordinate system given by this basis have covariance 0 – they are uncorrelated. Their variances are the numbers on the diagonal.

"PCA" means finding this basis and the numbers on the diagonal.

For visualization or dimensionality reduction, we can now choose those vectors belonging to the biggest eigenvalues – these are the directions in which we have the biggest variance, i.e. in which we can distinguish best. The process of finding and choosing such vectors is also called *feature extraction*.

PCA overview:

Start with given data $x^{(1)}, \ldots, x^{(k)} \in \mathbb{R}^n$.

1. Compute the mean value $\bar{x}_r$ of each coordinate. Assemble these numbers into the *mean vector* $\bar{x}$.
2. Center your data: Replace $x^{(1)}, \ldots, x^{(k)}$ by $x^{(1)} - \bar{x}, \ldots, x^{(k)} - \bar{x}$
3. Take the centered data vectors as columns of a data matrix D. Compute the covariance matrix $C = \frac{1}{k-1} D D^T$.
4. Compute the eigenvalues of C
5. For the biggest few eigenvalues, compute the eigenvectors.
6. Project your data to the subspace spanned by those eigenvectors.

Example 1.6.4. Consider the following four points in $\mathbb{R}^2$:

$$\begin{pmatrix}1\\1\end{pmatrix}, \begin{pmatrix}0\\-1\end{pmatrix}, \begin{pmatrix}2\\1\end{pmatrix}, \begin{pmatrix}-1\\-1\end{pmatrix}$$

We perform PCA to find the 1-dimensional subspace of $\mathbb{R}^2$ in which the projections of the above points have the biggest variance.

First we compute the mean of each coordinate:

$$\bar{x}_1 = \frac{1}{4}(1+0+2+(-1)) = \frac{1}{2}$$
$$\bar{x}_2 = \frac{1}{4}(1+(-1)+1+(-1)) = 0$$

Now we "center the data", i.e. we subtract the mean vector from each of the above data sample vectors:

$$\begin{pmatrix} \frac{1}{2} \\ 1 \end{pmatrix}, \begin{pmatrix} -\frac{1}{2} \\ -1 \end{pmatrix}, \begin{pmatrix} \frac{3}{2} \\ 1 \end{pmatrix}, \begin{pmatrix} -\frac{3}{2} \\ -1 \end{pmatrix}$$

This last step is not necessary. One can also simply calculate the covariance matrix coefficient by coefficient, using the definition of (co)variance, but centering is implicit in the definition of covariance.

With centered data, the mean of each coordinate is 0, and so the formula for covariance becomes

$$\mathrm{Cov}(r,s) := \frac{1}{k-1}\sum_{i=1}^{k}(x_r^{(i)} - \bar{x}_r)(x_s^{(i)} - \bar{x}_s) = \frac{1}{k-1}\sum_{i=1}^{k}(x_r^{(i)})(x_s^{(i)})$$

This last expression also occurs in the following way: If one forms the so-called data matrix, whose columns are the given sample vectors,

$$D := \begin{pmatrix} \frac{1}{2} & -\frac{1}{2} & \frac{3}{2} & -\frac{3}{2} \\ 1 & -1 & 1 & -1 \end{pmatrix}$$

and calculates $D \cdot D^T$ then one sees that $\mathrm{Cov}(r,s)$ exactly the entry at the position (r,s) of that matrix - up to the factor of $\frac{1}{k-1}$ (here $\frac{1}{3}$).

We thus obtain the covariance matrix

$$C := \frac{1}{3}D \cdot D^T = \frac{1}{3}\begin{pmatrix} \frac{1}{2} & -\frac{1}{2} & \frac{3}{2} & -\frac{3}{2} \\ 1 & -1 & 1 & -1 \end{pmatrix}\begin{pmatrix} \frac{1}{2} & 1 \\ -\frac{1}{2} & -1 \\ \frac{3}{2} & 1 \\ -\frac{3}{2} & -1 \end{pmatrix} = \frac{1}{3}\begin{pmatrix} 5 & 4 \\ 4 & 4 \end{pmatrix}$$

Now we determine the eigenvalues of this matrix. We can leave out the factor $\frac{1}{3}$ in front, because it changes each eigenvalue by exactly that factor and does not change the eigenvectors at all.

$$\chi_{3C}(x) = \det\begin{pmatrix} x-5 & -4 \\ -4 & x-4 \end{pmatrix} = (x-5)(x-4) - 16 = x^2 - 9x + 4$$

Thus eigenvalues are

$$x_{1/2} = \frac{9}{2} \pm \sqrt{\frac{81}{4} - 4} = \frac{9}{2} \pm \sqrt{\frac{81}{4} - \frac{16}{4}} = \frac{9 \pm \sqrt{65}}{2}.$$

The bigger of these eigenvalues is $\frac{9+\sqrt{65}}{2}$, so the projection to the eigenspace for this eigenvalue will have the biggest variance.

So we look for an eigenvector for the eigenvalue $\frac{9+\sqrt{65}}{2}$. We need to solve the system of linear equations

$$\begin{pmatrix} 5 & 4 \\ 4 & 4 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \frac{9 + \sqrt{65}}{2} \begin{pmatrix} x \\ y \end{pmatrix}$$

Since we are looking at an eigenvalue, the space of solutions is at least 1-dimensional, so one of the equations must be redundant and it suffices to look, e.g., at the first equation

$$5x + 4y = \frac{9 + \sqrt{65}}{2} x$$

This implies $y = \frac{-1+\sqrt{65}}{8} x$, and inserting, for example, $x = 1$ we obtain the eigenvector

$$\begin{pmatrix} 1 \\ \frac{-1+\sqrt{65}}{8} \end{pmatrix}$$

Thus the subvector space we were looking for is the space spanned by this vector. One could now proceed further, calculate the projections of the given sample vectors to this subvector space and see how much the variance reduced, but this would be even more tedious and is better done in Python.

Here is a beautiful forum post about PCA, with great animations.

A common question is why we have to center the data. A short answer is: We don't. We can simply compute the covariance matrix (which involves subtracting the mean, i.e. centering, on the way) and look for its eigenvectors.

1.7 Singular value decomposition

Theorem 1.7.1. *Let A be an $n \times m$-matrix. Then there are an orthogonal $n \times n$-matrix, a diagonal $n \times m$-matrix Σ and an orthogonal $m \times m$-matrix V such that*

$A = U\Sigma V^T$. The diagonal matrix can be chosen such that its entries are all non-negative and ordered by size with the biggest value in the upper left corner.

Proof. Consider the matrix A^TA. It is a symmetric $m \times m$-matrix, thus by Theorem 1.5.3 there is an orthogonal $m \times m$-matrix V such that

$$A^TA = V \begin{pmatrix} \lambda_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \lambda_m \end{pmatrix} V^T,$$

where the λ_i are the eigenvalues of A^TA. They are all non-negative: If v is an eigenvector for λ_i, then

$$\lambda_i\langle v, v\rangle = \langle v, \lambda_i v\rangle = \langle v, A^TAv\rangle = v^TA^TAv = \langle Av, Av\rangle \geq 0.$$

We can arrange the order of the λ_i such that $\lambda_i \geq \lambda_{i+1}$ by conjugating with permutation matrices. In particular all zeros then occur at the lower right of the diagonal.

By Prop. 1.3.8, the columns of V are orthogonal, and they are eigenvectors of A^TA. The images under A of these columns are also orthogonal: Let v_i, v_j be the ith and jth column vectors of V. Then

$$\langle Av_i, Av_j\rangle = (Av_i)^TAv_j = v_i^TA^TAv_j = v_i(\lambda_j v_j) = \lambda_j\langle v_i, v_j\rangle.$$

If $i \neq j$ then this last expression is 0, and so Av_i and Av_j are orthogonal. If $i = j$, then the last expression is λ_i (since $\langle v_i, v_i\rangle = 1$), i.e. $\|Av_i\|^2 = \lambda_i$ and so Av_i is a vector of length $\sqrt{\lambda_i}$.

So $\{Av_1, \ldots, Av_m\}$ is a set of orthogonal vectors in $\mathbb{R}^n$. Some of the Av_i could be zero – these we leave out, obtaining a set $\{Av_1, \ldots, Av_r\}$, where $r = rank\, A = rank\, A^TA$. We adjust the vector in this set to have length 1, obtaining the set $\{\frac{Av_1}{\|Av_1\|}, \ldots, \frac{Av_r}{\|Av_r\|}\}$ of orthonormal vectors. Finally we complete this set to an orthonormal basis of $\mathbb{R}^n$, and we define U to be the orthogonal matrix whose columns u_i are given by the vectors of this basis.

Setting Σ to be the $n \times m$-matrix with diagonal entries the non-zero $\|Av_i\| = \sqrt{\lambda_i}$ and zeros elsewhere, We now have, by construction, the equation $AV = U\Sigma$. To see this, it suffices to check that this equation holds after multiplication with the standard basis vectors of $\mathbb{R}^m$:

$$Ave_i = Av_i = \|Av_i\|\frac{Av_i}{\|Av_i\|} = \|Av_i\|u_i = U(\|Av_i\| \cdot e_i) = U\Sigma e_i$$

Finally, by multiplying from the right with $V^T = V^{-1}$ we obtain the decomposition $A = U\Sigma V^T$ from the claim. □

Definition 1.7.2. *A decomposition $A = U\Sigma V^T$ as in Theorem 1.7.1 is called a* singular value decomposition *of A.*

Remark 1.7.3. The matrix Σ in a singular value decomposition of A is uniquely determined by A. Its diagonal entries are called the *singular values* of A, and are denoted by $\sigma_1, \ldots, \sigma_r$.

The column vectors of V (i.e. the row vectors of V^T) are called *right singular vectors*, and the column vectors of U are called *left singular vectors*.

The matrices U and V, on the other hand, are *not* uniquely determined by A: It was visible from the proof of Theorem 1.7.1 that in the construction of U we had no choice regarding the non-zero columns of the form Av_i, but then arbitrarily completed these to a basis of $\mathbb{R}^n$.

Also, if some of the singular values of A are zero, then one has free choice of some of the column vectors of V. As an extreme case just take A to be the zero matrix, then Σ is the zero matrix and U and V can be chosen arbitrarily among orthogonal matrices of the right size.

Here you can find more details on how uniquely the matrices U and V are determined.

Examples 1.7.4. 1. $\begin{pmatrix} -5 & 0 \\ 0 & 2 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 5 & 0 \\ 0 & 2 \end{pmatrix} \begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix}$

2. $$\begin{pmatrix} 2 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 5 \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} 5 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}$$

3. $$\begin{pmatrix} \sqrt{2} + \frac{3}{2} & \frac{3}{\sqrt{2}} & 2 \\ \frac{3}{2} & \frac{3}{\sqrt{2}} & 0 \end{pmatrix} = \begin{pmatrix} \frac{1}{\sqrt{2}} & 1 \\ \frac{1}{\sqrt{2}} & 0 \end{pmatrix} \begin{pmatrix} 3 & 0 & 0 \\ 0 & 2 & 0 \end{pmatrix} \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \\ \frac{1}{\sqrt{2}} & 0 & \frac{1}{\sqrt{2}} \\ 0 & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{pmatrix}$$

SVD overview

We record the algorithm for finding an SVD of a given $n \times m$-matrix A, that can be extracted from the proof of Theorem 1.7.1:

1. Calculate $A^T A$

2. Find eigenvalues $\lambda_1, \ldots, \lambda_m$ and eigenvectors $v_1, \ldots, v_m$ of $A^T A$ of length 1

3. Possibly permute the eigenvectors to order the eigenvalues by size
4. set $V := (v_1, \dots, v_m)$ and $\Sigma := diag_{n\times m}(\sqrt{\lambda_1}, \dots, \sqrt{\lambda_{min\{m,n\}}})$
5. take the non-zero Av_i, resize and complete them to an orthonormal basis of $\mathbb{R}^n$
6. Set U to be matrix whose column vectors are these basis vectors

Example 1.7.5. For an example of a SVD computation see for example Deisenroth, Faisal, Ong: Mathematics for Machine Learning, Example. 4.13 (page 125)

Example 1.7.6. The completion to an orthonormal basis in step 5 only has to be done when the matrix in question has more rows than columns. *However, for such a matrix A one can avoid the completion by simply forming the singular value decomposition of A^T and in the end transposing the result!*

Nevertheless, to give an example we compute the singular value decomposition of

$$A := \begin{pmatrix} 3 & -2 \\ 1 & 1 \\ 2 & 3 \end{pmatrix}$$

Following the algorithm we compute

$$A^TA = \begin{pmatrix} 3 & 1 & 2 \\ -2 & 1 & 3 \end{pmatrix} \begin{pmatrix} 3 & -2 \\ 1 & 1 \\ 2 & 3 \end{pmatrix} = \begin{pmatrix} 14 & 1 \\ 1 & 14 \end{pmatrix}$$

We compute the eigenvalues of A^TA. The characteristic polynomial is this:

$$\chi_{A^TA}(x) = \det \begin{pmatrix} x-14 & -1 \\ -1 & x-14 \end{pmatrix} = (x-14)^2 - 1 = x^2 - 28x + 195$$

The eigenvalues are the roots of this polynomial, i.e.

$$14 \pm \sqrt{14^2 - 195} = 14 \pm 1 = \begin{cases} 15 \\ 13 \end{cases}$$

At this point we already know the singular values: They are the square roots of the above eigenvalues, namely $\sqrt{15}$ and $\sqrt{13}$.

We compute the eigenvectors for the eigenvalue 15:
An eigenvector for the eigenvalue 15 is the same thing as an element of

$$\begin{aligned}\ker(15 \cdot I - A^T A) &= \ker \begin{pmatrix} 15-14 & -1 \\ -1 & 15-14 \end{pmatrix} = \ker \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix} = \ker \begin{pmatrix} 1 & -1 \\ 0 & 0 \end{pmatrix} \\ &= \left\{ \begin{pmatrix} x \\ y \end{pmatrix} \middle| \, x = y \right\}\end{aligned}$$

Since we want to form an orthogonal matrix, we need to choose an eigenvector of norm 1, e.g. $v_1 := (\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}})^T$ (we also could have chosen $-v_1$, but in a 1-dimensional eigenspace there is not more choice than that).

Similarly, an eigenvector for the eigenvalue 13 is the same thing as an element of

$$\begin{aligned}\ker(13 \cdot I - A^T A) &= \ker \begin{pmatrix} 13-14 & -1 \\ -1 & 13-14 \end{pmatrix} = \ker \begin{pmatrix} -1 & -1 \\ -1 & -1 \end{pmatrix} = \ker \begin{pmatrix} -1 & -1 \\ 0 & 0 \end{pmatrix} \\ &= \left\{ \begin{pmatrix} x \\ y \end{pmatrix} \middle| \, x = -y \right\}\end{aligned}$$

An element of norm 1 of this set is for example $v_2 := (\frac{1}{\sqrt{2}}, -\frac{1}{\sqrt{2}})^T$. The matrix V has the vectors v_1, v_2 as columns, and the matrix V^T therefore has them as rows:

$$V^T = \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{pmatrix}$$

Now we compute the images of v_1, v_2 under A:

$$Av_1 = \begin{pmatrix} 3 & -2 \\ 1 & 1 \\ 2 & 3 \end{pmatrix} \begin{pmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{pmatrix} = \begin{pmatrix} \frac{1}{\sqrt{2}} \\ \frac{2}{\sqrt{2}} \\ \frac{5}{\sqrt{2}} \end{pmatrix}$$

$$Av_2 = \begin{pmatrix} 3 & -2 \\ 1 & 1 \\ 2 & 3 \end{pmatrix} \begin{pmatrix} \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} \end{pmatrix} = \begin{pmatrix} \frac{5}{\sqrt{2}} \\ 0 \\ -\frac{1}{\sqrt{2}} \end{pmatrix}$$

We start building an orthonormal basis using these vectors. They already are orthogonal (necessarily, see the proof of the singular value decomposition), but we still have to bring them down to size 1. Their norms currently are their respective singular values (again by construction, see the proof): $\|Av_1\| = \sqrt{15}$ and $\|Av_2\| = \sqrt{13}$, so we set

$$u_1 := \frac{1}{\sqrt{15}} A v_1 = \begin{pmatrix} \frac{1}{\sqrt{30}} \\ \frac{2}{\sqrt{30}} \\ \frac{5}{\sqrt{30}} \end{pmatrix}$$

$$u_2 := \frac{1}{\sqrt{13}} A v_2 = \begin{pmatrix} \frac{5}{\sqrt{26}} \\ 0 \\ -\frac{1}{\sqrt{26}} \end{pmatrix}$$

We now need to complete $\{u_1, u_2\}$ to an orthonormal basis of $\mathbb{R}^3$. For this choose any vector that completes this set to a basis, then subtract the projections to u_1, u_2 to make the three vectors orthogonal, finally bring the resulting vector to norm 1 (this is called the Gram-Schmidt process):

We choose the vector $e_2 = (0,1,0)^T$ to complete $\{u_1, u_2\}$ to a basis (you can check that $\{u_1, u_2, e_2\}$ is a basis e.g. by calculating the determinant of the matrix with these vectors as columns).
Now we form

$$\begin{aligned} u &:= e_2 - \langle e_2, u_1 \rangle u_1 - \langle e_2, u_2 \rangle u_2 \\ &= \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} - \frac{2}{\sqrt{30}} \begin{pmatrix} \frac{1}{\sqrt{30}} \\ \frac{2}{\sqrt{30}} \\ \frac{5}{\sqrt{30}} \end{pmatrix} - 0 \begin{pmatrix} \frac{1}{\sqrt{30}} \\ \frac{2}{\sqrt{30}} \\ \frac{5}{\sqrt{30}} \end{pmatrix} = \begin{pmatrix} -\frac{2}{30} \\ 1 - \frac{4}{30} \\ -\frac{10}{30} \end{pmatrix} = \begin{pmatrix} -\frac{2}{30} \\ \frac{26}{30} \\ -\frac{10}{30} \end{pmatrix} \end{aligned}$$

The norm of u is $\|u\| = \sqrt{\frac{2^2+26^2+10^2}{30^2}} = \frac{\sqrt{780}}{30}$, so we finally set

$$u_3 := \frac{1}{\|u\|} u = \frac{30}{\sqrt{780}} u = \begin{pmatrix} -\frac{2}{\sqrt{780}} \\ \frac{26}{\sqrt{780}} \\ -\frac{10}{\sqrt{780}} \end{pmatrix}$$

We have obtained the singular value decomposition

$$A = \begin{pmatrix} \frac{1}{\sqrt{30}} & \frac{5}{\sqrt{26}} & -\frac{2}{\sqrt{780}} \\ \frac{2}{\sqrt{30}} & 0 & \frac{26}{\sqrt{780}} \\ \frac{5}{\sqrt{30}} & -\frac{1}{\sqrt{26}} & -\frac{10}{\sqrt{780}} \end{pmatrix} \begin{pmatrix} \sqrt{15} & 0 \\ 0 & \sqrt{13} \\ 0 & 0 \end{pmatrix} \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{pmatrix}$$

Remark 1.7.7. A singular value decomposition is not unique. This is clear if one considers the SVD of the $n \times m$-matrix where all entries are zero: The diagonal matrix will be the zero matrix, and therefore any orthogonal matrices can be chosen for U and V.

Some sources for non-uniqueness that can be seen in the algorithm are these:

1. Column vectors of V that are in the kernel of A could be replaced by other vectors in the kernel (as long as they still are part of an orthonormal basis).

2. If AA^T has an eigenspace of dimension > 1, then there will be several choices of orthonormal basis for it.

3. If the images Av_i do not span all of the target vector space, then one has to complete the vectors $\frac{1}{\|Av_i\|}Av_i$ to an orthonormal basis, and there may be a lot choices for this.

One can show that these are essentially the only sources of ambiguity in the singular value decomposition. In particular the diagonal matrix Σ is completely determined by A. This also justifies calling the diagonal entries of Σ "the singular values of A".

Remark 1.7.8. The left hand matrix U in a singular value decomposition $A = U\Sigma V^T$ can be chosen by rescaling the images Av_i of the vectors v_i under A and completing the results to a basis. If you don't like having to choose anything, this last step can be replaced as follows:

A singular value decomposition of A^T is given by $A^T = (U\Sigma V^T)^T = V\Sigma^T U^T$. So to find a matrix U that fits on the left hand side of a singular value decomposition $A = U\Sigma V^T$, you can do the algorithm for A^T instead, i.e. find an orthonormal basis of eigenvectors of the symmetric matrix AA^T and take those as the column vectors of an orthogonal matrix U.

Now if we already had V obtained by forming a basis of eigenvectors of A^TA one should ask whether we can be sure that *this* U and *this* V really fit together into a single SVD $A = U\Sigma V^T$. The answer is: Where we had to choose bases of eigenspaces of dimension > 1, item 2. of Remark 1.7.7, the choices do not necessarily fit together.

However, this is the only possible problem. If we only had eigenspaces of dimension 1 as is usually the case, then U and V will fit together, possibly after multiplying some vectors with -1. This would follow from a more precise consideration of the sources of non-uniqueness mentioned in Remark 1.7.7. The other two sources of non-uniqueness, changing things in the kernel or completing a list of vectors to a basis will not affect the product of the three matrices.

Remark 1.7.9. For a matrix A of rank r, let

$$v^{(1)} = \begin{pmatrix} v_1^{(1)} \\ \vdots \\ v_n^{(1)} \end{pmatrix}, \dots, v^{(n)} = \begin{pmatrix} v_1^{(n)} \\ \vdots \\ v_n^{(n)} \end{pmatrix}$$

be the right singular vectors and

$$u^{(1)} = \begin{pmatrix} u_1^{(1)} \\ \vdots \\ u_m^{(1)} \end{pmatrix}, \dots, u^{(n)} = \begin{pmatrix} u_1^{(m)} \\ \vdots \\ u_m^{(m)} \end{pmatrix}$$

the left singular vectors of a singular value decomposition of A.

The SVD equation $A = U\Sigma V^T$ can also be read as $A = \sum_{i=1}^{r} \sigma_i u_i v_i^T$. Each of these summands is an $m \times n$-matrix of rank 1, so one can understand the SVD as a representation of A as a sum of rank 1 matrices.

To understand this, first note that we can decompose the matrix V^T as the sum of its rows:

$$V^T = \begin{pmatrix} v_1^{(1)} & \cdots & v_n^{(1)} \\ 0 & & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 0 \end{pmatrix} + \begin{pmatrix} 0 & & 0 \\ v_1^{(2)} & \cdots & v_n^{(2)} \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 0 \end{pmatrix} + \dots + \begin{pmatrix} 0 & \cdots & 0 \\ 0 & & 0 \\ \vdots & \ddots & \vdots \\ v_1^{(1)} & \cdots & v_n^{(1)} \end{pmatrix}$$

To compute $A = U\Sigma V^T$, we can multiply $U\Sigma$ with each of the above summands and then sum the results. The first summand gives

$$\begin{pmatrix} u_1^{(1)} & \cdots & u_1^{(m)} \\ u_2^{(1)} & & u_2^{(m)} \\ \vdots & \ddots & \vdots \\ u_m^{(1)} & \cdots & u_m^{(m)} \end{pmatrix} \begin{pmatrix} \sigma_1 & 0 & \cdots & & 0 & 0 \\ 0 & \sigma_2 & & & 0 & 0 \\ \vdots & & \ddots & & \vdots & \vdots \\ 0 & & & \sigma_r & 0 & 0 \\ \vdots & \cdots & & & 0 & 0 \\ \vdots & \cdots & & & 0 & 0 \\ 0 & 0 & \cdots & & 0 & 0 \end{pmatrix} \begin{pmatrix} v_1^{(1)} & \cdots & v_n^{(1)} \\ 0 & & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 0 \end{pmatrix}$$

$$= \begin{pmatrix} u_1^{(1)} & \cdots & u_1^{(m)} \\ u_2^{(1)} & & u_2^{(m)} \\ \vdots & \ddots & \vdots \\ u_m^{(1)} & \cdots & u_m^{(m)} \end{pmatrix} \begin{pmatrix} \sigma_1 v_1^{(1)} & \cdots & \sigma_1 v_n^{(1)} \\ 0 & & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 0 \end{pmatrix}$$

$$= \begin{pmatrix} \sigma_1 v_1^{(1)} u_1^{(1)} & \sigma_1 v_2^{(1)} u_1^{(1)} & \cdots & \sigma_1 v_n^{(1)} u_1^{(1)} \\ \sigma_1 v_1^{(1)} u_2^{(1)} & & & \sigma_1 v_n^{(1)} u_2^{(1)} \\ \vdots & & \ddots & \vdots \\ \sigma_1 v_1^{(1)} u_m^{(1)} & \cdots & \sigma_1 v_{n-1}^{(1)} u_m^{(1)} & \sigma_1 v_n^{(1)} u_m^{(1)} \end{pmatrix}$$

$$= \sigma_1 \begin{pmatrix} u_1^{(1)} \\ \vdots \\ u_m^{(1)} \end{pmatrix} \begin{pmatrix} v_1^{(1)} & \cdots & v_n^{(1)} \end{pmatrix} = \sigma_1 u^{(1)} v^{(1)\,T}$$

Likewise, the i-th summand gives $\sigma_i u^{(i)} v^{(i)\,T}$. Summing up all these results gives the claimed equation.

In applications the singular vectors sometimes aquire concrete meanings. See Deisenroth, Faisal, Ong: Mathematics for Machine Learning, Figure 4.10 and Example 4.14 (page 127) for a very good example involving movie ratings. Another example of application are word vector embeddings in natural language processing.

Low rank approximation by Singular Value Decompositions

The SVD allows to approximate a matrix by another matrix of the same size but of lower rank. This can reduce the complexity of computations and memory usage, and it can also help with clustering and be useful for other things.

Definition 1.7.10. *For a diagonal $n \times m$-matrix Σ we denote by $\Sigma^{(k)}$ the matrix obtained from Σ by keeping the first k diagonal entries and setting the others to zero.*

The rank k SVD-approximation *of a general $n \times m$-matrix A, is the matrix $A^{(k)} := U\Sigma^{(k)}V^T$ where $A = U\Sigma V^T$ is a singular value decomposition.*

One can check that the above $A^{(k)}$ does not depend on the choice of singular value decomposition of A, and thus is well-defined. Equivalently, if $A = \sum_{i=1}^{r} \sigma_i u_i v_i^T$ is the representation of Remark 1.7.9, one can define $A^{(k)} := \sum_{i=1}^{k} \sigma_i u_i v_i^T$, i.e. as the sum of the first k of the above summands.

Actually this is the best way of approximating a matrix by others of lower rank, in terms of two standard notions of matrix distance.

Definition 1.7.11. *(i) The* spectral norm *of an $n \times m$-matrix is defined as*

$$\|A\| := max\{\|Av\|_2 \ | \ v \in \mathbb{R}^m, \ \|v\|_2 = 1\}.$$

The spectral distance *between two $n \times m$-matrices A, B is $\|A - B\|$.*

(ii) The Frobenius norm *of a matrix A is $\|A\|_F := \sqrt{tr(A^TA)}$. The* Frobenius distance *between two $n \times m$-matrices A, B is $\|A - B\|_F$.*

Both spectral distance and Frobenius distance are invariant under multiplication with orthogonal matrices. This follows in the first case because orthogonal matrices don't change lengths and in the second case because of the cyclic invariance of the trace.

One can easily convice oneself that the spectral norm of a diagonal matrix with nonnegative diagonal entries is the biggest of these entries: Multiplying the diagonal matrix with entries $\sigma_1, \ldots, \sigma_n$ (all nonnegative, ordered by size) with the vector $x = (x_1, \ldots, x_n)^T$ gives the vector $(\sigma_1 x_1, \ldots, \sigma_n x_n)^T$ whose squared norm is $\sigma_1^2 x_1^2 + \ldots + \sigma_n^2 x_n^2$. The vector x of norm 1 that maximizes this expression is $(1, 0 \ldots, 0)^T$ – intuitively one puts all the weight of the norm 1 vector into the coefficient that gets multiplied with the biggest σ_i; any other distribution of the weights will result in a smaller vector (formally one can compute the maximum of $\sigma_1^2 x_1^2 + \ldots + \sigma_n^2 x_n^2$ under the constraint $x_1^2 + \ldots + x_n^2 = 1$, with the method of Lagrange multipliers to prove this - this was exercise 17). We will use this in the following.

Theorem 1.7.12. *[Eckart-Young-Mirsky theorem] The SVD-approximation $A^{(k)}$ is the rank k matrix that is closest to A both in the spectral and the Frobenius distance.*

Proof. We only give the proof for the spectral distance:

Let $A = U\Sigma V^T$ be a singular value decomposition. Then $A^{(k)} = U\Sigma^{(k)}V^T$. The spectral distance between A and $A^{(k)}$ is

$$\|A - A^{(k)}\| = \|U\Sigma V^T - U\Sigma^{(k)}V^T\| = \|U(\Sigma - \Sigma^{(k)})V^T\| = \|\Sigma - \Sigma^{(k)}\| = \sigma_{k+1}$$

We now show that given any matrix B of rank k, we have $\|A - A^{(k)}\| = \sigma_{k+1} \leq \|A - B\|$.

For this note that, since B has rank k, we can find a linear combination of the first $k+1$ columns of V, $w = \mu_1 v_1 + \ldots + \mu_{k+1}v_{k+1}$ such that $Bw = 0$. Since scaling w does not change the fact that $Bw = 0$, we can arrange it that $\mu_1^2 + \ldots + \mu_{k+1}^2 = 1$. With this one gets

$$\begin{aligned}\|A - B\|^2 &\geq \|(A - B)w\|^2 = \|Aw - Bw\|^2 = \|Aw\|^2 = \mu_1^2\sigma_1^2 + \ldots + \mu_{k+1}^2\sigma_{k+1}^2 \\ &\geq \mu_1^2\sigma_{k+1}^2 + \ldots + \mu_{k+1}^2\sigma_{k+1}^2 \\ &= (\mu_1 + \ldots + \mu_{k+1})\sigma_{k+1}^2 = \sigma_{k+1}^2 = \|A - A^{(k)}\|^2\end{aligned}$$

For the proof in the Frobenius distance case see Ben-Israel/Greville, Generalized inverses - Theory and applications, 2nd edition, page 213. According to the authors the Theorem should be called “Schmidt theorem”. □

This low-rank approximation can be used for data compression, e.g. by storing an image in a matrix and replacing it by one of lower rank, such that almost no quality is lost. The `jpeg` compression algorithm is based on this idea. Here is a Python notebook showing that idea.

Low rank approximation can also be used for image denoising: If an image is blurred by some random noise, replace it by a low rank approximation, and it will often look better. The idea is that random noise is something that can not easily be compressed in some shorter description, so if we force ourselves to give it a “shorter description” (i.e. an encoding by a lower rank matrix) the noise will be forced out. See e.g. the article Fan et al., Image denoising by low-rank approximation with estimation of noise energy distribution in SVD domain. That random noise cannot be compressed into a short description, is an idea that will be explained further in the part on Information Theory.

Another application is background extraction in videos. Here is the idea: Imagine you have a video of, say 200 frames, each of 700 × 500 pixels. Then you can see each frame as a vector with $700 \cdot 500 = 350000$ entries and the whole film as a 350000 × 200-matrix. We want to distinguish static background from movement in the front. If no movement was recorded, then this matrix would have rank 1, because each column would be the same. If something moves through the image, then this makes the rank of the matrix go up. But still, in most columns the background pixels would be visible, just not in those where something was briefly in front. If you go to the rank 1 approximation, the moving parts will be deleted (roughly, possibly with some other slight distortion), because this is the closest you can get to the given matrix by a rank 1 matrix. See Reitberger et al., Background Subtraction using Adaptive Singular Value Decomposition.

Remark 1.7.13. Singular value decomposition and PCA: If you want to perform PCA on some collection of data vectors with n features, instead of following our recipe from the PCA section, you can as well produce a singular value decomposition of the *centered* data matrix: If the centered data matrix decomposes as $X = U\Sigma V^T$, then the covariance matrix is given by

$$C = \frac{1}{n-1}X^TX = \frac{1}{n-1}V\Sigma^TU^TU\Sigma V^T\frac{1}{n-1}V\Sigma^T\Sigma V^T$$

and since $\Sigma^T\Sigma$ is a diagonal matrix, the princpal axes are the columns of V. See this forum post for more details.

1.8 Pseudoinverses and least square approximation

Definition 1.8.1. *A* pseudoinverse, *or* Moore-Penrose inverse, *of a matrix A is a matrix A^+ such that all of the following equations hold:*

(i) $AA^+A = A$

(ii) $A^+AA^+ = A^+$

(iii) $(AA^+)^T = AA^+$

(iv) $(A^+A)^T = A^+A.$

Theorem 1.8.2. *Every matrix has a unique pseudoinverse.*

Proof. Let A be a matrix.

Uniqueness: Suppose that B and C both satisfy the conditions for a pseudoinverse. Then

$$\begin{aligned} AB &\overset{(i)}{=} (ACA)B = (AC)(AB) \overset{(iii)}{=} (AC)^T(AB)^T \\ &= C^TA^T(AB)^T = C^T(ABA)^T \overset{(i)}{=} C^TA^T = (AC)^T \overset{(iii)}{=} AC \end{aligned}$$

Similarly we get $BA = CA$. Now we can conclude

$$B \overset{(ii)}{=} BAB = BAC = CAC \overset{(ii)}{=} C.$$

Existence: One easily checks that for a square diagonal matrix the pseudoinverse given by

$$\begin{pmatrix} d_1 & 0 & \cdots & & & 0 \\ 0 & \ddots & & & & \\ \vdots & & d_r & & & \\ & & & 0 & & \\ & & & & \ddots & \\ 0 & & \cdots & & & 0 \end{pmatrix}^+ = \begin{pmatrix} d_1^{-1} & 0 & \cdots & & & 0 \\ 0 & \ddots & & & & \\ \vdots & & d_r^{-1} & & & \\ & & & 0 & & \\ & & & & \ddots & \\ 0 & & \cdots & & & 0 \end{pmatrix}$$

If $m \geq n$, then an $n \times m$-diagonal matrix is of the form $A = \big(D|\mathbf{0}\big)$, where D is some diagonal square matrix and $\mathbf{0}$ is the zero matrix of size $n \times (m-n)$. We check that in this case the pseudoinverse is given by $A^+ = \begin{pmatrix} D^+ \\ \mathbf{0} \end{pmatrix}$.

Indeed, AA^+, resp. A^+A, is an $n \times n$- (resp. $m \times m$) diagonal matrix with 1s on the diagonal in the places where D has non-zero entries. Thus AA^+ and A^+A are symmetric (i.e. satisfy conditions (iii) and (iv) for pseudoinverses) and clearly also satisfy $AA^+A = A$ and $A^+AA^+ = A^+$ (conditions (i) and (ii)).

The corresponding considerations apply for the case $n \geq m$.

Finally, for a general matrix A, consider its singular value decomposition $A = U\Sigma V^T$. We will show that the pseudoinverse is given by $A^+ = V\Sigma^+U^T$ by checking the four defining properties directly:

(i) $AA^+A = U\Sigma V^T V\Sigma^+ U^T U\Sigma V^T = U\Sigma\Sigma^+\Sigma V^T = U\Sigma V^T = A$

(ii) $A^+AA^+ = V\Sigma^+ U^T U\Sigma V^T V\Sigma^+ U^T = V\Sigma^+\Sigma\Sigma^+ U^T = V\Sigma^+ U^T = A^+$

(iii) $(AA^+)^T = \left(U\Sigma V^T V\Sigma^+ U^T\right)^T = \left(U\Sigma\Sigma^+ U^T\right)^T = U\left(\Sigma\Sigma^+\right)^T U^T$
$= U\left(\Sigma\Sigma^+\right) U^T = U\Sigma V^T V\Sigma^+ U^T = AA^+$

(iv) $\left(A^+A\right)^T = \left(V\Sigma^+ U^T U\Sigma V^T\right)^T = \left(V\Sigma^+\Sigma V^T\right)^T$
$= V\left(\Sigma^+\Sigma\right)^T V^T = V\left(\Sigma^+\Sigma\right) V^T = V\Sigma^+ U^T U\Sigma V^T = A^+A$

□

Remark 1.8.3. Unlike for inverses it is not in general true that $(AB)^+ = B^+A^+$. For example

$$\text{for } A = \begin{pmatrix} 1 & 0 \end{pmatrix}, \quad B = \begin{pmatrix} 1 \\ 1 \end{pmatrix} \qquad \text{one has} \quad (AB)^+ = 1 \quad \text{and} \quad B^+A^+ = \frac{1}{2}$$

Explicit formulas for $(AB)^+$ in terms of A^+ and B^+ exist , as well as complete characterizations of when $(AB)^+ = B^+A^+$

Application of pseudoinverses: Pseudoinverses can be used to find the solution of smallest length to a system of linear equations. Even better, if the system of linear equations does not have a solution, one can use pseudoinverses to find the best approximation to a solution: The next theorem says that, given a possibly unsolvable system of linear equations $Ax = b$, then $z := A^+b$ comes closest to being a solution.

Theorem 1.8.4. *Let A be an $n \times m$-matrix and $b \in \mathbb{R}^n$. Define $z := A^+b$. Then we have for all $x \in \mathbb{R}^m$: $\|Ax - b\| \geq \|Az - b\|$.*

Proof. First note the following equality:

$$\begin{aligned} A^T(Az - b) &= A^T(AA^+b - b) = A^T(AA^+)^T b - A^T b \\ &= (AA^+A)^T b - A^T b \overset{(i)}{=} A^T b - A^T b = 0 \qquad (*) \end{aligned}$$

With this we can see that for any $x \in \mathbb{R}^m$

$$\begin{aligned}
\|Ax - b\|^2 &= \|A(x - z + z) - b\|^2 = \|A(x - z) + Az - b\|^2 \\
&= (A(x - z) + (Az - b))^T (A(x - z) + (Az - b)) \\
&= (A(x - z))^T A(x - z) + (A(x - z))^T (Az - b) \\
&\qquad + (Az - b)^T A(x - z) + (Az - b)^T (Az - b) \\
&= \|A(x - z)\|^2 + (x - z)^T \underbrace{A^T (Az - b)}_{=0, \text{ by } (*)} \\
&\qquad\qquad + (((Az - b)^T A(x - z))^T)^T + \|Az - b\|^2 \\
&= \|A(x - z)\|^2 + 0 + ((x - z)^T \underbrace{A^T (Az - b)}_{=0, \text{ by } (*)})^T + \|Az - b\|^2 \\
&= \|A(x - z)\|^2 + \|Az - b\|^2 \\
&\geq \|Az - b\|^2
\end{aligned}$$

□

Geometrically, Theorem 1.8.4 says that $Az = AA^+b$ is the point in the range of the linear map A that is closest to b.

In the case when the system of linear equations $Ax = b$ *does* have a solution, the pseudoinverse also gives a solution: If x is such a solution, then $Ax - b = 0$, and by the above we have $0 = \|Ax - b\| \geq \|Az - b\| \geq 0$, so $Az - b = 0$ and thus $z = A^+b$ is also a solution.

The approximate solution obtained via a pseudoinverse in Thm. 1.8.4 does not need to be unique. But it is the best solution, in the sense that is the vector of smallest size, as stated in the next theorem.

Thus even in the case where we have a solution, the pseudoinverse picks the best one.

Theorem 1.8.5. *Let A be an $n \times m$-matrix and $b \in \mathbb{R}^n$. Then among all $x \in \mathbb{R}^m$ that minimize $\|Ax - b\|$, the point $z := A^+b$ has the smallest possible norm.*

Proof. The proof is analogous to that of Theorem 1.8.4.

First suppose that $Ax = b$ is a system of linear equations that *has* a solution. We show that then $z := A^+b$ is the solution of minimal size. Let

x be any solution, i.e. suppose $Ax = b$. Then first note that

$$\begin{aligned} z^T(x-z) &= (A^+b)^T(x-z) \overset{(ii)}{=} (A^+AA^+b)^T(x-z) = (A^+Az)^T(x-z) \\ &= z^T(A^+A)^T(x-z) \overset{(iv)}{=} z^T(A^+A)(x-z) = z^TA^+(Ax-Az) \\ &= z^TA^+(b-b) = 0 \end{aligned}$$

In the passage to the last line we used that, when there exists a solution to $Ax = b$, then z must be one, i.e. $Az = b$.

Now we obtain

$$\begin{aligned} \|x\|^2 = \|(x-z)+z\|^2 &= ((x-z)+z)^T((x-z)+z) \\ &= (x-z)^T(x-z) + \underbrace{z^T(x-z)}_{=0} + (x-z)^Tz + z^Tz \\ &= \|x-z\|^2 + 0 + (\underbrace{z^T(x-z)}_{=0})^T + \|z\|^2 \\ &= \|x-z\|^2 + \|z\|^2 \geq \|z\|^2 \end{aligned}$$

Now for a general system, that has possibly no solution, we know from Theorem 1.8.4 that AA^+b is the closest point to b that is in the image of A. Note that by the properties of linear functions there is a unique such point[2].

So to find the $x \in \mathbb{R}^m$ of smallest norm that minimizes this distance, we have to find the $x \in \mathbb{R}^m$ of smallest norm that maps to AA^+b. From what we have just proved, we know that this is precisely A^+b. □

1.8.1 Application to linear approximation of a set of data points (Least squares method)

Given a set of data points $\tilde{d}^{(1)}, \ldots, \tilde{d}^{(k)} \in \mathbb{R}^n$, one can ask whether they sit, at least approximately, in a smaller dimensional affine subspace. By first calculating their mean value $\bar{d} = \frac{1}{k}\Sigma_{i=1}^k d_i$ and then subtracting that from every point, i.e. now considering the new points $d^{(1)} := \tilde{d}^{(1)} - \bar{d}, \cdots, d^{(k)} :=$

[2] To see this, one can parametrize the plane that is the image of A by some function $f\colon \mathbb{R}^{\operatorname{rank} A} \to \mathbb{R}$. Then the function measuring the squared distance from points in the plane to b, i.e. $\mathbb{R}^{\operatorname{rank} A} \to \mathbb{R}, t \mapsto \|f(t) - b\|^2$, is a quadratic function with exactly one critical point.

$\tilde{d}^{(k)} - \bar{d}$, one can center the points around zero. Now the question is whether the points $d^{(1)}, \ldots, d^{(k)}$ sit in a lower dimensional subvector space.

Unless the x_n-axis is contained in that subspace, the last coordinate $d_n^{(i)}$ of each vector $d^{(i)}$ then depends linearly on the first $n-1$ coordinates. That is, one has $d_n^{(i)} = d_1^{(i)} a_1 + \ldots + d_{n-1}^{(i)} a_{n-1}$, for each i and for some fixed $a_j \in \mathbb{R}$.

Phrased differently, one has

$$\begin{pmatrix} d_n^{(1)} \\ \vdots \\ d_n^{(k)} \end{pmatrix} = \begin{pmatrix} d_1^{(1)} & \cdots & d_{n-1}^{(1)} \\ \vdots & & \vdots \\ d_1^{(k)} & \cdots & d_{n-1}^{(k)} \end{pmatrix} \begin{pmatrix} a_1 \\ \vdots \\ a_{n-1} \end{pmatrix}$$

Now it is usually not true that given data in $\mathbb{R}^n$ fits into a lower dimensional subspace, but we can still try to find the $(n-1)$-dimensional subspace such that the sum of the distances of the $d_n^{(i)}$ to it is minimal.

This subspace will be given by a single defining equation $\tilde{a}_1 x_1 + \ldots + \tilde{a}_{n-1} x_{n-1} + \tilde{a}_n x_n = 0$, and unless $a_n = 0$ (which corresponds to the case when the subspace contains the x_n-axis) we can divide everything by $-\tilde{a}_n$ and then only need to look for $a_1 := -\frac{\tilde{a}_1}{\tilde{a}_n}, \ldots, a_{n-1} := -\frac{\tilde{a}_{n-1}}{\tilde{a}_n}$, giving us the subvector space defined by the equation $a_1 x_1 + \ldots a_{n-1} x_{n-1} - x_n = 0$.

We know that we can do this using the pseudoinverse of the above $k \times (n-1)$-matrix: Defining

$$d := \begin{pmatrix} d_n^{(1)} \\ \vdots \\ d_n^{(k)} \end{pmatrix}, \quad D := \begin{pmatrix} d_1^{(1)} & \cdots & d_{n-1}^{(1)} \\ \vdots & & \vdots \\ d_1^{(k)} & \cdots & d_{n-1}^{(k)} \end{pmatrix}, \quad a := \begin{pmatrix} a_1 \\ \vdots \\ a_{n-1} \end{pmatrix}$$

we need to compute $a = D^+ d$.

The above motivating story of the given points in $\mathbb{R}^n$ just led us to a setting of *linear regression* – i.e. a situation where we want to find coefficients which occur linearly in some function. One encounters this setting for example if one has k measurements of quantities $d_1, \ldots d_{n-1}$ and suspects a further quantity d_n to depend linearly on these.

Remark 1.8.6. In many texts the columns of the above matrix are required to be linearly independent. If they are not, then one says that the data has *multicollinearity*. In this case the user is advised to remove some columns until they become linearly independent. Then the recipe from exercise 5 applies and one can compute $D^+ = (D^TD)^{-1}D^T$.

However, we know from Theorem 1.8.4 that the pseudoinverse gives the best approximation always, whether we have multicollinearity or not.

Example 1.8.7. Exercise 2 was an example of the fact that, given n points in $\mathbb{R}^2$ with different x-coordinates $w_1, \ldots, w_n$, one can always find a unique polynomial of degree $n-1$ whose graph passes through all of them. The reason is that finding such a polynomial is equivalent to solving a linear system of equations, as in exercise 2, and that the matrix encoding this system is invertible: One has

$$\det \begin{pmatrix} w_1^{n-1} & w_1^{n-2} & \cdots & 1 \\ w_2^{n-1} & w_2^{n-2} & \cdots & 1 \\ & & \ddots & \\ w_n^{n-1} & w_n^{n-2} & \cdots & 1 \end{pmatrix} = \Pi_{1 \le i < j \le n}(w_j - w_i)$$

and since all the w_i are supposed to be different, this expression is non-zero (this determinant is called the Vandermonde determinant).

If we want a polynomial of lower degree, we may not be able to pass through all of the points, but we can still aim to minimize the (squared) distances of the points from the graph:

Given $(w_1, y_1), \ldots, (w_n, y_n) \in \mathbb{R}^2$, find a polynomial of degree k, $p(x) = a_kx^k + \ldots + a_1x + a_0$ such that the sum of the squared distances $(p(w_i) - y_i)^2$ becomes minimal.

For this set up the linear system

$$\begin{pmatrix} w_1^k & w_1^{k-1} & \cdots & 1 \\ w_2^k & w_2^{k-1} & \cdots & 1 \\ & & \ddots & \\ w_n^k & w_n^{k-1} & \cdots & 1 \end{pmatrix} \begin{pmatrix} a_k \\ \vdots \\ a_0 \end{pmatrix} = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix}$$

and find the best approximate solution for $a_1, \ldots, a_n$ using the least squares method.

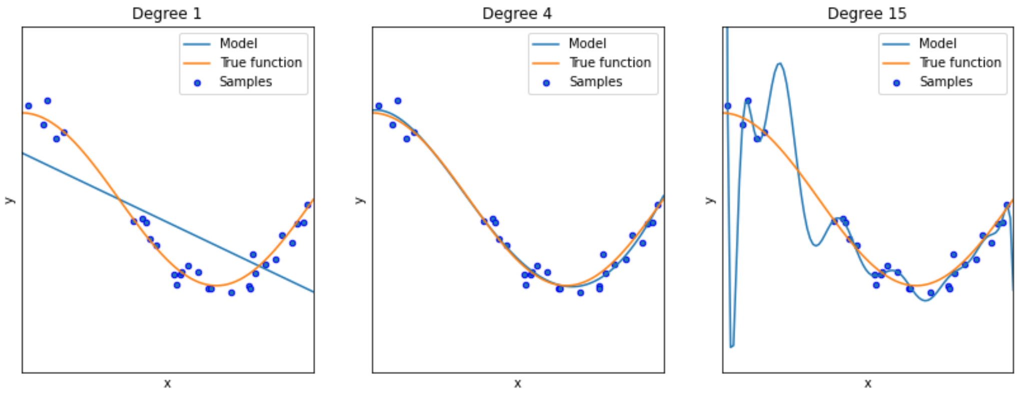
Choosing polynomials of lower degree does not just reduce computational complexity, much more importantly it helps to avoid the so-called *overfitting*:

1.8.2 Overfitting and Data splitting

We often try to model a function so that it best fits to some given training data. As seen above, with polynomials for example this can be achieved with perfect precision, if we allow a high enough degree. However, the training data is often not perfectly mirroring the true function that we try to approximate, but instead is blurred out by some "noise" (e.g. measurement errors).

If we try to adapt our model function too perfectly to the given data, then the noise determines a lot of the function's behaviour. The function will then reproduce very precisely our training data, but possibly approximate very badly any further data. Here is an example:

The true function is plotted in orange. The sample points were obtained by adding some random values (Gaussian distributed) to the function value. The blue curves show what functions we obtain if we look for the polynomial of degrees 1, 4 and 15 best fitting to the samples:



Picture produced with BSD licensed code from Scikit-learn

As you can see, the degree 15 polynomial shows some wild oscillating behaviour, which is necessary to precisely capture the sampling points. If now we were now trying to predict the value of our true function at a value at the left border of the shown interval, using the degree 15 polynomial could result in a wildly wrong guess, while using the degree 4 polynomial would be less wrong.

The process of adapting a model to reflect some training data is called *fitting the model*. If the phenomenon that we just described occurs, i.e. too good adaptation to the data leading to bad new predictions, this is called *overfitting*. Thus the degree 15 model here is an example of overfitting. The degree 1 model already performs poorly on the training data, which would be called *underfitting*.

In many applications one tries to find a "sweet spot" – a not too simple, but also not too complex model – so that prediction accuracy on the training data is ok, but overfitting is avoided.

In order to find that sweet spot a common strategy is not to use all of the available data for training, but instead *split the data*! One takes e.g. two thirds of the data samples, chosen randomly, to train a model and then tests the model on the remaining one third of the data. In this way one can estimate whether overfitting occurred or not.

In notebooks for exercise 14 you can see these ideas illustrated concretely.

1.9 Final remarks

Some remarks and takeaway lessons for this section on linear algebra:

- Linear algebra is largely about linear maps. Few functions in real life are linear and one might suspect that linear algebra is of limited use for finding such functions.

  But linear algebra is not just capable of finding linear maps approximating given data: The example of polynomial regression Ex. 1.8.7 clearly shows this. Instead we can use linear algebra to find (possibly non-linear) functions that depend linearly on some parameter! A polynomial is determined by its coefficients – it is a linear combination of the monomials $1, x, x^2, x^3, \ldots$ – and to find a polynomial means to find these coefficients, *on which it depends linearly*. The same would apply if, for example, we were trying to find a function that is a linear combination of different sine and cosine functions.

  More generally, one can extract linear dependencies occurring somewhere in the middle of the process of building some function and profitably use linear algebra just on that part. The generalized linear models that we will encounter later are an example, and so are (functions described by) neural networks.

- The consideration of matrices in linear algebra is motivated by the fact that they correspond to linear maps. All the constructions and operations on matrices come from this view point – for example one could hardly come up with the law for multiplication of matrices, if it were not precisely what describes composition of linear maps. Likewise, diagonalization of matrices and quadratic forms and singular value decomposition arise from a geometric viewpoint and by thinking about linear maps.

  However, matrices don't just arise through linear maps! Encoding an image by the gray scale values of its pixels gives rise to a matrix, as does counting the words in documents in natural language processing. Another example is the adjacency matrix of a graph. Once we have such a matrix, we can use all of our methods and theorems on it, ignoring the fact that they were inspired by thinking of linear maps.

  In this way linear algebra serves as a toolbox of things one can do with arrays of numbers, no matter what is their origin. It is sometimes in this spirit that linear algebra is used in machine learning. The matrix cookbook, well-known in machine learning circles, is a witness of this. However, without knowing some of the original intentions of linear algebra concerning vector spaces and linear maps, it is probably hard to know where to look for the right tool for a given problem.

  It is, in any case, good to be aware of both sides of the coin.

- Linear algebra is very useful in itself and we have seen some direct applications in this section. Equally important, however, is the role of linear algebra as a basic language in all of mathematics. Concepts, proofs and algorithms from other fields – graph theory, probability theory, number theory, really pretty much everything – often rely on linear algebra, and when a mathematical problem can be reduced to linear algebra it is usually a big step towards its solution. We will see illustrations of this point in the rest of the course.

A good reference on PCA, singular value decomposition, pseudoinverses and applications is the preprint Zhang: The Singular Value Decomposition, Applications and Beyond.

A continuation of these themes is given by *higher order singular value decompositions*: The singular value decomposition that we have seen for matrices, i.e. 2-dimensional arrays of numbers, also exists for higherdimensional arrays. Remember the low rank approximations of matrices encoding a grey scale image: This made it possible to achieve excellent approxiations to a given image using much less memory.

A color image is usually given as *three* matrices, one for the intensity of red, green and blue, respectively, at each pixel. So for an image with 300×400 pixels we would get a $300 \times 400 \times 3$-array. For a short film with 600 frames, each a greyscale picture of 300×400 pixels, we would get a $300 \times 400 \times 600$-array. And for a color film, with 3 layers per frame, we would get a 4-dimensional $300 \times 400 \times 3 \times 600$-array. Thus we have 3-, 4- and higherdimensional arrays of numbers and can ask for approximations of lower complexity for those.

Just like matrices correspond to linear maps, higherdimensional arrays correspond to multilinear maps. We have seen the case of a bilinear form, i.e. a bilinear map $f\colon V \times V \to \mathbb{R}$ that is linear in each variable. Such a map is always of the form $(v, v') \mapsto v^T A v'$, where the entries of the matrix A are the values $f(e_i, e_j)$ at pairs of basis vectors. A map to $g\colon U \times V \times W \to \mathbb{R}$, linear in each argument, would correspond to a cube of numbers given by the values on all tuples of basis vectors $g(e_i, e_j, e_k)$. The dimension of the array of numbers gets higher, both when we consider more arguments of our function, and when we consider functions going to $\mathbb{R}^n$ for $n > 1$.

The linear algebra that deals with multilinear maps and the associated higher dimensional arrays is called *tensor calculus*. The name of the famous Deep Learning library *Tensorflow* alludes to this.

Here is an article whose introduction lists some applications of the tensor singular value decomposition, and here is a very readable practical introduction with Python code.

A theme that we have not touched upon is *numerical linear algebra*: How to actually make a computer calculate with huge matrices is a field of active research and a pillar of machine learning. This is addressed in other modules of the master program. If you don't want to wait for such a course to be offered, check out the free online Python-based course Computational Linear Algebra for Coders!

If you find errors, points that are unclear to you, or have other suggestions for improvement of this manuscript, please tell me!

2 Matrix differential calculus

For now I won't write my own notes on matrix differential calculus. I will do it later, whenever I find the time.

- A reminder of usual differential calculus is given in Parr/Howard's The Matrix Calculus You Need For Deep Learning, but also in many other places.

- The lecture is following this series of blog posts:

  Frechet derivatives 1

  Frechet derivatives 2

  Frechet derivatives 3

  Frechet derivatives 4

  We really only covered the first of these blog posts, the others can serve as additional motivation and realistic use cases. **For the exam it is enough to know the content of the first of these posts, or equivalently the content of the lecture and the exercises 15a) and b)**.

- Another lecture on Matrix differential calculus has been given in the Machine Learning lecture course. Here is Prof. Harmeling's 2019 lecture on Matrix Differential Calculus. This is what you need in practice.

- Lists of rules and tricks for applying matrix differential calculus are given in the "cheat sheet" distributed in the Machine Learning class, and also here:

  Summary chapter of Magnus/Neudecker: Matrix Differential Calculus

  matrix cookbook

- The website matrixcalculus.org let's you compute matrix derivatives and even export the results into Python or LaTeX!

3 Convex optimization

A good source for this part of the course is the book Boyd/Vandenberghe, Convex Optimization, Cambridge University Press 2004, freely available under the given link. This chapter follows passages of that book, adding some own discussion.

3.1 Introduction

In Machine Learning and Data Science one often has to minimize or maximize functions. For example, in training a neural network, one tries to find the weights that minimize the error on the training data, and if given some data, one tries to find the probability distribution that is most likely to produce that data as a random sample, e.g. by the method "maximum likelihood estimation" (which already has the maximization task in its name). This is what this chapter is about.

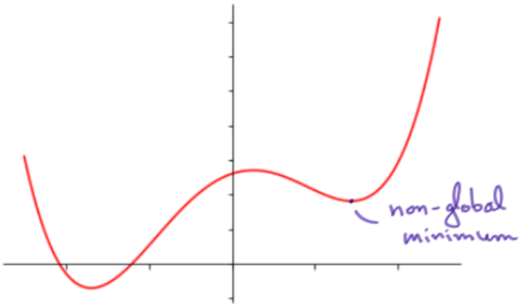
An optimization problem is a problem of the following type (where f is a function from some subset of $\mathbb{R}^n$ to $\mathbb{R}$):

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & x \in S \end{array}$$

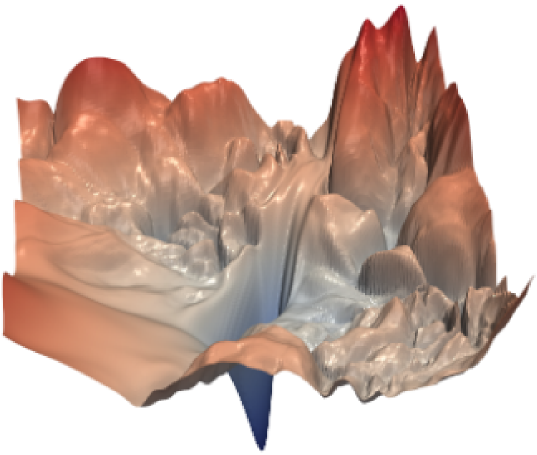
That is, we ask for which $x \in S$ the function f attains the smallest value. Sometimes we also want to maximize the function f, but that is equivalent to minimizing the function $-f$.

General optimization problems can quickly be intractable, because we simply don't have a good strategy to find the minimum. One general strategy for differentiable functions f is *gradient descent*: One starts at a randomly chosen point, looks which is the direction of steepest descent (which is the negative gradient), and takes a step in that direction (hoping it is still in S). This is repeated until the value seems not to get much smaller anymore.

For general functions, it is a major headache that gradient descent can lead one into a local minimum that is not a global minimum:

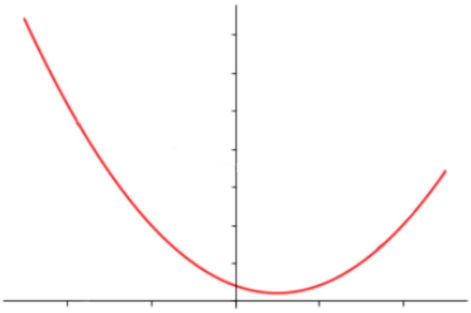


One counters this by doing the random initialization many times and hoping that one of the starting points is close enough to the global minimum that gradient descent leads there. Here is a picture of a neural network loss landscape illustrating the many non-global minima that might be there:



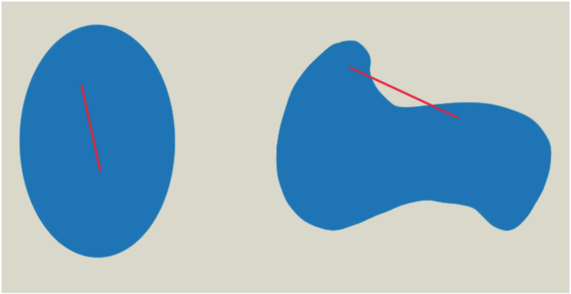
A neural network loss landscape, from Li et al.: Visualizing the Loss Landscape of Neural Nets. Code and further pictures and links here.

Convex optimization problems are a class of optimization problems that do not suffer from this problem. Roughly, a function is convex if its graph bulges outwards at all times:



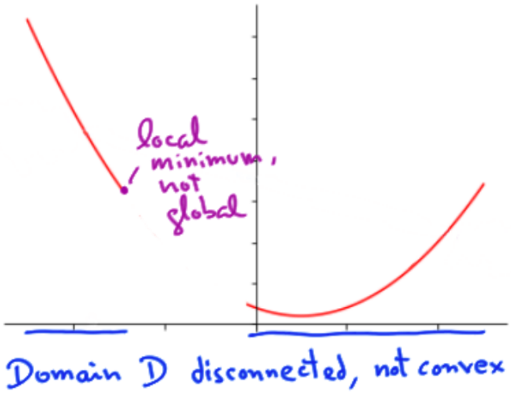
Such functions have at most one global minimum, so gradient descent can not lead into a non-global minimum.

A subset of $\mathbb{R}^n$ is convex, if the straight path between any two points goes entirely through that set:



A convex and a non-convex set.

A *convex optimization problem* is one where the function f is convex and the set given by the constraints, where the minimum is sought, is convex as well. Again with gradient descent in mind, one can see why the convexity of the domain is a desirable condition: In $\mathbb{R}$, convex sets are precisely the connected sets, i.e. single intervals. If our function was instead defined on a union of two disjoint intervals, it could be that we descend down one branch of the graph and stop at its lowest point, while the global minimum is a different branch:



So it is much easier to solve convex optimization problems. But in general one wants to optimize (i.e. maximize or minimize) all kinds of functions, not only convex ones. In concentrating on convex problems, are we just looking for the car keys under the streetlight?

No, because on the one hand convex optimization problems do arise in Machine Learning and Bayesian Statistics. There also are slight generalizations of convexity as in Remark 3.3.10 below, that can go a bit further, while still using techniques and intuition about convex optimization problems.

On the other hand out of a general optimization problem one can produce a convex optimization problem that helps finding information about the original problem. This is called Lagrange duality and we will see it later in this chapter.

3.2 Convex sets

Definition 3.2.1. *1. A set $C \subseteq \mathbb{R}^n$ is* convex, *if for all $x, y \in C$ and all $\theta \in [0,1]$ we have $\theta x + (1-\theta)y \in C$.*

2. If $x_1, \dots, x_n \in \mathbb{R}^n$ and $\theta_1, \dots, \theta_n \in \mathbb{R}_{\geq 0}$ are such that $\theta_1 + \dots + \theta_n = 1$, then $\theta_1 x_1 + \dots + \theta_n x_n$ is called a convex combination *of the points $x_1, \dots, x_n$.*

3. The convex hull of a set $W \subseteq \mathbb{R}^n$ is defined by

$$\mathbf{conv}\, W := \{\, \theta_1 x_1 + \dots + \theta_n x_n \mid x_i \in W,\, \theta_i \geq 0, \Sigma_i \theta_i = 1 \,\}$$

One can easily see that a set is convex if and only if it contains all convex combinations of any finite set of its points.

Arbitrary intersections of convex sets are convex, and so the intersection of all convex sets containing a given set W is convex again, and it is the smallest convex set containing W. This intersection is precisely $\mathbf{conv}\, W$.

Remark 3.2.2. One can show that a convex set C is also closed under infinite convex combinations in the following sense: If one has $x_i \in C$ $(i \in \mathbb{N})$ and θ_i $(i \in \mathbb{N})$ satisfy $\theta_i \geq 0$ and $\sum_{i=0}^{\infty} \theta_i = 1$, then $\sum_{i=0}^{\infty} \theta_i x_i \in C$.

From considering the finite approximations, it is immediate, that the infinite convex combination must lie in the closure of C. In case C is not closed, there is no problem: The limit point can not lie on a boundary point that does not belong to C. Intuitively, all the summands with positive θ_i and x_i lying in the interior drag the sum away from the boundary, so that one always ends up inside C.

More generally, if for a a convex set $C \subseteq \mathbb{R}^n$ and a function $p\colon C \to \mathbb{R}$ we have $\int_C p(x)\, dx = 1$, then $\int_C p(x) x\, dx \in C$.

Still more generally, if a random variable X takes values in a convex set $C \subseteq \mathbb{R}^n$ with probability 1, then its expected value EX lies in C. The previous two cases cover discrete and continuous random variables. Ignore this remark if your memory on probability theory is weak – we will cover this later.

Examples 3.2.3. (a) Open and closed balls in any norm on $\mathbb{R}^n$ are convex, e.g. $B_\epsilon^{\ell_2}(z) = \{x \in \mathbb{R}^n \mid \|x - z\| < \epsilon\}$, $\overline{B_\epsilon^{\ell_2}(z)} = \{x \in \mathbb{R}^n \mid \|x - z\| \leq \epsilon\}$. This follows from the triangle inequality (exercise).

(b) A vector $a \in \mathbb{R}^n$ and a $b \in \mathbb{R}$ define a *hyperplane*, namely the set $\{x \in \mathbb{R}^n \mid a^T x = b\}$. Alternatively, a hyperplane can be written as $\{x \in \mathbb{R}^n \mid a^T(x - x_0) = 0\}$ if one picks any $x_0 \in \mathbb{R}^n$ with $a^T x_0 = b$. Hyperplanes are convex (exercise).

(c) A hyperplane divides $\mathbb{R}^n$ into the two *closed halfspaces* $\{x \mid a^T x \leq b\}$ and $\{x \mid a^T x \geq b\}$. These are convex as well (exercise).

(d) Similarly a hyperplane gives rise to two *open halfspaces* $\{x \mid a^T x < b\}$ and $\{x \mid a^T x > b\}$, which are convex again (exercise).

(e) A *polyhedron* is the solution set of a finite number of linear equalities and inequalities, i.e. a set of the form

$$P = \{x \mid a_j^T x \leq b_j\, (j = 1 \ldots m),\, c_i^T x = d_i\, (i = 1 \ldots k)\}$$

Such a P is an intersection of hyperplanes and halfspaces and therefore convex. One can show that a bounded polyhedron is the convex hull of finitely many points, namely those points where a maximal number of the inequalities are equalities.

(f) The probability measures on a finite set $\{1, \dots, n\}$, i.e. the vectors $\{(p_1, \dots, p_n) \mid p_i \geq 0, \Sigma_{i=1}^n p_i = 1\}$ form a convex set. In fact this set is a polyhedron, called the *standard* $(n-1)$*-simplex*. It is the convex hull of the standard basis of $\mathbb{R}^n$.

A source of many further examples is the following simple observation:

Proposition 3.2.4. *Let $f \colon \mathbb{R}^n \to \mathbb{R}^k$ be an affine function, i.e. a function of the form $x \mapsto Ax + b$ for a $b \in \mathbb{R}^k$ and a $k \times n$-matrix A.*

(a) If $C \subseteq \mathbb{R}^n$ is convex, then $f(C)$ is convex, too.

(b) If $D \subseteq \mathbb{R}^k$ is convex, then $f^{-1}(D) = \{x \in \mathbb{R}^n \mid f(x) \in D\}$ is convex, too.

For example, the projection of a convex set onto some of its coordinates is convex.

Theorem 3.2.5 (Separating Hyperplane Theorem)**.** *Suppose $C, D \subseteq \mathbb{R}^n$ are convex and $C \cap D = \varnothing$. Then there exists a hyperplane separating C from D, i.e. an $a \in \mathbb{R}^n$, $a \neq 0$, and a $b \in \mathbb{R}$ such that $a^T x \leq b$ for all $x \in C$ and $a^T x \geq b$ for all $x \in D$.*

Proof. We will only consider the special case where C and D have a positive minimal distance

$$dist(C, D) := \inf\{\|u - v\| \mid u \in C, v \in D\} > 0$$

which is attained by two points $c \in C$, $d \in D$ In this case define $a := d - c$ and $b := \frac{1}{2}\big(\|d\|^2 - \|c\|^2\big)$. We claim that then the hyperplane $\{x \mid a^T x = b\}$ separates C and D, i.e. that the function $f(x) = a^T x - b$ is nonnegative on D and nonpositive on C.

For this observe that $f(x) = a^T x - b = (d - c)^T (x - \frac{1}{2}(d + c))$. Suppose there is a $u \in D$ such that $f(u) = (d - c)^T (u - \frac{1}{2}(d + c)) < 0$.

Rewriting $f(u)$ as

$$0 > f(u) = (d - c)^T (u - d + \frac{1}{2}(d - c)) = (d - c)^T (u - d) + \frac{1}{2}\|d - c\|^2$$

we see that $(d-c)^T(u-d) < 0$.

We now look at the intermediate points between u and d and at their squared distances from c:

$$\begin{aligned}\|d+t(u-d)-c\|^2 &= (d+t(u-d)-c)^T(d+t(u-d)-c)\\ &= d^Td + t(u-d)^T(d-c) + t(d-c)^T(u-d) + t^2\|u-d\|^2 + \|c\|^2\end{aligned}$$

Taking the derivative with respect to t, and inserting $t=0$, we get

$$\frac{d}{dt}\|d+t(u-d)-c\|^2_{t=0} = 2t(d-c)^T(u-d) < 0$$

Since this is negative, for some $0 < t < 1$, close enough to 0, we have

$$\|d+t(u-d)-c\|^2 < \|d-c\|^2$$

But this is a contradiction, because $d+t(u-d) = tu+(1-t)d$ belongs to D by convexity, but d was suppposed to be the point of D that is closest to c.

A similar argument gives $f(u) \leq 0$ for $u \in C$. □

A hyperplane that touches the boundary of a set C and that has all of C contained in one of its halfspaces is called a *supporting hyperplane* for C.

Proposition 3.2.6. *At every point x on the boundary of a convex set C, there exists a supporting hyperplane for C that contains x.*

Conversely, if a set C is closed, has non-empty interior and has a supporting hyperplane at every point of its boundary, then it is convex.

The proof is not hard, but I don't discuss it further. Prop. 3.2.6 implies that every closed convex set with nonempty interior is the intersection of halfspaces.

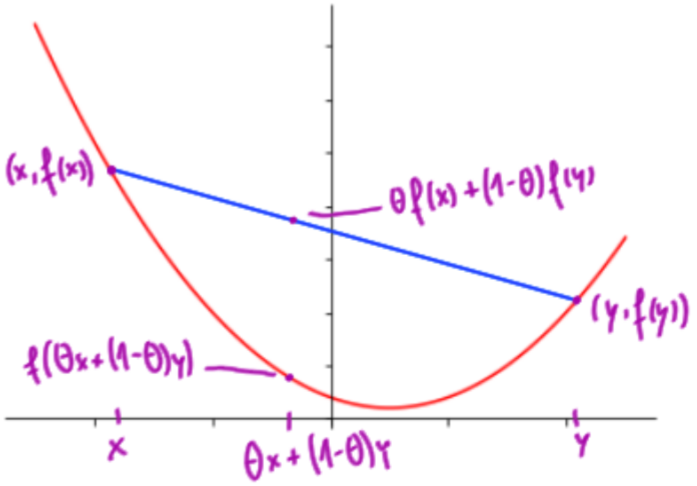
Remark 3.2.7. A further important way to recognize a set as convex is given by Prop. 3.5.5 below: Sublevel sets of convex functions are convex. But to make sense of this, we first have to introduce convex functions.

3.3 Convex functions

Definition 3.3.1. *A function $f\colon \mathbb{R}^n \supseteq D \to \mathbb{R}$ is* convex *if its domain D is a convex set and for every $x, y \in D$ and $\theta \in [0,1]$ we have $f(\theta x + (1-\theta)y) \leq \theta f(x) + (1-\theta)f(y)$.*

The function is called strictly convex *if strict inequality holds whenever $x \neq y$ and $\theta \neq 0, 1$.*
The function is called concave *(resp.* strictly concave*), if $-f$ is convex (resp. strictly convex).*

The condition for being a convex function can be visualized as the requirement that the straight line segment between two points on its graph must always lie above the graph:



A convex function

Remark 3.3.2. It follows immediately from the definition that a function $f\colon \mathbb{R}^n \supseteq D \to \mathbb{R}$ is *convex* if and only if its domain D is convex and the area above its graph (the so-called epigraph)

$$E_f := \{(x_1, \ldots, x_{n+1})^T \in \mathbb{R}^{n+1} \mid (x_1, \ldots, x_n)^T \in D \text{ and } x_{n+1} \geq f(x_1, \ldots, x_n)\}$$

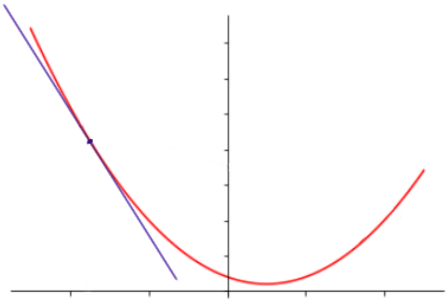
is a convex subset of $\mathbb{R}^{n+1}$.

Proposition 3.3.3. *(a) Suppose $f\colon D \to \mathbb{R}$ is differentiable. Then f is convex if and only if D is convex and for all $x, y \in D$ we have*

$$f(y) \geq f(x) + \nabla f(x)^T (y - x)$$

(b) Suppose $f\colon D \to \mathbb{R}$ is twice differentiable. Then if f is convex if and only if D is convex and the Hesse-matrix $H_f(x)$ is positive semidefinite for all $x \in D$.

Part (a) of this proposition says intuitively that the tangent line to the graph that goes into the direction of steepest descent lies completely below the graph:



Part (b) of the last proposition says intuitively that a function is convex, if and only if its graph is always bending upwards while moving towards points with bigger coordinates. The graph in the picture also illustrates this.

About the proof: (a): In the 1-dimensional case, that the tangent line at any point lies completely below the graph, means that the tangent lines form a family of supporting hyperplanes. So the epigraph is the intersection of the halfspaces of these tangent lines, hence convex. The same is true for the higherdimensional case: The right hand side is the first order approximation of f around x, and its graph is the tangent hyperplane of the graph of f. Again, the inequality says that the epigraph lies completely on one side of the tangent hyperplane. Thus altogether the family of these tangent hyperplanes is a family of supporting hyperplanes. The intersection of the corresponding halfspaces is the epigraph. So the epigraph is convex, and by Remark 3.3.2 the function is then convex.

(b): Very the Hesse matrix condition says that the graph is always curved upwards. It is then clear that the tangent hyperplane at a point is always below the whole graph, which implies condition (a).

To pin down this intuition more precisely: Intuitively, around every point there is a neighborhood in which the Taylor series terms of order 3 and higher are negligible. If the Hesse matrix is positive *definite*, then for a y in such a neighborhood, setting $\epsilon := y - x$ we have

$$\begin{aligned} f(y) = f(x+\epsilon) &= f(x) + \nabla f(x) \cdot \epsilon + \epsilon^T H_f \epsilon + o(\|\epsilon\|^3) \\ &\geq f(x) + \nabla f(x) \cdot \epsilon \end{aligned}$$

where the inequality holds, because $\epsilon^T H_f \epsilon > 0$ and the contribution of the $o(\|\epsilon\|^3)$ term is too small to change the inequality.

Now for an arbitrary pair of points x, y one can go from x to y in tiny enough steps to always have such an inequality, and conclude the overall inequality from (a).

This argument works only for very regular functions – e.g. if there is a radius r such that the Taylor series is a always a good enough approximation in a neighborhood of radius r around a point (then we can really pass between two points in finitely many steps as suggested).

Ask me for a more formal proof if you want to see one. ♡

Remark 3.3.4. Here is an alternative, and more formal, proof of Prop. 3.3.3 (a), directly in terms of the definition of convexity. First suppose that $f\colon D \to \mathbb{R}$ is defined on a subset $D \subset \mathbb{R}$ of the real numbers.

Suppose that f is convex. Then we need to show the inequality of Prop. 3.3.3 (a). So let $x, y \in D$. Since D is convex, we know that for every $\theta \in [0,1]$ the convex combination $(1-\theta)x + \theta y$ is also in D. It will be convenient to rewrite $(1-\theta)x + \theta y = x + \theta(y-x)$.

From the convexity of f we know

$$f(x + \theta(y-x)) \leq (1-\theta) f(x) + \theta f(y)$$

Dividing this equation by θ gives

$$\frac{f(x+\theta(y-x))}{\theta} \leq \frac{(1-\theta)}{\theta} f(x) + f(y) = \frac{1}{\theta} f(x) - f(x) + f(y)$$

This implies

$$f(y) \geq f(x) + \frac{f(x+\theta(y-x)) - f(x)}{\theta}$$

The fraction on the right hand side is exactly the difference quotient known from the limit definition of derivative. So taking $\lim_{\theta\to 0}$ gives the desired inequality

$$f(y) \geq f(x) + f'(x) \cdot (y-x)$$

For the opposite direction suppose f satisfies $f(b) \geq f(a) + f'(a)(b-a)$ for all $a, b \in D$. Let $x, y \in D$ and $\theta \in [0,1]$, and abbreviate $z := \theta x + (1-\theta)y$ for the convex combination. The supposed inequality with $b = x$ and $a = z$ gives $f(x) \geq f(z) + f'(z)(x-z)$ and multiplying with θ gives

$$\theta f(x) \geq \theta f(z) + \theta f'(z) \cdot (x-z)$$

Likewise the supposed inequality with $b = y$ and $a = z$ gives $f(y) \geq f(z) + f'(z)(y-z)$ and multiplying with $1-\theta$ gives

$$(1-\theta)f(y) \geq (1-\theta)f(z) + (1-\theta)f'(z) \cdot (y-z)$$

Summing these two inequalities yields

$$\begin{aligned}\theta f(x) + (1-\theta)f(y) \geq \underbrace{\theta f(z) + (1-\theta)f(z)}_{=f(z)} \\ + \theta f'(z) \cdot (x-z) + (1-\theta)f'(z) \cdot (y-z) \\ = f(z) + f'(z) \cdot (\theta x - \theta z) + f'(z) \cdot ((1-\theta)y - (1-\theta)z) \\ = f(z) + f'(z)(\theta x + (1-\theta)y - \theta z - (1-\theta)z) \\ = f(z) + f'(z)(z-z) = f(z) = f(\theta x + (1-\theta)y)\end{aligned}$$

In the higherdimensional case $f\colon D \to \mathbb{R}$ with $D \subseteq \mathbb{R}^n$ one can reduce to the onedimensional case as follows: Given two points $x, y \in D$, consider the composed function $\ell^{x,y}\colon [0,1] \to D \to \mathbb{R}$ given by $\theta \mapsto f(\theta x + (1-\theta)y)$. It should be clear that f is convex if and only if for every pair of points x, y the function $\ell^{x,y}$ is convex: After all, convexity is *defined* via line segments...

Remark 3.3.5. The statement of Prop. 3.3.3(a) says something remarkable: For a differentiable convex function, its first order Taylor approximation at any point gives a lower bound for all points! Thus local information at a single point can tell us something about all other points (i.e. global information). The next corollary is is an example of this.

Recall that a critical point of a differentiable function is point where the differential is zero.

Corollary 3.3.6. *A critical point of a convex differentiable function is a global minimum.*

Proof. Follows directly from Prop. 3.3.3(a). □

In particular a local minimum of a convex differentiable function is always a global minimum. This latter statement is actually true for *all* convex functions – a proof is given in Boyd/Vandenberghe' book, Section 4.2.2 (p.138).

Proposition 3.3.7. *Let $f\colon \mathbb{R}^n \supseteq D \to \mathbb{R}$ be a function and D a convex set then the following properties are equivalent to the definition Def. 3.3.1 of a convex function*

1. *for every $x, y \in D$ and $\theta \in [0,1]$ we have (Definition 3.3.1)*
$$f(\theta x + (1-\theta)y) \leq \theta f(x) + (1-\theta) f(y),$$
2. *if $x_1, \dots, x_n \in C$, $\theta_1, \dots, \theta_n \in \mathbb{R}$ with $\theta_i \geq 0$ and $\sum_i \theta_i = 1$ then*
$$f(\sum_i \theta_i x_i) \leq \sum_i \theta_i f(x_i),$$
3. *if $X \subset C$, $g\colon X \to \mathbb{R}$ with $g(x) \geq 0$ for all $x \in X$ and $\int_X g(x)dx = 1$ then*
$$f\left(\int_X g(x) x dx\right) \leq \int_X f(x) g(x) dx,$$
4. *if $X\colon \Omega \to \mathbb{R}^n$ is an $\mathbb{R}^n$-valued random variable and $P(X \in D) = 1$ then*
$$f(EX) \leq E(f(X)).$$
(here E denotes the expectation – if you forgot your probability theory, ignore this example)

All of these inequalities are called Jensen's inequality.

Convention about the domain: We define a new function $\tilde{f}\colon \mathbb{R}^n \to \mathbb{R} \cup \{\infty\}$ by
$$\tilde{f}(x) := \begin{cases} f(x) & x \in D \\ \infty & x \notin D \end{cases}.$$
and will often use $\tilde{f}$ instead of f, but still write "f".

Examples 3.3.8. (for working with $\tilde{f}$)

1. If f is convex then $\tilde{f}(\theta x + (1-\theta)y) \leq \theta \tilde{f}(x) + (1-\theta)\tilde{f}(y)$ for all $x, y \in \mathbb{R}$, hence also $\tilde{f}$ is convex.

2. For two functions $f_1\colon D_1 \to \mathbb{R}$ and $f_2\colon D_2 \to \mathbb{R}$ and their pointwise sum $f_1 + f_2\colon D_1 \cap D_2 \to \mathbb{R}$ we have
$$\tilde{f}_1 + \tilde{f}_2 = \widetilde{f_1 + f_2}.$$

Examples 3.3.9. (Convex functions)

1. Any two times differentiable function f with positive semidefinite Hessian matrix H_f, e.g. x^a with $a \geq 1, a \leq 0$.
2. Any norm $\|\cdot\| : \mathbb{R}^n \to \mathbb{R}$, since
$$\|\theta x + (1-\theta)y\| \leq \|\theta x\| + \|(1-\theta)y\| = \theta\,\|x\| + (1-\theta)\,\|y\|\,.$$
3. The function $f\colon \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} \mapsto \max\{x_1, \ldots, x_n\}$ is convex, since
$$\begin{aligned} f(\theta x + (1-\theta)y) &= \max_i\{\theta x_i + (1-\theta)y_i\} \\ &\leq \theta \max_i\{x_i\} + (1-\theta)\max_i\{y_i\} \\ &= \theta f(x) + (1-\theta)f(y), \end{aligned}$$
4. If $f_1, \ldots, f_n$ is convex and $w_1, \ldots, w_n \geq 0$ then $w_1 f_1 + \ldots + w_n f_n$ is convex, because
$$\begin{aligned} & w_1 f_1(\theta x + (1-\theta)y) + \ldots + w_n f_n(\theta x + (1-\theta)y) \\ & \leq w_1(\theta f_1(x) + (1-\theta)f_1(y)) + \ldots + w_n(\theta f_n(x) + (1-\theta)f_n(y)) \\ & = \theta(w_1 f_1(x) + \ldots + w_n f_n(x)) + (1-\theta)(w_1 f_1(y) + \ldots + w_n f_n(y)) \end{aligned}$$

   This extends to infinite sums and integrals: For all $y \in A$ let $f(x,y)$ be a convex function in x and $w(y) \geq 0$, then
$$g(x) := \int_A w(y) f(x,y) dy \quad \text{is convex.}$$
5. Let f be a convex function, then $g(x) := f(Ax + b)$ is convex.

6. If f_1, f_2 are two convex functions, then their pointwise maximum $g(x) := \max\{f_1(x), f_2(x)\}$ is convex. More generally speaking, if $x \mapsto f(x,y)$ is convex for all $y \in A$, then $g(x) := \sup_{y \in A} f(x,y)$ is a convex function. E.g. the Matrix norm: For fixed $u \in \mathbb{R}^m, v \in \mathbb{R}^n$ the function $\mathbb{R}^{m+n} \to \mathbb{R}$, $X \mapsto u^T X v$ is linear and therefore convex. Hence $\mathbb{R}^{m+n} \to \mathbb{R}$, $X \mapsto \sup\{u^T X v \mid \|u\| = \|v\| = 1\}$ is convex.

7. Let $f(x) := h(g(x))$ with $g\colon \mathbb{R}^n \supseteq D \to \mathbb{R}$, $h\colon \mathbb{R} \supseteq A \to \mathbb{R}$ being two functions. and h convex, then

   - if h is convex, $\tilde{h}$ is non-decreasing and g convex, then f is convex
   - if h is convex, $\tilde{h}$ is non-increasing and g concave, then f is convex
   - if h is concave, $\tilde{h}$ is non-decreasing and g concave, then f is concave
   - if h is concave, $\tilde{h}$ is non-increasing and g convex, then f is concave

   The first of these four propositions can be shown as follows: let $x, y \in \operatorname{dom} g$ and $g(x), g(y) \in \operatorname{dom} h$, then since g is convex we have

   $$g(\theta x + (1-\theta)y) \leq \theta g(x) + (1-\theta) g(y).$$

   h is assumed to be convex, therefore $\operatorname{dom} h$ is a convex set and hence the right hand side of this inequality lies in $\operatorname{dom} h$. Same is true for the left hand side, since $\tilde{h}$ is non-decreasing. We now can apply h to the above inequality and get

   $$\begin{aligned} h(g(\theta x + (1-\theta)y)) &\underbrace{\leq}_{h \text{ non.decr.}} h(\theta g(x) + (1-\theta) g(y)) \\ &\underbrace{\leq}_{h \text{ convex}} \theta h(g(x)) + (1-\theta) h(g(y)) \end{aligned}$$

Remark 3.3.10. Many functions of interest are not convex. There are some generalizations that still share good properties with convex functions and allow some techniques of convex optimization to be used. Examples (let $D \subseteq \mathbb{R}^n$ be a convex set):

(a) A function $f\colon D \to \mathbb{R}$ is *quasiconvex* if for every $x, y \in D$ and $\theta \in [0,1]$ we have $f(\theta x + (1-\theta)y) \leq \max\{f(x), f(y)\}$. Like for convexity, there are criteria for quasiconvexity in terms of the first and second order differentials, als well as in terms of the epigraph. See Boyd/Vandenberghe, Section 3.4.

(b) A function $f\colon \mathbb{R}^n \supseteq D \to \mathbb{R}$ is called *log-convex* if for all $x \in D$ we have $f(x) > 0$ and for all $x, y \in D$, $\theta \in [0,1]$ we have $f(\theta x + (1-\theta)y) \geq f(x)^\theta f(y)^{1-\theta}$. Equivalently $f(x) > 0$ for all $x \in D$ and $\log f$ is convex. This is useful in probability theory. See Boyd/Vandenberghe, Section 3.5.

(c) Once can generalize the meaning of "$\leq$" in the definition of a convex function, bringing lots of additional examples into the picture. See Boyd/Vandenberghe, Section 3.6.

3.4 Reminder/Crash course: Constrained optimization via Lagrange multipliers

The Lagrange multiplier method is not officially part of the course material. Instead it belongs to the things that we supposed you would know already. It will not be asked in the exam, but it can provide good intuition for the discussion of convex optimization problems that will follow.

Suppose we are looking for a maximum of a function $f\colon D \to \mathbb{R}$ on some domain D, and suppose there exists such a maximum at all (this is e.g. the case if D is bounded and closed). Then two different things can happen:

1. The maximum could be in the interior of D. In this case we know what to do: The maximum is necessarily at a critical point (i.e. a point $p \in D$ with $\nabla f(p) = 0$), so we look for critical points. For each critical point we check whether it is a maximum, e.g. by checking whether the Hesse matrix is negative definite.

2. The maximum could be on the boundary of D.

So to find a maximum on all of D, one simply checks for maxima in the interior, then for maxima on the boundary, then compares which ones are bigger. The question is how to find the maxima on the boundary.

The method of Lagrange multipliers can help to find the maximum or minimum of a function on a subset of $\mathbb{R}^n$ that is *given by an equation*. Very often the boundary of the domain D is given by an equation. For example, if we look for maxima on the unit ball in $\mathbb{R}^2$, $B_1(0) = \{x \in \mathbb{R}^2 \mid \|x\| \leq 1\}$ its boundary is given by

$$\{x \in \mathbb{R}^2 \mid \|x\| = 1\} = \{(x,y) \in \mathbb{R}^2 \mid x^2 + y^2 = 1\}.$$

I won't formulate the method in full generality or precision; I just give an example that I hope is guidance enough for solving exercises of this type, and for making sense of later discussions.

You do not have to memorize the procedure below, because it will be nicely packaged inside the general setup for convex optimization. But having seen it will help you understand the idea of that general procedure. The justification of the method is included in the discussion of the KKT conditions in Section 3.7.

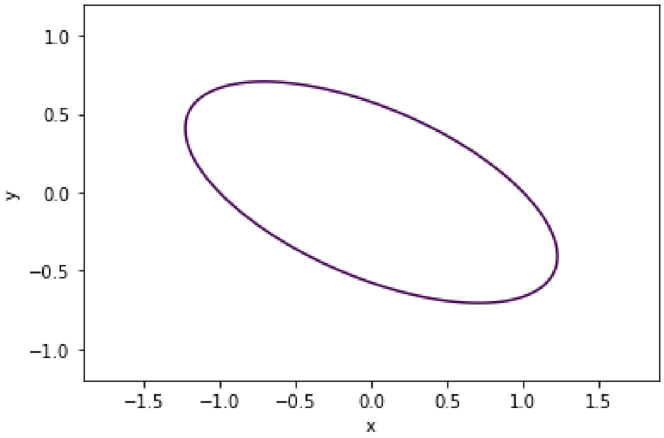
Suppose you want to find the points maximizing the function

$$f\colon \mathbb{R}^2 \to \mathbb{R}, \quad (x,y)^T \mapsto 2x + y$$

inside the set

$$S := \{(x,y)^T \in \mathbb{R}^2 \mid g(x,y) := x^2 + 2xy + 3y^2 = 1\}.$$

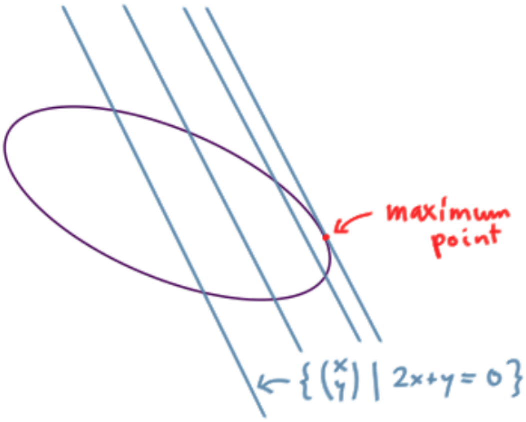
This set S is what is called a *level set*, i.e. the set of points where the function g takes a certain value.



The level set $S \subseteq \mathbb{R}^2$

(If you want to know how to draw this level set with Python, here is the code.)

We can draw the level sets of f into this picture (in this case they are straight lines, as f is linear). The goal of maximizing f inside S can be rephrased as finding the intersection points of the biggest value level set of f with S:

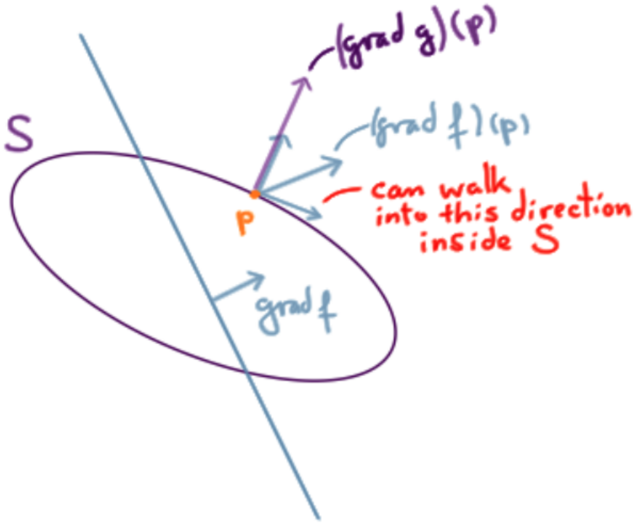


maximal point = intersection of maximal level set of f with S

How do we find this point (or in general *these points*, as there can be several)?

The graphical idea is that if you are at some point $p \in S$, to achieve a bigger value of f you want to walk into the direction of the *gradient of f* at p, $\nabla f(p)$, as that is the direction of biggest increment. But you are only allowed to walk inside of S!

So you can decompose the gradient of f, into a part that is tangent to the level set S and a part that is orthogonal to the level set S. Being orthogonal to S means precisely pointing into the direction of $\nabla g(p)$!



Walking along S to increase f.

Now you walk a bit into the direction of the tanget-to-S part of $\nabla f(p)$ and the value of f will increase. The only points p where you can not do that, are those where the tanget-to-S part of $\nabla f(p)$ is zero. Equivalently: those points where ∇f points into the direction of ∇g, or in equations: $\nabla f(p) = \lambda \nabla g(p)$ for some $\lambda \in \mathbb{R}$. Thus we look for points p satisfying this equation.

First we compute the gradients of f and g:

$$\nabla f(x,y) = \begin{pmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{pmatrix} = \begin{pmatrix} 2 \\ 1 \end{pmatrix} \qquad \nabla g(x,y) = \begin{pmatrix} \frac{\partial g}{\partial x} \\ \frac{\partial g}{\partial y} \end{pmatrix} = \begin{pmatrix} 2x+2y \\ 2x+6y \end{pmatrix}$$

Now we try and find the solutions of the equations

$$\begin{pmatrix} 2 \\ 1 \end{pmatrix} = \nabla f(x,y) = \lambda \nabla g(x,y) = \lambda \begin{pmatrix} 2x+2y \\ 2x+6y \end{pmatrix}$$

i.e. of

$$\begin{aligned} 2\lambda x + 2\lambda y &= 2 \qquad (*) \\ 2\lambda x + 6\lambda y &= 1 \qquad (**) \end{aligned}$$

These are two equations with the three indeterminates x, y, λ, but we also have a third equation that we can use:

$$x^2 + 2xy + 3y^2 = 1 \qquad (***)$$

From equation $(*)$ we get

$$x = \frac{1}{\lambda} - y$$

(note that equation $(*)$ implies that $\lambda \neq 0$, so we can divide by λ).

Substituting this expression for x into equation $(**)$ we get

$$\begin{aligned} 1 &= 2\lambda(\frac{1}{\lambda} - y) + 6\lambda y \\ &= 2 - 2\lambda y + 6\lambda y \ = \ 2 - 4\lambda y \\ \Rightarrow \quad & 4\lambda y = 1 \\ \Rightarrow \quad & y = \frac{1}{4\lambda} \end{aligned}$$

Substituting this back into our expression of x we get

$$x = \frac{2}{\lambda} - \frac{1}{4\lambda} = \frac{8-1}{4\lambda} = \frac{7}{4\lambda}$$

Now inserting our expressions for x and y in terms of λ into equation $(***)$ we get

$$\begin{aligned} 1 &= \left(\frac{7}{4\lambda}\right)^2 + 2\left(\frac{7}{4\lambda}\right)\left(\frac{1}{4\lambda}\right) + 3\left(\frac{1}{4\lambda}\right)^2 \\ &= \frac{49}{16\lambda^2} + 2\frac{7}{16\lambda^2} + 3\frac{1}{16\lambda^2} \\ &= \frac{49+14+3}{16\lambda^2} = \frac{66}{16\lambda^2} = \frac{33}{8\lambda^2} \end{aligned}$$

Hence we have $\lambda^2 = \frac{33}{8}$, i.e. the solutions

$$\lambda = \pm\sqrt{\frac{33}{8}} = \pm\frac{1}{2}\sqrt{\frac{33}{2}}$$

For $\lambda = \frac{1}{2}\sqrt{\frac{33}{2}}$ we get the point with coordinates

$$x_1 = \frac{7}{4\lambda} = \frac{7}{4}\frac{1}{\frac{1}{2}\sqrt{\frac{33}{2}}} = \frac{7}{2}\sqrt{\frac{2}{33}} \quad \text{and } y_1 = \frac{1}{4\lambda} = \frac{1}{2}\sqrt{\frac{2}{33}}$$

Likewise for $\lambda = -\frac{1}{2}\sqrt{\frac{33}{2}}$ we get the point with coordinates

$$x_2 = -\frac{7}{2}\sqrt{\frac{2}{33}} \quad \text{and } y_2 = -\frac{1}{2}\sqrt{\frac{2}{33}}$$

One sees that $f(x_1, y_1) = \frac{15}{2}\sqrt{\frac{2}{33}} > 0$ and $f(x_2, y_2) = -\frac{15}{2}\sqrt{\frac{2}{33}} < 0$, hence f achieves its maximum in S at the point (x_1, y_1).

Remark 3.4.1. Note: If we have $\nabla f(p) = \lambda \nabla g(p)$ with $\lambda > 0$, then the gradients of f and g point into the same direction, which means that we walked as far as we could to *in*crease f. Thus a solution to this equation will give us a local *maximum* of f under the constraint $g(x) = 0$.

If we have $\nabla f(p) = \lambda \nabla g(p)$ with $\lambda < 0$, then the gradients of f and g point into the same direction, which means that we walked as far as we could to *de*crease f. Thus a solution to this equation will give us a local *minimum* of f under the constraint $g(x) = 0$.

3.5 Optimization Problems

Definition 3.5.1. *An* optimization problem *(in standard form), also sometimes called a* nonlinear programming problem, *is a problem of the form*

$$\begin{array}{lll} \textit{minimize} & f(x) & \\ \textit{subject to} & g_i(x) \leq 0 & i = 1, \ldots, m \\ & h_j(x) = 0 & j = 1, \ldots, k \end{array}$$

The function f is called objective function *(or* cost function*). The (in)equalities $g_i(x) \leq 0$, resp. $h_j(x) = 0$, are called* inequality constraints, *resp.* equality constraints, *and the functions g_i, h_j are called the* constraint functions. *All of these functions are supposed to be defined on subsets of $\mathbb{R}^n$ and take values in $\mathbb{R}$.*

A point that lies in the domains of the objective and all constraint functions, and that satisfies all constraints, is called a feasible point. *The problem is called a* feasible problem, *if there is a feasible point. The* optimal value *is the number*

$$p^* := \inf\{f(x) \mid g_i(x) \leq 0, h_j(x) = 0\},$$

if it exists, otherwise we set $p^* = -\infty$, if the set is unbounded below, and $p^* = \infty$ if the set is empty. An optimal point *is a feasible point x^* such that $f(x^*) = p^*$.*

One can express the equality constraints equivalently by two inequalities, so one can always formulate an optimization problem without equalities.

Remark 3.5.2. In the machine learning literature optimal points are often expressed as

$$x^* = \mathtt{argmin}(f(x))$$

In general there can be *several* arguments minimizing the function f, a whole set of them! In this respect the notation $x^* \in \mathtt{argmin}(f(x))$ would be better, while the above notation with "$=$" suggests uniqueness. This slight inaccuracy is harmless in all cases I know, I just mention the abuse of notation so it doesn't confuse you in the future.

Remark 3.5.3. In practice one formulates optimization problems in the most readable way, and not necessarily strictly in the standard form of Def. 3.5.1; for example like this:

$$\begin{array}{rc} \text{maximize} & e^{-ax^2+4y} \\ \text{subject to} & 17 \leq x^2+y^2 \leq 23 \\ & x+y=6 \end{array}$$

But one can always reformulate such a problem into an equivalent one in standard form, by the following operations:

- change "maximize" into "minimize" and multiply the objective function by (-1)
- switch $\geq$ into $\leq$ by multiplying both sides with -1
- subtract the right hand side of an equality or an $\leq$-inequality to get 0 on one side

In the above example we would get the following result:

$$\begin{array}{rc} \text{minimize} & -e^{-ax^2+4y} \\ \text{subject to} & 17-(x^2+y^2) \leq 0 \\ & x^2+y^2-23 \leq 0 \\ & x+y-6=0 \end{array}$$

This reformulated problem clearly has the same set of feasible points and the same optimal point(s), and its optimal value is the negative of the optimal value of the original problem. The latter is unavoidable if one changes maximization into minimization, but clearly the information one gets out of a solution is the same.

There are other operations that do not change the set of feasible points and the optimal point(s), and even not the optimal value. For example one could square the equality constraint:

$$\begin{array}{ll} \text{minimize} & -e^{-ax^2+4y} \\ \text{subject to} & 17 - x^2 + y^2 \leq 0 \\ & x^2 + y^2 - 23 \leq 0 \\ & (x+y-6)^2 = 0 \end{array}$$

We do, however, *not* consider this an equivalent problem! The reason will become apparent in Sections 3.6 and 3.7. In short: from a formulation in standard form we can derive some associated functions and conditions, and also another optimization problem (the so-called *dual problem*) and these depend not just on the set of feasible points given by the constraints, but really on the choice of constraint functions!

How do you then know which formulation is the right one? The answer is: There is no right one! Instead, it is a good thing, and an element of freedom for you who want to optimize something, that you can choose different formulations of the essentially same problem: They give you different angles from which to view and solve the problem.

Definition 3.5.4. *A* convex optimization problem *is an optimization problem such that the functions $f, g_1, \dots, g_m$ are convex and the functions $h_1, \dots, h_k$ are affine, i.e. it is a problem of the form*

$$\begin{array}{lll} \textit{minimize} & f(x) & \\ \textit{subject to} & g_i(x) \leq 0 & i = 1, \dots, m \\ & a_j^T x - b_j = 0 & j = 1, \dots, p \end{array}$$

With convex functions $f, g_1, \dots, g_m$, vectors $a_j \in \mathbb{R}^n$ and numbers $b_j \in \mathbb{R}$.

It may look puzzling that we ask the inequality constraints to be convex but the equality constraints to be affine. The next two items will explain this.

Proposition 3.5.5. *Let $D \subseteq \mathbb{R}^n$ $g\colon D \to \mathbb{R}$ be a convex function and $a \in \mathbb{R}$. Then the sublevel set $S_a^g := g^{-1}((-\infty, a]) := \{x \in D \mid g(x) \leq a\}$ is convex.*

Proof. Let $x, y \in S_a$ and $\theta \in [0,1]$. Then

$$g(\theta x + (1-\theta)y) \leq \theta g(x) + (1-\theta)g(y) \leq \theta a + (1-\theta)a = a$$

So $\theta x + (1-\theta)y \in S_a$. Hence S_a is convex. □

Corollary 3.5.6. *The set of feasible points of a convex optimization problem is convex.*

Proof. Since the objective and constraint functions are convex, their domains are convex, and so is the intersection of all their domains – call it D. The solution set of each equality constraint $h_j(x) = 0$ is a hyperplane – call it H_j – and hence convex. The solution set of each inequality constraint $g_i(x) \leq 0$ is the sublevel set $S_0^{g_i}$ and convex by Prop, 3.5.5. The set of feasible points is the intersection

$$\mathcal{F} := D \cap S_0^{g_1} \cap \ldots S_0^{g_m} \cap H_1 \cap \ldots \cap H_k$$

and therefore convex. □

Thus the reason that we ask the equality constraint functions to be convex is, in short, that the only functions whose level sets are convex are affine functions.

Alternatively, if we reformulate the problem as one with only inequalities, each equality constraint function will appear twice, once as it is, and once with a minus sign. Both of these should be convex. But the only convex functions g such that $-g$ is convex too, are the affine functions.

So convex optimization problems are indeed precisely problems of minimizing a convex function on a convex set, as stated in the introduction to this chapter.

Definition 3.5.7. *A* linear optimization problem, *or* linear programming problem, *is an optimization problem such that the functions $f, g_1, \ldots, g_m, h_1, \ldots, h_k$ are affine*

Remark 3.5.8. The term "programming" in the (non-)linear programming problems does not refer to programming a computer. It rather comes from "program" in the sense of "cinema program", i.e. it refers to scheduling and planning. This is because linear programming was invented (by

George Dantzig) to optimize the logistics of the US Air Force (at a time when computers were still very uncommon).

In case you have heard the folklore story of a professor writing an open problem on the blackboard, and a student arriving late at class, thinking it's the homework for that week and solving it – it's true, that student was George Dantzig and it was actually *two* open problems.

Example 3.5.9. Imagine a casino owner trying to invent a game in which players can bet on events $E_1, \ldots, E_n$. The revenue of the casino in case of event E_i is $u_i \in \mathbb{R}$ (it could be negative, if the customer wins and the casino loses). The revenues form a vector $u = (u_1, \ldots, u_n)$ Now the owner wants to set up the game such that events $E_1, \ldots, E_n$ happen with probabilities $p_1, \ldots, p_n$, respectively. The vector of probabilities $p = (p_1, \ldots, p_n)$ should be chosen such that the expected revenue $E(p) = \sum_{i=1}^{n} p_i u_i = p^T u$ is maximal. This is the optimization problem

$$\begin{array}{ll} \text{maximize} & E(p) \\ \text{subject to} & p_i \geq 0, \;\; p_i \leq 1 \qquad i = 1, \ldots, m \\ & p_1 + \ldots + p_n = 1 \end{array}$$

To bring the problem into the standard form prescribed by Def. 3.5.1, we have to rewrite it as follows:

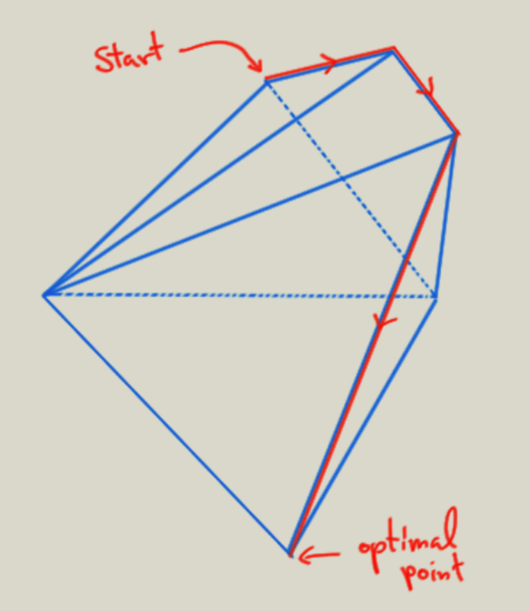
$$\begin{array}{ll} \text{minimize} & -E(p) \\ \text{subject to} & -p_i \leq 0, \;\; p_i - 1 \leq 0 \quad i = 1, \ldots, m \\ & (1, \ldots, 1) \cdot p - 1 = 0 \end{array}$$

Note that to get the problem into the form of the definition we have to introduce minus signs in several places and subtract the bounds of inequalities. The function $E(p)$ is linear, and so are the projections $\mathrm{pr}_i \colon (p_1, \ldots, p_n) \mapsto p_i$, and thus $\mathrm{pr}_i - 1$ are affine functions. Therefore the above problem is a linear optimization problem in the sense of Def. 3.5.4. It is not a very interesting one because the solution would just be to assign probability 1 to the event with the biggest revenue, however, the casino story would more realistically come with extra constraints like a minimum variance to keep the game interesting.

Remark 3.5.10. Linear optimization problems are abundant and solving them is a powerful technique employed in many areas, for example everywhere in economy, but also in chemical engineering, network design, biology and game theory.

Linear optimization problems are the most tractable of all optimization problems. They can for example be solved by the so-called simplex algorithm:

It should be intuitively plausible (and it can be proved) that a linear function, if it attains a maximum on a polyhedron, then that maximum is attained on a vertex. To find an optimal point, one first has to find a vertex; vertices are the points where all the constraints are satisfied and a maximal number of the inequalities are actual equalities. I a vertex is not an optimal point, then there is an edge leading to a vertex where the objective function has a higher value. One travels to the other vertex on that edge and repeats.



There is a lot more to be said about linear optimization, e.g. there are methods to reformulate the problem into a linear optimization problem where all constraints are given by equalities, and to solve that problem by simple matrix manipulations.

In the master program AI and Data Science there is a course on linear optimization that is frequently offered and where you can learn more about this.

There can be many equivalent ways of expressing an optimization problem, in the sense that solutions of one problem can be easily transformed into solutions of the other – see section 4.1.3 (page 130) of Boyd/vandenberghe's book for a long list of examples.

We finish the section by revisiting the Lagrange multiplier method for convex optimization problems:

Proposition 3.5.11. *Consider the convex optimization problem*

$$\begin{array}{lll} \textit{minimize} & f(x) & \\ \textit{subject to} & g_i(x) \leq 0 & i = 1, \dots, m \\ & a_j^T x - b_j = 0 & j = 1, \dots, p \end{array}$$

Let M be the set of feasible points. Suppose that the objective function f is differentiable. Then a point $x \in M$ is optimal if and only if

$$\nabla f(x)^T (y - x) \geq 0 \quad \textit{for all } y \in M \qquad (*)$$

Proof. By the first order condition for convexity of Prop. 3.3.3, we have for all x, y in the domain of f:

$$f(y) \geq f(x) + \nabla f(x)^T (y - x)$$

Thus if $x \in M$ satisfies the condition $(*)$, then clearly we have $f(y) \geq f(x)$ for all $y \in M$.

To prove the converse, suppose that x does not satisfy $(*)$, i.e. there exists a $y \in M$ with $\nabla f(x)^T (y - x) < 0$. Since M is convex, for all $\theta \in [0, 1]$ the points $\theta y + (1 - \theta) x$ are in M as well. Put differently, we have a path $z \colon [0, 1] \to M$, $z(t) := \theta y + (1 - \theta) x$. The (one-sided) differential of the composed function $f(z(t))$ at $t = 0$ is given by

$$(\frac{d}{dt} f(z(t)))(0) = \lim_{t \to 0} \frac{f(z(t)) - f(x)}{t} = (\nabla f)(x)^T (y - x) < 0$$

Here in the last equality we get the derivative as directional derivative by multiplying the gradient with a vector in the direction of the path. This implies that for a small enough $t > 0$ we have $f(z(t)) < f(x)$ and thus $f(x)$ is not a minimum. □

Prop. 3.5.11 reproduces the method of Lagrange multipliers in the case of a convex optimization problem: The condition

$$\nabla f(x)^T (y - x) \geq 0 \quad \text{for all } y \in M \qquad (*)$$

can be satisfied precisely if either $\nabla f(x) = 0$, in which case we have a critical point and therefore a global minimum by Cor. 3.3.6, or $\nabla f(x) \neq 0$ and in this case x must be on the boundary of M, and all $y \in M$ must be in the positive halfspace defined by $\nabla f(x)$, i.e. $\nabla f(x)$ defines a tangent line to M. These are precisely the kinds of points eligible for minima according to the Lagrange multiplier method.

3.6 Lagrange duality

Definition 3.6.1. *Consider the optimization problem*

$$\begin{array}{lll} \textit{minimize} & f(x) & \\ \textit{subject to} & g_i(x) \leq 0 & i = 1, \ldots, m \\ & h_j(x) = 0 & j = 1, \ldots, p \end{array}$$

Let $D := (\operatorname{dom} f) \cap (\operatorname{dom} g_1) \cap \ldots \cap (\operatorname{dom} g_m) \cap (\operatorname{dom} h_1) \cap \ldots \cap (\operatorname{dom} h_p) \subseteq \mathbb{R}^n$ *be the biggest set where the objective function and all constraint functions are defined. Let* $\mathcal{F} \subseteq D$ *be the set of feasible points.*

We define, for $x \in D, \lambda \in \mathbb{R}^m, \nu \in \mathbb{R}^p$ *the* Lagrangian

$$L(x, \lambda, \nu) := f(x) + \sum_{i=1}^{m} \lambda_i g_i(x) + \sum_{j=1}^{p} \nu_j h_j(x)$$

and the Lagrange dual function *as*

$$q(\lambda, \nu) := \inf_{x \in D} L(x, \lambda, \nu)$$

The vectors λ *and* ν *are called the* dual variables*.*

For each fixed x, the function $L(x, \lambda, \nu)$ is affine, and hence the Lagrange dual function, being an infimum of affine (hence concave) functions, is concave. The domain of the Lagrange dual function g is the subset of $\mathbb{R}^m \times \mathbb{R}^p$ where the infimum defining g exists.

The Lagrange dual function gives lower bounds for the optimal value p^* of the optimization problem.

Proposition 3.6.2. *For all* $\lambda \in \mathbb{R}^m$ *with* $\lambda_i \geq 0$ *and all* $\nu \in \mathbb{R}^p$ *we have* $q(\lambda, \nu) \leq p^*$

Proof. Let λ, ν be as in the statement. For every *feasible* point $x \in \mathcal{F}$ we have $g_i(x) \leq 0$, hence $\lambda_i g_i(x) \leq 0$. and $h_i(x) = 0$. Therefore

$$L(x, \lambda, \nu) = f(x) + \underbrace{\sum_{i=1}^{m} \lambda_i g_i(x)}_{\leq 0} + \underbrace{\sum_{j=1}^{p} \nu_j h_j(x)}_{=0} \leq f(x)$$

and thus $q(\lambda, \nu) = \inf_{x \in D} L(x, \lambda, \nu) \leq \inf_{x \in \mathcal{F}} L(x, \lambda, \nu) \leq \inf_{x \in \mathcal{F}} f(x) = p^*$ □

In view of Prop. 3.6.2 it is natural to ask for the best lower bound that the dual function can give us. This constitutes the dual problem of a given optimization problem:

Definition 3.6.3. *Given an optimization problem with dual function $q(\lambda, \nu)$, the* Lagrange dual problem, *or simply* dual problem, *is the optimization problem*

$$\begin{array}{ll} \textit{maximize} & q(\lambda, \nu) \\ \textit{subject to} & \lambda_i \geq 0 \end{array}$$

A point $(\lambda, \nu) \in \mathbb{R}^m \times \mathbb{R}^p$ is feasible for the dual problem *if $\lambda_i \geq 0$ and $q(\lambda, \nu) \neq -\infty$. We denote by d^* the optimal value of the dual problem.*

The original problem is called the primal problem, *to distinguish it from the dual problem.*

By the remark after Def. 3.6.1, the dual function is concave, so the dual problem is always a concave maximization problem (equivalently: after bringing it into standard form it becomes a convex optimization problem).

The following fact, called weak duality, is an immediate consequence of Prop. 3.6.2:

Fact 3.6.4 ("weak duality"). $d^* \leq p^*$

Definition 3.6.5. *The number $p^* - d^*$ is called the* duality gap *of the original problem. If it is zero, i.e. if $d^* = p^*$, then we say that* strong duality *holds for the original problem.*

Remark 3.6.6. Remember the chain of inequalities with which we established weak duality in the proof of Prop. 3.6.2:

$$q(\lambda, \nu) = \inf_{x \in D} L(x, \lambda, \nu) \leq \inf_{x \in \mathcal{F}} L(x, \lambda, \nu) \leq \inf_{x \in \mathcal{F}} f(x) = p^*$$

Note that $\inf_{x \in \mathcal{F}} L(x, 0, 0) = \inf_{x \in \mathcal{F}} f(x) = p^*$. Thus, if in the definition of the dual function we would only search through the set of feasible points $\mathcal{F}$, there would be no duality gap. In other words, the duality gap exclusively arises from searching through the bigger domain D, and the possibility of achieving lower values of the Lagrangian outside of $\mathcal{F}$.

The benefit of considering the dual problem ist that it is often easier to solve – for example it is always convex – and it still gives information about the original problem: At least a lower bound for the optimal value, and in the case of strong duality even the exact optimal value.

It is therefore of interest when we have strong duality.

The following is an important class of examples from linear optimization.

Example 3.6.7. For a feasible *linear* optimization problem, one can show that the dual of the dual problem is the primal problem. Weak duality gives us $d^* \leq p^*$, if we see p^* as the optimal value of the primal problem. On the other hand weak duality gives us $p^* \leq d^*$ if we see p^* as the optimal value for the dual of the dual. Therefore linear optimization problems have strong duality.

Theorem 3.6.8 (Slater's condition)**.** *Suppose the primal problem is convex and there exists a feasible point x for which the inequalities involving non-affine functions hold strictly (i.e. $g_i(x) < 0$ for all g_i except possibly the ones which are affine functions). Then we have strong duality and the optimal values are actually attained. In other words: there are a primal feasible x^* and a dual feasible (λ^*, ν^*) such that $f(x^*) = q(\lambda^*, \nu^*)$.*

Proof. For a proof see Boyd/Vandenberghe, Convex Optimization, Section 5.3.2. □

Note that the feasible point occurring in the above statement is not required to have any relation to the optimal points – we just want something to be there in the interior of our convex set of feasible points.

Note: Different formulations of the same primal problem can lead to different dual problems:

Example 3.6.9. Consider the problem

$$\begin{array}{rc} \text{minimize} & x - \log x \\ \text{subject to} & x \leq 4 \end{array}$$

The Lagrangian is

$$L(x, \lambda) = x - \log x + \lambda(x - 4)$$

The first and second derivatives of $L(x, \lambda)$ along x are $\frac{\partial}{\partial x} L(x, \lambda) = 1 - \frac{1}{x} + \lambda$ and $\frac{\partial}{\partial x \partial x} L(x, \lambda) = \frac{1}{x^2}$. The latter is always > 0, so for every λ the function $x \mapsto L(x, \lambda)$ is convex, and takes its minimum at the place where the first derivative is 0.

Solving $0 = \frac{\partial}{\partial x}L(x,\lambda) = 1 - \frac{1}{x} + \lambda$ for x gives $x = \frac{1}{1+\lambda}$. Thus the Lagrange dual function is

$$\begin{aligned} q(\lambda) &= \inf_x L(x,\lambda) = L(\frac{1}{1+\lambda},\lambda) \\ &= \frac{1}{1+\lambda} - \log(\frac{1}{1+\lambda}) + \frac{\lambda}{1+\lambda} - 4\lambda \\ &= 1 - 4\lambda + \log(1+\lambda) \end{aligned}$$

The problem

$$\begin{array}{ll} \text{minimize} & f(x) := x - \log x \\ \text{subject to} & \quad x^2 \leq 16 \end{array}$$

has the same objective function and the same set of feasible points as the first problem: The logarithm is only defined for positive real numbers and for $x > 0$ the conditions $x^2 \leq 16$ and $x \leq 4$ are equivalent.

The Lagrangian is $L(x,\lambda) = x - \log x + \lambda(x^2 - 16)$. We have $\frac{\partial}{\partial x}L(x,\lambda) = 1 - \frac{1}{x} + 2x\lambda$ and $\frac{\partial}{\partial x \partial x}L(x,\lambda) = \frac{1}{x^2} + 2\lambda$, which is > 0 for $\lambda \geq 0$ (which is the only case we consider because of the dual feasibility condition).

Thus again we find the minimum by solving $0 = \frac{\partial}{\partial x}L(x,\lambda) = 1 - \frac{1}{x} + 2x\lambda$, which is equivalent (since $x \neq 0$) to $0 = 2x^2\lambda + x - 1$, which is equivalent to $x^2 + \frac{1}{2\lambda}x - \frac{1}{2\lambda}$. We get

$$x = -\frac{1}{4\lambda} \pm \sqrt{\frac{1}{16\lambda^2} + \frac{1}{2\lambda}} = \frac{-1 \pm \sqrt{1+8\lambda}}{4\lambda}$$

Since $\frac{-1-\sqrt{1+8\lambda}}{4\lambda} < 0$ and we only consider positive x, the minimum of $L(-,\lambda)$ is attained at $x = \frac{-1+\sqrt{1+8\lambda}}{4\lambda}$(which is > 0) and its value is

$$\begin{aligned} q(\lambda) = L(\frac{-1+\sqrt{1+8\lambda}}{4\lambda},\lambda) &= \frac{-1+\sqrt{1+8\lambda}}{4\lambda} - \log\left(\frac{-1+\sqrt{1+8\lambda}}{4\lambda}\right) \\ &\quad + \lambda(\frac{1}{16\lambda^2} - \frac{\sqrt{1+8\lambda}}{8\lambda^2} + \frac{1+8\lambda}{16\lambda^2} - 16) \end{aligned}$$

Example 3.6.10. Consider an unconstrained problem of the form

$$\text{minimize} \quad f(Ax+b)$$

Since there are no constraints, the dual function depends on no variables, i.e. it is constant and has value p^* (the optimal value of the primal problem). In particular we have strong duality, but that is not very useful.

On the other hand, reformulating the problem as

$$\begin{array}{ll} \text{minimize} & f(z) \\ \text{subject to} & Ax + b = z \end{array}$$

we obtain the Lagrangian

$$L(x, z, \nu) = f(z) + \nu^T(Ax + b - z).$$

The dual function $q(\nu)$ is obtained by minimizing over x and z. For any fixed z, minimization over x gives $-\infty$, unless $\nu^T A = 0$. If $\nu^T A = 0$, the minimization gives

$$q(\nu) = b^T \nu + \inf_z \{ f(z) - \nu^T z \}$$

Thus the dual problem becomes

$$\begin{array}{ll} \text{maximize} & b^T\nu + \inf_z\{f(z) - \nu^T z\} \\ \text{subject to} & A^T \nu = 0 \end{array}$$

which is potentially more useful. An example of this will appear in an exercise.

3.6.1 Duality gaps at pairs of feasible points

Given a dual feasible point (λ, v) (i.e. $\lambda_i \geq 0$), we know that $p^* \geq q(\lambda, v)$, and therefore $-p^* \leq -q(\lambda, v)$. If we have a primal feasible x, then adding $f(x)$ on both sides gives us

$$f(x) - p^* \leq f(x) - q(\lambda, v)$$

Here the right hand side is what we can compute without already having solved the optimization problem. It is called *the duality gap at* $(x, (\lambda, v))$. The left hand side tells us how far we still are, at the point x, from the optimal value of the primal problem.

Put differently, since we always have

$$q(\lambda, v) \leq d^* \leq p^* \leq f(x)$$

we obtain an interval in which the primal and dual optimal values must lie,

$$p^*, d^* \in [q(\lambda, v), f(x)]$$

and the duality gap at $x, (\lambda, v)$ is the length of this interval.

This suggests an "algorithm" for approximating the optimal value:

1. Choose an $\epsilon > 0$, a distance within which you want to be from the optimal value.

2. Then randomly choose feasible starting points x_1 and (λ_1, v_1).

3. Perform gradient descent with respect to f, produce a sequence of feasible points x_n, with descending values $f(x_n)$. Simultaneously, by gradient ascent with respect to g produce a sequence (λ_n, v_n) with growing values $q(\lambda_n, v_n)$.

4. When $f(x_n) - q(\lambda_n, v_n) \leq \epsilon$ stop - you now know that $f(x_n)$ is less than ϵ away from the optimal value.

Of course, for this algorithm to terminate for any ϵ, we need strong duality, e.g. by Slater's condition. More seriously, we need to specify how precisely we want to do the gradient descent; for example, we have to make sure that it always gives us feasible points. The usual problem of gradient descent is that one can get stuck in local minima without even knowing it. In the presence of strong duality, we now do have a way of knowing (or at least suspecting) it: If our x-values don't change much for a while and maybe even oscillate around a point, but the duality gap stays big, then we might be stuck in a local minimum and can for example try starting the gradient descent randomly in a different region, or increase step size to get out of the local valley.

Remark 3.6.11. Even for convex functions formulating the dual problem and monitoring the duality gap can be useful in practice:

If we were to perform gradient descent without the dual problem, we need to formulate a stopping criterion, saying when we are content with our current achieved value. Not knowing the optimal value, the only criterion that we have is the size of the gradient. When the gradient is very close to 0, smaller than some previously chose ϵ, we might suspect that we are close to the unique critical point and say that we stop.

But consider the convex function $x \mapsto -\log x$. Its derivative is $x \mapsto -\frac{1}{x}$, which converges to 0 very fast, and we would quickly stop. But $-\log x$ is unbounded below, and we could get an arbitrarily better value by continuing our gradient descent. One can make this into a real example, with an actual optimal value, either by bounding the domain or by changing the definition of the function for high enough x, from $\log x$ to something that eventually increases.

3.6.2 Conceptual understanding of Lagrange duality

Why, of all functions, is the Lagrangian an appropriate function for getting information about a given optimization problem? The results about the Lagrangian and the dual problem justify their use, but is there a systematic way on which one could arrive at the Lagrangian?

It is a general idea in mathematics to view a problem that one wants to solve, as one in a family of problems. This can allow to approach the original problem from different angles.

Embedding the objective function into a family of objective functions

One way of seeing a given optimization problem as one in a family of problems, is to see the objective function as one in a parametrized family of functions. That is, one chooses a function $\phi(x, \lambda)$ of two variables (or two groups of variables) such that $f(x) = \phi(x, 0)$. We can then invoke the general max-min inequality:

Proposition 3.6.12 (max-min inequality). *Let $\phi\colon X \times Y \to \mathbb{R}$ be any function. Then*

$$\sup_{y \in Y} \inf_{x \in X} f(x, y) \leq \inf_{x \in X} \sup_{y \in Y} f(x, y).$$

Proof. Define $q(y) := \inf_{x \in X} \phi(x, y)$. Then we have

$$\begin{aligned}
&\forall x\ \forall y\colon \quad q(y) \leq \phi(x, y) \\
&\Rightarrow \quad \forall x\colon \ \sup_y q(y) \leq \sup_y \phi(x, y) \\
&\Rightarrow \quad \sup_y q(y) \leq \inf_x \sup_y \phi(x, y) \\
&\Rightarrow \quad \sup_y \inf_x \phi(x, y) \leq \inf_x \sup_y \phi(x, y)
\end{aligned}$$

□

In the case of the Lagrangian, if we start with $X = \mathcal{F}$, the set of feasible points, and $Y = (\mathbb{R}_{\geq 0})^m \times \mathbb{R}^p$ (for m inequality constraints and p equality constraints), we obtain

$$\sup_{(\lambda,\nu)\in Y} \inf_{x\in\mathcal{F}} L(x,\lambda,\nu) \leq \inf_{x\in\mathcal{F}} \sup_{(\lambda,\nu)\in Y} L(x,\lambda,\nu)$$

Now for a feasible $x \in \mathcal{F}$ we have $\sup_{(\lambda,\nu)\in Y} L(x,\lambda,\nu) = \sup_{(\lambda,\nu)\in Y} f(x) + \sum_{i=1}^m \lambda_i g_i(x) + \sum_{j=1}^p \nu_j h_j(x) = f(x)$, because for a feasible point x we have

- $h_j(x) = 0$, so the ν_j don't matter at all,
- $g_i(x) \leq 0$, so $\lambda_i = 0$ achieves the maximum among all $\lambda_i \geq 0$

Thus in the case of the Lagrangian we obtain the objective function f back on the right hand side. Now, as mentioned in Remark 3.6.6, we can attain equality by setting $\lambda = \nu = 0$ on the left hand side: $L(x,0,0) = f(x)$, so $\inf_{x\in\mathcal{F}} L(x,0,0) = p^*$. Thus the max-min equality in itself is not very useful here, but we arrive at Lagrange duality by relaxing the requirement $x \in \mathcal{F}$ and allowing all $x \in D$...

You can maybe imagine that there are other possibilities of exploiting the general idea of tweaking max-min inequality into being useful. For an example, see chapter 3 of these Optimization course notes by Parrilo and Lall.

Thus one answer to the question why the Lagrangian is the right choice for a family of functions containing our objective function is: It is not *the* right choice – there are also others.

Another way of understanding Lagrange duality is to see the dual problem as a *perturbation* of the original problem, arising by varying the constraints.

From this point of view it is very natural that different formulations of a given optimization problem give different Lagrangians: A formulation of an optimization problem specifies not just the problem, but also directions of perturbation.

We discuss this point of view in two ways:

Lagrangian as relaxation of the constraints 1

The method of Lagrange multipliers from Section 3.4 is incorporated into the Lagrangian and the search for its infimum in x. For simplicity consider a problem with a single ineqality constraint:

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & g(x) \leq 0 \end{array}$$

with f and g differentiable functions (but not necessarily convex). In this case the Lagrangian becomes $L(x, \lambda) = f(x) + \lambda g(x)$. If x^* is an optimal point for the problem, then there are two possibilities: $g(x) < 0$ (i.e. the point lies in the interior of the feasible region) or $g(x) = 0$.

In the first case $g(x) < 0$, we can find the minimum just by searching critical points, i.e. solutions of $\nabla f(x) = 0$, as you would do it for an unconstrained problem. In this case, equivalently, we simply treat the function $L(x, 0) = f(x)$ as an unconstrained function to be minimized by looking for critical points.

In the second case $g(x^*) = 0$, our optimal point lies on the boundary of the feasible region. For finding such a point we have the method of Lagrange multipliers. For a fixed $\lambda > 0$ we can again search minima of $L(x, \lambda)$ by looking for critical points:

$$(\nabla L)(x^*, \lambda) = \nabla f(x^*) + \lambda \nabla g(x^*) = 0$$

Rewriting this condition gives $\nabla f(x^*) = -\lambda \nabla g(x^*)$. For *positive* λ, this means that $\nabla f(x^*)$ and $\nabla g(x^*)$ point in *opposite* directions, which is exactly the condition for finding a minimum – see the last paragraph of Section 3.4. *This is the reason why we put the constraint $\lambda \geq 0$ into the dual problem!* The dual function is exactly $q(\lambda) = \inf_x L(x, \lambda)$, it gives back that minimum in question.

However, in finding critical points of the function $x \mapsto L(x, \lambda)$ for a fixed λ we do not necessarily only get points with $g(x) = 0$: The equality $\nabla f(x^*) = -\lambda \nabla g(x^*)$ gives us n equations in the n variables of $x = (x_1, \ldots, x_n)$. In Section 3.4 we insisted on $g(x) = 0$ and with this had an extra equation to pin down the additional variable λ. Now, on the contrary, we have λ fixed, can use it to pin down x and then compute $g(x) = c$. Nothing forces $c \leq 0$.

It is in this sense that the dual function embodies a relaxation of the original problem: We now also consider x that lie outside the feasible points.

Lagrangian as relaxation of the constraints 2

Another view point on the Lagrangian, explaining the interaction with *in*equality constraints, is that it arises by relaxing the hard constraints on our given problem by "soft constraints":

Define the indicator function of the nonpositive real numbers by

$$I_{\leq 0}(r) := \begin{cases} 0 & \text{if } r \leq 0 \\ \infty & \text{if } r > 0 \end{cases}$$

Then the optimization problem

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & g_i(x) \leq 0 \end{array}$$

can be equivalently rewritten as the unconstrained problem

$$\text{minimize} \quad f(x) + \textstyle\sum_i I_{\leq 0}(g_i(x))$$

Indeed, the indicator function becomes ∞ outside the set of feasible points, so that this function does not take its minimum there. It becomes 0 inside the set of feasible points, so that there we have the original objective function.

One could say that $I_{\leq 0}$ places a penalty on points outside the feasible set, in fact the harshest possible penalty. Now for given $\lambda_i \geq 0$ the function

$$f(x) + \sum_i \lambda_i(g_i(x))$$

arises from the original objective function by adding terms that one can also understand as placing a penalty in infeasible points (and causing a preference for feasible points): For feasible points these new terms become negative (or, on the boundary, zero) and for infeasible points they become positive. Thus if the goal is to minimize this function, points in the feasible sets are more likely to achieve it, but infeasible points are also considered.

The larger the λ_i, the more weight is given to these penalty terms, and the indicator function *outside the feasible set* can be seen as the limit for $\lambda_i \to \infty$.

From this viewpoint, one could say that in considering the Lagrangian we allow ourselves to step over the boundary of the set of feasible points and also take into account the behaviour of our functions "out there".

Geometric viewpoints

First, here is a source with great pictures of what the functions from Lagrange duality do:

Lagrange duality – Blog post by Zhiliang Zhou.

A geometric interpretation of the dual function, also providing insight into what the dual problem tells us, is given in these lecture notes by Tibshirani – they contain great graphics (alternatively see Boyd/Vandenberghe, Convex Optimization, Section 5.3). This geometric point of view is tightly connected with conjugate functions, which are treated in Section 3.8, but not in enough depth to get the full picture. It is also the best way of understanding Slater's condition. Unfortunately there is no time in this course to dwell on this.

The top voted answer at this forum post wonderfully explains the dual problem via conjugate functions, without going much into geometry:

Intuition behind the dual problem – Forum post by littleO

The other answers there are also worth checking out.

3.7 The KKT conditions

Suppose we have the primal optimal point x^* and the dual optimal point (λ^*, v^*), and suppose that we have strong duality. Then we have the following inequalities:

$$\begin{aligned} f(x^*) = q(\lambda^*, v^*) &= \inf_{x \in D} \left(f(x) + \sum_{i=0}^{m} \lambda_i^* g_i(x) + \sum_{j=1}^{p} v_j^* h_j(x) \right) \\ &\overset{(A)}{\leq} f(x^*) + \sum_{i=0}^{m} \lambda_i^* g_i(x^*) + \sum_{j=1}^{p} v_j^* h_j(x^*) \\ &\overset{(B)}{\leq} f(x^*) \end{aligned}$$

The inequality (A) holds, because the feasible point x^* occurs in the set D over which the infimum is taken.

The inequality (B) holds for the following reasons: As x^* is feasible, we have $h_j(x^*) = 0$ and the last group of summands is zero. Since $g_i(x^*) \leq 0$ (x^* being feasible), and $\lambda_i \geq 0$ ((λ, v) being feasible), the middle summands are at ≤ 0.

Now since we have the same terms at the beginning and the end of the chain of inequalities, all the inequalities must in fact be equalities.

This implies $\sum_{i=1}^m \lambda_i^* g_i(x^*) = 0$. Since each each summand is ≤ 0, in fact each summand must be $= 0$:

$$\lambda_i g_i(x^*) = 0 \quad (i = 1, \ldots, m)$$

This condition is called *complementary slackness*. The name may be better explained by rephrasing the condition as

$$\lambda_i > 0 \;\Rightarrow\; g_i(x^*) = 0 \quad \text{and} \quad g_i(x^*) < 0 \;\Rightarrow\; \lambda_i = 0,$$

which says that if the dual constraint is satisfied with some slack, i.e. not lying exactly on the boundary, then the primal constraint can not have slack, and vice versa.

Now suppose additionally that $f, g_1, \ldots, g_m, h_1, \ldots, h_p$ are differentiable. Since the inequality (A) is actually an equality, we know that x^* minimizes the function $L(x, \lambda^*, v^*)$ (as a function just of x, with the fixed optimal dual values λ^*, v^*).

This implies that the gradient (with respect to x) is zero at x^*:

$$0 = \big(\nabla L(-, \lambda^*, v^*)\big)(x^*) = \nabla f(x^*) + \sum_{i=1}^m \lambda_i^* \big(\nabla g_i\big)(x^*) + \sum_{j=1}^p v_j^* \big(\nabla h_j\big)(x^*)$$

Definition 3.7.1. *The* Karush-Kuhn-Tucker conditions, *or* KKT conditions, *are the following conditions that must hold at every pair $x^*, (\lambda^*, v^*)$ of optimal primal and dual points:*

$$\begin{aligned} g_i(x^*) &\leq 0 && \textit{(x^* primal feasible)} \\ h_j(x^*) &= 0 && \\ \lambda_i^* &\geq 0 && \textit{((λ, v) dual feasible)} \\ \lambda_i g_i(x^*) &= 0 && \textit{(complementary slackness)} \\ \nabla L(-, \lambda^*, v^*)(x^*) &= 0 && \textit{(gradient condition)} \end{aligned}$$

As we proved in this section, the KKT conditions are always necessary for a pair pf primal and dual points to be optimal. For convex problems they are also sufficient:

Proposition 3.7.2. *Suppose the primal problem is convex with differentiable objective and constraint functions. If $\tilde{x}$, $(\tilde{\lambda}, \tilde{v})$ satisfy the KKT conditions, then they are primal, resp. dual, optimal points with duality gap zero.*

Proof. First, since the primal problem is convex, the functions h_j are affine, so the function $\sum_{j=1}^{p} \tilde{v}_j h_j$ is also affine, in particular convex.

Thus $L(x, \tilde{\lambda}, \tilde{v}) = f(x) + \sum_{i=1}^{m} \tilde{\lambda}_i g_i(x) + \sum_{j=1}^{p} \tilde{v}_j h_j(x)$, being a positive linear combination of convex functions, is also convex (positivity of the $\tilde{\lambda}_i$ follows from the dual feasibility condition).

For a convex function, the gradient being zero implies that we have a minimum, so by the gradient condition $L(x, \tilde{\lambda}, \tilde{v})$ attains its minimum at $\tilde{x}$. In other words: the inequality (A) is an equality.

The primal equality feasibility condition implies that $\sum_{j=1}^{p} \tilde{v}_j h_j(\tilde{x}) = 0$. The complementary slackness condition implies that $\sum_{i=0}^{m} \tilde{\lambda}_i g_i(\tilde{x}) = 0$. Together, these facts imply that inequality (B) is an equality.

But (A) and (B) being equalities implies that the duality gap is zero. □

Example 3.7.3. Eventually here should be an example of a convex optimization problem where we make use of the KKT conditions... For now: Search the internet – there are plenty.

For an arbitrary – possibly non-convex – problem, the KKT conditions are still necessary in many cases. More precisely:

Definition 3.7.4. *Consider a set of inequality constraints g_i and equality constraints h_j, and let x be a point where these functions are defined. Let $I := \{i \mid g_i(x) = 0\}$ (in words: I contains the indices of those inequality constraints for which x lies on the boundary). The point x is called* regular, *if it is feasible and the set $\{\nabla g_i(x) \mid i \in I\}$ is linearly independent.*

Theorem 3.7.5. *Let $f, g_1, \ldots, g_n, h_1, \ldots, h_k$ be differentiable functions. If x^* is a local minimizer of f subject to $g(x) \leq 0$ and $h_j(x) = 0$, and a regular point, then there exist $\lambda = (\lambda_1, \ldots, \lambda_n) \in \mathbb{R}^n$, $\lambda_i \geq 0$ and $\mu = (\mu_1, \ldots \mu_k) \in \mathbb{R}^k$ such that the pair $(x, (\lambda, \mu))$ satisfies the KKT conditions.*

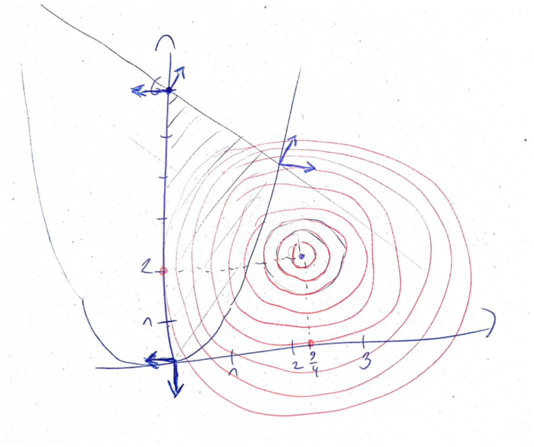
See Chong/Żak, An Introduction to Optimization, 4th edition, Chapter 21 for this theorem and an accompanying graphic.

In many cases, e.g. whenever we know that all points are regular, Theorem 3.7.5 gives us a search strategy for optimal points. One can narrow down the set of potentially optimal points using the KKT conditions. If there are only finitely many points left, it suffices to check at which one the minimum is achieved.

Example 3.7.6. Consider the problem

$$\begin{array}{rc} \text{maximize} & f(x,y) := (x-\frac{9}{4})^2 + (y-2)^2 \\ \text{subject to} & y - x^2 \geq 0 \\ & x + y \leq 6 \\ & y, x \geq 0 \end{array}$$

The problem is not convex: The Hesse matrix of the objective function is negative definite. The objective function is the squared distance from the point $(\frac{9}{4}, 2)$. Sketching the set of feasible points makes it clear that we have *bounded* set of feasible points, i.e. it doesn't stretch infinitely long into any direction, so it should be intuitively clear that the distance from the point $(\frac{9}{4}, 2)$ can be maximized somewhere[1].



From the sketch it is visible that the maximum should be achieved at the upper left corner of the feasible points: This is the feasible point that

[1] More mathematically: The boundary of the set of feasible points is always part of that set, i.e. it is *closed*. It is a theorem from analysis that a real valued function on a closed and bounded (a.k.a. compact) subset of $\mathbb{R}^n$ always achieves a maximum, so this problem must have a solution.

is farthest from the point $(\frac{9}{4}, 2)$. We will see how to get to this result by calculation, using the above theorem.

First note that all feasible points are regular: For points in the interior, the set of gradients whose linear independence we have to check is empty. The empty set is linearly independent. For points which are on exactly one boundary segment, the set of gradients contains only one element, and it suffices to check that that gradient is non-zero. There are only three points where two boundary segments meet. At those points we have to check that the two gradients of the constraint functions point into different directions. You can see in the sketch that that it the case.

Thus all feasible points are regular, and the KKT conditions are necessary for a local maximum under the given constraints. We first bring the problem into the standard form

$$\begin{array}{ll} \text{minimize} & -(x-\frac{9}{4})^2 - (y-2)^2 \\ \text{subject to} & \quad x^2 - y \leq 0 \\ & x + y - 6 \leq 0 \\ & \quad -x \leq 0 \end{array}$$

Here we left out the condition $y \geq 0$ (resp. $-y \leq 0$) because it follows from the first condition $x^2 - y \leq 0$, which is equivalent to $y \geq x^2$.

First we consider the Lagrangian. Note that there is no variable v, because we do not have equality constraints.

$$L((x,y),\lambda) = -(x-\frac{9}{4})^2 - (y-2)^2 + \lambda_1(x^2-y) + \lambda_2(x+y-6) + \lambda_3(-x)$$

We are looking for minimizing points, and we know that these must satisfy the gradient condition:

$$(I) \quad \frac{\partial}{\partial x} L((x,y),\lambda) = -2(x-\frac{9}{4}) + 2\lambda_1 x + \lambda_2 - \lambda_3 \overset{!}{=} 0$$

$$(II) \qquad \frac{\partial}{\partial y} L((x,y),\lambda) = -2(y-2) - \lambda_1 + \lambda_2 \overset{!}{=} 0$$

The complementary slackness conditions are:

$$\begin{array}{rll} (III) & \lambda_1(x^2-y) & = 0 \\ (IV) & \lambda_2(x+y-6) & = 0 \\ (V) & \lambda_3(-x) & = 0 \end{array}$$

Altogether these are five equations with five variables, so there is some hope of having finitely many solutions and simply check which ones are feasible, and which ones give the maximum.

Case 1: $\lambda_1 = 0$
Then (II) implies $\lambda_2 = 2(y-2)$.

Case 1.1: $\lambda_2 = 0$.
Then we know $y = 2$ from the previous equation.

> Case 1.1.1: $\lambda_3 = 0$. Then (I) gives us $x = \frac{9}{4}$. Since $x^2 (\frac{9}{4})^2 > 2 = y$, the point $(\frac{9}{4}, 2)$ is not feasible.
>
> Case 1.1.2: $\lambda_3 \neq 0$. Then complementary slackness gives us $-x = 0$, i.e. $x = 0$. The point $(0,2)$ is feasible and we have $f(0,2) = (\frac{9}{4})^2$.

Case 1.2.: $\lambda_2 \neq 0$. Then complementary slackness gives us $x + y - 6 = 0$, i.e. $y = 6 - x$.

> Case 1.2.1: $\lambda_3 = 0$. Then (I) gives us $\lambda_2 = 2(x - \frac{9}{4})$. Substituting this, and the above expression for y into (II), we get $-2(6-x-2) + 2x - \frac{9}{2} = 0$, which simplifies to $-\frac{25}{2} + 4x = 0$, implying $x = \frac{25}{8}$, $y = 6 - \frac{25}{8} = \frac{27}{8} < x^2$, so we have no feasible point.
>
> Case 1.2.2: $\lambda_3 \neq 0$. Then complementary slackness gives us $x = 0$, hence $y = 6$. The point $(0,6)$ is feasible and we have $f(0,6) = (\frac{9}{4})^2 + 16$.

Case 2: $\lambda_1 \neq 0$
Then complementary slackness implies $y = x^2$.

Case 2.1: $\lambda_2 = 0$.
Then (II) implies $\lambda_1 = -2x^2 + 4$

Case 2.1.1: $\lambda_3 = 0$. Then (I) and our expression for λ_1 give us $0 = -2(x - \frac{9}{4}) + 2(-2x^2 + 4)x = -4x^3 + 6x + \frac{9}{2}$. So x is a solution of this polynomial, or equivalently of the polynomial $x^3 - \frac{3}{2}x - \frac{9}{8}$. By guessing, or applying the algorithm for degree 3 polynomials, one finds that $x = \frac{3}{2}$ is a solution. By polynomial division, or again from the algorithm, one sees that this is the only real solution. We obtain the feasible point $(\frac{3}{2}, \frac{9}{4})$. $f(\frac{3}{2}, \frac{9}{4}) = \frac{9}{16} + \frac{1}{16} = \frac{5}{8}$. This is smaller than $f(0,6)$.

Case 2.1.2: $\lambda_3 \neq 0$. Then complementary slackness gives us $x = 0$, hence $y = x^2 = 0$. The point $(0,0)$ is feasible and we have $f(0,0) = \left(\frac{9}{4}\right)^2 + 4$.

Case 2.2: $\lambda_2 \neq 0$. Then complementary slackness gives us $x + y - 6 = 0$, i.e. $x + x^2 - 6 = 0$. Solving this quadratic equation results in $x = -\frac{1}{2} \pm \sqrt{\frac{1}{4} + \frac{24}{4}}$ which is -3, violating the constraint $x \geq 0$ or 2. We obtain the feasible point $(2,4)$, with $f(2,4) = \frac{1}{16} + 4$. No need to consider further subcases.

Thus checking the KKT conditions we boiled down the possible candidates for a minimum to five feasible points. Among these, the one with the biggest value is $(0,6)$, namely $f(0,6) = \left(\frac{9}{4}\right)^2 + 16$. We could simply check the values to get to this conclusion.

Remark: If you check, you will see that the point $(0,2)$ is part of the KKT pair $((0,2),(0,0,\frac{9}{2}))$. If the problem was convex then Prop. 3.7.2 (saying that the KKT conditions are sufficient for a primal/dual pair to be optimal) would apply, and tell us that the point $(0,2)$ is optimal. We have seen that this is not the case, which shows that Prop. 3.7.2 does not hold for non-convex problems in general.

3.8 Conjugate functions (optional)

Definition 3.8.1. *Let $f\colon \mathbb{R}^n \supseteq D \to \mathbb{R}$ be any function. The* conjugate function *of f is defined as*

$$f^*(y) := \sup_{x \in D}(y^T x - f(x))$$

with $\operatorname{dom} f^* = \{y \in \mathbb{R}^n \mid y^T x - f(x)$ *is bounded above, i.e. its supremum exists*$\}$.

The map $f \mapsto f^*$ is also called the *Legendre transform* or the *Legendre-Fenchel transform*.

Observation 3.8.2. For each $x \in D$ the function $y \mapsto y^T x - f(x)$ is affine, and thus in particular convex. So f^* is a supremum of a family of convex functions, hence convex.

Examples 3.8.3. 1. Let's look at the function $f(x) = ax + b$. Then $y^T x - f(x) = yx - ax - b$ is bounded if and only if $y = a$. In this case the function is constant with value b. For $y \neq a$ the function becomes $(y - a)x - b$, a linear and hence unbounded function, so the supremum in question does not exist. Therefore $\operatorname{dom} f^* = \{a\}$ and $f^*(a) = -b$.

2. Let $f(x) = -\log(x)$. Then $xy + \log(x)$ is bounded if and only if $y < 0$. Furthermore this function takes its supremum at $x = -\frac{1}{y}$, since this is the solution to $\frac{d}{dx}(xy + \log(x)) = y + \frac{1}{x} \overset{!}{=} 0$. Hence we get $\operatorname{dom} f^* = \{y \in \mathbb{R} \mid y < 0\}$ and

$$f^*(y) = \sup_x (xy + \log x) = -\frac{1}{y}y + \log(-\frac{1}{y}) = -1 - \log(-y).$$

3. (Negative entropy) Let $f(x) = x\log(x)$ with $\operatorname{dom} f = \mathbb{R}_+$. Then $xy - x\log(x)$ is bounded above on $\mathbb{R}_+$ for all y, because $x\log(x)$ grows faster than xy. So

$$\operatorname{dom} f^* = \mathbb{R}.$$

Furthermore, a solution to $\frac{d}{dx}(xy - x\log(x)) = y - \log(x) - 1 \overset{!}{=} 0$ is given by $x = e^{y-1}$ and since $xy - x\log(x)$ is concave $x = e^{y-1}$ is the maximum of $xy - x\log(x)$. Hence

$$f^*(y) = e^{y-1}y - e^{y-1}\log(e^{y-1}) = e^{y-1}(y - (y-1)) = e^{y-1}.$$

4. Let $f(x) = \frac{1}{2}x^T Ax + b^T x + c$, with a positive definite (hence invertible) matrix A. Then $f^*(y) = \sup_x (y^T x - \frac{1}{2}x^T Ax - b^T x - c)$. For a given y, to find out the supremum of this function, we compute the differential of the function $h(x) := y^T x - \frac{1}{2}x^T Ax - b^T x - c = -\frac{1}{2}x^T Ax + (y - b)^T x - c$ and see for which x it vanishes. We use the method of section

2:

$$\begin{aligned}
f(x+\epsilon) &= -\frac{1}{2}(x+\epsilon)^T A(x+\epsilon) + (y-b)^T(x+\epsilon) - c \\
&= \underbrace{-\frac{1}{2}x^T Ax + (y-b)^T x - c}_{\text{constant in } \epsilon} \underbrace{- \frac{1}{2}\epsilon^T Ax - \frac{1}{2}x^T A\epsilon + (y-b)^T\epsilon}_{\text{linear in } \epsilon} \underbrace{- \frac{1}{2}\epsilon^T A\epsilon}_{o(\epsilon)} \\
&= -\frac{1}{2}x^T Ax + (y-b)^T x - c \underbrace{-\frac{1}{2}\langle\epsilon, Ax\rangle - \frac{1}{2}\langle Ax, \epsilon\rangle}_{=\langle -Ax,\epsilon\rangle} + \langle y-b, \epsilon\rangle + o(\epsilon) \\
&= -\frac{1}{2}x^T Ax + (y-b)^T x - c + \langle -Ax + y - b, \epsilon\rangle + o(\epsilon)
\end{aligned}$$

So the derivative is $-Ax + y - b$, and we are interested in the x where it is 0:

$$-Ax + y - b = 0 \quad \Rightarrow \quad x = A^{-1}(y-b)$$

Since A is positive definite and occurs with negative sign, the values of h go to $-\infty$ for large norms of x, so the critical value is a supremum. The x that we found we can now insert into the function h and obtain

$$\begin{aligned}
h(A^{-1}(y-b)) &= -\frac{1}{2}(y-b)^T A^{-1} A A^{-1}(y-b) + (y-b)^T A^{-1}(y-b) - c \\
&= \frac{1}{2}(y-b)^T A^{-1}(y-b) - c
\end{aligned}$$

5. Let $f = \|\cdot\| : \mathbb{R}^n \to \mathbb{R}$ be some norm. The dual norm $\|\cdot\|_*$ of $\|\cdot\|$ is defined as

$$\|\cdot\|_* : \mathbb{R}^n \to \mathbb{R}\,,\; y \mapsto \sup\{y^T z \;\mid\; \|z\| \leq 1\}.$$

For these norms one can prove the inequality $y^T x \leq \|x\|\,\|y\|_*$. It is a fact that the dual norm of the ℓ_p-norm is the ℓ_q-norm, for the q satisfying $\frac{1}{p} + \frac{1}{q} = 1$ – then the above inequality is the so-called Hölder inequality, which you may have seen before.

We can use this to compute the conjugate function of the norm:

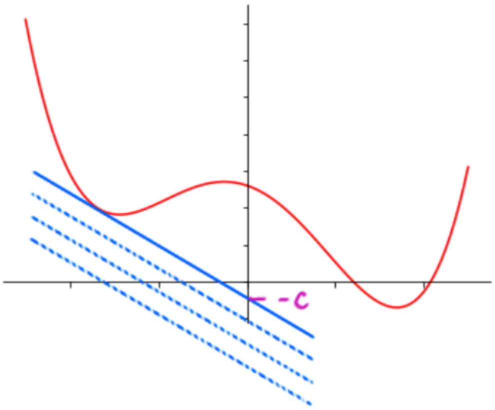
$$f^*(y) = \begin{cases} 0 & \text{if } \|y\|_* \leq 1 \\ \infty & \text{otherwise} \end{cases}$$

Proof: If $\|y\|_* > 1$, there exists a z, $\|z\| < 1$ with $y^T z > 1$. Then

$$y^T(tz) - \|tz\| = t(y^T z - \|z\|) \xrightarrow{t\to\infty} \infty$$

which shows that the supremum is equal to infinity. If $\|y\|_* \leq 1$, then because of the Hölder inequality $y^T x \leq \|x\| \, \|y\|_*$, we have $y^T x - \|x\| \leq 0$ for all x. Hence for $x = 0$ we get the maximum.

A graphical account of the function f^* goes as follows[2]: Given a vector y, we are trying to cage in the epigraph of f with hyperplanes of slope y. This means we are looking at all affine functions with slope y, i.e. of the form $x \mapsto \langle y, x\rangle - c$ whose values are always below those of f[3]. The only free choice we have in this is the constant term c of the affine function. If we want to cage in the epigraph as tightly as possible, we want the smallest possible c (smallest because of the minus sign in front of the c – we want to shift the hyperplane *up*):



Now for such an affine function below f we have the inequality

$$\begin{aligned} f(x) &\geq \langle y, x\rangle - c \quad \forall x \in V \\ \iff c &\geq \langle y, x\rangle - f(x) \quad \forall x \in V \\ \iff c &\geq \sup_x \ \langle y, x\rangle - f(x) \end{aligned}$$

[2] Following this Stackexchange post

[3] In the usual presentation of an affine function $x \mapsto \langle y, x\rangle + c$ we would put a plus sign in front of the constant term c, but clearly this way is equivalent as we vary throgh all possible c.

so the best choice of c is

$$f^*(y) = \sup_x \quad \langle y, x\rangle - f(x).$$

In summary: $f^*(y)$ is minus the amount by which we can shift up the graph of a linear map of slope y until we touch the graph of f.

Why should we care about these hyperplanes that touch the graph of our given function? Well, for a convex function we know that the epigraph is convex, hence is the intersection of halfspaces, and the hyperplanes defining these halfspaces are tangent hyperplanes, since they touch the graph.

It should be graphically plausible that for a convex function (imagine a differentiable one, for simplicity), for each vector y there is a unique tangent hyperplane has normal vector y. And the tangent plane, given the normal vector determines, and is determined by, the value at 0.

Since a function is determined by its graph, and the graph of a convex function is determined by its touching hyperplanes, it should be plausible that the conjugate function, which encodes all about these touching hyperplanes, knows all about the original function. This is indeed true, see Theorem 3.8.6 below.

Definition 3.8.4. *A function $f\colon \mathbb{R}^n \supseteq D \to \mathbb{R}$ is called* closed *if its epigraph $\{(x,t) \in \mathbb{R}^n \times \mathbb{R} \mid x \in D, t \geq f(x)\}$ is closed.*

Example 3.8.5. Consider the two functions $f_1, f_2\colon \mathbb{R}_{>0} \to \mathbb{R}$ given by $f_1(x) := \frac{1}{x}$ and $f_2(x) := x \log x$. The function f_1 is closed, the function f_2 not (because the point $(0,0)$, indeed the whole y-axis, belongs to the closure of its epigraph, but not to the epigraph itself, as the function is not defined at 0).

We can now ask what happens if we take the conjugate of the conjugate:

Theorem 3.8.6. *(a) For all $x \in D$ we have $f^{**}(x) \leq f(x)$.*

(b) The epigraph of f^{**} is the smallest convex closed set containing the epigraph of f.

(c) In particular, if f was already closed and convex (hence had a convex closed epigraph) we have $f^{**}(x) = f(x)$.

Proof. (a) By definition we have $f^*(y) = \sup_x(\langle y, x\rangle - f(x))$, hence for all x, y we have $f^*(y) \geq \langle y, x\rangle - f(x)$. Solving for $f(x)$, we get that for all x, y we have $f(x) \geq \langle y, x \rangle - f^*(y)$. This implies that $f(x) \geq \sup_y(\langle y, x \rangle - f^*(y)) = f^{**}(y)$, where the latter equality holds by definition of conjugates, now applied to f^*.

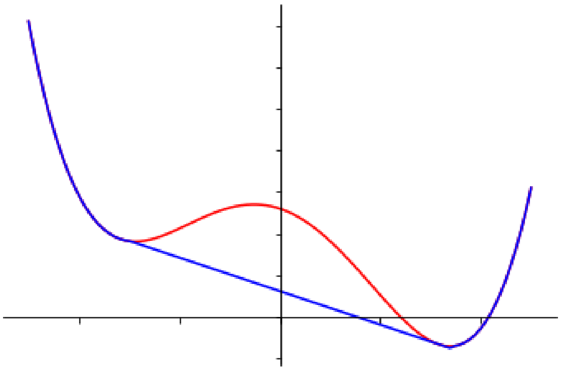
(b) Taking the supremum of functions, as we do in the definition of the conjugate, corresponds to taking intersections of epigraphs. Hence the epigraph of f^{**} is the intersection of all closed halfspaces containing the epigraph of f. As an intersection of closed convex sets it is closed and convex again, and clearly it is the smallest such set.

(c) Clear from the previous items.

□

Since an inclusion of epigraphs corresponds to a pointwise inequality between functions, we could also conclude statement (a) from statement (b) in Thm. 3.8.6.

Visually, the effect of passing to the double conjugate is that of filling out the concave bulges of the graph:



original function (red) and its double conjugate (blue)

Remark 3.8.7. There is an axiomatic characterization of the Legendre transform as the only order reversing duality of convex functions up to some simple transformations.

Summary: Given an arbitrary optimization problem with a possibly *non-convex* objective function f, you can take the double conjugate f^{**} to obtain a convex function with *the same optimal value*: It is graphically clear that the lowest point of the epigraph and the lowest point of its convex closure have the same height. There can arise new *optimal points*, however. Still, knowing the optimal value can already be useful in itself, depending on your application.

3.8.1 Dual Problem And Conjugate Function

Often one can express the Lagrange dual function in terms of the conjugate function. Here is an easy class of examples:

Proposition 3.8.8. *For an optimization problem with linear constraints, i.e. of the form*

$$\begin{array}{lll} \textit{minimize} & f(x) & \\ \textit{subject to} & a_i^T x - b_i \leq 0 & i = 1, \dots, m \\ & c_j^T x - d_j = 0 & j = 1, \dots, p \end{array}$$

We let A, resp. C, be the matrix with rows given by the a_i, resp c_j, and b, d the vectors with entries b_i, c_j.

Then the Lagrange dual function of the problem is given by

$$g(\lambda, \nu) = -b^T\lambda - d^T\nu - f^*(-A^T\lambda - C^T\nu).$$

Proof.

$$\begin{aligned} q(\lambda, \nu) &= \inf_{x \in D} \{f(x) + \lambda^T(Ax - b) + \nu^T(Cx - d)\} \\ &= -b^T\lambda - d^T\nu + \inf_{x \in D} \{f(x) + (A^T\lambda + C^T\nu)^T x\} \\ &= -b^T\lambda - d^T\nu - \sup_{x \in D} \{-f(x) - (A^T\lambda + C^T\nu)^T x\} \\ &= -b^T\lambda - d^T\nu - f^*(-A^T\lambda - C^T\nu) \end{aligned}$$

□

In the situation of Prop. 3.8.8, the domain of $q(\lambda, \nu)$ is thus $\operatorname{dom} g = \{(\lambda, \nu) \mid -A^T\lambda - C^T\nu \in \operatorname{dom} f^*\}$.

Example 3.8.9. Consider the problem

$$\begin{array}{ll} \text{minimize} & \|x\| \\ \text{subject to} & Cx = d \end{array}$$

where $\|.\|$ is any norm. We know from Example 3.8.3.5 that the dual function of the norm is given by

$$f^*(y) = \begin{cases} 0 & \text{if } \|y\|_* \leq 1 \\ \infty & \text{otherwise} \end{cases}$$

By Prop. 3.8.8 we get

$$q(\nu) = -d^T\nu - f^*(-C^T\nu) = \begin{cases} -d^T\nu & \text{if } \|C^T\nu\|_* \leq 1 \\ -\infty & \text{otherwise} \end{cases}$$

Example 3.8.10. Consider the problem

$$\begin{array}{lll} \text{minimize} & f(x) := \sum_{i=1}^n x_i \log(x_i) & \\ \text{subject to} & \quad a_i^T x - b_i \leq 0 & i = 1, \dots, m \\ & \quad \sum_{i=1}^n x_i = 1 & \end{array}$$

where we set $\operatorname{dom} f := (\mathbb{R}_{>0})^n$. We have seen in Example 3.8.3.3 that the dual function of $f(u) = u \cdot log(u)$ is $f^*(y) = e^{y-1}$. Since our f is a sum of such functions of different variables and since the formation of the supremum in the definition of the conjugate function is independent for each variable, we get $f^*(y) = \sum_{i=1}^n e^{y_i - 1}$ with domain all of $\mathbb{R}^n$. With Prop. 3.8.8 we obtain

$$q(\lambda, \nu) = -b^T\lambda - \nu - \sum_{i=1}^n e^{-a_i^T\lambda - \nu - 1} = -b^T\lambda - \nu - e^{-\nu-1} \sum_{i=1}^n e^{-a_i^T\lambda}$$

where a_i is the ith column of A.

The relationship between the conjugate function and the dual problem can be pinned down by varying the constraint limits (which is a way of perturbing a given problem):

Consider convex functions $f, g_1, \dots, g_m$, all defined on $\mathcal{D} \subseteq \mathbb{R}^n$. Then every sequence of real numbers $u = (u_1, \dots, u_m)^T$ determines a convex optimization problem:

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & g_i(x) \leq u_i \quad i = 1, \dots, m \end{array}$$

We will denote the set of feasible points for this problem by $\mathcal{F}_u$. Note that every convex optimization problem is of this form, because we can express an equality as a pair of inequalities.

This defines a function

$$p\colon \mathbb{R}^m \supseteq \mathcal{E} \to \mathbb{R}, \qquad p(u) := \inf_{x \in \mathcal{F}_u \cap \mathcal{D}} f(x).$$

Abbreviating $g(x) := (g_1(x), \ldots, g_m(x))$, the dual function of the convex optimization problem given by $u \in \mathbb{R}^m$ is

$$\begin{aligned} q(\lambda) &= \inf_{x \in \mathcal{D}} \{f(x) + \lambda^T(g(x) - u)\} \\ &= \inf_{x \in \mathcal{D}} \{f(x) + \lambda^T g(x)\} - \lambda^T u \\ &= s(\lambda) - \lambda^T u \end{aligned}$$

where $s(\lambda) := \inf_{x \in \mathcal{D}} \{f(x) + \lambda^T g(x)\}$. For later use we also define

$$\widetilde{s}(\lambda) := \begin{cases} \inf_{x \in \mathcal{D}} \{f(x) + \lambda^T f(x)\} & \text{if } \lambda_i \geq 0 \text{ for all } i \\ \infty & \text{otherwise} \end{cases}$$

On the other hand we have

$$\begin{aligned} p(u) &= \inf_{x \in \mathcal{F}_u \cap \mathcal{D}} f(x) \\ &= \inf_{x \in \mathcal{F}_u \cap \mathcal{D}} \{f(x) + \sup_{\lambda \in \mathbb{R}^m_{\geq 0}} \inf_{x \in \mathcal{F}_u \cap \mathcal{D}} \lambda^T(g(x) - u)\} \\ &= \sup_{\lambda \in \mathbb{R}^m_{\geq 0}} \inf_{x \in \mathcal{F}_u \cap \mathcal{D}} \{f(x) + \lambda^T(g(x) - u)\} \\ &= \sup_{\lambda \in \mathbb{R}^m_{\geq 0}} q(\lambda) - \lambda^T u \\ &= \sup_{\lambda \in \mathbb{R}^m_{\geq 0}} \lambda^T(-u) - (-s(\mu)) \\ &= \sup_{\lambda \in \mathbb{R}^m} \lambda^T(-u) - (-\widetilde{s}(\mu)) \\ &= -\widetilde{s}^*(-u) \end{aligned}$$

Here the step to the second line is valid because for any x, we have $\sup_{\lambda \in \mathbb{R}^m_{\geq 0}} \lambda^T(f(x) - u) = 0$, attained at $\lambda = 0$, as for $\lambda \in \mathbb{R}^m_{\geq 0}$ we always have $\lambda^T(f(x) - u) \leq 0$. The step to the third line is valid because the first summand does not depend on λ.

The step to the fourth line is valid by definition of s, and because $\lambda^T u$ does not depend on x. The remaining steps just repackage the requirement $\lambda \in \mathbb{R}^m_{\geq 0}$ into the the definition of $\widetilde{s}$, in order to match the expression exactly with the definition of conjugate function.

3.9 Multi-objective Optimization

Sometimes one wants to minimize several functions $f_1, \dots, f_q \colon \mathbb{R}^n \supseteq D \to \mathbb{R}$ simultaneously (possibly under constraints). This is called a multi-objective optimization problem. An optimal point fur such a problem would be a feasible point x^* satisfying $\forall y \colon f_i(x^*) \leq f_i(y)$, $i = 1, \dots, q$. Usually such an optimal point does not exist: The typical situation is that if one f_i is minimized at x then the other f_j are not minimized at that x. If an optimal point *does* exist, one says that the objectives f_i are *non-competing*.

Definition 3.9.1. *Consider a multi-objective optimization problem with objective functions $f_1, \dots, f_q \colon \mathbb{R}^n \supseteq D \to \mathbb{R}$ and set of feasible points $\mathcal{F}$. A point x is called* Pareto optimal *for this problem, if, whenever $f_i(y) \leq f_i(x)$ for all $i = 1 \dots q$, then $f_i(y) = f_i(x)$.*

In other words, a point x is Pareto optimal, if whenever one value $f_i(y)$ is lower than $f_i(x)$, then for another objective function f_j we must have $f_j(y) > f_j(x)$.

Usually there is a whole set of Pareto optimal points, and staying within this set one can make the value of one f_i lower at the cost of making the value of another f_j bigger. Studying the interplay between these effects is called a *trade-off analysis*.

One can see the objective functions as a map $f = (f_1, \dots, f_q)$ from the set $\mathcal{F}$ of feasible points to $\mathbb{R}^q$. The image of the Pareto optimal points will then be those points for which one can not lower the value (i.e. travel parallel to some coordinate axis towards smaller values) of any coordinate without leaving the image of f:

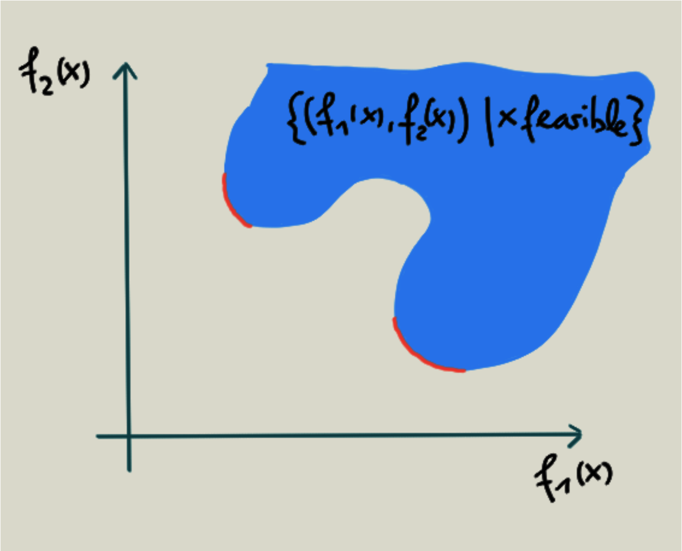


Image of the feasible points marked in blue, images of Pareto optimal points marked in red

Example 3.9.2. Suppose you want to invest a fixed amount of money T into some stocks, government bonds or similar financial products. Typically the products that promise more return are also more volatile, i.e. have a bigger variation around the expected return, in short a bigger variance.

investment nr.	**expected return**	**variance**
i	p_i	Σ_{ii}
1	17%	35%
2	13%	10%
3	7%	5%
3	3%	0%

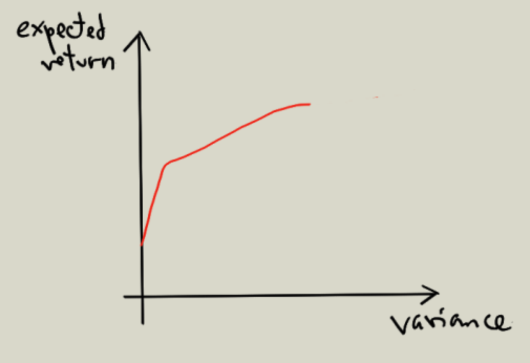
Often stocks prices are not independent, e.g. if climate protection taxes are raised, the gasoline car producer's stocks might fall while the solar cell producer's stocks will rise. So there are also covariances between the stocks, which, together with the variances from the table, form a covariance matrix Σ.

Now typically an investor wants to maximize returnand minimize variance (the latter to avoid bancrupcy and be able to plan financially).

Denote the amount of money that we want to invest into product i by x_i, and abbreviate the vectors of these amounts by $x := (x_1, \ldots, x_4)^T$. Then the question of how to invest becomes multi-objective optimization problem

$$
\begin{array}{rc}
\text{minimize} & -(p_1, \ldots, p_4) \cdot \begin{pmatrix} x_1 \\ \vdots \\ x_4 \end{pmatrix} \\
\text{and} & x^T \Sigma x \\
\text{subject to} & (1, \ldots, 1) \cdot \begin{pmatrix} x_1 \\ \vdots \\ x_4 \end{pmatrix} = T
\end{array}
$$

The curve of Pareto optimal points might look like this:



Return and variance at the Pareto optimal points

It is up to the investor which Pareto optimal point they want to choose, and there is no purely mathematical answer here. A common choice is to look for a "knee", from which on, in this case, raising the variance results in smaller gains of expected return.

The question is now how to find the Pareto optimal points, and ideally a parametrization of them. One idea is to turn the multi-objective optimization problem into a single objective optimization problem.

Scalarization of a multi-objective problem: Instead of having several independent objectives we can form a single objective function as a weighted sum. That is, we pick weights $\lambda_i \geq 0$ and pose the problem

$$
\begin{array}{rc}
\text{minimize} & \lambda_1 f_1(x) + \ldots + \lambda_q f_q(x) \\
\text{subject to} & \text{(same constraints)}
\end{array}
$$

A high weight in front of one of an objective functions means that we place more importance on it. Only the relative importance matters: If we multiply every λ_i with a fixed constant c then the optimal points of this problem do not change. Often one therefore chooses a $\lambda_j \neq 0$ and divides the objective function by λ_j. The coefficient in front of f_i is then $\frac{\lambda_i}{\lambda_j}$. It means that, among the Pareto optimal points, decreasing f_i by some amount α leads to an increase of f_j by $\frac{\lambda_i}{\lambda_j}\alpha$.

Note that an optimal point x^* of this problem is a Pareto optimal point for the multi-objective problem: If there was a possibility to lower the value of one f_i without raising the values of the others, then x^* would not be optimal.

Now suppose that the f_i and the constraint functions are convex. Then the objective function of the above problem is a positive linear combination of convex functions, and hence convex. So the above problem is a convex optimization problem and has a single solution x^*_λ (supposing there is a solution at all). Varying the parameters λ_i gives us a function $\mathbb{R}^q \to \mathbb{R}^n$, $\lambda = (\lambda_1, \ldots, \lambda_q) \mapsto x^*_\lambda$. Ths is a parametrization of the set of Pareto optimal points!

Example 3.9.3 (Regularized least squares)**.** Consider the following multi-objective optimization problem:

$$\begin{array}{rc} \text{minimize} & f_1(x) := \|Ax - b\|^2 \\ \text{and} & f_1(x) := \|x\|^2 \\ \text{subject to} & \text{no constraints} \end{array}$$

We can rewrite the objective functions as $f_1(x) = (Ax-b)^T(Ax-b) = x^TA^TAx - 2b^TAx + b^Tb$ and $f_2(x) = x^Tx = x^TIx$.

The weighted sum scalarization of this problem then asks to minimize the following function:

$$\lambda_1 \cdot f_1(x) + \lambda_2 \cdot f_2(x) = x^T(\lambda_1 A^TA + \lambda_2 I)x - 2\lambda_1 b^TAx + \lambda_1 b^Tb$$

Note that A^TA is positive semidefinite and I is positive definite, so $\lambda_1 A^TA + \lambda_2 I$ is positive definite. Hence the function $x \mapsto x^T(\lambda_1 A^TA + \lambda_2 I)x$ is convex and the whole objective function is a positive linear combination of this function and an affine function, hence convex. So we have an unconstrained convex optimization problem and the optimal point is the point where the gradient vanishes.

To determine the gradient, first note that the derivative of the function $g(x) := x^T Cx$ for a symmetric matrix C is given by $2Cx$: We have

$$\begin{aligned}(x^T + \epsilon^T)C(x+\epsilon) &= x^T Cx + \epsilon^T Cx + x^T C\epsilon + \epsilon^T C\epsilon \\ &= x^T Cx + 2x^T C\epsilon + o(\|\epsilon\|) \\ &= g(x) + \langle 2Cx, \epsilon \rangle + o(\|\epsilon\|)\end{aligned}$$

Therefore the gradient of our weighted sum objective function is

$$\nabla_x(\lambda_1 f_1 + \lambda_2 f_2)(x) = 2(\lambda_1 A^T A + \lambda_2 I)x - 2\lambda_1 A^T b$$

Setting this to 0 gives

$$x = \left(\lambda_1 A^T A + \lambda_2 I\right)^{-1} \lambda_1 A^T b$$

if the inverse matrix occurring here exists. Note that for $\lambda_2 \neq 0$ the inverse does indeed exist, because the matrix in question is positive definite, as remarked before.

Supposing $\lambda_1 \neq 0$ we can divide our objective function by λ_1, and setting $\mu = \frac{\lambda_2}{\lambda_1}$ we get the optimal point

$$x(\mu) = \left(A^T A + \mu I\right)^{-1} A^T b$$

This is a parametrization of the Pareto curve (since here it is one dimensional), i.e. the set of Pareto optimal points (minus the extreme case where $\lambda_1 = 0$).

It is worth considering the two extreme cases:

$\mu \to \infty$: This corresponds to the case $\lambda_1 = 0, \lambda_2 = 1$. The scalarized problem is then simply to minimize $\|x\|^2$, and the only solution is $x = 0$.

$\mu \to 0$: This corresponds to the case $\lambda_1 = 1, \lambda_2 = 0$. The scalarized problem is then to minimize $\|Ax - b\|^2$, and there may be many solutions. However, the limit solution from our formula is $\left(A^T A\right)^{-1} A^T b$ (if the inverse of $A^T A$ exists). This is the solution via the pseudoinverse that we know from Theorem 1.8.4. We also know from Theorem 1.8.5 that this particular solution is the one with the smallest norm! That is, in the limit we did not just get a the solution that minmizes first objective function, but we among all such solutions we got the one that gives the smallest value for the second objective function. This observation should make sense if you look again at the picture of the Pareto optimal points above, and imagine travelling towards an end point of the red line segment.

We don't go further into the topic of Multi-objective Optimization. For a very readable introduction see the last chapter of Chong/Żak, An Introduction to Optimization, 4th edition.

3.10 Regularization

Regularization means adding a second objective to a given optimization objective to aid in the optimization task.

Most frequently this second objective is to minimize the size of the solution, i.e. to minimize $\|x\|$ for some norm. And most frequently this is done by scalarization, i.e. by replacing the objective function $f_0(x)$ with $f_0(x) + \delta\|x\|$ for some weight δ.

Here are some reasons for wanting to do so:

Numerical stability: Keeping the numbers small helps to avoid buffer overflows and other trouble with floating point arithmetic.

Reducing sensitivity to parameters: Often the optimization problem in question contains parameters that themselves only arose as estimates from real world data. For example in the task of minimizing $\|Ax - b\|^2$ the matrix A typically comes from observations of the real world which are themselves prone to error. For big x this error will be propagated much more, while for small x it will not cause as much effect.

Encouraging sparsity for numerical effectiveness: Optimization problems become vastly more demanding in terms of computer power as the number of variables rises. It is therefore helpful to be able to set a few variables to zero and stop considering them as variables.

Minimizing $\|x\|$ of course encourages all entries of the vector x to go closer to zero. If some entry becomes close enough to zero during gradient descent, one can decide to keep it equal to zero thus obtaining a lower dimensional problem.

Avoiding overfitting: Remember the example of polynomial regression, where, given some points, we looked for coefficients of a polynomial such that the points lie approximately on its graph.

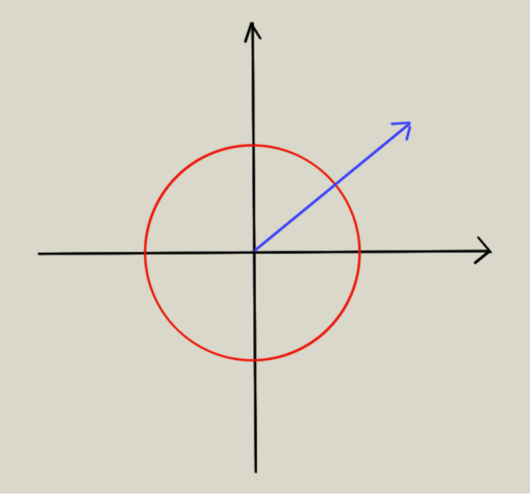
Given samples $(x_1, y_1), \dots, (x_k, y_k)$, we found our coefficients a_i by minimizing the squared error $e(a_0, \dots, a_n) := \sum_{j=0}^{k}(\sum_{i=0}^{n} a_i x_j^i - y_j)^2$. If we want to favor polynomials of smaller degree we could for example instead minimize the function $e(a_0, \dots, a_n) + \sum_{i=0}^{n} i \cdot a_i^2$. This aims to keep the coefficients of the polynomial small. Additionally, the higher the degree the more weight the coefficient has, so that the coefficients of the high degree monomials are made smaller by the optimization.

If one decides to simply delete monomomials whose coefficient is below some threshold value, this leads to smaller degree polynomials. In any case, the higher degree monomials will have less influence on the outcome, due to their smaller coefficients, which can avoid overfitting.

Ensuring invertibility: Remember that the least squares problem of minmizing $\|Ax - b\|$ has an easy solution if A has full rank: it is $(A^T A)^{-1} A^T b$. In Example 3.9.3 we saw that the regularized Problem has the solution $(A^T A + \delta I)^{-1} A^T b$ and that now the matrix in question is *always* invertible (because it is positive definite). This is not the only such use of regularization. More generally one can ensure so-called generic properties, i.e. properties that almost always are there, of the objects at hand by regularization. In our example being invertible means having non-zero determinant – intuitively of all the uncountably many values that a matrix can have as determinant only zero is a problem, ans as we vary the problem a bit we can safely stay away from that very special case.

Once you decided to keep your solution vector small, the question is which norm to use for your second objective function. Perhaps surprisingly, the choice of norm does make a difference!

This becomes clearer when you consider how the shape of the unit ball tells you which vector are considered smaller by that norm. The unit ball of a norm $\| - \|$ completely determines that norm as follows: Take the boundary of the unit ball $\partial B := \{x \mid \|x\| = 1\}$, and for a given vector z consider the $r \geq 0$ such that $r \cdot z \in B$. Then $\|z\| = r^{-1}$:

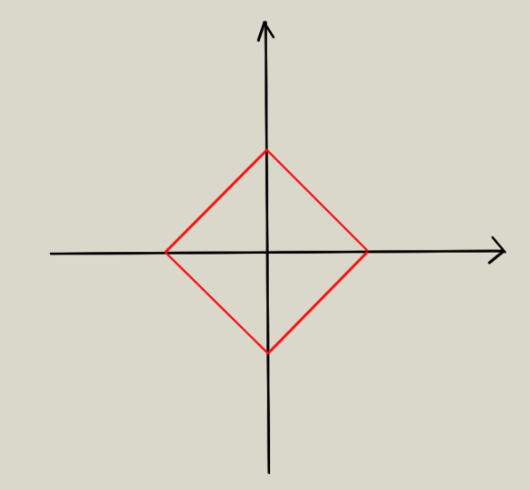


A vector of length 2 needs to be multiplied with $r = \frac{1}{2}$ to lie on the unit ball

An obvious and frequent choice of norm is the Euclidean norm. Regularization with respect to this norm is called *Tikhonov regularization*.

For a less symmetric unit ball than the Euclidean one, the corresponding norm considers those vectors smaller that point in directions where the unit ball bulges out more: One has to retract them less to reach the boundary of the unit ball.

Now if one looks at the unit ball of the ℓ_1-norm, one sees that for making a vector ℓ_1-smaller, it is beneficial to move it closer (i.e. euclidean closer) to the nearest axis:



ℓ_1-unit ball

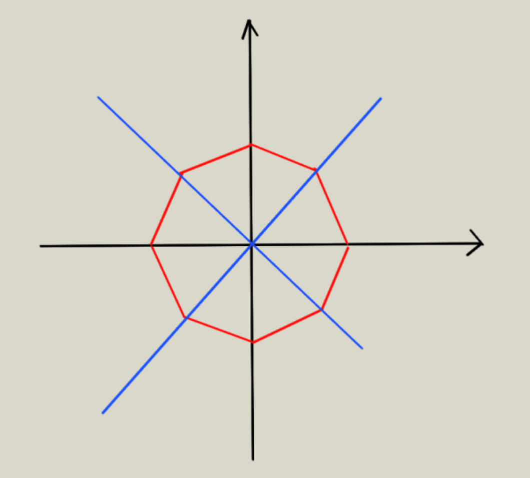
But being closer to the axis means that one coordinate is closer to zero. This is why regularization with respect to the ℓ_1-norm encourages sparsity!

For a concrete example statement see e.g. Ng, A. Y., Feature selection, L1 vs. L2 regularization, and rotational invariance. Twenty-First International Conference on Machine Learning - ICML '04

The article considers the setting of a logistic regression where many of the feature variables are irrelevant: Say there are n feature variables, but r of them are already sufficient to obtain a model with small squared error.

The above article shows that with ℓ_1-regularization one needs much fewer samples to get close to a sparse model than with ℓ_2-regularization. The precise statement is somewhat technical.

Another norm that has recently been considered for regularization is the norm whose unit ball is a regular octagon, with vertices on the axes and on the diagonals:



Another unit ball

Encouraging smallness with respect to this norm makes vectors move closer either to the axes or to the diagonals. Being on a diagonal means that two coordinates are equal. In this case one has to store one number less and again in effect achieved a dimension reduction by 1. Indeed it is a technique in training neural networks to encourage the "coupling of weights", i.e. to ensure that the weights of two different connections between neurons have the same value. This can be reinforced by octagon regularization.

A final question might be: Why regularize with a norm at all? One can easily come up with "unit balls" of non-norms that encourage weight coupling and sparsity even more. One reason is when one has a convex optimization problem that one wants to regularize: Since norms are also convex, regularizing with a norm will result in a convex optimization problem, which is always good. But otherwise different non-norm regularization terms might be an interesting thing to investigate.

Keeping the solutions small or sparse is not the only goal one can pursue by regularization. Consider a function approximation problem: Typically an observed function is given as a time series, i.e. a long vector of values at evenly spaced points in time. The observed function might be disturbed by noise, and therefore be very shaky, while from our external knowledge we know that our function should be smooth.

We can model smoothness by saying that first and second derivatives should not be too big. For function of which we only know finitely many sample values x_i, $i = 1 \dots n$ we can approximate the first derivative simply by the finite differences $x_{i+1} - x_i$. To keep those small means to regularize by the sum of squared differences $d(x) := \sum_{i=1}^{n-1}(x_{i+1} - x_i)^2$. This can in turn be expressed as $d(x) = \|Dx\|^2 = x^T D^T DX$ where

$$D = \begin{pmatrix} -1 & 1 & 0 & \cdots & 0 & 0 & 0 \\ 0 & -1 & 1 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & -1 & 1 & 0 \\ 0 & 0 & 0 & \cdots & 0 & -1 & 1 \end{pmatrix}$$

The second order derivative is expressed as the variation in the first order derivatives, i.e. by the differences $(x_{i+1} - x_i) - (x_i - x_{i-1}) = (x_{i+1} - 2x_i + x_{i-1})$. So to ensure small second derivatives, and therefore not too harsh variation, one regularizes by the term $s(x) := \sum_{i=1}^{n-2}(x_{i+1} - 2x_i + x_{i-1})^2$. This term can also be expressed as $s(x) = \|Sx\|^2 = x^T S^T Sx$ where

$$S = \begin{pmatrix} 1 & -2 & 1 & 0 & \dots & 0 & 0 & 0 & 0 \\ 0 & 1 & -2 & 1 & \dots & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -2 & \dots & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \dots & -2 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & \dots & 1 & -2 & 1 & 0 \\ 0 & 0 & 0 & 0 & \dots & 0 & 1 & -2 & 1 \end{pmatrix}$$

One see that both smoothness regularization terms give rise to easy quadratic optimization problems on their own.

It should be intuitively clear that smoothness regularization can also help avoid overfitting: Just look at the pictures in Section 1.8.2 – overfitting is associated with wild jumpy behaviour; the contrary of smoothness.

3.11 Final remarks

Optimization is a wide field, and we have only touched a corner of it. To mention just one direction that we did not see at all: We always assumed our constraints to give continuous regions of space. But one can also ask about optimization in discrete regions of space, e.g. finding the minimum value of a function on all points with integer coordinates. This is for example covered in the course on Linear Optimization of Prof. Klau, frequently offered in the AI and Data Science program.

A resource for a systematic overview and further reading are the Guide pages of the NEOS Optimization Server. Check out the "Types of optimization problems" and the "Taxonomy" pages, to get an impression.

In general a real world optimization problem starts with having to choose a *model* – you first need to know which function you want to optimize. This part is not purely mathematical and often requires intuition and knowledge of your domain of application. We did not touch upon this part at all and just started with a given function.

Once you have a concrete function that you want to optimize, you have to recognize into which classes of optimization problems it falls, and then can look what algorithms are available to solve them.

The NEOS Server is a free resource where you can find long lists of optimization algorithms, and implementations of them, and you can even submit your optimization problem to their server and let them solve it. It is probably more powerful than your laptop at home, but for a very demanding problem you should maybe rather check out high performance computing resources of HHU or other places.

The NEOS guide pages also offer a list of case studies to learn from, some of them interactive.

Concerning optimization software packages: The two packages that featured in the exercise sheets, CVXOPT and Pyomo are actually used in industry and academic research.

CVXOPT is capable of treating general *(quasi)convex* optimization problems, also if they involve non-differentiable objective and constraint functions. In using CVXOPT you have to specify a solver you want to use, which requires some knowledge of the available and appropriate solvers.

To avoid that, an alternative is the CVXPY package – it automatically chooses its solvers. It comes with its own solver functions, but you can also force it to call CVXOPT instead. Another bonus is that the syntax is very close to the standard mathematical notation that you have seen in this chapter.

Pyomo is really a "frontend" with a nice syntax, that calls other solver libraries. Here again you have to know which library to call. The library IPOPT, featured in an exercise, can treat possibly non-convex problems with *differentiable* objective and constraint functions, so that works in very many cases, but it may not be the most efficient choice for special cases.

Not a software package, but maybe even better: The GENO project website let's you enter an optimization problem and returns Python code that solves it!

Mathematical modelling, optimization and Machine Learning: Mathematical modelling means expressing a real world situation in mathematical terms, e.g. with the goal of understanding the situation better, or predicting future developments, or finding solutions to some engineering or finance or other decision problem. Often a mathematical model has to be set up to obtain an optimization problem in the first place, whose solution then yields an explanation (e.g. determining the minimal energy state of a physical system might explain its behaviour), or prediction (finding the linear/polynomial/periodic/etc. function that best fits to some given data gives predictions for new data), or solution (which medical treatment has the biggest chance of success).

In traditional mathematical modelling one needs to find mathematical expressions of reality oneself. This has the advantage that one knows precisely which components of the mathematical model correspond to which aspects of reality, and what leads to the obtained solution. It also limits the method, because one needs to already have a grasp on the appropriate mathematics and some insights into the situation to be modelled and what is crucial about it. It would, for example, be very hard to come up from scratch with a mathematical definition of when an image shows a cat!

In Machine Learning, in contrast, one does not need to find mathematical expressions of reality: One simply gathers a lot of examples and counterexamples and then *trains* a model - the mathematical expression modelling reality arises in this process! However, one still has to do some secondary mathematical modelling: One has to choose a loss function, which often amounts to choosing a metric or some other notion of distance between two samples of the given data. And one can choose a regularization term, e.g. a smoothing term reflecting the expectation that the data in question should come from a smooth function. Or one chooses a statistical model based on very general assumptions and then only has to ask what are the precise parameters. Such general assumptions also go into choosing an architecture of a neural network (e.g. fully connected versus convolutional). So one is not completely off the hook, but it is sufficient to understand the situation in much broader strokes than when one does mathematical modelling in a stricter sense.

That said, many Machine Learning practitioners agree that it is a healthy attitude to try to solve a problem without Machine Learning first. It has even been called the first rule of Machine Learning.

For some light reading, and interesting speculation on how Optimization is intertwined with Deep Learning check out this Blog post by Francesco Orabona!

Further reading and exercises: A very readable book (easier than Boyd/Vandenberghe) to study on your own is this:

Chong/Żak, An Introduction to Optimization, 4th edition.

The exercises on convex optimization in this book are at a level that you should be able to tackle now.

For some direct applications of the contents of this chapter in Machine Learning see for example Section 7.1 of Bishop's book Pattern Matching and Machine Learning – you will meet Lagrangians, KKT conditions and the dual problem, all conspiring to get to the setup of support vector machines.

On the practical side, here you can see an implementation of support vector machines using CVXOPT.

What you should take away from this chapter:

- notion of convex set
- notion of convex function
- optimization problem in standard form
- notion of convex optimization problem
- Lagrangian, dual function and dual problem of an optimization problem
- Slater's condition
- KKT conditions
- How to tackle a problem by looking for the points satisfying the KKT conditions

The main thing among these, *for the exam*, is the last one. For reading Machine Learning literature, knowing the KKT conditions is also pretty important, but Lagrange duality also appears sometimes.

4 Probability Theory

Probability Theory and Statistics are at the heart of AI and Data Science. In this chapter we quickly review the basics.

Good resources for this chapter:
Rozanov: Probability Theory: A concise course – not free, but good because of its conciseness.

For probability theory in general, but also for the stochastic processes that will appear in a later chapter, a great source is: Grimmett/Stirzaker, Probability and Random processes – not free.

Bertsekas, Tsitsiklis: Introduction to Probability, 2nd edition – here a free shorter version with two chapters less
Here is the course homepage of a course at MIT following the book of Bertsekas and Tsitsiklis, with lecture notes, videos, exercises and summaries.

For removing all doubts that this has to do with Machine Learning, but also as a good and thorough source in itself, look at Bishop: Pattern recognition and Machine Learning

For a terse but mathematically complete treatment, a good source is Rosenthal: First Look At Rigorous Probability Theory However, we will not look into the foundations of probability rigorously beyond the basic definitions, and the other sources are better adapted to our approach.

A fantastic resource is the Random website! It contains a complete development from the very basic notions up to quite advanced topics, with full proofs, a good collection of examples, exercises and animations illustrating random experiments.

4.1 Probability spaces

The precise (Kolmogorov style) mathematical formulation of axiomatic probability theory can appear somewhat technical and forbidding. Don't worry

– in practice one doesn't really use these definitions. They are necessary for the basic setup, but then they vanish under the hood and you only use higher level concepts: It's like bytecode – it needs to be somewhere in the background for your programs to run, but you do the programming in a high level language like Python.

What I just wrote is especially true for the notion of σ-algebra: We really only state it for completeness, but we will then ignore it in the rest of the text, without any harm.

Definition 4.1.1. *Let Ω be a set. A collection of subsets $\mathcal{A} \subseteq \wp(\Omega)$ is called a* σ-algebra *on Ω if*

(i) $\Omega \in \mathcal{A}$

(ii) $A \in \mathcal{A} \;\Rightarrow\; \bar{A} \in \mathcal{A}$

(iii) $A_1, A_2, A_3, \ldots \in \mathcal{A} \;\Rightarrow\; \bigcup_{n \in \mathbb{N}} A_n \in \mathcal{A}$

A pair $(\Omega, \mathcal{A})$ consisting of a set and a σ-algebras $\mathcal{A}$ on Ω is called a measurable space*.*

Remark 4.1.2. Clearly $\wp(\Omega)$ itself is a σ-algebra. One can easily verify that arbitrary intersections of σ-algebras on a set Ω are a σ-algebra again. Thus starting with any collection of subsets of Ω there is a smallest σ-algebra containing all those sets: one simply takes the intersection of all σ-algebras containing those sets.

Important examples: The σ-algebra on $\mathbb{R}$ generated by all intervals $[a, b]$, and more generally the σ-algebra on $\mathbb{R}^n$ generated by all products of intervals. These are called the *Borel σ-algebras* on $\mathbb{R}^n$ and denoted by $\mathcal{B}(\mathbb{R}^n)$. Still more generally one can generate a σ-algebra from the open sets of any metric space.

Definition 4.1.3. *Let $(\Omega, \mathcal{A})$, $(\Omega', \mathcal{A}')$ be measurable spaces. A map $f\colon \Omega \to \Omega'$ is called* measurable *if for all $A \in \mathcal{A}'$ we have $f^{-1}(A) \in \mathcal{A}$.*

Example 4.1.4. Any continuous map $f\colon \mathbb{R}^n \to \mathbb{R}^n$ is a measurable map $(\mathbb{R}^n, \mathcal{B}(\mathbb{R}^n)) \to (\mathbb{R}^n, \mathcal{B}(\mathbb{R}^n))$. This follows, because it is sufficient to check the measurability condition on preimages on the generators of a σ-algebra, because preimages of closed sets are closed and because all closed sets belong to $\mathcal{B}(\mathbb{R}^n)$. But there are many more measurable maps than continuous maps – it is actually hard to construct a non-measurable map.

Definition 4.1.5. *A triple* $(\Omega, \mathcal{A}, \mathrm{P})$ *is called* probability space *if* $\mathcal{A}$ *is a* σ-*algebra on* Ω *and* P *is a* probability measure *on* $\mathcal{A}$, *i.e. a map* $\mathrm{P}\colon \mathcal{A} \to [0,1] \subseteq \mathbb{R}$ *such that*

(i) $\mathrm{P}(\Omega) = 1$

(ii) $\mathrm{P}(\bar{A}) = 1 - \mathrm{P}(A)$

(iii) If $A_1, A_2, A_3, \ldots \in \mathcal{A}$ *are disjoint, then* $\mathrm{P}(\bigcup_{n\in\mathbb{N}} A_n) = \Sigma_{n\in Nb}\, \mathrm{P}(A_n)$

The elements of Ω *are called* outcomes *or* elementary events. *Subsets* $A \subseteq \mathcal{A}$ *are called* events. *Probability measures are also called* probability distributions.

Note that the third property of probability measures is only required for *countable* collections of disjoint sets. One also says that the measure satisfies *countable additivity*.

Remark 4.1.6. Ideally one would like to associate a probability $\mathrm{P}(A) \in [0,1]$ to *every* subset $A \in \wp(\Omega)$ and not just in some sub-σ-algebra of $\wp(\Omega)$. However, there are results from measure theory, saying that it can sometimes simply be impossible to consistently define a probability measure. A mind-boggling instance of this phenomenon is the Banach-Tarski paradox.

So instead one defines P just on some σ-algebra $\mathcal{A} \subseteq \wp(\Omega)$, i.e. a collection of subsets where defining a probability measure is possible. In practice one can always choose an $\mathcal{A}$ containing all subsets of interest.

We will soon stop mentioning $\mathcal{A}$. Any subsets of Ω that we talk about, will implicitly be assumed to be contained in $\mathcal{A}$.

Proposition 4.1.7. *Probability measures satisfy the following properties:*

(i) $\mathrm{P}(\emptyset) = 0$

(ii) $\mathrm{P}(A \cup B) = \mathrm{P}(A) + \mathrm{P}(B) - \mathrm{P}(A \cap B)$

(iii) if $A \subseteq B$ *then* $\mathrm{P}(A) \leq \mathrm{P}(B)$

(iv) if $A_n \supseteq A_{n+1}$, *then* $\mathrm{P}(\bigcap_{n\in\mathbb{N}} A_n) = lim_{n\in\mathbb{N}}\, \mathrm{P}(A_n)$

(v) if $A_n \subseteq A_{n+1}$, *then* $\mathrm{P}(\bigcup_{n\in\mathbb{N}} A_n) = lim_{n\in\mathbb{N}}\, \mathrm{P}(A_n)$

Proof. The proofs can be found in Rozanov, chapter 2. However, for now it is more important than knowing a proof, that the statements make sense to you intuitively. □

4.2 Discrete and continuous probability spaces

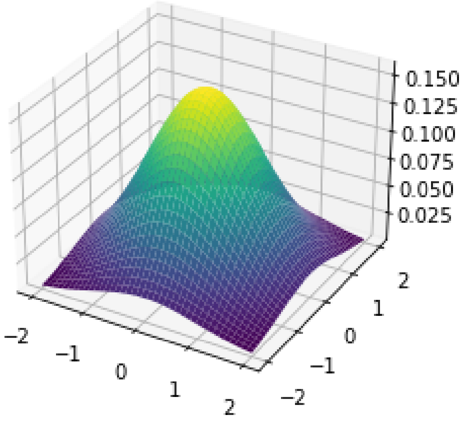
Example 4.2.1. In an experiment where two fair dice are thrown one could set $\Omega = \{(a,b) \mid a,b \in \{1,\ldots,6\}\ \}$, $\mathcal{A} = \wp(\Omega)$, $\mathrm{P}(\{(a,b)\}) := \frac{1}{36}$ for all a, b. By property (iii) of a probability measure, P is then determined for every subset, because arbitrary subsets of Ω are disjoint unions of singleton subsets.

Remark 4.2.2. Example 4.2.1 is an example of a *discrete* probability space - i.e. one with a finite or countably infinite set of elementary events $\Omega = \{\omega_i \mid i \in \mathbb{N}\}$. On these spaces every point can be given a probability $\mathrm{P}(\{\omega_i\}) = p_i$ and one has $\Sigma_{i \in \mathbb{N}} p_i = \mathrm{P}(\Omega) = 1$. Thus one can (and has to) associate to every subset $A \subseteq \Omega$ the probability $\mathrm{P}(A) = \Sigma_{\omega_i \in A} p_i$. It follows that in this case one can take the σ-algebra to be the whole power set; $\mathcal{A} = \wp(\Omega)$.

The function $\Omega \to [0,1], \omega_i \mapsto p_i$ is called a *probability mass function*

Example 4.2.3. In an experiment where a dart is thrown, in an attempt to hit the origin in $\mathbb{R}^2$, one might set $\Omega = \mathbb{R}^2$, $\mathcal{A} = \mathcal{B}(\mathbb{R}^2)$ and $\mathrm{P}(A) := \int_A f$, where $f(x,y) := \frac{1}{2\pi} \exp\left(-\frac{x^2+y^2}{2}\right)$ Of course one needs to show that $\int_{\mathbb{R}^2} f = 1$ so that this does in fact define a probability measure. This is done below in Example 4.4.8.

If a probability measure is defined in this way as an integral, the integrand f is called the *density function*. Here is a plot of the above f (it is the density function of a bivariate normal distribution, see later):



Density function of a bivariate normal distribution

A probability space of this type is sometimes called *continuous* probability space, to distinguish it from the discrete case. Note that the probability of a single event $a \in \Omega = \mathbb{R}^2$ in our example is $\int_{\{a\}} f = 0$. This is the real distinguishing feature from the discrete case.

We will almost only speak of discrete and continuous probability spaces (and later discrete and continuous random variables), often giving proofs only in one of the two cases.

One can see discrete probability spaces, after identifying their elements with a discrete subset $\{a_i \mid i \in \mathbb{N}\}$ of $\mathbb{R}$, as limit cases of probability spaces defined by density functions: Around each point a_i put an interval $I_i := (a_i - \epsilon, a_i + \epsilon)$ and define a density function by

$$f_\epsilon(x) := \begin{cases} \frac{1}{2\epsilon} P(a_i) & \text{if } x \in I_i \\ 0 & \text{otherwise} \end{cases}$$

As ϵ gets smaller, the regions where the function f_ϵ is non-zero get smaller and concentrate around the elements a_i. In the limit we are only left with the elements a_i. This is just an intuitive picture – see also this animation on Wikipedia – but it can be made rigorous (the limit of these density functions is no longer a function, but a *distribution* in the sense of functional analysis).

Vice versa, continuous probability spaces can be approximated by discrete ones: One can chop up $\mathbb{R}^n$ into little hypercubes of edge length ϵ and approximate a density function by density functions that are constant on each cube. As ϵ gets smaller, the aproximation gets better and in the limit we have our given density function back.

General probability spaces always consist of a discrete and a continuous part – the precise distinction here is between atomic and non-atomic measures. Any measure decomposes into an atomic and a non-atomic part (by Thm. 2.1 here).

There is a way of treating all probability spaces uniformly via measure theory, but we will avoid the extra language and theory needed for that.

4.3 Conditional probability and independence of events

Definition 4.3.1. *Given a probability space and two events $A, B \subseteq \Omega$, the* conditional probability, *or* probability of A given B, *is* $P(A \mid B) := \frac{P(A \cap B)}{P(B)}$

For a fixed event B one obtains a new probability measure $A \mapsto P(A \mid B)$. One should think of this conditional probability measure $P(- \mid B)$ as a rescaling of the original probability measure to account for the fact that one is sure of the event B.

Theorem 4.3.2 (Bayes' Theorem)**.** $P(A \mid B) = \frac{P(B|A)P(A)}{P(B)}$

Proof. Just substitute the definition of the probability of B given A, $P(B \mid A) = \frac{P(A \cap B)}{P(A)}$, into the right hand side of the claim:

$$P(A \mid B) = \frac{P(A \cap B)}{P(B)} = \frac{P(B \mid A)P(A)}{P(B)}$$

□

A particularly useful form of Bayes' theorem is the following. We write $C \coprod D$ for the *disjoint* union of sets C, D.

Proposition 4.3.3. *Let $\Omega = A_1 \coprod \ldots \coprod A_n$ be a decomposition of Ω into disjoint subsets. Then we have for any $B \subseteq \Omega$:*

$$P(A_j \mid B) = \frac{P(B \mid A_j)P(A_j)}{\Sigma_i P(B \mid A_i)P(A_i)}$$

Proof. The given decomposition of Ω allows us to decompose any set B as $B = B \cap A_1 \coprod \ldots \coprod B \cap A_n$. This gives us

$$\begin{aligned} P(A_j \mid B) &= \frac{P(A_j \cap B)}{P(B)} = \frac{P(B \mid A_j)P(A_j)}{\sum_{j=1}^{n} P(B \cap A_j)} \\ &= \frac{P(B \mid A_j)P(A_j)}{\Sigma_i P(B \mid A_i)P(A_i)} \end{aligned}$$

□

Example 4.3.4. Suppose we have a blood test, testing for a disease of a patient. Denote the events that the test is positive (i.e. indicates that the patient has the disease) by T, and the event that the patient has the disease by D. From the development of the test typically the following conditional probabilities are known: $P(T \mid D)$ (*sensitivity*, i.e. the probability that the test detects the disease if it is present) and $P(\overline{T} \mid \overline{D})$ (specificity, i.e. the probability that the test does not produce false alarms).

Also from general medical research the probability $P(D)$, i.e. the frequency of the disease among the population might be known.

After applying the test, however, one is typically interested in the opposite conditional probabilities: Suppose that the test shows a positive result, what is the probability $P(D \mid T)$ that the patient actually has the disease? And if one has a negative result, what is the probability the patient has the disease anyway?

It is easy to come up with shocking examples of tests with great sensitivity and specificity, but huge probabilities of false positive tests.

Definition 4.3.5. *Two events* A, B *are* independent *if* $P(A \mid B) = P(A)$

Observation 4.3.6. Two events A, B are *independent* if and only if $P(A \cap B) = P(A) \cdot P(B)$ if and only if $P(B \mid A) = P(B)$, which means that none has any influence over the other.

4.4 Random variables

Definition 4.4.1. *A* random variable *on a probability space* $(\Omega, \mathcal{A}, \mathrm{P})$ *is a measurable function* $(\Omega, \mathcal{A}) \to (\Omega', \mathcal{A}')$ *to some other measurable space.*

Remark 4.4.2. By far the most common are $\mathbb{R}^n$-valued random variables, i.e. measurable functions $(\Omega, \mathcal{A}) \to (\mathbb{R}^n, \mathcal{B}(\mathbb{R}^n))$. Mostly we will have $n = 1$.

In fact, in much of the literature the expression "random variable" only means $\mathbb{R}$-valued random variable!

In practice checking measurability is never an issue: Most often either measurability follows from continuity (see Example 4.1.4), or we are talking about a discrete probability space, where every subset is measurable (see Remark 4.2.2) and hence every function out of it is measurable.

Definition 4.4.3. *A random variable on a probability space induces a probability measure on its range: if* $X\colon (\Omega, \mathcal{A}, \mathrm{P}) \to (\Omega', \mathcal{C})$ *is measurable, then we can assign to it a probability measure* P_X *on* Ω' *by setting* $\mathrm{P}_X\colon \mathcal{C} \ni C \mapsto \mathrm{P}(X^{-1}(C))$*. This probability measure* P_X *is called the* probability distribution *of* X*.*

One writes $\mathrm{P}(X = a) := \mathrm{P}_X(\{a\}) = \mathrm{P}(\{\omega \in \Omega \mid X(\omega) = a\})$ *and* $\mathrm{P}(X \in A) := \mathrm{P}_X(A) = \mathrm{P}(\{\omega \in \Omega \mid X(\omega) \in A\})$*.*

Definition 4.4.4. *A random variable X is called* discrete*, resp.* continuous*, if its range together with the probability distribution* P_X *is a discrete, resp. continuous, probability space.*

Remark 4.4.5. A random variable X is discrete if it assumes only finitely many or countably many values $\{a_i \mid i \in \mathbb{N}\}$. In this case, its probability distribution is given by $\mathrm{P}_X(A) = \Sigma_{a_i \in A} \mathrm{P}(X = a_i)$. Thus P_X is determined by the individual values $\mathrm{P}(X = a_i)$, which together satisfy $\Sigma_{i \in \mathbb{N}} \mathrm{P}(X = a_i) = 1$ — the range of X becomes a discrete probability space with probability mass function $a_i \mapsto \mathrm{P}(X = a_i)$.

An $\mathbb{R}^n$-valued random variable X is continuous if its probability distribution is of the form $\mathrm{P}_X(A) = \int_A f$ for some $\mathbb{R}_{\geq 0}$-valued function f on $\mathbb{R}^n$. In this case $\mathrm{P}(X = a) = 0$ for any $a \in \mathbb{R}^n$, since an integral over a one point set is always zero.

In the special case of a continuous $\mathbb{R}$-valued random variable X one has $\mathrm{P}(a \leq X \leq b) := \mathrm{P}_X([a,b]) = \mathrm{P}(\{\omega \in \Omega \mid a \leq X(\omega) \leq b\}) = \int_a^b f(x)dx$. Here it maybe becomes particularly clear that $\mathrm{P}(X = a) = \mathrm{P}(a \leq X \leq a) = \int_a^a f(x)dx = 0$.

Remark 4.4.6. We can and will often work with a random variable entirely in terms of its distribution, without using the function at all. Accordingly, in the literature some times one reads statements like "let (X_1, X_2, X_3, X_4) be an $\mathbb{R}^4$-valued random variable with density function $f(x_1, x_2, x_3, x_4)$..." with domain Ω and the map $\Omega \to \mathbb{R}^4$ giving the random variable left unspecified.

If you insist that you want a random variable in the sense of the formal definition, you can simply take the identity map on the probability space with underlying set the range of the "random variable", some appropriate σ-algebra, and the probability distribution defined by the given density – here it would look like $(\mathbb{R}^4, \mathcal{B}(\mathbb{R}^4), \mathrm{P} : \mathbb{R}^4 \supseteq A \mapsto \int_A f)$.

Example 4.4.7. In Example 4.2.1 we can define the random variable $S \colon \Omega \to \mathbb{R}, (a,b) \mapsto a+b$. It can assume values from 2 to 12 and we can calculate $S^{-1}(2), \dots, S^{-1}(12)$, obtaining a discrete probability distribution P_S on the range $\{2, \dots, 12\}$ of S:

We simply have to count for each value, how many of the 36 possible results on the dice sum up to that value:

	1	**2**	**3**	**4**	**5**	**6**
1	(1,1)	(1,2)	(1,3)	(1,4)	(1,5)	(1,6)
2	(2,1)	(2,2)	(2,3)	(2,4)	(2,5)	(2,6)
3	(3,1)	(3,2)	(3,3)	(3,4)	(3,5)	(3,6)
4	(4,1)	(4,2)	(4,3)	(4,4)	(4,5)	(4,6)
5	(5,1)	(5,2)	(5,3)	(5,4)	(5,5)	(5,6)
6	(6,1)	(6,2)	(6,3)	(6,4)	(6,5)	(6,6)

Now we can calculate the induced probability distribution P_S as e.g. in $\mathrm{P}_S(\{5\}) = \mathrm{P}(S^{-1}(5)) = \mathrm{P}(\{(4,1),(3,2),(2,3),(1,4)\}) = \frac{4}{36}$.

Altogether we obtain $\mathrm{P}_S(\{2\}) = \frac{1}{36}, \mathrm{P}_S(\{3\}) = \frac{2}{36}, \mathrm{P}_S(\{4\}) = \frac{3}{36}, \mathrm{P}_S(\{5\}) = \frac{4}{36}$, $\mathrm{P}_S(\{6\}) = \frac{5}{36}$, $\mathrm{P}_S(\{7\}) = \frac{6}{36}$, $\mathrm{P}_S(\{8\}) = \frac{5}{36}$, $\mathrm{P}_S(\{9\}) = \frac{4}{36}$, $\mathrm{P}_S(\{10\}) = \frac{3}{36}$, $\mathrm{P}_S(\{11\}) = \frac{2}{36}$, $\mathrm{P}_S(\{12\}) = \frac{1}{36}$.

Example 4.4.8. The dart player of Example 4.2.3 would be interested the random variable $D : \mathbb{R}^2 \to \mathbb{R}_{\geq 0}$ given by $(x,y) \mapsto \sqrt{x^2+y^2}$ measuring how far the dart hits from the center. For the following remember the notation $\overline{B_d}(0) := \{(x,y) \in \mathbb{R}^2 \mid \sqrt{x^2+y^2} \leq d\}$.
The probability that the dart hits at a distance less than d from the center is

$$\begin{aligned} P(D \leq d) &= P_D(\{z \in \mathbb{R}_{\geq 0} \mid z \leq d\}) = P_D([0,d]) = P(D^{-1}([0,d])) \\ &= P(\{(x,y) \in \mathbb{R}^2 \mid \sqrt{x^2+y^2} \leq d\}) = P(\overline{B_d}(0)) \\ &= \int_{\overline{B_d}(0)} \frac{1}{2\pi} \exp(-\frac{x^2+y^2}{2}) \end{aligned}$$

where P_D denotes the induced probability measure on $\mathbb{R}_{\geq 0}$.

To compute this last integral, remember the change of variables formula for integrals: Let Φ be a differentiable bijection between two regions of $\mathbb{R}^n$, with differentiable inverse. Let $U, V \subseteq \mathbb{R}^n$ such that $\Phi(U) = V$. Finally let Jac_Φ be the Jacobian of Φ. Then we have

$$\int_{\Phi(U)} f = \int_U (f \circ \Phi) \cdot |\det Jac_\Phi|$$

We use this for transforming our integral into polar coordinates, i.e. we take

$$\Phi\colon [0,2\pi) \times \mathbb{R}_{\geq 0} \to \mathbb{R}^2, \quad (\varphi, r) \mapsto \begin{pmatrix} \cos\varphi & -\sin\varphi \\ \sin\varphi & \cos\varphi \end{pmatrix} \begin{pmatrix} r \\ 0 \end{pmatrix} = \begin{pmatrix} r\cos\varphi \\ r\sin\varphi \end{pmatrix}$$

The determinant of the Jacobian is

$$\begin{aligned} \det Jac_\Phi = \det \begin{pmatrix} \frac{\partial(r\cos\varphi)}{\partial\varphi} & \frac{\partial(r\cos\varphi)}{\partial r} \\ \frac{\partial(r\sin\varphi)}{\partial\varphi} & \frac{\partial(r\sin\varphi)}{\partial r} \end{pmatrix} &= \det \begin{pmatrix} -r\sin\varphi & \cos\varphi \\ r\cos\varphi & \sin\varphi \end{pmatrix} \\ &= -r(\sin\varphi)^2 - r(\cos\varphi)^2 = -r((\sin\varphi)^2 + (\cos\varphi)^2) = -r \end{aligned}$$

and we have $\Phi([0,2\pi] \times [0,r]) = B_r(0)$. Change of variables gives us

$$\begin{aligned} &\int_{\overline{B_d}(0)} \frac{1}{2\pi} \exp(-\frac{x^2+y^2}{2}) \\ &= \int_{[0,2\pi]\times[0,d]} \frac{1}{2\pi} \exp(-\frac{r^2}{2})|r| \,\mathrm{d}r\,\mathrm{d}\varphi \\ &= \int_0^{2\pi} \int_0^d \frac{1}{2\pi} \exp(-\frac{r^2}{2})|r| \,\mathrm{d}r\,\mathrm{d}\varphi \\ &= \int_0^{2\pi} 1 \,\mathrm{d}\varphi \cdot \int_0^d \frac{1}{2\pi} \exp(-\frac{r^2}{2}) r \,\mathrm{d}r \\ &= 2\pi \cdot \left(-\frac{1}{2\pi} \exp(-\frac{r^2}{2}) \right)\Bigg|_{r=0}^{r=d} \\ &= 1 - \exp(-\frac{d^2}{2}) \end{aligned}$$

So altogether we obtain $P(D \leq d) = 1 - \exp(-\frac{d^2}{2})$.

Note that in the limit case $d \to \infty$ this shows that the density function of Example 4.2.3 is in fact a density function, i.e. its integral over all of $\mathbb{R}^2$ is equal to 1.

What we just calculated, was the so-called cumulative distribution function of the random variable D:

Definition 4.4.9. *Let X be an $\mathbb{R}$-valued random variable. The function*

$$F_X \mathbb{R} \to \mathbb{R}, \quad r \mapsto P_X(X \leq r)$$

is called the cumulative distribution function *(or short* cdf*) of X.*

Remark 4.4.10. The cumulative distribution function completely determines the distribution of the random variable: A general interval $[a,b]$ fits into the equation of sets $(-\infty,b] = (-\infty,a]\coprod[a,b]$, therefore we have $P(a \leq X \leq b) = P_X([a,b]) = P((-\infty,b]) - P((-\infty,a]) = F_X(b) - F_X(a)$. A general Borel measurable subset of $\mathbb{R}$ can be approximated by intervals and its probability is determined by the probability of intervals.

Example 4.4.11. In the situation of Example 4.4.8 define

$$g(r) := \begin{cases} 0 & \text{if } r \leq 0 \\ \exp(-\frac{r^2}{2})r & \text{if } r \geq 0 \end{cases}$$

From the calculation in Example 4.4.8 one could see that the cumulative distribution function, for $d \geq 0$, is given by

$$P(D \leq d) = \int_0^d \exp(-\frac{r^2}{2})r\,\mathrm{d}r = \int_{-\infty}^d g(r)\,\mathrm{d}r$$

The latter formula is also true for negative d. It follows that $P(a \leq D \leq b) = \int_a^b g(r)\,\mathrm{d}r$. Approximating a general (Borel measurable) subset of $\mathbb{R}$ by intervals, one can conclude that D is a continuous random variable with density function g.

Definition 4.4.12. *Given a probability space whose set of elementary events is a product of two sets, $(C \times D, P)$, one gets a random variable $C \times D \to D$ given by the projection. The distribution of that random variable, which is a probability measure on D, is called the* marginal distribution *with respect to D.*

Example 4.4.13. In the dart example Ex. 4.2.3 define $X : \mathbb{R}^2 \to \mathbb{R}$ to be the x-coordinate of the point where the dart hits. To compute the induced probability distribution, note that for a subset $A \subseteq \mathbb{R}$ we have $X^{-1}(A) = \{(x,y) \mid x \in A\}$ Thus the induced probability distribution is $\mathrm{P}_X \colon \mathbb{R} \supseteq A \mapsto \mathrm{P}(X^{-1}(A)) = \int_{A\times\mathbb{R}} \frac{1}{2\pi} \exp\left(-\frac{x^2+y^2}{2}\right) = \int_A \int_{-\infty}^{\infty} \frac{1}{2\pi} \exp\left(-\frac{x^2+y^2}{2}\right) dydx$. We see that the induced probability measure is again given by integrating over a density function (namely the function $x \mapsto \int_{-\infty}^{\infty} \frac{1}{2\pi} \exp\left(-\frac{x^2+y^2}{2}\right) dy$) and thus we have a continuous random variable. We will later calculate this density function more explicitly.

4.5 Independence, Expectation, Covariance

Definition 4.5.1. *The joint distribution of two random variables $X\colon \Omega \to T$, $Y\colon \Omega \to S$, defined on the same probability space $(\Omega, \mathcal{A}, P)$, is the probability measure $P_{X,Y}$ that associates to $A \subseteq T, B \subseteq S$ the number*

$$P(X \in A, Y \in B) := P(\{\omega \in \Omega \mid X(\omega) \in A \text{ and } Y(\omega) \in B\}).$$

Definition 4.5.2. *Two random variables $X, Y\colon \Omega \to M$ are independent if and only if for all sets $A, B \subseteq M$ the events $(X \in A) := \{\omega \in \Omega \mid X(\omega) \in A\}$ and $(Y \in B) := \{\omega \in \Omega \mid Y(\omega) \in B\}$ are independent.*

In other words, for all $A, B \subseteq M$, we want

$$P(X \in A, Y \in B) = P(X \in A) \cdot P(Y \in B)$$

Example 4.5.3. Two $\mathbb{R}$-valued random variables X, Y are independent if and only if $\{a \leq X \leq b\}$ and $\{c \leq Y \leq d\}$ are independent events for all $a, b, c, d \in \mathbb{R}$. It suffices to check this for intervals because these generate measurable sets.

For continuous independent random variables it is easy to compute the density function of their joint distribution:

Proposition 4.5.4. *Let X, Y be independent continuous $\mathbb{R}$-valued random variables with density functions f, g. Then the joint distribution of X, Y is continuous with density function $f \cdot g$.*

Proof.

$$\begin{aligned} P_{X,Y}(A, B) &= P(X \in A, Y \in B) = P(X \in A) \cdot P(Y \in B) \\ &= \int_A f dx \cdot \int_B g dy = \int_{A \times B} f \cdot g \, dx dy \end{aligned}$$

where the second equation uses the hypothesis that X, Y are independent and the last equation uses Fubini's theorem. □

Definition 4.5.5 (expectation for discrete and continuous random variables)**.** *Let X be an $\mathbb{R}$-valued random variable on a probability space $(\Omega, \mathcal{A}, \mathrm{P})$. The* expectation*, or* expected value *of X, denoted by $\mathrm{E}(X)$, is the real number defined as follows:*

If X is discrete:

$\mathrm{E}(X) := \Sigma_{r\in\mathbb{R}} \;\; r\cdot \mathrm{P}(X = r)$ if the sum converges absolutely, i.e. if $\Sigma_{r\in\mathbb{R}} |r|\,\mathrm{P}(X = r)$ converges.

If X is continuous with density function f:
$\mathrm{E}(X) := \int_{\mathbb{R}} x \cdot f(x)dx$ if $\int_{-\infty}^{\infty} |x| \cdot f(x)dx$ exists.

The expectation of an $\mathbb{R}^n$-valued random variable $\mathbf{X} = (X_1, \dots, X_n)^T$ is $E\mathbf{X} := (EX_1, \dots, EX_n)^T$.

Note that in the discrete case the sum is countable, since only for countably many $r \in \mathbb{R}$ we have $\mathrm{P}(X = r) \neq 0$.

If in the above definition one obtains the density function as the limit of piecewise constant functions then the expectation in the continuous case is the limit of the expectations of the approximating discrete random variables – see Rozanov, Section 4.8 for this.

Remark 4.5.6. The expectation is defined for general $\mathbb{R}$-valued random variables via approximation by discrete variables. In the case of a continuous random variable the above definition is then a theorem. We will only need continuous and discrete random variables.

Also there are some fine distinctions whether the expectation exists, and sometimes one allows it to be infinite, but we will just assume that the absolute convergence conditions in the above definition hold whenever we use expectations, e.g. in Theorem 4.5.10 below.

Example 4.5.7. The expected value of the sum of two dice rolls is 7. To see this one can use the definition and the calculations of Example 4.4.7. Alternatively one can declare the result of the first die and the result of the second die to be two different random variables X and Y, calculate each of their expectations to be 3.5, and then use Theorem 4.5.10 below to calculate $E(X + Y)$.

Example 4.5.8. The expected value of the x-coordinate of the dart of Examples 4.2.3 and 4.4.13 is $\int_{-\infty}^{\infty} x{\cdot}f(x)dx$, where $f(x) = \int_{-\infty}^{\infty} \frac{1}{2\pi} \exp\left(-\frac{x^2+y^2}{2}\right) dy$ is the density function. This can be seen to be zero by symmetry: We have $f(x) = f(-x)$, so $\int_{-\infty}^{\infty} x{\cdot}f(x)dx = \int_{-\infty}^{0} x{\cdot}f(x)dx + \int_{0}^{\infty} x{\cdot}f(x)dx = -\int_{0}^{\infty} x{\cdot}f(x)dx + \int_{0}^{\infty} x{\cdot}f(x)dx = 0$, provided the last integral converges. In fact it does, but we will not go into this.

Example 4.5.9. Consider an $\mathbb{R}^4$-valued random variable (X_1, X_2, X_3, X_4) with density function

$f(x_1, x_2, x_3, x_4) := c\cdot(x_1^4 + x_2^2 + x_3^2 + x_4^2) \cdot \chi_I$, where $c := \frac{2^5-1+5\cdot 3^4}{3^5}$ and χ_I is the indicator function of the region $I := [\frac{1}{3}, \frac{2}{3}] \times [0, 1]^3$, i.e. $\chi_I(x) = 1$ if $x \in I$ and $= 0$ otherwise. We compute the expectation of the $\mathbb{R}$-valued random variable $X_1 \cdot X_2$:

$$\begin{aligned}
\mathrm{E}(X_1 \cdot X_2) &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} x_1 x_2 f(x_1, x_2, x_3, x_4) \cdot \chi_I \; dx_1 dx_2 dx_3 dx_4 \\
&= \int_{\frac{1}{3}}^{\frac{2}{3}}\int_0^1\int_0^1\int_0^1 x_1 x_2 f(x_1, x_2, x_3, x_4) \; dx_4 dx_3 dx_2 dx_1 \\
&= \int_{\frac{1}{3}}^{\frac{2}{3}}\int_0^1\int_0^1 c\left(x_1 x_2 (x_4 x_1^4 + x_4 x_2^2 + x_4 x_3^2 + \frac{1}{3}x_4^3)\right)\Big|_0^1 \; dx_3 dx_2 dx_1 \\
&= c\int_{\frac{1}{3}}^{\frac{2}{3}}\int_0^1\int_0^1 x_1 x_2 (x_1^4 + x_2^2 + x_3^2 + \frac{1}{3}) \; dx_3 dx_2 dx_1 \\
&= c\int_{\frac{1}{3}}^{\frac{2}{3}}\int_0^1 x_1 x_2 (x_1^4 + x_2^2 + \frac{2}{3}) \; dx_2 dx_1 \\
&= c\int_{\frac{1}{3}}^{\frac{2}{3}}\int_0^1 x_1^5 x_2 + x_1 x_2^3 + \frac{2}{3} x_1 x_2 \; dx_2 dx_1 \\
&= c\int_{\frac{1}{3}}^{\frac{2}{3}} \frac{1}{2}x_1^5 + \frac{1}{4}x_1 + \frac{1}{3}x_1 \; dx_1 \\
&= c \cdot (\frac{2^4}{3^5} + \frac{1}{6} + \frac{2}{9} - \frac{1}{2 \cdot 3^5} - \frac{1}{12} - \frac{1}{3^6})
\end{aligned}$$

Theorem 4.5.10. *Expectation is linear: If X,Y are $\mathbb{R}$-valued random variables and $a, b \in \mathbb{R}$, one can form a new random variable $aX + bY$, namely the function $\Omega \to \mathbb{R}, \omega \mapsto aX(\omega) + bY(\omega)$. The expectation of this new random variable is described as follows:*

$$E(aX + bY) = aE(X) + bE(Y)$$

Proof. We only give the proof in the case of a discrete probability space $(\Omega, \mathcal{A}, \mathrm{P})$. In this case let $\{a_i \mid i \in \mathbb{N}\} \subseteq \mathbb{R}$ be the (countable) set of values of X.

$$\begin{aligned}
\mathrm{E}(X) &= \Sigma_{r\in\mathbb{R}} \; r\cdot\mathrm{P}(X = r) \\
&= \Sigma_{i\in\mathbb{N}} \; a_i\cdot\mathrm{P}(X = a_i) \\
&= \Sigma_{i\in\mathbb{N}} \; a_i\cdot(\Sigma_{\{\omega \mid X(\omega)=a_i\}} \mathrm{P}(\omega)) \\
&= \Sigma_{\omega\in\Omega} \; X(\omega)\,\mathrm{P}(\omega)
\end{aligned}$$

Likewise we have $\mathrm{E}(Y) = \Sigma_{\omega\in\Omega}\; Y(\omega)\,\mathrm{P}(\omega)$. Now one sees

$$\begin{aligned} E(aX+bY) &= \Sigma_{\omega\in\Omega}\;(aX(\omega)+bY(\omega))\,\mathrm{P}(\omega) \\ &= a(\Sigma_{\omega\in\Omega}\;X(\omega)\,\mathrm{P}(\omega)) + b(\Sigma_{\omega\in\Omega}\;Y(\omega)\,\mathrm{P}(\omega)) \\ &= aE(X)+bE(Y) \end{aligned}$$

where we are allowed to reorder the sum because of absolute convergence. The continuous case follows by approximating the given random variables by ones defined on discrete probability spaces. □

Corollary 4.5.11. *For an $\mathbb{R}^n$-valued random variable $\mathbf{X} = (X_1,\ldots,X_n)^T$ and an $m\times n$-matrix A one has $E(A\mathbf{X}) = AE(\mathbf{X})$.*

Proof. Let $A = (a_{ij})$. Then we have

$$E(A\mathbf{X}) = E\begin{pmatrix} a_{11}X_1+\ldots+a_{1n}X_n \\ \vdots \\ a_{m1}X_1+\ldots+a_{mn}X_n \end{pmatrix} = \begin{pmatrix} E(a_{11}X_1+\ldots+a_{1n}X_n) \\ \vdots \\ E(a_{m1}X_1+\ldots+a_{mn}X_n) \end{pmatrix}$$

$$\overset{Thm.4.5.10}{=} \begin{pmatrix} a_{11}EX_1+\ldots+a_{1n}EX_n \\ \vdots \\ a_{m1}EX_1+\ldots+a_{mn}EX_n \end{pmatrix} = AE\mathbf{X}$$

□

Definition 4.5.12. *The* covariance *of two $\mathbb{R}$-valued random variables X, Y is defined as* $\mathrm{Cov}(X,Y) := E((X-EX)(Y-EY))$

The variance *of an $\mathbb{R}$-valued random variable X is defined as* $\mathrm{Var}(X) := E((X-E(X))^2) = \mathrm{Cov}(X,X)$.

Remark 4.5.13. To make sense of Def. 4.5.12, one has to interpret the number $E(X)$ as a constant random variable taking this number as its value.

Remark 4.5.14. Variance measures how much a random variable deviates from its expected value. It is always ≥ 0: only the absolute value of the deviation is measured.

Covariance of two random variables X, Y measures whether the deviations of X and Y from their expected values happen into the same directions, i.e. whether when X grows bigger than its expectation so does Y (positive covariance), or when X becomes smaller then Y becomes bigger (negative covariance), or there is no such tendency (covariance close to zero).

Lemma 4.5.15. $\operatorname{Var}(X) = E(X^2) - (EX)^2$

Proof. We have

$$\begin{aligned}\operatorname{Var}(X) &:= E((X - E(X))^2) = E(X^2 - 2XEX + (EX)^2) \\ &= E(X^2) - 2(EX)(EX) + (EX)^2 = E(X^2) - (EX)^2\end{aligned}$$

where in the passage to the second line we used the linearity of expectation. □

Example 4.5.16. Consider an $\mathbb{R}^3$-valued random variable (X_1, X_2, X_3) with density function $f(x_1, x_2, x_3) := (x_1^2 + x_2^2 + x_3^2) \cdot \chi_I$, where χ_I is the indicator function of the unit cube $I := [0,1]^3$, i.e. $\chi_I(x) = 1$ if $x \in I$ and $= 0$ otherwise. That is, the density function is given by $f(x_1, x_2, x_3) := (x_1^2 + x_2^2 + x_3^2)$ in the unit cube and is 0 otherwise.

We compute the variance of the $\mathbb{R}^3$-valued random variable $W := \frac{8X_1X_2X_3}{\sqrt{X_1^2+X_2^2+X_3^2}}$ by the formula of Lemma 4.5.15. First we need to compute expectation.

$$
\begin{aligned}
EW &= \int_0^1 \int_0^1 \int_0^1 \frac{8xyz}{\sqrt{x^2+y^2+z^2}}(x^2+y^2+z^2)\; dxdydz \\
&= 8\int_0^1 \int_0^1 \int_0^1 xyz(x^2+y^2+z^2)^{\frac{1}{2}}\; dxdydz \\
&= 4\int_0^1 \int_0^1 yz \int_0^1 2x(x^2+y^2+z^2)^{\frac{1}{2}}\; dxdydz \\
&= 4\int_0^1 \int_0^1 yz \left[\frac{2}{3}(x^2+y^2+z^2)^{\frac{3}{2}}\right]_{x=0}^{x=1} dydz \\
4\frac{2}{3}&\int_0^1 \int_0^1 yz \left[(1+y^2+z^2)^{\frac{3}{2}} - (y^2+z^2)^{\frac{3}{2}}\right] dydz \\
&= 2\frac{2}{3}\int_0^1 z \int_0^1 2y \left[(1+y^2+z^2)^{\frac{3}{2}} - (y^2+z^2)^{\frac{3}{2}}\right] dydz \\
&= 2\frac{2}{3}\int_0^1 z\frac{2}{5} \left[(1+y^2+z^2)^{\frac{5}{2}} - (y^2+z^2)^{\frac{5}{2}}\right]_{y=0}^{y=1} dz \\
&= 2\frac{2}{3}\frac{2}{5}\int_0^1 z \left[(1+y^2+z^2)^{\frac{5}{2}} - (y^2+z^2)^{\frac{5}{2}}\right]_{y=0}^{y=1} dz \\
&= \frac{2}{3}\frac{2}{5}\int_0^1 2z \left[(2+z^2)^{\frac{5}{2}} - 2(1+z^2)^{\frac{5}{2}} + (z^2)^{\frac{5}{2}}\right] dz \\
&= \frac{2}{3}\frac{2}{5}\frac{2}{7} \left[(2+z^2)^{\frac{7}{2}} - 2(1+z^2)^{\frac{7}{2}} + (z^2)^{\frac{7}{2}}\right]_{z=0}^{z=1} \\
&= \frac{2}{3}\frac{2}{5}\frac{2}{7} \left[3^{\frac{7}{2}} - 2\cdot 2^{\frac{7}{2}} + 1^{\frac{7}{2}} - 2^{\frac{7}{2}} + 2{\cdot}1^{\frac{7}{2}} - 0^{\frac{7}{2}}\right] \\
&= \frac{2}{3}\frac{2}{5}\frac{2}{7} \left[3^{\frac{7}{2}} - 3\cdot 2^{\frac{7}{2}} + 3\right]
\end{aligned}
$$

$$
\begin{aligned}
E(W^2) &= \int_0^1 \int_0^1 \int_0^1 \frac{64x^2y^2z^2}{x^2+y^2+z^2}(x^2+y^2+z^2)\; dxdydz \\
&= 64\int_0^1 \int_0^1 \int_0^1 x^2y^2z^2\; dxdydz \\
&= 64\int_0^1 \int_0^1 y^2z^2 \left[\frac{1}{3}x^3\right]_{x=0}^{x=1} dydz = 64\int_0^1 \int_0^1 y^2z^2\frac{1}{3}\; dydz = \ldots = \frac{64}{27}
\end{aligned}
$$

$$
Var(W) = E(W^2) - (EW)^2 = \frac{64}{27} - \left(\frac{2}{3}\frac{2}{5}\frac{2}{7}\left[3^{\frac{7}{2}} - 3\cdot 2^{\frac{7}{2}} + 3\right]\right)^2
$$

Proposition 4.5.17. *Covariance is symmetric and bilinear: We have*

- $\mathrm{Cov}(X,Y) = \mathrm{Cov}(Y,X)$
- $\mathrm{Cov}(aX + bY, Z) = a\,\mathrm{Cov}(X,Z) + b\,\mathrm{Cov}(Y,Z)$

Proof. Exercise. □

Remark 4.5.18. By Prop. 4.5.17 covariance almost has the properties of an inner product. The only axiom that is not satisfied is that $\mathrm{Cov}(X,X) = \mathrm{Var}(X) = 0$ does not imply $X = 0$ – see the corrsponding exercise on Sheet 7. The random variables with $\mathrm{Var}(X) = 0$ form, however, a subvector space of all random variables, and the covariance becomes an inner product on the quotient space. The notion of "length" of a random variable X associated to that inner product is $\|X\| = \sqrt{\mathrm{Cov}(X,X)} = \sqrt{\mathrm{Var}(X)} =: \sigma_X$. This is known as the *standard deviation* of X. The analogue of the formula $\cos\varphi = \frac{\langle v,w\rangle}{\|v\|\cdot\|w\|}$ features the *correlation coefficient* $\rho_{XY} := \frac{\mathrm{Cov}(X,Y)}{\sigma_X\sigma_Y}$.

We record the numbers motivated in the last remark in their own definition:

Definition 4.5.19. *(i) The* standard deviation *of X is $\sigma_X := \sqrt{\mathrm{Var}(X)}$.*

(ii) The correlation coefficient *of X and Y is defined as $\rho_{XY} := \frac{\mathrm{Cov}(X,Y)}{\sigma_X\sigma_Y}$.*

(iii) Two random variables X, Y are called uncorrelated *if $\mathrm{Cov}(X,Y) = 0$.*

Remark 4.5.20. For the inner product picture of Remark 4.5.18 you should not imagine random variables X, Y as vectors whose entries are the possible values of X, resp. Y – this would anyway not be a complete picture of the random variable because the respective probabilities would not be recorded anywhere. Rather you should see a random variable as a map $\Omega \to \mathbb{R}$ for some probability space Ω, which can be understood as a vector with entries $X(\omega)$, going through the $\omega \in \Omega$, i.e. of length the size of Ω. For a finite Ω this really is a finite length vector, and covariance of centered (i.e. with expectation 0) random variables is an inner product between vectors in the usual sense, given by a diagonal matrix whose entry a_{ii} is the probability of the i-th elementary event ω_i. To compute covariance in this way the two random variables have to be given as maps from the same set Ω.

Now often you are not given a probability space Ω, but only two lists of values and their respective probabilities (or two density functions in the continuous case). But you can always arrange such an Ω!

Here is an example: Suppose X takes values $1, 2, 3$ with probabilities $\frac{1}{2}, \frac{1}{3}, \frac{1}{6}$ and Y takes values $-1, 2$ with probabilities $\frac{1}{4}, \frac{3}{4}$. To be able to talk about covariance at all, i.e. to make sense of the expectation $E((X - EX)(Y - EY))$ you need to know the joint distribution: If the joint probability mass function is $f(x, y)$, the covariance is

$$\operatorname{Cov}(X, Y) = \sum_{x \in \{1,2,3\}} \sum_{y \in \{-1,2\}} f(x, y) \cdot (x - EX) \cdot (y - EY).$$

Let's say that the joint probability mass function is given by this table:

$X \backslash Y$	-1	2
1	$\frac{3}{24}$	$\frac{9}{24}$
2	$\frac{1}{24}$	$\frac{7}{24}$
3	$\frac{2}{24}$	$\frac{2}{24}$

Let $\Omega := \{\omega_{(1,-1)}, \omega_{(2,-1)}, \omega_{(3,-1)}, \omega_{(1,2)}, \omega_{(2,2)}, \omega_{(3,2)}\}$ and set

$$\begin{aligned}
&X(\omega_{(1,-1)}) = X(\omega_{(1,2)}) = 1 \\
&X(\omega_{(2,-1)}) = X(\omega_{(2,2)}) = 2 \\
&X(\omega_{(3,-1)}) = X(\omega_{(3,2)}) = 3 \\
&Y(\omega_{(1,-1)}) = Y(\omega_{(2,-1)}) = Y(\omega_{(3,-1)}) = -1 \\
&Y(\omega_{(1,2)}) = Y(\omega_{(2,2)}) = Y(\omega_{(3,2)}) = 2
\end{aligned}$$

In general, if X takes values in a set $\mathcal{X}$ and Y takes values in a set $\mathcal{Y}$, you define $\Omega := \mathcal{X} \times \mathcal{Y}$, take the joint probability mass function $f(x, y)$ to define $P((x, y)) := f(x, y)$ and define $X \colon \Omega = \mathcal{X} \times \mathcal{Y} \to \mathcal{X}$, resp. $Y \colon \Omega = \mathcal{X} \times \mathcal{Y} \to \mathcal{Y}$ to be the projection to the first, resp. second, component. This gives a probability space, and the vector space of random variables on this probability space is where an inner product for X and Y is defined.

You can see that for random variables with finitely many possible values you get a probability space Ω with finitely many elements, and covariance really becomes an inner product on $\mathbb{R}^n$ (up to an isomorphism), where n is the size of Ω.

For discrete random variables with infinitely many possible values you get the same story, including the inner product, just with infinite sums.

For continuous random variables you replace the joint probability mass function with a joint density function, you get an uncountable probability space $\Omega = \mathcal{X} \times \mathcal{Y}$ and you get an inner product of the form $\langle g, h \rangle = \int_\Omega g \cdot h$.

If you have not seen this before it may look strange, but it is actually a usual kind of inner product on Hilbert spaces.

Remark 4.5.21. Covariance measures correlation, not (in)dependence! As an example consider e.g. the random variable X that is uniformly distributed on $[-1,1]$ and the random variable $Y := X^2$. We have $\mathrm{Cov}(X,Y) = E((X-EX)(Y-EY)) = E(XY) - EX \cdot EY = E(XY) = \int_{-1}^{1} x^3 \frac{1}{2} - 0 \cdot EY = 0$, but $P(Y \leq \frac{1}{4} \mid -\frac{1}{2} \leq X \leq \frac{1}{2}) = 1 \neq P(Y \leq \frac{1}{4})$.

Definition 4.5.22. *Given an $\mathbb{R}^n$-valued random variable $\mathbf{X} = (X_1, \dots, X_n)^T$ we define its* covariance matrix $\mathrm{Cov}(\mathbf{X})$ *to be the $n \times n$-matrix whose entry at the place (i,j) is* $\mathrm{Cov}(X_i, X_j) = E((X_i - EX_i)(X_j - EX_j))$.

More generally, the covariance matrix of two $\mathbb{R}^n$-valued random variables $\mathbf{X}, \mathbf{Y}$ is defined to be the matrix $\mathrm{Cov}(\mathbf{X}, \mathbf{Y})$ *whose entry at the place (i,j) is* $\mathrm{Cov}(X_i, Y_j)$.

The following expression of the covariance matrix is useful in computations:

Proposition 4.5.23. $\mathrm{Cov}(\mathbf{X}, \mathbf{Y}) = E((\mathbf{X} - E\mathbf{X})(\mathbf{Y} - E\mathbf{Y})^T)$

Proof. First, the expression on the right hand side is to be understood as the expectation of a *matrix valued* random variable – this means simply that we take the expectation of each of its entries, thus obtaining a matrix of real numbers.

The matrix of random variables in question is

$$(\mathbf{X} - E\mathbf{X})(\mathbf{Y} - E\mathbf{Y})^T = \begin{pmatrix} X_1 - EX_1 \\ \vdots \\ X_n - EX_n \end{pmatrix} \cdot \begin{pmatrix} Y_1 - EY_1, & \cdots & , Y_n - EY_n \end{pmatrix}$$

$$= \begin{pmatrix} (X_1 - EX_1)(Y_1 - EY_1) & \dots & (X_1 - EX_1)(Y_n - EY_n) \\ \vdots & \ddots & \vdots \\ (X_n - EX_n)(Y_1 - EY_1) & \dots & (X_n - EX_n)(Y_n - EY_n) \end{pmatrix}$$

Now taking the expectation of each entry we get exactly the covariance matrix. □

Remark 4.5.24. The fact that we can diagonalize the covariance matrix (because it is symmetric, by the Principal Axis Theorem 1.5.3) means that we can always find a basis of linear combinations of our features which are mutually uncorrelated!

For an example see the computation of the covariance of the Gaussian in the next section.

Remark 4.5.25 (Higher moments)**.** The expectation EX and the variance $\mathrm{Var}(X) = E(X^2) - (EX)^2$ feature the first two of the sequence of numbers $E(X^n)$, that one can associate to any $\mathbb{R}$-valued random variable. The number

$$E(X^n) = \begin{cases} \sum_{i=0}^{\infty} a_i^n P(a_i) & \text{if } X \text{ discrete} \\ \int_{-\infty}^{\infty} x^n f(x)\,\mathrm{d}x & \text{if } X \text{ continuous with density function } f \end{cases}$$

is called the *n-th moment* of X.

If all moments exist, i.e. if none of these sums/integrals diverges, then one can assemble them into the so-called *moment generating function*, defined by $M_X(t) := E(e^{tX}) = E(\Sigma_{k\in\mathbb{N}} \frac{t^k X^k}{k!}) = \Sigma_{k\in\mathbb{N}} \frac{t^k E(X^k)}{k!}$. This is at first just a formal power series, not necessarily used as a function, which stores the moments in its coefficients. It can be usefully employed, regardless of whether it converges anywhere or not[1].

If, however, the function converges for all t in some neighbourhood of 0, then there is at most one probability distribution with this moment generating function, so in this case all moments together determine the distribution of the random variable! This appears e.g. as Theorem 30.1 of Billingsley, Probability and Measure. If you happen to know about those, there is also an easy explanation in terms of Laplace transforms. This is the basis of one of the possible proofs of the central limit theorem.

The 3rd moment $E(X^3)$ encodes the *skewness* of the random variable. It measures how asymmetric the distribution is around the mean. Similar things could be said about higher odd moments.

The 4th moment $E(X^4)$ encodes the *kurtosis* of the random variable. It measures how likely a distribution is to produce outliers, far away from its expectation. Similar things could be said about higher even moments.

An immediate application of moments is this: Suppose you have a function g that can be expressed as a converging power series, $g(x) = \sum_{i=0}^{\infty} a_i x^i$ – this e.g. true of the logarithm and exponential functions. If now X is a random variable, then the expectation of the transformed random variable $g(X)$ is $E(g(X)) = \sum_{i=0}^{\infty} a_i E(X^i)$, and thus a function of the moments of X.

[1] This is an instance of the general combinatorial technique of generating functions.

More generally, one can sometimes compute also the higher moments of $E(g(X)^n)$ using the power series representation, and thereby determine the whole moment generating function $M_{g(X)}(t)$. From this one can try to get back $g(X)$, in the best case simply by recognizing $M_{g(X)}(t)$ as the moment generating function of some known distribution, or else by calculating the so-called inverse Laplace transform).

Another application is also known as "the method of moments". Here one exploits that from given data one can easily extract a good estimate of the moments (one just has to average the powers of the different samples). The bare numbers one gets out of this might themselves already be used for numerical approximation of the density function. But one is in better shape if one already has a guess that the sought distribution belongs to a certain parametrized family of distributions. In this case one often can calculate the concrete parameters from finitely many moments, and thus get a sharper estimate. More on this in the Statistics part of the course.

4.6 The algebra of random variables

So $\mathbb{R}$-valued random variables are (measurable) maps from some probability space to $\mathbb{R}$. In practice the map is what gives the random variable its meaning, or its context, but the induced probability distribution on $\mathbb{R}$ is all one needs to work with it.

The map viewpoint makes the following fact apparent: If $(\Omega, \mathcal{A}, P)$ is a probability space, then the set of all maps $\Omega \to \mathbb{R}$ forms an $\mathbb{R}$-vector space, with respect to pointwise addition and scalar multiplication. The set of all measurable maps is a subvector space. If, as I invite you to do, choose to ignore the issues of σ-algebras and measurability, then we are just talking about the vector space of all maps $\Omega \to \mathbb{R}$ that occurred in Example 1.1.4.

Much more generally, given any measurable function $h\colon \mathbb{R}^n \to \mathbb{R}$ and n random variables $X_1, \ldots, X_n\colon \Omega \to \mathbb{R}$, we can form the new random variable

$$h(X_1, \ldots, X_n)\colon \Omega \xrightarrow{(X_1,\ldots,X_n)} \mathbb{R}^n \xrightarrow{h} \mathbb{R}, \quad \omega \mapsto h(X_1(\omega), \ldots, X_n(\omega))$$

In particular we can multiply random variables, take powers, exponentials, and apply any other piecewise continuous function [2]. Even more

[2] In fact one can axiomatize this behaviour of the collection of all random variables on

generally we can transform $\mathbb{R}^n$-valued random variables into $\mathbb{R}^m$-valued random variables using (measurable) functions $\mathbb{R}^n \to \mathbb{R}^m$.

The question then arises, how to describe the pointwise operations that we obviously have in the function space picture, in the picture of an $\mathbb{R}$-valued random variable as a distribution on $\mathbb{R}$. For example, given random variables X and Y, how can one compute the distributions of the new random variables $X + Y$ or $X \cdot Y$ from the distributions of X and Y? And if the random variables are continuous, how can one get their density functions?

Proposition 4.6.1. *Suppose X is a random variable taking values in $[a, b] \subseteq \mathbb{R}$ and $g\colon [a, b] \to [c, d] \subseteq \mathbb{R}$ is a (measurable) strictly increasing function (i.e. $r < s \Rightarrow g(r) < g(s)$). Define a new random variable by $Y := g(X)$. Then*

(i) The cumulative distribution function of Y is given by $F_Y(y) = F_X(g^{-1}(y))$.

(ii) If X is continuous and g^{-1} has a continuous derivative, then Y is continuous with density function $\frac{\partial F_Y}{\partial y}(y) = f_X(g^{-1}(y)) \cdot (g^{-1})'(y)$

Proof. (i) $F_Y(y) = P(Y \leq y) = P(g(X) \leq y) = P(X \leq g^{-1}(y)) = F_X(g^{-1}(y))$

(ii) If X is continuous with density function f_X, then by (i)

$$\begin{aligned} F_Y(y) = F_X(g^{-1}(y)) &= \int_a^{g^{-1}(y)} f_X(x)\,\mathrm{d}x \\ &= \int_{g^{-1}(c)}^{g^{-1}(y)} f_X(x)\,\mathrm{d}x = \int_c^y f_X(g^{-1}(x)) \cdot (g^{-1})'(x)\,\mathrm{d}x \end{aligned}$$

By the fundamental theorem of Calculus we have

$$\frac{\partial F_Y}{\partial y}(y) = f_X(g^{-1}(y)) \cdot (g^{-1})'(y)$$

□

a fixed probability space, obtaining the notion of commutative von Neumann-algebra. Substantial parts of probability theory can be built on the grounds of just these axioms, and their non-commutative version can serve as an environment for the theory of random matrices.

This gives an easy strategy to calculate density functions of invertibly transformed univariate random variables: Apply part (i) of Prop. 4.6.1 and derive the function obtained in this way – for this you don't really have to remember the formula of part (ii).

Example 4.6.2. Let X be a continuous $\mathbb{R}$-valued random variable with density function

$$f_X(x) = \begin{cases} 4x^3 & \text{if } x \in [0,1] \\ 0 & \text{otherwise} \end{cases}$$

and let $Y := \sqrt{X}$. The cumulative distribution function of Y is

$$\begin{aligned} F_Y(y) &= P(Y \leq y) = P(\sqrt{X} \leq y) = P(X \leq y^2) = \int_0^{y^2} 4x^3 \, \mathrm{d}x \\ &= x^4 \mid_0^{y^2} = y^8 \end{aligned}$$

The density function of Y is therefore $\frac{\partial F_Y}{\partial y} = 8y^7$.

More generally we can use the transformation formula for higherdimensional integrals

Proposition 4.6.3. *Suppose X is a continuous random variable, with density function f_X, taking values in $U \subseteq \mathbb{R}^n$. Suppose $g\colon U \to V \subseteq \mathbb{R}^m$ is an invertible function with differentiable inverse and define a new random variable by $Y := g(X)$. Then Y is continuous with density function*

$$f_Y(y) = f_X(g^{-1}(y)) \cdot |\det D(g^{-1})(y)|$$

where $D(g^{-1})$ is the differential of the inverse function.

Proof.

$$\begin{aligned} P(Y \in A) &= P(g(X) \in A) = P(X \in g^{-1}(A)) \\ &= \int_{g^{-1}(A)} f_X = \int_A f_X(g^{-1}(y)) \cdot |\det D(g^{-1})(y)| \end{aligned}$$

where the last equality holds by the higherdimensional change of variables formula. □

Example 4.6.4. See Deisenroth/Faisal/Ong, Ex. 6.17

One can not only compute the distributions of random variables that were transformed by an invertible operation. As an extremely important example we mention the distribution of the sum of two random variables:

Proposition 4.6.5. *Let X, Y be two $\mathbb{R}$-valued random variables. Define $Z := X + Y$.*

(i) If X, Y are discrete, taking values in $\mathcal{X} := \{x_i \mid i \in \mathbb{N}\}$, $\mathcal{Y} := \{y_i \mid i \in \mathbb{N}\}$ respectively, and their joint probability mass function on $\mathcal{X} \times \mathcal{Y}$ is $f(x, y)$, then the probability mass function of Z is $h(z) = \sum_{x \in \mathcal{X}} f(x, z - x)$[3].

(ii) If additionally X, Y are independent with probability mass functions f, g, respectively, then the probability mass function of Z is

$$h(z) = (f * g)(z) := \sum_{x \in \mathcal{X}} f(x) g(z - x).$$

(iii) If X, Y are continuous, with joint density function $f(x, y)$, then the density function of Z is $h(z) = \int_{-\infty}^{\infty} f(x, z - x)\, dx$.

(iv) If additionally X, Y are independent with density functions f, g, respectively, then the density function of Z is

$$h(z) = (f * g)(z) := \int_{-\infty}^{\infty} f(x) g(z - x)\, dx.$$

Proof. *Discrete case:*

$$\begin{aligned} f_Z(z) &= P(Z = z) = P(X + Y = z) \\ &= \sum_{x \in \mathcal{X}} P(X = x, Y = z - x) = \sum_{x \in \mathcal{X}} f(x, z - x) \end{aligned}$$

where the passage to the second line is justified by decomposing the event $X + Y = z$ into the disjoint events "$X = x$ and $Y = z - x$" for all $x \in \mathcal{X}$.

In the case where X, Y are independent, we have $f(x, z - x) = f(x) g(z - x)$ by definition of independence, because the probability mass functions give the probabilities of particular events (namely those which correspond to 1-element sets).

Continuous case: For $A \subseteq \mathbb{R}$ let $B := \{(x, y) \in \mathbb{R}^2 \mid x + y \in A\}$. Then

$$P(Z \in A) = P(X + Y \in A) = P((X, Y) \in B) = \int_B f(x, y)$$

[3] If $z - x \notin \mathcal{Y}$ set $f(x, z - x) = 0$.

Now we can use the transformation formula for higherdimensional integrals: We use the map $\mathbb{R}^2 \to \mathbb{R}^2$ given by $(x,y) \mapsto (x, x+y)$. Its inverse is $(x,z) \mapsto (x, z-x)$ and the Jacobian of the inverse is

$$J := \begin{pmatrix} 1 & -1 \\ 0 & 1 \end{pmatrix}$$

Clearly $\det J = 1$. Furthermore, the inverse sends the set $\mathbb{R} \times A$ to B.

The transformation formula gives us

$$\begin{aligned} P(Z \in A) &= \int_B f(x,y) = \int_{\mathbb{R}\times A} f(x, z-x) \cdot |\det J| \, \mathrm{d}x \, \mathrm{d}z \\ &= \int_A \int_{-\infty}^{\infty} f(x, z-x) \, \mathrm{d}x \, \mathrm{d}z \end{aligned}$$

This shows that the density function of Z is the one from the claim. In the case where X, Y are independent, we have $f(x, z-x) = f(x)g(z-x)$ by Prop. 4.5.4. □

The function $f * g$ appearing in Prop. 4.6.5 is called the *convolution* of f and g. This operation is important also outside of probability theory. From the fact that it corresponds to addition of random variables, it follows immediately that for density functions f, g, h one has $(f * g) * h = f * (g * h)$ and $f * g = g * f$.

The same is true for arbitrary functions for which the convolution exists, for example because of the Convolution theorem which says that the Fourier transform turns convolution into pointwise multiplication of functions and vice versa.

We will mention convolution again in the context of the Central Limit Theorem. See Remark 4.9.10 for links to some great blog posts, with graphics and animations illustrating convolution.

For an overview and illustrations around transformations of random variables see also the section on transformations of the Random website.

For a more general strategy for computing distributions of transformed random variables (that we will not cover in this course), see Remark 4.5.25.

In view of the previous remark, it can be desirable to compute the moments of a transformed random variable $g(X)$ without knowing its distribution. For the first moment, i.e. the expectation, this is easy (we already used this in Remark 4.5.21):

Proposition 4.6.6. *Let* $\mathbf{X}$ *be a continuous* $\mathbb{R}^n$*-valued random variable with density function* f *and let* $g\colon \mathbb{R}^n \to \mathbb{R}^m$ *be measurable. The expectation of* $\mathbf{Y} := g(\mathbf{X})$ *is*

$$EY = \int_{\mathbb{R}^n} g(x)f(x)dx$$

Proof. This is exactly the definition of expectation of the random variable $g(X)\colon \Omega \xrightarrow{\mathbf{X}} \mathbb{R}^n \xrightarrow{g} \mathbb{R}^m$ □

A case where we can easily compute expectation and covariance of $g(\mathbf{X})$ is that of an affine transformation $g(x) = Ax + \mathbf{b}$:

Proposition 4.6.7. *Let* $\mathbf{X}$ *be an* $\mathbb{R}^n$*-valued random variable,* A *an* $m \times n$*-matrix and* $\mathbf{b}$ *a vector in* $\mathbb{R}^m$*. Then the random variable* $\mathbf{Y} := A\mathbf{X} + \mathbf{b}$ *has expectation* $E\mathbf{Y} = AE\mathbf{X} + \mathbf{b}$ *and covariance matrix* $\operatorname{Cov}(\mathbf{Y}) = A\operatorname{Cov}(\mathbf{X})A^T$*.*

Proof. The first statement follows immediately from the linearity of expectation: $E\mathbf{Y} = E(A\mathbf{X} + \mathbf{b}) = AE\mathbf{X} + E\mathbf{b} = AE\mathbf{X} + \mathbf{b}$.

The second statement also uses the linearity of expectation:

$$\begin{aligned}
\operatorname{Cov}(\mathbf{Y}) &= E[(\mathbf{Y} - E\mathbf{Y})(\mathbf{Y} - E\mathbf{Y})^T] \\
&= E[((A\mathbf{X} + \mathbf{b}) - E(A\mathbf{X} + \mathbf{b}))((A\mathbf{X} + \mathbf{b}) - E(A\mathbf{X} + \mathbf{b}))^T] \\
&= E[(A\mathbf{X} + \mathbf{b} - (AE\mathbf{X} + \mathbf{b}))(A\mathbf{X} + \mathbf{b} - (AE\mathbf{X} + \mathbf{b}))^T] \\
&= E[(A\mathbf{X} - AE\mathbf{X})(A\mathbf{X} - AE\mathbf{X})^T] \\
&= E[A(\mathbf{X} - E\mathbf{X})(\mathbf{X} - E\mathbf{X})^T A^T] \\
&= AE[(\mathbf{X} - E\mathbf{X})(\mathbf{X} - E\mathbf{X})^T]A^T = A\operatorname{Cov}(\mathbf{X})A^T
\end{aligned}$$

Here the first equality is the content of Prop. 4.5.23. □

As a final observation we note that independence can not be destroyed by applying functions to random variables:

Proposition 4.6.8. *Let* X, Y *be independent random variables, and let* g, h *be any functions. Then* $g(X)$ *and* $h(Y)$ *are also independent.*

Proof. Let A, B be subsets of the ranges of g, h, respectively. Then we have

$$\begin{aligned}
P(g(X) \in A, h(Y) \in B) &= P(X \in g^{-1}(A), Y \in h^{-1}(B)) \\
&\overset{(*)}{=} P(X \in g^{-1}(A)) \cdot P(Y \in h^{-1}(B)) = P(g(X) \in A) \cdot P(h(Y) \in B)
\end{aligned}$$

where the equality $(*)$ comes from the independence of X and Y. □

4.7 Conditional expectation

Remember that conditional probability was defined by $P(A \mid B) = \frac{P(A \cap B)}{P(B)}$. For discrete random variables X, Y and a value b in the range of Y with $P(Y = b) \neq 0$, we can define
$P(X \in A \mid Y = b) := \frac{P(X \in A\,, Y=b)}{P(Y=b)}$, thus obtaining a new random variable called $(X \mid Y = b)$ – you can imagine it as a "slice" of the joint distribution of X and Y).

We would like to have something similar for continuous random variables. However, for $\mathbb{R}^n$-valued random variables X, Y, typically the probability $P(Y = b)$ that Y takes some fixed value is zero. Still one often wants to ask about the distribution of X once the value of Y is known. One uses that $P(|Y - b| < \epsilon)$ is often non-zero and defines:

$$P(X \in A \mid Y = b) \;:=\; \lim_{\epsilon \to 0} \frac{P(X \in A,\ |Y - b| < \epsilon)}{P(|Y - b| < \epsilon)} \qquad \text{(if this limit exists)}$$

Example 4.7.1. Consider $\mathbb{R}$-valued random variables X, Y having joint density function

$$f(x,y) = \begin{cases} x + y & \text{if } (x,y) \in [0,1] \times [0,1] \\ 0 & \text{otherwise} \end{cases}$$

Then

$$P(\frac{1}{2} < X < 1 \mid Y = \frac{1}{3}) \;=\; \lim_{\epsilon \to 0} \frac{P(X \in [\frac{1}{2}, 1],\ |Y - \frac{1}{3}| < \epsilon)}{P(|Y - \frac{1}{3}| < \epsilon)} \;=\; \lim_{\epsilon \to 0} \frac{\int_{\frac{1}{3}-\epsilon}^{\frac{1}{3}+\epsilon} \int_{\frac{1}{2}}^{1} x + y \ \ dx\, dy}{\int_{\frac{1}{3}-\epsilon}^{\frac{1}{3}+\epsilon} \int_{0}^{1} x + y \ \ dx\, dy}$$

$$= \lim_{\epsilon \to 0} \frac{\int_{\frac{1}{3}-\epsilon}^{\frac{1}{3}+\epsilon} [\frac{1}{2}x^2 + yx]_{\frac{1}{2}}^{1}\ dy}{\int_{\frac{1}{3}-\epsilon}^{\frac{1}{3}+\epsilon} [\frac{1}{2}x^2 + yx]_{0}^{1}\ dy} \;=\; \lim_{\epsilon \to 0} \frac{\int_{\frac{1}{3}-\epsilon}^{\frac{1}{3}+\epsilon}\ y + \frac{1}{2} - \frac{1}{2}y - \frac{1}{8}\ dy}{\int_{\frac{1}{3}-\epsilon}^{\frac{1}{3}+\epsilon}\ y + \frac{1}{2}\ dy} \;=\; \lim_{\epsilon \to 0} \frac{\int_{\frac{1}{3}-\epsilon}^{\frac{1}{3}+\epsilon}\ \frac{1}{2}y + \frac{3}{8}\ dy}{\int_{\frac{1}{3}-\epsilon}^{\frac{1}{3}+\epsilon}\ y + \frac{1}{2}\ dy}$$

$$= \lim_{\epsilon \to 0} \frac{[\frac{1}{4}y^2 + \frac{3}{8}y]_{\frac{1}{3}-\epsilon}^{\frac{1}{3}+\epsilon}}{[\frac{1}{2}y^2 + \frac{1}{2}y]_{\frac{1}{3}-\epsilon}^{\frac{1}{3}+\epsilon}} \;=\; \lim_{\epsilon \to 0} \frac{\frac{1}{4}(\frac{1}{9} + \frac{2}{3}\epsilon + \epsilon^2) + \frac{3}{8}(\frac{1}{3} + \epsilon) - \frac{1}{4}(\frac{1}{9} - \frac{2}{3}\epsilon + \epsilon^2) - \frac{3}{8}(\frac{1}{3} - \epsilon)}{\frac{1}{2}(\frac{1}{9} + \frac{2}{3}\epsilon + \epsilon^2) + \frac{1}{2}(\frac{1}{3} + \epsilon) - \frac{1}{2}(\frac{1}{9} - \frac{2}{3}\epsilon + \epsilon^2) - \frac{1}{2}(\frac{1}{3} - \epsilon)}$$

$$= \lim_{\epsilon \to 0} \frac{\frac{2}{6}\epsilon + \frac{6}{8}\epsilon}{\frac{2}{3}\epsilon + \epsilon} \;=\; \lim_{\frac{26}{24}\epsilon \to 0} \frac{\epsilon}{\frac{5}{3}\epsilon} \;=\; \frac{13}{20}$$

The map $A \mapsto P(X \in A \mid Y = b)$ is again a probability measure on the range of X. If X, Y are continuous random variables with joint density function $f(x,y)$, then one can see that this probability measure is again continuous. For simplicity we consider $\mathbb{R}$-valued random variables X, Y and one-sided intervals. We write $g(y) := \int_{-\infty}^{\infty} f(x,y)\, dx$ for the marginal density function of Y.

$$\begin{aligned} P(X \le a \mid Y = b) \;&=\; \lim_{\epsilon \to 0} \frac{P(X \le a,\ |Y-b| < \epsilon)}{P(|Y-b| < \epsilon)} \;=\; \lim_{\epsilon \to 0} \frac{\int_{-\infty}^{a} \int_{b}^{b+\epsilon} f(x,y)\; dy\, dx}{\int_{-\infty}^{\infty} \int_{b}^{b+\epsilon} f(x,y)\; dy\, dx} \\ &= \lim_{\epsilon \to 0} \frac{\frac{1}{\epsilon} \int_{-\infty}^{a} \int_{b}^{b+\epsilon} f(x,y)\; dy\, dx}{\frac{1}{\epsilon} \int_{b}^{b+\epsilon} \int_{-\infty}^{\infty} f(x,y)\; dx\, dy} \;=\; \lim_{\epsilon \to 0} \frac{\int_{-\infty}^{a} \frac{1}{\epsilon} \int_{b}^{b+\epsilon} f(x,y)\; dy\, dx}{\frac{1}{\epsilon} \int_{b}^{b+\epsilon}\ g(y)\; dy} \\ &= \frac{\int_{-\infty}^{a} \lim_{\epsilon \to 0} \frac{1}{\epsilon} \int_{b}^{b+\epsilon} f(x,y)\; dy\, dx}{\lim_{\epsilon \to 0} \frac{1}{\epsilon} \int_{b}^{b+\epsilon}\ g(y)\; dy} \;=\; \frac{\int_{-\infty}^{a} f(x,b)\; dx}{g(b)} \;=\; \int_{-\infty}^{a} \frac{f(x,b)}{g(b)} dx \end{aligned}$$

Switching the limit into the first integral is allowed by Lebesgue's dominated convergence theorem. The limits of the inner integrals are, by the fundamental theorem of calculus, the evaluations at the integrands, which explains the second to last step: More precisely we use

$$\begin{aligned} \lim_{\epsilon \to 0} \frac{1}{\epsilon} \int_{b}^{b+\epsilon} f(x)\, dx &= \lim_{\epsilon \to 0} \frac{\int_{-\infty}^{b+\epsilon} f(x)\, dx \ - \ \int_{-\infty}^{b} f(x)\, dx}{\epsilon} \\ &= \lim_{\epsilon \to 0} \frac{F(b+\epsilon) - F(b)}{\epsilon} = F'(b) = f(b) \end{aligned}$$

where the equality before the last holds because $F(x) := \int_{-\infty}^{x} f(t)\, dt$ is an antiderivative of $f(x)$.

Thus the density function of the new probability measure is $f(x \mid b) := \frac{f(x,b)}{g(b)} = \frac{f(x,b)}{\int_{-\infty}^{\infty} f(x,b)\, dx}$. We will also consider its behaviour as b varies, i.e. see it as a function of two variables:

Definition 4.7.2. *The function* $f(x \mid y) := \frac{f(x,y)}{g(y)} = \frac{f(x,y)}{\int_{-\infty}^{\infty} f(x,y)\, dx}$ *is called* the conditional density function*.*

For discrete random variables with joint probability mass function $f(x,y)$, we use the same notation to denote the conditional probability mass function $f(x \mid y) := \frac{f(x,y)}{g(y)} = \frac{f(x,y)}{\Sigma_x f(x,y)}$ *(where the sum in the denominator ranges over all x in the range of X).*

For every fixed b in the range of Y we thus obtain a new random variable $(X|Y=b)$: In the continuous case it is given by the conditional density function, as discussed above. In the discrete case it is given by the probability distribution of the beginning of the section.

So for every b we have a random variable and with it all the usual numbers and operations we have for random variables, for example expectation and covariance matrix. Now if we take the expectation while letting b vary, we obtain a new random variable:

Definition 4.7.3. *The* conditional expectation $E(X \mid Y)$ *is the random variable* $\omega \mapsto E((X \mid Y = Y(\omega))$.

Concretely the value of this random variable for $Y(\omega) = b$ *is*

$$E(X \mid Y = b) = \begin{cases} \Sigma_x \; x \cdot f(x \mid b) & \textit{if } X \textit{ is discrete} \\ \int_{-\infty}^{\infty} x \cdot f(x \mid b)\, dx & \textit{if } X \textit{ is continuous} \end{cases}$$

Example 4.7.4. For the joint distribution of Example 4.7.1 we get

$$E(X \mid Y = y) \;=\; \int_0^1 x \cdot f(x \mid y)\, dx \;=\; \int_0^1 x \cdot \frac{x+y}{\int_0^1 x + y\, dx}\, dx \;==\; \int_0^1 x \cdot \frac{x+y}{\frac{1}{2}+y}\, dx$$

$$= \frac{1}{\frac{1}{2}+y} \int_0^1 x^2 + xy\; dx \;=\; \frac{1}{\frac{1}{2}+y} \left[\frac{1}{3}x^3 + \frac{1}{2}x^2 y \right]_{x=0}^{1} \;=\; \frac{\frac{1}{3}+\frac{1}{2}y}{\frac{1}{2}+y}$$

Remark 4.7.5. Likewise we can define the conditional variance of $\mathbb{R}$-valued random variables as $Var(X \mid Y) := E(X^2 \mid Y) - (E(X \mid Y))^2$.

Proposition 4.7.6. *The expectation of the random variable* $E(X \mid Y)$ *is equal to the expectation of the random variable* X*:*

$$E_y(E_x(X \mid Y)) = E_x(X)$$

(here we put the subscripts to the Es to clarify which variable the expectation is taken over)

Proof. The continuous $\mathbb{R}$-valued case goes as follows:

Denote by $h(x) := \int_{-\infty}^{\infty} f(x,y)\, dy$ the marginal density of X.

$$E_y(E_x(X \mid Y)) \;=\; E_y\left(\int_{-\infty}^{\infty} x f(x \mid y)\, dx\right) \;=\; \int_{-\infty}^{\infty} \left(\int_{-\infty}^{\infty} x f(x \mid y)\, dx\right) g(y)\; dy$$

$$= \int_{-\infty}^{\infty} \left(\int_{-\infty}^{\infty} x\, f(x \mid y) g(y)\, dx\right)\; dy \;=\; \int_{-\infty}^{\infty} \left(\int_{-\infty}^{\infty} f(x \mid y) g(y)\, dy\right) x\; dx$$

$$= \int_{-\infty}^{\infty} \left(\int_{-\infty}^{\infty} \frac{f(x,y)}{g(y)} g(y)\, dy\right) x\; dx \;=\; \int_{-\infty}^{\infty} x\, h(x)\; dx \;=\; E_x X$$

□

Example 4.7.7. We consider an $\mathbb{N}$-valued random variable X that counts the number of visitors of a website during an hour, and another $\mathbb{N}$-valued random variable Y that counts the number of people that click on an ad.

The number of visitors to a website in a given time interval is known to be Poisson distributed: If there are n visitors per hour on average, then the probability of having k visitors in h hours is $P_X(X = k) = e^{-n}\frac{n^k}{k!}$. The expected value of the Poisson distribution is

$$\begin{aligned}\sum_{k=0}^{\infty} k{\cdot}e^{-n}\frac{n^k}{k!} &= e^{-n}\sum_{k=1}^{\infty} k\frac{n^k}{k!} = n \cdot e^{-n}\sum_{k=1}^{\infty}\frac{n^{k-1}}{(k-1)!} \\ &= n \cdot e^{-n}\sum_{k=0}^{\infty}\frac{n^k}{k!} = n \cdot e^{-n}e^{n} = n\end{aligned}$$

as it should be.

Each visitor has a probability of p of clicking on an ad (and consequently a probability of $1 - p$ of ignoring it). Thus, if we have k visitors, the probability that m of them click on the ad is
$P_Y(Y = m) = \binom{k}{m}p^m(1-p)^{k-m}$ (this is the binomial distribution). The expectation of the binomial distribution is $\sum_{m=0}^{k} m \cdot \binom{k}{m}p^m(1-p)^{k-m} = kp$

Thus the expected number of people clicking on the ad per hour is:

$$E_y(Y) = E_x(E_y(Y \mid X)) = E_x(pX) = pE_x(X) = p \cdot n$$

Proposition 4.7.8. *Assuming all the following conditional expectations exist, we have, for $\mathbb{R}$-valued random variables X, Y, Z, real numbers a, b and functions $g\colon \mathbb{R} \to \mathbb{R}$*

(i) $E(a \mid Y) = a$

(ii) $E(aX + bZ \mid Y) = aE(X \mid Y) + bE(Z \mid Y)$

(iii) $E(X \mid Y) = E(X)$ *if X, Y are independent*

(iv) $E(X \cdot g(Y) \mid Y) = g(Y)E(X \mid Y)$

(v) $E(X \mid Y, g(Y)) = E(X \mid Y)$

The following theorem says that $E(X \mid Y)$ is the best possible approximation of X by a function of Y!

Theorem 4.7.9. *Let X, Y be $\mathbb{R}$-valued random variables and $g\colon \mathbb{R} \to \mathbb{R}$ a measurable function. Then $E((X - E(X \mid Y))^2) \leq E((X - g(Y))^2)$, and equality holds if and only if $g(Y) = E(X \mid Y)$ with probability* 1*.*

Proof. We have

$$\begin{aligned}E((X-g(Y))^2) &= E((X-E(X|Y)+E(X|Y)-g(Y))^2)\\ &= E((X-E(X|Y))^2)+E((E(X|Y)-g(Y))^2)+2E((X-E(X|Y))(E(X|Y)-g(Y)))\\ &= E((X-E(X|Y))^2)+E((E(X|Y)-g(Y))^2)+2E((X-E(X|Y))h(Y))\end{aligned}$$

where $h(Y) := E(X \mid Y) - g(Y)$. It suffices to show that the last summand is zero (because squares are positive). This can be seen as follows:

$$\begin{aligned}E(h(Y)E(X \mid Y)) &= \int h(y)E(X \mid Y = y) f_Y(y)\, dy\\ &= \int h(y) \left(\int x \cdot f(x \mid y)\, dx \right) f_Y(y)\, dy\\ &= \int \int x \cdot h(y) \cdot f(x,y)\, dx\, dy = E(h(Y)X)\end{aligned}$$

where we denote the joint density of X and Y by $f_{X,Y}(x,y)$, the marginal density of Y by $f_Y(y)$ and the conditional distribution by $f(x \mid y)$.

Therefore the last summand is:
$2E((X - E(X|Y))h(Y)) = 2E(X \cdot h(Y)) - 2E(E(X|Y)h(Y)) = 0$ □

4.8 Gaussian distributions

Definition 4.8.1 (Gaussian distribution)**.** *A* Gaussian distribution*, or* normal distribution*, is a continuous distribution on $\mathbb{R}^n$ whose density function is of the form*

$$p(\mathbf{x}) = (2\pi)^{-\frac{n}{2}} det(\boldsymbol{\Sigma})^{-\frac{1}{2}} exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right)$$

where $\boldsymbol{\mu}$ is a vector in $\mathbb{R}^n$ (called the mean vector*) and $\boldsymbol{\Sigma}$ is a positive definite symmetric $n \times n$-matrix (called the* covariance matrix*).*

One also writes $\mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}, \boldsymbol{\Sigma})$, or sometimes just $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, for the above $p(\mathbf{x})$, emphasizing its dependence on $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$.

If an $\mathbb{R}^n$-valued random variable X has a Gaussian distribution with mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$, one writes $X \sim \mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}, \boldsymbol{\Sigma})$.

The $\mathbb{R}^n$-valued standard Gaussian distribution *or* standard normal distribution *is the Gaussian distribution with $\boldsymbol{\mu} = \mathbf{0}$ (the zero vector) and $\boldsymbol{\Sigma} = \mathbf{I}$ (the unit matrix).*

As an example we have seen the bivariate standard Gaussian distribution as a model for the darts experiment Ex. 4.2.3 and we have shown in Example 4.4.8 that the integral over the bivariate standard Gaussian is in fact 1. We record this statement as a lemma.

Lemma 4.8.2. *The 2-dimensional (or bivariate) Gaussian with zero vector and diagonal matrix*

$$\mathcal{N}\left(x \,\middle|\, \begin{pmatrix}0\\0\end{pmatrix}, \begin{pmatrix}1 & 0\\0 & 1\end{pmatrix}\right) = \frac{1}{2\pi}\exp\left(-\frac{x^2+y^2}{2}\right)$$

is a density function.

Proof. We have shown this by direct calculation, using polar coordinates, in Example 4.4.8. □

To show that the integral over an arbitrary Gaussian distribution is 1, one has to follow a non-obvious path through several little useful results.

Lemma 4.8.3. *The 1-dimensional (or univariate) Gaussian with first parameter 0 $\mathcal{N}(x|0,\sigma^2) = \frac{1}{\sqrt{2\pi}}\frac{1}{\sigma}\exp\left(-\frac{x^2}{2\sigma^2}\right)$ is a density function.*

Proof. First we treat the case $\sigma = 1$. Consider $z := \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{x^2}{2}\right)$.

$$\begin{aligned}
z^2 &= \left(\int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}} \exp\left(\frac{-x^2}{2}\right) dx\right)^2 \\
&= \frac{1}{2\pi} \left(\int_{-\infty}^{\infty} \exp\left(\frac{-x^2}{2}\right) dx\right) \left(\int_{-\infty}^{\infty} \exp\left(\frac{-y^2}{2}\right) dy\right) \\
&= \frac{1}{2\pi} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \exp\left(\frac{-x^2}{2}\right) \cdot \exp\left(\frac{-y^2}{2}\right) dx\, dy \\
&= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \frac{1}{2\pi} \exp\left(\frac{-(x^2+y^2)}{2}\right) dx\, dy \\
&= \int_{\mathbb{R}^2} \frac{1}{2\pi} \exp\left(\frac{-(x^2+y^2)}{2}\right) \overset{4.8.2}{=} 1
\end{aligned}$$

So $z = \pm 1$. Since the integrand is always positive, we know $z > 0$, and hence $z = 1$.

Now for general σ the function $h(x) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{x^2}{2\sigma^2}\right)$ arises from $\mathcal{N}(x|0,1) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{x^2}{2}\right)$ by substituting x with $\frac{x}{\sigma}$.

From the substitution formula for integrals we get

$$\begin{aligned}
\int_{\mathbb{R}} h(x)\, \mathrm{dx} &= \int_{\mathbb{R}} \mathcal{N}(\frac{x}{\sigma}|0,1)\, \mathrm{dx} \\
&= \int_{\mathbb{R}} \mathcal{N}(x|0,1)\sigma\, \mathrm{dx} = \sigma \cdot \int_{\mathbb{R}} \mathcal{N}(x|0,1)\, \mathrm{dx} = \sigma \cdot 1 = \sigma
\end{aligned}$$

Here the factor σ appearing in the step to the second line is the derivative of the inverse of the function $x \mapsto \frac{x}{\sigma}$ from prescribed by the transformation formula.

It follows that $\int_{\mathbb{R}} \mathcal{N}(x|0,\sigma)\, \mathrm{dx} = \int_{\mathbb{R}} \frac{1}{\sigma} h(x)\, \mathrm{dx} = 1$. □

Lemma 4.8.4. *The function* $\mathcal{N}(\mathbf{0}, \mathbf{D})$ *with a* diagonal matrix $\mathbf{D}$ *is a density function.*

Proof. Let $\mathbf{D}$ have diagonal entries $d_1, \ldots, d_n$. Then for $\mathbf{x} = (x_1, \ldots, x_n) \in \mathbb{R}^n$ we have $(\mathbf{x} - \mathbf{0})^T \mathbf{D}^{-1} (\mathbf{x} - \mathbf{0}) = d_1^{-1} x_1^2 + \ldots + d_n^{-1} x_n^2$.

Suppose by induction on the dimension n that we already know that Gaussians with diagonal matrix in dimensions $n-1$ are density functions (i.e. have integral 1). We did the base case for dimensions 1 in Lemma 4.8.3. We have

$$\begin{aligned}
&\int_{\mathbb{R}^n} \mathcal{N}(x \mid \mathbf{0}, \mathbf{D}) \\
&\quad = \int_{\mathbb{R}^n} (2\pi)^{-\frac{n}{2}} (d_1 \cdot \ldots \cdot d_n)^{-\frac{1}{2}} exp\left(-\frac{1}{2}(d_1^{-1}x_1^2 + \ldots + d_n^{-1}x_n^2)\right) dx_n \ldots dx_1 \\
&\quad \overset{(1)}{=} (2\pi)^{-\frac{n}{2}} (d_1 \cdot \ldots \cdot d_n)^{-\frac{1}{2}} \int_{\mathbb{R}^n} exp\left(-\frac{x_1^2}{2d_1}\right) \cdot exp\left(-\frac{1}{2}(d_2^{-1}x_2^2 + \ldots + d_n^{-1}x_n^2)\right) dx_1 \ldots dx_n \\
&\quad \overset{(2)}{=} \int_{\mathbb{R}} (2\pi d_1)^{-\frac{1}{2}} exp\left(-\frac{x_1^2}{2d_1}\right) dx_1 \\
&\qquad\qquad \cdot \int_{\mathbb{R}^{n-1}} ((2\pi)^{n-2} d_2 \ldots d_n)^{-\frac{1}{2}} exp\left(-\frac{1}{2}(d_2^{-1}x_2^2 + \ldots + d_n^{-1}x_n^2)\right) dx_2 \ldots dx_n \\
&\quad \overset{(3)}{=} \int_{\mathbb{R}} \mathcal{N}(x \mid \mathbf{0}, d_1) dx_1 \cdot \int_{\mathbb{R}^{n-1}} \mathcal{N}(x \mid \mathbf{0}, diag(d_2, \ldots, d_n)) dx_2 \ldots dx_n = 1 \cdot 1 = 1
\end{aligned}$$

where the last step uses the induction hypothesis that the Gaussians with diagonal matrix up to dimensions $n-1$ are density functions. □

Lemma 4.8.5. *For an $\mathbb{R}^n$-valued random variable $\mathbf{X} = (X_1, \ldots, X_n)^T$ that has Gaussian distribution $\mathcal{N}(\mathbf{0}, \mathbf{D})$ with a* diagonal matrix $\mathbf{D}$, *the components X_i are jointly independent and have a Gaussian distribution.*

Proof. Again let $\mathbf{D}$ have diagonal entries $d_1, \ldots, d_n$. Again for $\mathbf{x} = (x_1, \ldots, x_n) \in \mathbb{R}^n$ we have $(\mathbf{x} - \mathbf{0})^T \mathbf{D}^{-1} (\mathbf{x} - \mathbf{0}) = d_1^{-1}x_1^2 + \ldots + d_n^{-1}x_n^2$.

For subsets $A_1, \ldots A_n \subseteq \mathbb{R}$ we have

$$P(X_1 \in A, \ldots, X_n \in A_n) = \int_{A_1 \times \ldots \times A_n} \mathcal{N}(\mathbf{x} \mid \mathbf{0}, \mathbf{D}) dx_1 \ldots dx_n$$

$$= \int_{A_1} \cdots \int_{A_n} (2\pi)^{-\frac{n}{2}} (d_1 \cdot \ldots \cdot d_n)^{-\frac{1}{2}} exp\left(-\frac{1}{2}(d_1^{-1}x_1^2 + \ldots + d_n^{-1}x_n^2)\right) dx_n \ldots dx_1$$

$$\overset{(1)}{=} (2\pi)^{-\frac{n}{2}} (d_1 \cdot \ldots \cdot d_n)^{-\frac{1}{2}} \int_{A_1} \cdots \int_{A_n} exp\left(-\frac{x_1^2}{2d_1}\right) \cdot \ldots \cdot exp\left(-\frac{x_n^2}{2d_n}\right) dx_n \ldots dx_1$$

$$\overset{(2)}{=} (2\pi)^{-\frac{n}{2}} (d_1 \cdot \ldots \cdot d_n)^{-\frac{1}{2}} \int_{A_1} exp\left(-\frac{x_1^2}{2d_1}\right) dx_1 \cdot \ldots \cdot \int_{A_n} exp\left(-\frac{x_n^2}{2d_n}\right) dx_n$$

$$\overset{(3)}{=} \int_{A_1} (2\pi d_1)^{-\frac{1}{2}} exp\left(-\frac{x_1^2}{2d_1}\right) dx_1 \cdot \ldots \cdot \int_{A_n} (2\pi d_n)^{-\frac{1}{2}} exp\left(-\frac{x_n^2}{2d_n}\right) dx_n$$

$$\overset{(4)}{=} \int_{A_1} (2\pi d_1)^{-\frac{1}{2}} exp\left(-\frac{x_1^2}{2d_1}\right) dx_1 \cdot \int_{\mathbb{R}^{n-1}} \mathcal{N}(x_2, \ldots, x_n \mid \mathbf{0}, diag(d_2, \ldots, d_n) dx_2 \ldots dx_n)$$

$$\cdot \ldots \cdot$$

$$\int_{A_n} (2\pi d_n)^{-\frac{1}{2}} exp\left(-\frac{x_n^2}{2d_n}\right) dx_n \cdot \int_{\mathbb{R}^{n-1}} \mathcal{N}(x_1, \ldots, x_{n-1} \mid \mathbf{0}, diag(d_1, \ldots, d_{n-1})) dx_1 \ldots dx_{n-1}$$

$$\overset{(5)}{=} \int_{A_1} \left(\int_{\mathbb{R}^{n-1}} \mathcal{N}(x_1, x_2, \ldots, x_n \mid \mathbf{0}, \mathbf{D}) dx_2 \ldots dx_n\right) dx_1$$

$$\cdot \ldots \cdot$$

$$\int_{A_n} \left(\int_{\mathbb{R}^{n-1}} \mathcal{N}(x_1, x_2, \ldots, x_n \mid \mathbf{0}, \mathbf{D}) dx_1 \ldots dx_{n-1}\right) dx_n$$

$$\overset{(6)}{=} P(X_1 \in A_1) \cdot \ldots \cdot P(X_n \in A_n)$$

Explanations of the single steps:

(1) the exponential law turning addition to multiplication

(2) during integration over one variable, the other variables form constant factors that we can pull out

(3) redistribute the initial factors over the several integrals

(4) the integral over the density function of an $n-1$-dimensional Gaussian over all of $\mathbb{R}^{n-1}$ equals 1 by Lemma 4.8.4.

(5) pull th second integral in each row into the first one and merge the integrands using the exponential law – the outcome is the diagonal Gaussian with matrix **D**

(6) each row in the previous term is an integral over the marginal density function for an X_i.

It is also visible from the computation that X_i has Gaussian distribution with probability function $P(X_i \in A)) = \int_A (2\pi d_i)^{-\frac{1}{2}} exp\left(-\frac{x_i^2}{2d_i}\right) dx_i$: Just look at equations (4) to (6) in the special case $A_1 = \ldots = A_{i-1} = A_{i+1} = \ldots = A_n$. □

Lemma 4.8.6. *The variance of a standard normally distributed, $\mathbb{R}$-valued random variable is* 1.

Proof. The variance is

$$\begin{aligned}
&\int_{-\infty}^{\infty} x^2 \frac{1}{\sqrt{2\pi}} \exp\left(\frac{-x^2}{2}\right) dx \\
\overset{(1)}{=}& 2\int_{0}^{\infty} x^2 \frac{1}{\sqrt{2\pi}} \exp\left(\frac{-x^2}{2}\right) dx \\
=& 2\frac{1}{\sqrt{2\pi}} \int_{0}^{\infty} \underbrace{x}_{g} \cdot \underbrace{x \exp\left(\frac{-x^2}{2}\right)}_{h'} dx \\
\overset{(2)}{=}& \frac{2}{\sqrt{2\pi}} \left(\underbrace{\left[-x \exp\left(\frac{-x^2}{2}\right)\right]_0^{\infty}}_{g \cdot h} - \int_0^{\infty} \underbrace{-\exp\left(\frac{-x^2}{2}\right)}_{g' \cdot h} dx \right) \\
=& \frac{2}{\sqrt{2\pi}} \int_0^{\infty} \exp\left(\frac{-x^2}{2}\right) dx = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} \exp\left(\frac{-x^2}{2}\right) dx \overset{4.8.3}{=} 1
\end{aligned}$$

□

We still have to justify the names *mean vector* and *covariance matrix* for the entities $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ in the definition of a Gaussian random variable. For the mean vector this is left as an exercise: $\boldsymbol{\mu}$ actually is the expectation (or mean) of the random variable.

Proposition 4.8.7. *Let* $\mathbf{X} = (X_1, \ldots, X_n)^T \sim \mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}, \boldsymbol{\Sigma})$ *for a positive definite matrix* $\boldsymbol{\Sigma}$*. Then* $\mathrm{Cov}(\mathbf{X}) = \boldsymbol{\Sigma}$*.*

Proof. To compute the covariance of our Gaussian random variable, we perform an affine transformation: Let H be a solution to $H^T H = \boldsymbol{\Sigma}^{-1}$ with an invertible matrix H – it exists because $\boldsymbol{\Sigma}$ is positive definite and invertible. We will consider the new random vector $\mathbf{Y} := H(\mathbf{X} - \boldsymbol{\mu})$.

Note that for $y = Hx - \boldsymbol{\mu}$ we have

$$y^T y = (x - \boldsymbol{\mu})^T H^T H (x - \boldsymbol{\mu}) = (x - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (x - \boldsymbol{\mu}).$$

Furthermore, because of $H^T H = \boldsymbol{\Sigma}^{-1}$, we have $\det(H) = \sqrt{\det(\boldsymbol{\Sigma})}^{-1}$.

Now by Prop. 4.6.3 (and using that the derivative of the function $Hx - \boldsymbol{\mu}$ is H), the density function of $\mathbf{Y}$ is given by
$p(y) = \frac{1}{\det(H)} p(x) = \frac{1}{\det(H)} \frac{1}{(2\pi)^{n/2} \det(\boldsymbol{\Sigma})^{1/2}} \exp\left(-\frac{1}{2}(x - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (x - \boldsymbol{\mu})\right) = \frac{1}{(2\pi)^{n/2}} \exp\left(-\frac{1}{2} y^T y\right)$

Thus $\mathbf{Y}$ has standard normal distribution, and we know from Lemma 4.8.5 that the individual coordinates Y_i are independent and have univariate Gaussian distributions and from Lemma 4.8.6 that the variance of each coordinate is 1.

Thus $\mathrm{Cov}(Y_i, Y_j) = 0$ for all $i \neq j$ and $\mathrm{Var}(Y_i) = 1$ for all i: This means that the covariance matrix of $\mathbf{Y}$ is the identity matrix $\mathbf{I}$.

From Prop. 4.6.7 we obtain $\mathbf{I} = \mathrm{Cov}(\mathbf{Y}) = \mathrm{Cov}(H(\mathbf{X} - \boldsymbol{\mu})) = H\mathrm{Cov}(\mathbf{X})H^T$, and thus $\mathrm{Cov}(\mathbf{X}) = H^{-1}(H^T)^{-1} = (H^T H)^{-1} = (\boldsymbol{\Sigma}^{-1})^{-1} = \boldsymbol{\Sigma}$. □

Proposition 4.8.8. *Let* $\mathbf{X} \sim \mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}, \boldsymbol{\Sigma})$*. Then* $EX = \boldsymbol{\mu}$*.*

Proof. Exercise. □

Remark 4.8.9. The collection of Gaussian distributions $\mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}, \boldsymbol{\Sigma})$, parametrized by positive definite matrices Σ and vectors $\boldsymbol{\mu}$ of all dimensions, has a number of convenient closure properties:

1. The sum of independent (multivariate) Gaussian random variables is Gaussian. In other words: The convolution of two Gaussian density functions is Gaussian again.

2. The marginals of a multivariate Gaussian random variable are Gaussian again

3. The conditional distribution obtained from one Gaussian distribution conditioned on another Gaussian distribution is Gaussian again.

The proofs, and the explicit calculations of the new mean and covariance matrices resulting from the above operations, can for example be found in Secion 2.3 of Bishop's book "Pattern recognition and Machine Learning or in these short notes.

It follows that affine functions of Gaussian random variables are Gaussian: If $X \sim \mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}, \boldsymbol{\Sigma})$ is Gaussian, then so is $AX + b$.

Since by Prop.4.6.7 $E(AX+b) = AEX + b$ and $\mathrm{Cov}(AX+b) = A\,\mathrm{Cov}(X)A^T$, Prop. 4.8.7 and Prop. 4.8.8 imply the explicit formula $AX + b \sim \mathcal{N}(\mathbf{x} \mid A\boldsymbol{\mu} + b, A\boldsymbol{\Sigma}A^T)$.

Remark 4.8.10. For $\mathbb{R}$-valued random variables with a Gaussian joint distribution, being uncorrelated (i.e. having covariance 0) implies that they are independent: Having covariance 0 means that the covariance matrix of the joint distribution is diagonal, by Prop. 4.8.7. Now independence follows from Lemma 4.8.5.

4.9 Central Limit Theorem

We now turn to the central limit theorem(s), which is one of several reasons why Gaussian random variables are so important. We start with three results that will help understanding the statement of the Central Limit Theorem, and that important in their own right.

Proposition 4.9.1 (Markov's inequality)**.** *Let X be an $\mathbb{R}_{\geq}$-valued random variable. Then $P(X \geq a) \leq \frac{EX}{a}$*

Proof.

$$\begin{aligned} EX &= \underbrace{P(X<a)}_{\geq 0} \cdot \underbrace{E(X|X<a)}_{\geq 0,\text{ since } X \geq 0} + P(X \geq a) \cdot E(X|X \geq a) \\ &\geq P(X \geq a) \cdot E(X|X \geq a) \geq P(X \geq a) \cdot a \end{aligned}$$

□

Proposition 4.9.2 (Chebychev's inequality)**.** *Let X be random variable and $\mu := EX$. Then for all $\epsilon > 0$ we have*

$$P(|X - \mu| \geq \epsilon) \leq \frac{\mathrm{Var}(X)}{\epsilon^2}$$

Proof.

$$P(|X-\mu| \geq \epsilon) = P((X-\mu)^2 \geq \epsilon^2) \overset{4.9.1}{\leq} \frac{E((X-\mu)^2)}{\epsilon^2} = \frac{\text{Var}(X)}{\epsilon^2}$$

□

Definition 4.9.3. *A set* $\{X_1, \ldots, X_n\}$ *of random variables is* independent and identically distributed*, abbreviated* i.i.d.*, if each random variable has the same probability distribution and any random variable is independent of all the others, i.e. for all paiwise different indices* $1 \leq i, j_1, \ldots, j_k \leq n$ *and all sets* $M, M_1, \ldots, M_k$ *one has* $P(X_i \in M \mid X_{j_1} \in M_1, \ldots, X_{j_k} \in M_k) = P(X_i \in M)$.

The requirement of independence of the set of random variables as given above is strictly stronger than just asking for any pair of random variables X_i, X_j to be independent. There do for example exist triples of random variables which are pairwise independent, but not independent. Here is a list of examples.

Theorem 4.9.4 (Weak law of large numbers)**.** *Let* X_i*,* $i \in \mathbb{N}$ *be a sequence of i.i.d. random variables. Let* $\bar{X}_n := \frac{1}{n}(X_1 + \ldots + X_n)$*,* $\mu := EX_i$*,* $\sigma^2 := \text{Var}(X_i)$ *(*μ *and* σ^2 *don't depend on i because of the identical distributions). Then for all* $\epsilon > 0$ *we have*

$$\lim_{n \to \infty} P(|\bar{X}_n - \mu| < \epsilon) = 1$$

Proof. We suppose $\sigma < \infty$ in this proof, but that is actually not necessary.

We have $E\bar{X}_n = E(\frac{1}{n}(X_1 + \ldots + X_n)) = \frac{1}{n}(EX_1 + \ldots + EX_n) = \frac{1}{n} \cdot \cdot \mu = \mu$ and

$$\text{Var}(\bar{X}_n) = \text{Var}(\frac{1}{n}(X_1 + \ldots + X_n)) = \frac{1}{n^2}\text{Var}(X_1 + \ldots + X_n) = \frac{1}{n^2} \cdot n \cdot \sigma^2 = \frac{\sigma^2}{n}.$$

From Chebychev's inequality 4.9.2 we get $P(|\bar{X}_n - \mu| \geq \epsilon) \leq \frac{\sigma^2}{n \cdot \epsilon^2}$. This last expression clearly tends to 0 for $n \to \infty$. Hence $P(|\bar{X}_n - \mu| < \epsilon) = 1 - \frac{\sigma^2}{n \cdot \epsilon^2}$ converges to 1 for $n \to \infty$. □

Definition 4.9.5. *Let* X *and* $X_1, X_2, X_3, \ldots$ *be a sequence of* $\mathbb{R}^n$*-valued random variables. One says that the sequence* $(X_n)_{n \in \mathbb{N}}$ converges in distribution *to* X*, or* converges weakly *to* X*, if for all sets* $A \subseteq \mathbb{R}^n$ *with* $P(X \in \partial A) = 0$ *we have* $\lim_{n \in \mathbb{N}} P(X_n \in A) = P(X \in A)$.

Remark 4.9.6. The sequence $(X_n)_{n\in\mathbb{N}}$ converges in distribution to X, if and only if for all $\mathbb{R}$-valued, continuous, bounded functions $H\colon \mathbb{R}^n \to \mathbb{R}$ we have $\lim_{n\in\mathbb{N}} E(H(X_n)) = E(H(X))$.

One can make sense of the notion of weak convergence for random variables taking values in other measurable spaces than $\mathbb{R}^n$.

The *Central Limit Theorem* is one of the high points of probability theory. Roughly, it says that if you take the average of many i.i.d. random variables, the resulting new random variable will have a distribution close to a Gaussian distribution.

Theorem 4.9.7 (Central Limit Theorem)**.** *Let $\{X_i\}_{i\in\mathbb{N}}$ be a set of i.i.d. random variables. Let $\mu := EX_i$ be the mean and $\sigma^2 := \mathrm{Var}(X_i)$ the variance (μ and σ^2 don't depend on i because of the identical distributions, we suppose they are $< \infty$).*

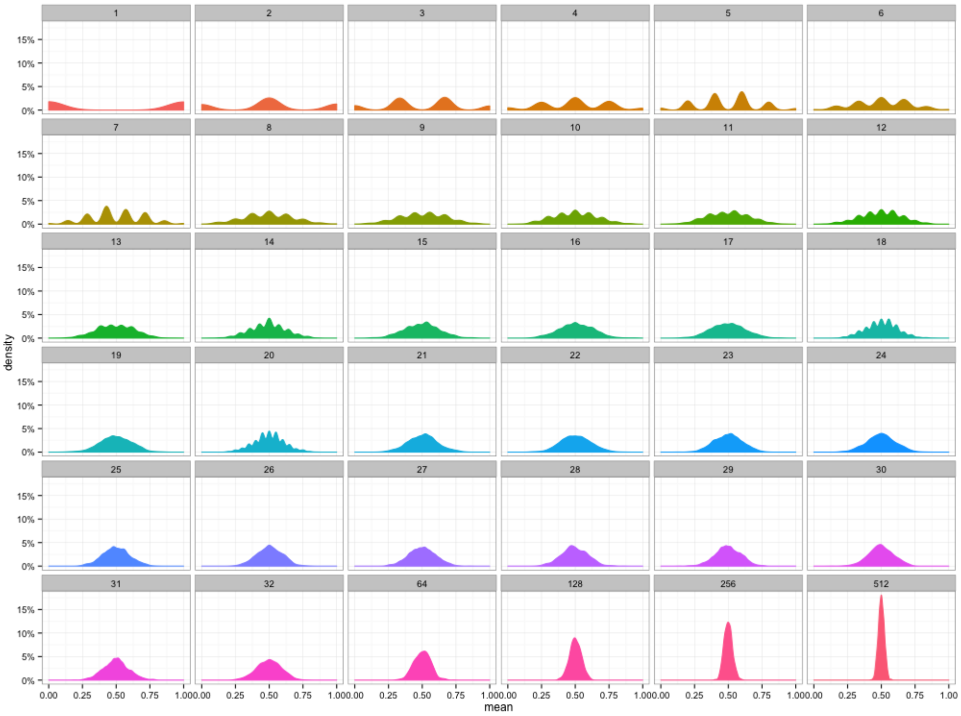
Then the sequence of random variables $S_n := \frac{1}{\sqrt{n}}\Sigma_{i=1}^n (X_i - \mu)$ converges in distribution to a random variable with normal distribution $\sim \mathcal{N}(0, \sigma^2)$.

This form of the central limit theorem may seem strange at first, and not reflect the intuitive phrasing offered before the theorem. In particular the definition of S_n begs an explanation, so here is one:

If we take the usual average $\bar{X}_n = \frac{1}{n}(X_1 + \ldots + X_n)$, then, as we saw in the proof of Theorem 4.9.4, we get

$$\mathrm{Var}(\bar{X}_n) = \mathrm{Var}(\frac{1}{n}(X_1 + \ldots + X_n)) = \frac{1}{n^2}\mathrm{Var}(X_1 + \ldots + X_n) = \frac{1}{n^2}\cdot n \cdot \sigma^2 = \frac{\sigma^2}{n}.$$

So the variance of the sequence $\bar{X}_n$ tends to 0, and it wouldn't be meaningful to say that the limit is a Gaussian distribution. Nevertheless you can see on the way that the shapes become more and more Gaussian. The following picture shows the distributions of the averages $\bar{X}_n$ where each X_i has the density function shown in the upper left picture:



Averages of some i.i.d. random variables. CC licensed by Wikimedia contributor Daniel Resende

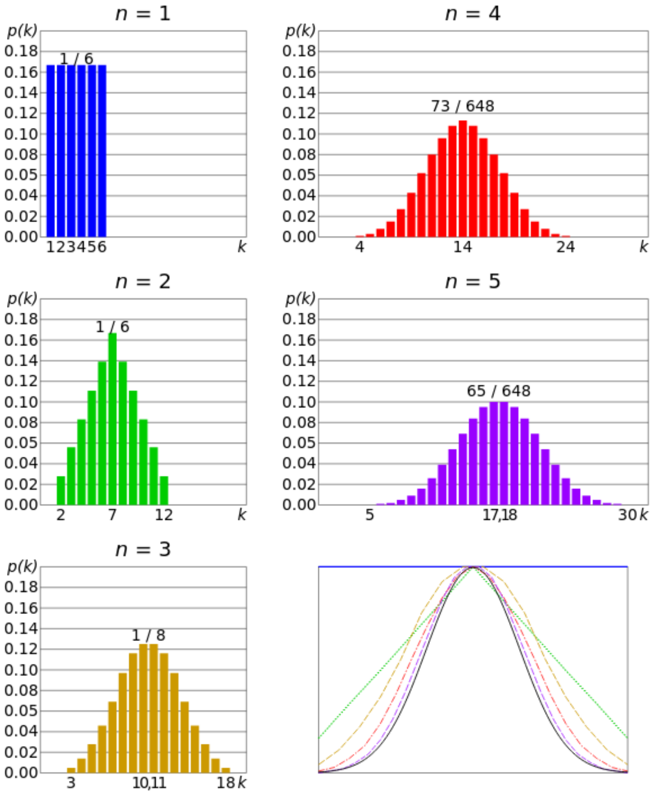
You can see the Gaussian shapes appearing, but also getting thinner and thinner.

If instead we take $\widetilde{X}_n := \frac{1}{\sqrt{n}}(X_1 + \ldots + X_n)$, then we get

$$\mathrm{Var}(\widetilde{X}_n) = \mathrm{Var}(\frac{1}{\sqrt{n}}(X_1+\ldots+X_n)) = \frac{1}{n}\mathrm{Var}(X_1+\ldots+X_n) = \frac{1}{n}\cdot n\cdot\sigma^2 = \sigma^2.$$

So the variance stays constant and we seem to have a chance that the distributions of the $\widetilde{X}_n$ converge to something with that variance. But now the expectation of $\widetilde{X}_n$ is $E\widetilde{X}_n = \frac{1}{\sqrt{n}}(n\mu) = \sqrt{n}\mu$, which goes to ∞ for rising n, so the sequence $(\widetilde{X}_n)_{n\in\mathbb{N}}$ will not converge (in distribution) anywhere.

Here is an illustration of the sum of n dice (i.e. uniformly distributed random variables taking values $1,\ldots,6$). There is no averaging, i.e. multiplication by $\frac{1}{n}$ or $\frac{1}{\sqrt{n}}$, and you see that the expectation (here: the peak of the hill) is moving to higher and higher numbers with rising n. You can see the shape approaching the shape of a Gaussian, but also wandering to the right along the x-axis. The same behaviour would be visible for the $\widetilde{X}_n$, just with a slower movement to the right:



Sum of n dice for several n. CC licensed by Wikimedia contributor Cmglee

To prevent the hill from wandering off to ∞, we exchange the X_i for their centered versions $X_i - \mu$. This gives the definition of S_n from Theorem 4.9.7. Only now the distributions of the S_n have the chance of converging anywhere at all – and they do.

The central limit theorem is a major reason why the normal distributions are so important: In practice it can be taken to say that many independent influences like little measurement errors add up to something Gaussian. Hence the error term in a statistical model is often taken to be Gaussian distributed (with mean zero).

We give another version where the random variables are not assumed to be identically distributed, but still independent. This much greater flexibility also comes at a little cost: In Thm. 4.9.7 we supposed that the variance, which can be given by an integral or an infinite sum, does not diverge, i.e. $E(|X_i - \mu_i|^2) < \infty$. Here we need a slightly stronger assumption. It is technical, but don't get hung up on it. You should just know that identical distributions are not necessary, if your independent distributions are well behaved.

Theorem 4.9.8 (Another Central Limit Theorem)**.** *Let $\{X_i\}_{i\in\mathbb{N}}$ be a set of independent random variables, each with finite expectation μ_i and finite variance σ_i^2. Let $s_n^2 := \sum_{i=1}^n \sigma_i^2$. If for some $\delta > 0$ the condition*

$$\lim_{n\to\infty} \frac{1}{s_n^{2+\delta}} \sum_{i=1}^{n} E(|X_i - \mu_i|^{2+\delta}) = 0$$

is satisfied, then the sequence of random variables $\bar{X}_n := \frac{1}{s_n}\sum_{i=1}^n (X_i - \mu_i)$ converges in distribution to a standard normal random variable.

In practice it is often easiest to check the condition for $\delta = 1$.

Remark 4.9.9. In other versions of the central limit theorem one can relax the assumption of independence, and there is also a version for $\mathbb{R}^n$-valued random variables. You can see a number of these variants on the Wikipedia page on the central limit theorem.

A notable version that is not directly mentioned there is the Berry-Esseen theorem, which also gives information on how fast the density functions in question converge to a Gaussian.

Remark 4.9.10. The central limit theorem (in the given version) talks about averaging i.i.d. random variables. On the level of density functions, averaging over n i.i.d. random variables with density function f corresponds to taking the n-fold convolution $\frac{1}{n} f * f * \ldots * f$. The central limit theorem hinges on the purely analytical fact that convolution powers of functions tend towards Gaussian density functions.

For some thoughts on this, information and experiments how fast one gets close to a Gaussian distribution, and great animations, see this little series of blog posts by Maxwell Peterson:

The central limit theorem in terms of convolutions
Convolution as smoothing
How long does it take to become Gaussian?

Remark 4.9.11. The central limit theorem is usually taken as a justification that "noise" or "error" terms in statistical models are assummed to have a Gaussian distribution. For example, a *linear model* in statistics makes the assumption that for an $\mathbb{R}^n$-valued random variable Y one has $Y = A \cdot X + \epsilon$, where A is an $n \times m$-matrix modelling the linear dependence on another $\mathbb{R}^m$-valued random variable X, and ϵ is a $\mathbb{R}^m$-valued random variable with Gaussian distribution, modelling the noise or measurement errors. We will discuss this in more details in the statistics chapter.

Notation	set-theoretic interpretation	probabilistic interpretation	dice example	dart example
$\omega \in \Omega$	element	elementary event / sample	outcome (2,5)	dart hit in (0.4, 0.2)
$\mathcal{A}$	collection of subsets	collection of events	all subsets	Borel subsets
$A \subseteq \Omega$	subset	event	$\{(2,1),(2,2),(2,3),(2,4),(2,5),(2,6)\}$ = "first die shows a 2"	$\{x \in \mathbb{R}^2 \mid \|x\| \leq \frac{1}{2}\}$ = "dart hits within $\frac{1}{2}$cm from center"
$\bar{A}$	complement of A	event A does not occur	"first die does not show a 2"	"dart hits more than $\frac{1}{2}$cm from center"
$A \cup B$	union of A and B	event A or B or both occur		
$A \cap B$	intersection of A and B	both events A or B occur		
Ω	the whole set	certain event	any result on the dice	dart hits anywhere
$\emptyset$	empty set	impossible event		
$A \cap B = \emptyset$	A and B are disjoint	A and B cannot happen simultaneously	first die shows 2 and 3	dart hits below and above center
$\bigcup_{n=1}^{\infty} A_n$	union of the sets A_n	at least one of the events A_n occurs		
$\bigcap_{n=1}^{\infty} A_n$	intersection of the sets A_n	all of the events A_n occur		
$lim\,sup_n\ A_n$	$\bigcap_{n=1}^{\infty} \bigcup_{k=n}^{\infty} A_n$	infinitely many of the events A_n occur		
$lim\,inf_n\ A_n$	$\bigcup_{n=1}^{\infty} \bigcap_{k=n}^{\infty} A_n$	all but finitely many of the events A_n occur		

5 Statistics

Statistics comes in two flavours, *descriptive statistics* and *inferential statistics*. Descriptive Statistics basically denotes the activity of summarizing given data, for example by calculating the mean and the empirical variance, or drawing a scatter plot. In inferential statistics one assumes that the data arises as values of some random variable and tries to find out things about its distribution. The assumptions on what kind of distribution one deals with are what constitutes a *statistical model*.

There is not that much to say about descriptive statistics, and inferential statistics is the deeper and more useful variant, but sometimes all one needs is a descriptive summary: Think for example of the mean and median given to the students after an exam – one does not really want to model the exam situation as a random experiment, all one needs is some summarizing data that helps students to judge where they are in relation to the others.

For the basic notions of Statistics, with concrete examples, see the introduction to the Statistics part of the Random website

Good statistics books are:

Casella/Berger: Inferential Statistics

Wasserman: All of Statistics

Wasserman's book should be downloadable from the university network (through which you can surf via VPN with your Student ID).

Rao: Advanced Statistical Inference (course notes)

5.1 Descriptive Statistics

Descriptive Statistics denotes the activity of summarizing, and possibly displaying, given data. One does this by computing certain values directly from the data. Typically, data comes as a sequence, also called a *sample*, $(X^{(1)}, \ldots, X^{(n)})$ of results, where each $X^{(i)} = (X_1^{(i)}, \ldots, X_k^{(i)})$ is a tuple of values. The entries of this tuple are called *features* or *attributes*.

Here the individual entries $X_j^{(i)}$ can be any kinds of values: e.g. numerical values, like the petal length in the famous iris data set, or categorical values like the iris species in the iris data set. The possible ranges of these values can be continuous or discrete domains. For the iris data set we would have $n = 150$, corresponding to the 150 specimens of which petal/sepal length and width and species were determined.

Definition 5.1.1. *A* statistic *is just any function of the data.*

We list some common examples:

1. One thing one can always compute is the *frequency*, or *empirical probability*, of each possible value: If the value a of the jth attribute is taken n_a times, then the frequency is $\frac{n_a}{n}$. Together, the frequencies of all possible values of an attribute form a probability distribution, called the *empirical distribution*. For attributes with continuous ranges it can make more sense to count the frequencies of the events that attributes fall into certain intervals, instead of looking at each individual value.

2. For an $\mathbb{R}$-valued attribute taking values $a_1, \dots, a_n$, one can take its *mean* $m_a := \frac{1}{n}\sum_{i=1}^n a_i$. One can easily see that this is the expectation of the empirical distribution on the values $a_1, \dots, a_n$.

3. Still for an $\mathbb{R}$-valued attribute taking values $a_1, \dots, a_n$, one can compute its *empirical variance* $\frac{1}{n-1}\sum_{i=1}^n (a_i - m_a)^2$. It is a measure for how far the values spread out around their mean.

   Similarly, given a further attribute taking values $b_1, \dots, b_n$ and with mean m_b, one can compute their *empirical covariance* $\frac{1}{n-1}\sum_{i=1}^n (a_i - m_a)(b_i - m_b)$. It is a measure for how much the two sets of values depend linearly on each other.

   With this, given an $\mathbb{R}^n$-valued attribute, we can compute its *empirical covariance matrix*. This is exactly what we did for PCA in Section 1.6.

4. A *median* of an $\mathbb{R}$-valued attribute taking values $a_1, \dots, a_n$, is any value such that number such that half of the points are bigger and half of the points are smaller. If n is uneven, one takes the median to be the middle point. The median has the advantage of being extremely robust against outliers.

These statistics can be given meaning in themselves, as indicated, but their meaning will become more transparent if we see them as *estimators*, in the setting of inferential statistics.

As a final warning, let it be said that even descriptive statistics is not as simple as it may sound now. The problem is to decide *which statistics* convey meaningful information about the data. This is not a mathematical problem, but involves careful consideration of the subject matter at hand. A famous simple example is the Berkeley student admission data, which is an instance of Simpson's paradox.

This paradox can be explained by confounding variables, a concept that we will not address, but which will be thoroughly analyzed in the lecture course on Causality of the AI Master program.

5.2 Inferential Statistics and Estimators

The typical setup for Inferential Statistics is this: One assumes that the sample $(X^{(1)}, \ldots, X^{(n)})$ arose by repeating n times a random experiment yielding the tuples $X^{(i)}$. One assumes that the different experiments do not influence each other and each obeys the same probability rules. This is formalized by requiring that $\{X^{(1)}, \ldots, X^{(n)}\}$ is a set of *independent and identically distributed* random variables. We recall Definition 4.9.3:

Definition. The random variables $X_1, \ldots, X_n$ are called *independent and identically distributed*, short *i.i.d.*, if

(a) each random variable has the same probability distribution

(b) for all pairwise different indices $1 \leq i, j_1, \ldots, j_k \leq n$ and all sets $S, S_1, \ldots, S_k$ one has $P(X_i \in S \mid X_{j_1} \in S_1, \ldots, X_{j_k} \in S_k) = P(X_i \in S)$.

The concept of a set of i.i.d. random variables is so important, it even has a Wikipedia page.

The random variables $X^{(i)}$ are also called the *explanatory variables* or *observables*.

In practice one tries to determine relevant values associated to the distribution, using the sample, which consist of concrete outcomes of a random experiment. For example one could try to estimate the expectation of the distribution, and one would typically compute the mean of the values obtained in a random experiment to obtain such an estimate.

To formally construct and evaluate such methods of estimation, one comes up with a recipe – like computing the mean – and applies that not to actual data but to the random variables $(X^{(1)}, \ldots, X^{(n)})$. The result is another random variable, and we can judge its properties as an *estimator* of the desired value.

In our example one would compute the mean of the random variables: $M_n := \frac{1}{n}(X_1 + \ldots + X_n)$

This is new random variable now describes our calculation recipe prior to obtaining any actual values in random experiments and that allows an analysis of how good such a recipe works. One calls such a random variable an *estimator* of whatever value one hopes to determine with it. For example, M_n is an estimator of the expectation of X_i (remember, all the X_j have the same expectation).

Estimators are not a new mathematical concept, and there is no definition of just a plain notion of estimator in itself – an estimator is simply a random variable. Instead it makes sense to say that we want to regard the random variable Y as an estimator *of* something.

To make it clear what is intended to be estimated by an estimator one uses the following notation: If θ the value one wishes to estimate, then an estimator for θ is denoted by $\hat{\theta}$.

Thus the empirical mean estimator of above could also be denoted by $\widehat{EX}$.

Definition 5.2.1. *Let $\hat{\theta}$ be an estimator for the vector $\theta \in \mathbb{R}^n$ (this means nothing more than that we are given a random variable $\hat{\theta}$ and a vector θ which we use in the following definitions).*

(i) For a given sample $\omega \in \Omega$ the error *of $\hat{\theta}$ is $e(\omega) := \hat{\theta}(\omega) - \theta$.*

(ii) The mean squared error *of $\hat{\theta}$ is $MSE(\hat{\theta}) := E(\|\hat{\theta} - \theta\|^2)$.*

(iii) The bias *of $\hat{\theta}$ is $B(\hat{\theta}) := E(\hat{\theta}) - \theta$.*

(iv) The estimator $\hat{\theta}$ is unbiased *if $B(\hat{\theta}) = 0$.*

An estimator for a number θ associated to some random variables $X_1, \ldots, X_n$ is linear*, if it is an affine function of these random variables , i.e. if $\hat{\theta} = b_0 + b_1 X_1 + \ldots + b_n X_n$ for some $b_1, \ldots, b_n \in \mathbb{R}$.*

Note that the mean squared error and the bias are notions associated to a pair consisting of an $\mathbb{R}^n$-valued random variable and a vector in $\mathbb{R}^n$. The question of whether an estimator is linear is only well-defined in the presence of some random variables $X_1, \dots, X_n$ on which the estimator is supposed to depend.

Example 5.2.2. The mean $M_n =$ as an estimator of the expectation is unbiased: We have $EM_n = E(\frac{1}{n}(X_1 + \ldots + X_n)) = \frac{1}{n}(EX_1 + \ldots + EX_n) = \frac{1}{n}(EX_1 + \ldots + EX_1) = EX_1$. Here we used that all the X_i are identically distributed, and hence $EX_i = EX_1$ for all i.

More generally, for any $\lambda_1, \dots, \lambda_n \in \mathbb{R}$ with $\sum_{i=1}^n \lambda_i = 1$ the random variable $M_{\lambda_1,\dots,\lambda_n} := \sum_{i=1}^n \lambda_i X_i$ is an unbiased estimator of the expectation: $EM_{\lambda_1,\dots,\lambda_n} := \sum_{i=1}^n \lambda_i EX_i = \sum_{i=1}^n \lambda_i EX_1 = EX_1$.

Example 5.2.3. To estimate the variance of the X_i a natural idea is to take $\frac{1}{n}\sum_{i=1}^n \left(X_i - \frac{1}{n}\sum_{j=1}^n X_j\right)^2$. This estimator is biased:

$$
\begin{aligned}
\mathrm{E}\left[\frac{1}{n}\sum_{i=1}^n\left(X_i - \frac{1}{n}\sum_{j=1}^n X_j\right)^2\right] &= \frac{1}{n}\sum_{i=1}^n \mathrm{E}\left[X_i^2 - \frac{2}{n}X_i\sum_{j=1}^n X_j + \frac{1}{n^2}\sum_{j=1}^n X_j \sum_{k=1}^n X_k\right] \\
= \frac{1}{n}\sum_{i=1}^n \Bigg[\frac{n-2}{n}\mathrm{E}\left[X_i^2\right] - \frac{2}{n}\sum_{j\neq i}\mathrm{E}\left[X_iX_j\right] &+ \frac{1}{n^2}\sum_{j=1}^n\sum_{k\neq j}\mathrm{E}\left[X_jX_k\right] + \frac{1}{n^2}\sum_{j=1}^n \mathrm{E}\left[X_j^2\right]\Bigg] \\
= \frac{1}{n}\sum_{i=1}^n \Bigg[\frac{n-2}{n}\left(\mathrm{Var}(X_i) + EX_i^2\right) &- \frac{2}{n}(n-1)EX_i^2 + \frac{1}{n^2}n(n-1)EX_i^2 \\
&+ \frac{1}{n}\left(\mathrm{Var}(X_i) + EX_i^2\right)\Bigg] \\
= \frac{n-1}{n}\mathrm{Var}(X_i).
\end{aligned}
$$

Here in the step to the third row we used that for *independent* random variables X, Y we have $E(XY) = (EX)(EY)$ (verify this as an exercise!), and that any X_i, X_j are pairwise independent and identically distributed, i.e. we can replace any EX_j, resp. $E(X_j^2)$, by EX_i, resp. $E(X_i^2)$.

To obtain an unbiased estimator one can instead use $\frac{1}{n-1}\sum_{i=1}^n \left(X_i - \overline{X}\right)^2$ This estimator differs from the previous one by a factor of $\frac{n}{n-1}$ and therefore removes the biasing factor.

In the previous example we could see that for growing n the expectation of the estimator $\frac{1}{n}\sum_{i=1}^{n}\left(X_i - \frac{1}{n}\sum_{j=1}^{n} X_j\right)^2$ tends towards $\mathrm{Var}(X_i)$, so for large sample sizes the bias is negligible. This shows that one does not have to discard biased estimators necessarily.

Remark 5.2.4. Estimators do not need to be given in terms of closed formulas of the random variables $X^{(i)}$. They can just as well be defined through some algorithm involving the $X^{(i)}$ – an example is given in Exercise 34.

One would like estimators to have low bias, low mean squared error and low variance. All of these three things have different meaning, but are related as follows:

Proposition 5.2.5. *Let $\hat{\theta}$ be an estimator for θ. Then*

$$MSE(\hat{\theta}) = \mathrm{Var}(\hat{\theta}) + (B(\hat{\theta}))^2$$

Proof. Exercise. □

Intuitively there are two sources of contribution to the mean squared error, i.e. the total failure of $\hat{\theta}$ just being equal to θ (seen as a constant random variable): One is the variance, i.e. the failure of $\hat{\theta}$ just being constant, and one is the bias, arising if $\hat{\theta}$ varies not around θ but around some other value.

Definition 5.2.6. *An estimator $\hat{\theta}$ for θ is called* best unbiased estimator *if among all unbiased estimators it is the one with the smallest variance.*

An estimator $\hat{\theta}$ for θ is called best linear unbiased estimator *(abbreviated BLUE) if among all unbiased estimators that are linear in the $X^{(i)}$ it is the one with the smallest variance.*

Example 5.2.7. A linear estimator for the expectation is necessarily of the form $\sum_{i=1}^{n}\lambda_i X_i$. It is unbiased if and only if $\sum_{i=1}^{n}\lambda_i = 1$, see Example 5.2.2.

We now have

$$\begin{aligned}
\operatorname{Var}(\sum_{i=1}^{n}\lambda_i X_i) &= E((\sum_{i=1}^{n}\lambda_i X_i)^2) - (E(\sum_{i=1}^{n}\lambda_i X_i))^2 \\
&= E(\sum_{i=1}^{n}\lambda_i^2 X_i^2 + \sum_{i=1}^{n}\sum_{j\neq i}\lambda_i\lambda_j X_i X_j) - \sum_{i=1}^{n}\sum_{j=1}^{n}\lambda_i\lambda_j EX_i EX_j \\
&= \sum_{i=1}^{n}\lambda_i^2 E(X_i^2) + \sum_{i=1}^{n}\sum_{j\neq i}\lambda_i\lambda_j EX_i EX_j \\
&\qquad - \sum_{i=1}^{n}\lambda_i^2 E(X_i)^2 - \sum_{i=1}^{n}\sum_{j\neq i}\lambda_i\lambda_j EX_i EX_j \\
&= \sum_{i=1}^{n}\lambda_i^2 E(X_i^2) - \sum_{i=1}^{n}\lambda_i^2 E(X_i)^2 = \sum_{i=1}^{n}\lambda_i^2 \operatorname{Var}(X_i) \\
&= \operatorname{Var}(X_i)(\sum_{i=1}^{n}\lambda_i^2)
\end{aligned}$$

Again we used that for independent random variables $E(XY) = (EX)(EY)$, and in the last step we used that all X_i have the same variance and therefore the factor that we pulled out is independent of i. Now using the method of Lagrange multipliers, one can find the $\lambda_1, \ldots, \lambda_n$ that minimize this expression subject to the constraint $\sum_{i=1}^{n}\lambda_i = 1$. The solution is $\lambda_1 = \ldots = \lambda_n = \frac{1}{n}$. Hence the usual mean is the best linear unbiased estimator of the expectation.

Remark 5.2.8. In Machine Learning there is another relation between bias and variance: When one trains a machine learning model to reproduce given data, one could see this as defining an estimator.

The bias measures how well the given function of the observables can approximate the true function at all. Underfitting, i.e. bad performance on the training data means that one has a high bias.

The variance can be understood as measuring how different the model could be if one had different training data. Overfitting, i.e. bad generalization to new data, often goes hand in hand with high variance (think of the very wiggly curves that we obtained for polynomials of high degrees).

These two complementary tendencies are known as the *bias-variance tradeoff*. Starting with a low model complexity (e.g. low degree polynomials) one first has a high bias and as the model complexity increases, the bias decreases and the variance can rise. For a long time the consensus was that one should try to hit a sweet spot in between. Recent observations, however, indicate that one should re-evaluate this: Here is a blog post summarizing this.

For the next definition we invoke the sample size n. Estimators really come in sequences; one typically gives an estimator $\hat{\theta}_n$ for every possible sample size n. One also calls the whole sequence simply an estimator (instead of "sequence of estimators").

Definition 5.2.9. *An estimator $\hat{\theta}_n$ (where $\hat{\theta}_n$ is an estimator for sample size n) is* consistent *if it converges in probability to the constant random variable with value θ, i.e. if $\lim_{n\to\infty} P(|\hat{\theta}_n - \theta| > \epsilon) = 0$.*

We can establish a sufficient criterion for consistency using **Chebyshev's Inequality**, Prop. 4.9.2**:** *For any $\mathbb{R}$-valued random variable X we have $P(|X - EX| \geq \epsilon) \leq \frac{\mathrm{Var}(X)}{\epsilon^2}$.*

Corollary 5.2.10. *An unbiased estimator $\hat{\theta}_n$ satisfying $\lim_{n\to\infty} \mathrm{Var}(\hat{\theta}_n) = 0$ is consistent.*

Example 5.2.11 (Law of large numbers)**.** By Example 5.2.2 the average $M_n := \frac{1}{n}\sum_{i=1}^{n} X_i$ is an unbiased estimator of the expectation. By Example 5.2.7 the variance of M_n is $\frac{1}{n}\mathrm{Var}(X_i)$, which clearly converges to 0 for growing n. Hence the mean is a consistent estimator

The fact that the mean is consistent is also called the **Weak law of large numbers**, see Theorem 4.9.4.

Parametric versus nonparametric Statistics

The examples of estimators in the previous section could be defined directly in terms of the sample, without any assumptions on the probability distribution that gave rise to the sample. They were examples of estimators from *nonparametric statistics* (see the Wikipedia page on Nonparametric statistics for some more refined explanation). One usually gets further if one can assume that the probability distribution in question comes from some parametrized family of probability distributions. One can then use, and try to to estimate, those parameters. This is called *parametric statistics*.

A *statistical model* is, as a first approximation, simply such a parametrized family of probability distributions, which one assumes contains the distributions of the random variables $X^{(i)}$. An example is the family of Gaussian distributions $\mathcal{N}(x \mid \boldsymbol{\mu}, \boldsymbol{\Sigma})$, parametrized by pairs $(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, where $\boldsymbol{\mu}$ is a vector and $\boldsymbol{\Sigma}$ is a positive definite matrix. When we come to regression, we will adopt a slightly more refined notion of statistical model.

Under the assumption that the unknown distribution comes from this family, one would for example like to determine the actual values of the parameters. In the Gaussian example, one could take the mean and covariance, since these are the only parameters needed to determine a Gaussian distribution. In a Bernoulli experiment, modelling just a single experiment with probability p of success and $1 - p$ of failure, one would try to determine p. A standard method is to look for the p under which the oberved data would have the highest probability One can do this by seeing the the probability as a two parameter function, i.e. also as a function of p, and deriving by p while inserting the observed values into the other variable – this is called the maximum likelihood method and is covered in the next section.

The maximum likelihood method illustrates how a parameter can be used (it gives us something to derive by), but of course it relies on the assumption that the data was generated by a distribution from a given family. In case one cannnot justify such an assumption, one can still do a lot, however. We will not cover any non-parametric statistics in this course. For an overview of methods see the book Wasserman: All of non-parametric statistics. For an approach to non-parametric statistics that is close to what we cover here, see the Empirical Likelihood method.

5.3 Maximum Likelihood estimators

Maximum likelihood estimation is the following procedure: For a random variable X whose probability distribution we want to find

1. Choose a parametrized family of probability distributions to which you suppose X belongs. Call the parameter θ and the range of possible values Θ.

2. Take n samples of your random variable

3. Choose the $\theta \in \Theta$ which makes your observed samples most likely.

Technically, as usual, we encode the taking of n samples as having n i.i.d. random variables $X_1, \ldots, X_n$, all having the distribution of X. Thus if X has density function (or probability mass function) $f(x;\theta)$ (depending on some parameter θ), then the joint distribution has density function $\Pi_{i=0}^{n} f(x_i;\theta)$.

Definition 5.3.1. *(i) The* likelihood function *associated to values $x_1,\dots,x_n$ of the random variables $X_1,\dots,X_n$ is the function*

$$l_{X,x_1,\dots,x_n}\colon \Theta \to \mathbb{R}_+,\ \ \theta \mapsto \Pi_{i=0}^{n} f(x_i;\theta)$$

(ii) The log likelihood function *is the function*

$$\ell_{X,x_1,\dots,x_n}\colon \Theta \to \mathbb{R}_+,\ \ \theta \mapsto \ln(\Pi_{i=0}^{n} f(x_i;\theta)) = \Sigma_{i=0}^{n} \ln(f(x_i;\theta))$$

(iii) A maximum likelihood estimator, *or* MLE, *of θ is a random variable of the form*

$$\hat{\theta}_{ML}\colon\ \omega \mapsto \operatorname{argmax}_\theta \ell_{X,X_1(\omega),\dots,X_n(\omega)}(\theta)$$

sending an event ω to a θ that maximizes the likelihood function.

(iv) More generally, a maximum likelihood estimator *of a function $g(\theta)$ of θ is a random variable of the form*

$$\hat{\theta}_{ML}\colon\ \omega \mapsto g(\operatorname{argmax}_\theta \ell_{X,X_1(\omega),\dots,X_n(\omega)}(\theta)).$$

Important point: Maximum likelihood estimators depend on the choice of a model! We can not just compute a maximum likelihood estimator for, for example, the expectation of the distribution of some i.i.d. random variables – we need to choose a family of distributions depending on some parameters, then maximum likelihood estimate those parameters, and then compute the mean using the estimated parameters.

Example 5.3.2. The maximum likelihood estimator for the mean value of an $\mathbb{R}$-valued random variable with normal distribution is calculated as follows: The likelihood function is

$$\mu \mapsto \left(\frac{1}{\sqrt{2\pi}\sigma}\right)^n exp\left(-\frac{1}{2\sigma^2}\Sigma_{i=0}^{n}(x_i-\mu)^2\right)$$

Since the logarithm is a monotonically increasing function, finding the μ that maximizes the likelihood function is equivalent to finding the μ that maximizes the log-likelihood function. The latter is

$$\ell\colon \mu \mapsto -n\ln(\sqrt{2\pi}\sigma) - \frac{1}{2\sigma^2}\Sigma_{i=0}^{n}(x_i-\mu)^2$$

The maximum is attained where the derivative with respect to μ is 0:

$$0 \overset{!}{=} \frac{d}{d\mu}\ell = \frac{2}{\sigma^2}\Sigma_{i=0}^{n}(x_i - \mu) = \frac{2}{\sigma^2}((\Sigma_{i=0}^{n}x_i) - n\mu)$$

Clearly this is the case for $\mu = \frac{1}{n}\Sigma_{i=0}^{n}x_i$. This gives us back the estimator we already knew.

Example 5.3.3. The family of Poisson distributions is a family of discrete distributions of $\mathbb{N}$-valued random variables X defined by

$$P(X = k) = \frac{\lambda^k e^{-\lambda}}{k!}$$

where $\lambda > 0$ can be any real number.

The Poisson distribution is used model events per fixed time or space interval, for example

- the number of flaws on a length of cable
- the number of people per hour clicking on a certain link on a homepage
- the number of customers per hour entering a shop

Suppose we are in the third situation and measure clicks per hour on a link on our homepage. After having measured this for n days, giving click numbers $x_1, \dots, x_n$, we decide to estimate the parameter λ of our Poisson distribution.

The likelihood function is

$$L(\lambda, x_1, \dots, x_n) = \Pi_{i=1}^{n}\frac{\lambda^{x_i}e^{-\lambda}}{x_i!}$$

The log-likelihood function is therefore

$$\begin{aligned}
\ell_{x_1,\dots,x_n}(\lambda) &= \ln\left(\Pi_{i=1}^{n}\frac{\lambda^{x_i}e^{-\lambda}}{x_i!}\right) \\
&= \sum_{i=1}^{n}\left[\ln\left(\lambda^{x_i}\right) + \ln\left(e^{-\lambda}\right) - \ln\left(x_i!\right)\right] \\
&= \sum_{i=1}^{n}\left[x_i\ln(\lambda) - \lambda - \ln\left(x_i!\right)\right] \\
&= -n\lambda + \ln(\lambda)\sum_{i=1}^{n}x_i - \sum_{i=1}^{n}\ln\left(x_i!\right)
\end{aligned}$$

To find the maximum we determine the critical points (turns out there is only one):

$$\frac{d}{d\lambda}\ell_{x_1,\dots,x_n}(\lambda) = -n + \frac{1}{\lambda}\sum_{i=1}^{n} x_i \overset{!}{=} 0$$

This implies: $\lambda = \frac{1}{n}\sum_{i=1}^{n} x_i$. Thus the maximum likelihood estimator is the sample mean. This makes some sense as the parameter λ for the Poisson distribution is both the mean and the variance.

Example 5.3.4. For a normally distributed $\mathbb{R}^n$-valued random variable, we compute the maximum likelihood estimator for the covariance matrix. The density function is

$$\mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}, \boldsymbol{\Sigma}) = (2\pi)^{-\frac{n}{2}}\det(\boldsymbol{\Sigma})^{-\frac{1}{2}}\exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right)$$

The density function for a sample of size N is thus a product of N such factors. The log-likelihood function is

$$\begin{aligned}\ell(\mathbf{x}_1,\dots,\mathbf{x}_N \mid \boldsymbol{\mu}, \boldsymbol{\Sigma}) &= \sum_{i=1}^{N}\left(-\frac{n}{2}\ln(2\pi) - \frac{1}{2}\ln\det(\boldsymbol{\Sigma}) - \frac{1}{2}(\mathbf{x}_i-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\mathbf{x}_i-\boldsymbol{\mu})\right)\\ &= -\frac{N}{2}\ln\det(\boldsymbol{\Sigma}) - \frac{1}{2}\sum_{i=1}^{N}(\mathbf{x}_i-\boldsymbol{\mu})^T\,\boldsymbol{\Sigma}^{-1}\,(\mathbf{x}_i-\boldsymbol{\mu}) + \text{ const}\\ &= -\frac{N}{2}\ln\det(\boldsymbol{\Sigma}) - \frac{1}{2}\sum_{i=1}^{N}\text{ trace}\left(\boldsymbol{\Sigma}^{-1}\,(\mathbf{x}_i-\boldsymbol{\mu})\,(\mathbf{x}_i-\boldsymbol{\mu})^T\right) + \text{ const}\\ &= -\frac{N}{2}\ln\det(\boldsymbol{\Sigma}) - \frac{1}{2}\text{ trace}\left(\boldsymbol{\Sigma}^{-1}\sum_{i=1}^{N}\left[(\mathbf{x}_i-\boldsymbol{\mu})\,(\mathbf{x}_i-\boldsymbol{\mu})^T\right]\right) + \text{ const}\end{aligned}$$

We calculate the derivative with respect to $\boldsymbol{\Sigma}$ using the following rules from matrix differential calculus:

$$\frac{\partial}{\partial A}\ln\det(A) = A^{-1} \qquad \text{and} \qquad \frac{\partial}{\partial A}\text{ trace }(A^{-1}B) = -(A^{-1})BA^{-1}$$

We obtain

$$\frac{\partial}{\partial\boldsymbol{\Sigma}}\ell(\mathbf{x}_1,\dots,\mathbf{x}_N \mid \boldsymbol{\mu}, \boldsymbol{\Sigma}) = -\frac{N}{2}\boldsymbol{\Sigma}^{-1} - \frac{1}{2}\sum_{i=1}^{N}\left(-\boldsymbol{\Sigma}^{-1}\,(\mathbf{x}_i-\boldsymbol{\mu})\,(\mathbf{x}_i-\boldsymbol{\mu})^T\,\boldsymbol{\Sigma}^{-1}\right)$$

Looking for critical points we get

$$0 \overset{!}{=} -\frac{N}{2}\boldsymbol{\Sigma}^{-1} - \frac{1}{2}\sum_{i=1}^{N}\left(-\boldsymbol{\Sigma}^{-1}\left(\mathbf{x}_i - \boldsymbol{\mu}\right)\left(\mathbf{x}_i - \boldsymbol{\mu}\right)^T \boldsymbol{\Sigma}^{-1}\right)$$

Multiplying the equation from the right with $2\boldsymbol{\Sigma}$ and bringing the first summand to the other side we get

$$N\mathbf{I} = -\sum_{i=1}^{N}\left(-\boldsymbol{\Sigma}^{-1}\left(\mathbf{x}_i - \boldsymbol{\mu}\right)\left(\mathbf{x}_i - \boldsymbol{\mu}\right)^T\right) = \boldsymbol{\Sigma}^{-1}\left(\sum_{i=1}^{N}\left(\mathbf{x}_i - \boldsymbol{\mu}\right)\left(\mathbf{x}_i - \boldsymbol{\mu}\right)^T\right)$$

and finally, solving for $\boldsymbol{\Sigma}$, we obtain the Maximum Likelihood estimator

$$\widehat{\boldsymbol{\Sigma}} = \frac{1}{N}\left(\sum_{i=1}^{N}\left(\mathbf{x}_i - \boldsymbol{\mu}\right)\left(\mathbf{x}_i - \boldsymbol{\mu}\right)^T\right)$$

This estimator, being a (matrix-valued) random variable in itself, also has a distribution. It is called the Wishart distribution.

Maximum likelihood estimators are often biased. Here is an example:

Example 5.3.5. The uniform distribution on the interval $[0, \theta]$ has density function

$$f(x, \theta) = \begin{cases} \frac{1}{\theta} & \text{if } 0 \leq x \leq \theta \\ 0 & \text{otherwise} \end{cases}$$

The expectation of a uniformly distributed random variable on $[0, \theta]$ is $\frac{\theta}{2}$. A natural estimator for θ is therefore $2 \cdot \frac{1}{n}\Sigma_{i=1}^{n} X_i$. Clearly it is unbiased.

We now consider the maximum likelihood estimator for θ. The joint distribution of n independent random variables $X_1, \ldots, X_n$ distributed uniformly on $[0, \theta]$ has density function

$$\begin{aligned} l(x_1, \ldots, x_n, \theta) = \Pi_{i=1}^{n} f(x, \theta) &= \begin{cases} \frac{1}{\theta^n} & \text{if } 0 \leq x_i \leq \theta \ \forall i \\ 0 & \text{otherwise} \end{cases} \\ &= \begin{cases} \frac{1}{\theta^n} & \text{if } 0 \leq max\{x_1, \ldots, x_n\} \leq \theta \\ 0 & \text{otherwise} \end{cases} \end{aligned}$$

The θ for which this function assumes its maximum is the smallest one for which it is non-zero, i.e. $max\{x_1, \ldots, x_n\}$. So $\hat{\theta}_{ML} = \max\{X_1, \ldots, X_n\}$.

Now for simplicity we will take $n = 2$ and observe

$$\begin{aligned}
E(\max\{X_1, X_2\}) &= \int_0^\theta \int_0^\theta \max\{x_1, x_2\} \frac{1}{\theta^2} \\
&= \frac{1}{\theta^2} \left(\int_0^\theta (\int_0^x x \, dy) \, dx + \int_0^\theta (\int_0^y y \, dx) \, dy \right) \\
&= \frac{1}{\theta^2} \left(\int_0^\theta x^2 \, dx + \frac{1}{\theta^2} \int_0^\theta y^2 \, dy \right) \\
&= \frac{1}{\theta^2} \left(\left[\frac{1}{3} x^3 \right]_0^\theta + \left[\frac{1}{3} y^3 \right]_0^\theta \right) = \frac{1}{\theta^2} \frac{2}{3} \theta^3 = \frac{2}{3} \theta
\end{aligned}$$

Here for the passage to the second line we divided the square into the triangle where $\max\{x_1, x_2\} = x_1$ and the triangle where $\max\{x_1, x_2\} = x_2$.

For example for $\theta = 1$ the expectation of a uniformly distributed random variable on $[0, 1]$ is $\frac{1}{2}$, but the maximum likelihood estimate would tend to give us $\frac{2}{3}$, which is far off.

For general n we get $\frac{n-1}{n}\theta$, so that for big n the bias becomes bigger.

The following theorem asserts that under reasonable conditions any maximum likelihood estimator is consistent. Rather than reading the precise statement, you should take a look at the proof idea.

Theorem 5.3.6. *Let $\Theta \subseteq \mathbb{R}^n$ be a compact set, $f(x, \theta)$, $\theta \in \Theta$ a family of density functions such that for $\theta \neq \theta'$ the distributions defined by $f(x, \theta) \neq f(x, \theta')$ are different. Suppose that the function $f(x, \theta)$ is continuous in θ for almost all x. Let $\theta_0 \in \Theta$. Suppose finally that there exists a $K(x)$ such that $E_{f(x,\theta_0)} K(x) < \infty$ and $\ln f(x, \theta) - \ln f(x, \theta_0) \leq K(x)$ for all x, θ.*

Then any maximum likelihood estimator for the distribution given by $f(x, \theta_0)$ is consistent.

About the proof: The maximum likelihood estimate is the maximizer of the modified log-likelihood function $L_n(x_1, \ldots, x_n)(\theta) := \frac{1}{n} \sum_{i=1}^n logf(x_i, \theta)$ (modified by the extra factor $\frac{1}{n}$ which of course does not change where the maximum occurs).

Define

$$L(\theta) := E_{f(x,\theta_0)} logf(x, \theta) = \int logf(x, \theta) \cdot f(x, \theta_0)$$

Note that $L(\theta)$ does not depend on the sample $x_1, \dots, x_n$.

Since all the summands in the L_n come from densities of independent copies of the same distribution, by the law of large numbers we have $\lim_{n\to\infty} L_n(\theta) = E_{f(x,\theta_0)} log f(x,\theta) = L(\theta)$.

Next observe that for any θ we have $L(\theta) \leq L(\theta_0)$: This is because

$$\begin{aligned} L(\theta) - L(\theta_0) &= E_{f(x,\theta_0)}(log f(X,\theta) - log f(X,\theta_0)) = E_{f(x,\theta_0)}\left(log \frac{f(X,\theta)}{f(X,\theta_0)}\right) \\ &\leq E_{f(x,\theta_0)}\left(\frac{f(X,\theta)}{f(X,\theta_0)} - 1\right) = \int \left(\frac{f(X,\theta)}{f(X,\theta_0)} - 1\right) f(X,\theta_0) \\ &= \int f(X,\theta) - \int f(X,\theta_0) = 1 - 1 = 0 \end{aligned}$$

Thus θ_0 is the maximizer of L.

Altogether we have: That $\hat{\theta}_n$ is the maximizer of L_n, by definition, $L_n \to L$ pointwise, and θ_0 is the maximizer of L. From these three facts it follows that $\theta_n \to \theta_0$. ♡

Example 5.3.5 is a case where the conditions of the theorem are not satisfied.

Sufficient statistics

As in the last section we consider a random variable $\mathbf{X}$ with density function $f(\mathbf{x}, \theta)$ depending on some parameter θ. We encode the taking of n samples as having n i.i.d. random variables $X_1, \dots, X_n$, all having the distribution of $\mathbf{X}$.

A *statistic* on $X_1, \dots, X_n$ is simply some function $\tau(X_1, \dots, X_n)$ computed from $X_1, \dots, X_n$. For example if the X_i are real variables then their average $\tau(X_1, \dots, X_n) = \frac{1}{n}\sum_{i=1}^n X_i$ is a statistic on $X_1, \dots, X_n$.

Definition 5.3.7. *A statistic* $\mathbf{t} = \tau(X_1, \dots, X_n)$ *on* $X_1, \dots, X_n$ *is called a* sufficient statistic *for* θ *if one can find two functions* g *and* h *such that the likelihood function of* $\mathbf{X}$ *can be written as*

$$l(\mathbf{x}, \theta) = g(\theta, \mathbf{t}) \cdot h(\mathbf{x}).$$

Note that h depends only on the data $\mathbf{x}$ and not on θ. Estimating θ can therefore be reduced to solving the equation

$$\frac{\partial}{\partial\theta}(\ln(g(\theta, \mathbf{t})) + \ln(h(\mathbf{x}))) = \frac{1}{g(\theta, \mathbf{t})} \cdot \frac{\partial}{\partial\theta} g(\theta, \mathbf{t}) = 0,$$

which depends only on θ and $\mathbf{t}$ not $\mathbf{x}$.

Remark 5.3.8. Sufficiency is related to the concept of data reduction. Suppose that $\mathbf{X}$ takes values in $\mathbb{R}^n$. If we can find a sufficient statistic that takes values in $\mathbb{R}^i$, then we can reduce the original data vector (whose dimension is usually large) to the vector of statistics (whose dimension is usually much smaller) with no loss of information about the parameter θ.

Example 5.3.9. If $X_1, \ldots, X_n$ are uniformly distributed on the interval $[0, \theta]$ then $\tau(X_1, \ldots, X_n) = \max(X_1, \ldots, X_n) := m$ is a sufficient statistic for θ:

$$l(\mathbf{x}, \theta) = \frac{1}{\theta^n} \cdot I_{[m,\infty)}(\theta),$$

where $I_{[m,\infty)}(\theta) = \begin{cases} 1 & \theta \in [m, \infty) \\ 0 & \text{otherwise} \end{cases}$ and $h(\mathbf{x}) = 1$.

Example 5.3.10. Let $X_1, \ldots, X_n$ be normal distributed variables with expectation μ and variance σ^2. If σ^2 is known then $\tau(X_1, \ldots, X_n) = \sum_{i=1}^n X_i := t$ is a sufficient statistic for μ:

$$\begin{aligned} l(\mathbf{x}, \mu) &= \left(\frac{1}{\sqrt{2\pi}\sigma}\right)^n \exp\left\{-\frac{1}{2\sigma^2}\sum_{i=1}^n (x_i - \mu)^2\right\} \\ &= \left(\frac{1}{\sqrt{2\pi}\sigma}\right)^n \exp\left\{\frac{\mu}{\sigma^2}\sum_{i=1}^n x_i - \frac{n\mu^2}{2\sigma^2}\right\} \exp\left\{-\frac{1}{2\sigma^2}\sum_{i=1}^n x_i^2\right\} \end{aligned}$$

with

$$g(\mu, t) = \left(\frac{1}{\sqrt{2\pi}\sigma}\right)^n \exp\left\{\frac{\mu}{\sigma^2}t - \frac{n\mu^2}{2\sigma^2}\right\} \text{ and } h(\mathbf{x}) = \exp\left\{-\frac{1}{2\sigma^2}\sum_{i=1}^n x_i^2\right\}.$$

See the section on Sufficient Statistics of the Random website for more information.

5.4 Maximum a posteriori estimators

Maximum a posteriori (MAP) estimators are another estimation technique, using ideas inspired by Bayes' theorem.

In maximum likelihood estimation, the parameter θ is assumed to be an unknown, but fixed quantity, about which one tries to derive a guess from the observed data.

In Bayesian estimation, of which MAP estimation is an example, we start not with no knowledge at all, but with some prior assumption on what are more likely and what are less likely values of θ. This assumption is given in the form of a probability distribution on the possible values of θ, called the *prior distribution*. The prior distribution expresses the experimenter's belief before the data are seen.

Then one takes a sample from the unknown distribution at hand and "updates" the belief about the prior distribution, obtaining the so-called *posterior distribution*. This is done using Bayes' formula. Since that is about conditional probabilities, one writes the density functions (resp. probability mass functions [pmf] in the discrete case) $f(x, \theta)$ in question as $f(x \mid \theta)$, thinking of this function as the density function (resp. pmf) of a random variable X under the condition that the parameter is θ. The updating formula then expresses the new belief after having taken the data into account, which is denoted as $f(\theta \mid x)$:

$$f(\theta \mid x) := \frac{f(x \mid \theta)h(\theta)}{f(x)}$$

Here $h(\theta)$ is the density/pmf of the prior distribution, and $f(x)$ describes the absolute probability to observe the data x, taking into account all possible values of θ:

$$f(x) = \begin{cases} \sum_\theta f(x \mid \theta)h(\theta) & \text{(discrete case)} \\ \int f(x \mid \theta)h(\theta)\, \mathrm{d}\theta & \text{(continuous case)} \end{cases}$$

One now searches for the θ maximizing $f(\theta \mid x)$ – the map that assigns to given data this maximizing θ, i.e.

$$\hat{\theta} \colon x \mapsto \operatorname{argmax}_\theta f(\theta \mid x),$$

is called the *Maximum a posteriori estimator*, or short *MAP* estimator.

Note that this makes it unnecessary to compute $f(x)$ – it is just a constant factor that does not influence for which θ the maximum is taken. As always when we use the expression "argmax", we implicitly assume that an optimal point of the corresponding optimization problem exists, and if there exists several, that we have a procedure to choose one of them.

Example 5.4.1. Imagine a shell game, i.e. that game where one person (Player One) shifts around some hats or cups, and the other person (Player Two) has to find the coin that is under one of them.

Let's say that a game consists of three identical rounds. Each round can be won or lost (i.e. Player Two finds the coin or not), and Player One wins with probability θ. The discrete random variable X counting the number of rounds that Player One wins, then takes the values $0, 1, 2, 3$ with probabilities

$$f(k \mid \theta) = \begin{cases} \binom{3}{x}\theta^x(1-\theta)^{3-x} & \text{if } x \in \{0,1,2,3\} \\ 0 & \text{otherwise} \end{cases}$$

This distribution is called the Binomial distribution for probability θ.

It is reasonable to assume that Player One is not at a disadvantage, or otherwise she wouldn't set a up a shell game on the street. So $\frac{1}{2} \leq \theta \leq 1$. Since we have no further information that would lead to preferring one θ over another, we take the uniform distribution on the interval $[\frac{1}{2}, 1]$ as a prior:

$$h(\theta) := \begin{cases} 2 & \text{if } \theta \in [\frac{1}{2}, 1] \\ 0 & \text{otherwise} \end{cases}$$

Now we observe the scores $x_1, \dots, x_n$ of n games (i.e. we get a sample of size n). The joint distribution of parameters and outcomes, i.e. the numerator of the fraction defining the posterior distribution, is

$$f(\theta, x_1, \dots, x_n) = h(\theta)\Pi_{i=1}^n f(x_i \mid \theta) = 2\Pi_{i=1}^n \binom{3}{x_i}\theta^{x_i}(1-\theta)^{3-x_i}$$

for $x_i \in \{0,1,2,3\}$ and $\theta \in [\frac{1}{2}, 1]$ and 0 otherwise. The posterior distribution is thus

$$f(\theta \mid x_1, \dots, x_n) = \frac{h(\theta)\Pi_{i=1}^n f(x_i \mid \theta)}{f(x_1, \dots, x_n)} = \frac{2\Pi_{i=1}^n \binom{3}{x_i}\theta^{x_i}(1-\theta)^{3-x_i}}{f(x_1, \dots, x_n)}.$$

For concreteness, let's suppose we observed 2 games with outcomes $x_1 = 2$ and $x_2 = 1$. Then the joint distribution is

$$\begin{aligned} f(\theta, 2, 1) &= 2\binom{3}{2}\theta^2(1-\theta)^1\binom{3}{1}\theta^1(1-\theta)^2 \\ &= 2\binom{3}{2}\binom{3}{1}\theta^3(1-\theta)^3 = 18\,\theta^3(1-\theta)^3 \end{aligned}$$

The marginal distribution, i.e. the denominator of the fraction defining the posterior distribution, is

$$\begin{aligned}
f(2,1) &= \int_{\frac{1}{2}}^{1} f(\theta,2,1)\mathrm{d}\theta = \int_{\frac{1}{2}}^{1} 18\,\theta^3(1-\theta)^3\mathrm{d}\theta \\
&= 18\int_{\frac{1}{2}}^{1} \theta^3\left(1-3\theta+3\theta^2-\theta^3\right)d\theta \\
&= 18\int_{\frac{1}{2}}^{1} \left(\theta^3-3\theta^4+3\theta^5-\theta^6\right)d\theta \\
&= 18\left[\frac{1}{4}\theta^4-\frac{3}{5}\theta^5+\frac{1}{2}\theta^6-\frac{1}{7}\theta^7\right]_{\frac{1}{2}}^{1} \\
&= 18\left[\frac{1}{4}-\frac{3}{5}+\frac{1}{2}-\frac{1}{7}-\left(\frac{1}{2^6}-\frac{3}{5\cdot 2^5}+\frac{1}{2^7}-\frac{1}{7\cdot 2^7}\right)\right] \\
&= 18\left[\frac{5\cdot 7-2^2\cdot 7+2\cdot 5\cdot 7-2^2\cdot 5}{2^2\cdot 5\cdot 7}-\frac{5\cdot 7\cdot 2-3\cdot 7\cdot 2^2+5\cdot 7-5}{2^7\cdot 5\cdot 7}\right] \\
&= 18\left[\frac{57\cdot 2^5-2^4}{2^7\cdot 5\cdot 7}\right] = 18\left[\frac{57\cdot 2-1}{2^3\cdot 5\cdot 7}\right] = \frac{9\cdot 113}{140}
\end{aligned}$$

We would not have needed to calculate this, in order to find the maximum a posteriori estimator, and just did it for concreteness.

The posterior distribution is now

$$f(\theta \mid 2,1) = \frac{18\,\theta^3(1-\theta)^3}{\frac{9\cdot 113}{140}} = \frac{280}{113}\,\theta^3(1-\theta)^3.$$

Now we need to look for the $\theta \in [\frac{1}{2},1]$ that maximize this function. To find critical points we set

$$0 \overset{!}{=} \frac{\partial}{\partial\theta}f(\theta \mid 2,1) = \frac{280}{113}\left(3\theta^2-12\theta^3+15\theta^4-6\theta^5\right)$$

Equivalently, we need to solve the equation $0 = 3-12\theta+15\theta^2-6\theta^3$. One solution is given by$\theta = 1$. Polynomial division gives $3-12\theta+15\theta^2-6\theta^3 = (\theta-1)(-6\theta^2+9\theta-3)$. The roots of the second factor are the roots of $\theta^2-\frac{3}{2}\theta+\frac{1}{2}$, which are $\frac{3}{4}\pm\sqrt{\frac{9}{16}-\frac{1}{2}} = \frac{3}{4}\pm\frac{1}{4}$, i.e. $\frac{1}{2}$ and 1. So we got two critical points to check, $\frac{1}{2}$ and 1. We have $f(1 \mid 2,1) = 0$ and $f(\frac{1}{2} \mid 2,1) = \frac{2^2\cdot 5\cdot 7}{113}\left(\frac{1}{2}\right)^6 > 0$.

The critical points here happened to be the endpoints of our interval $[\frac{1}{2}, 1]$ of admissible values. Otherwise we would still have had to compare the values at the critical points with the values at the endpoints and take the largest.

Altogether we obtain the maximum a posteriori estimate $\hat{\theta} = \frac{1}{2}$.

Remark 5.4.2. In comparison to maximum likelihood estimation, in maximum a posteriori estimation the prior introduces a tendency to place the estimate closer to the maximum of the prior (assuming for simplicity that there is a single maximum).

One therefore has to choose the prior carefully and shold really be able to justify this tendency that it introduces. If one has such justifications, then one can choose a so-called "informative prior", that contains the information that one wants to use. Sometimes one distinguishes between *subjective Bayesianism*, in which the prior expresses the beliefs of the experimenter, independently of any evidence, and *objective Bayesianism* where the prior is required to be backed up by previous trusted studies, or other kinds of sufficiently objective infomation.

In general, if one only has very sparse information, one should look for an "uninformative prior" – these are priors which do not imply any unwarranted tendencies. Here is a catalog of uninformative priors. It can be made more precise in which sense a prior is uninformative, but we will not discuss this now.

An example of an uninformative prior is the uniform distribution on some interval. When taking this prior, one should trust in *some* information, namely the information that the parameter in question belongs to the interval. But there is no implied tendency as to what values are more likely within that interval. For example, if the parameter in question is a probability p from some other experiment, then one clearly knows that $p \in [0, 1]$.

If the prior $h(x)$ is a uniform distribution, then within the interval where it is non-zero, one has

$$\operatorname{argmax}_\theta f(\theta \mid x) = \operatorname{argmax}_\theta f(x \mid \theta) \frac{h(\theta)}{f(x)} = \operatorname{argmax}_\theta f(x \mid \theta)$$

because $\frac{h(\theta)}{f(x)}$ is a constant, independent of θ. But $f(x \mid \theta)$ is precisely the likelihood function! Hence, up the the issue of restriction to an interval we have seen:

For the uniform prior, maximum a posteriori estimation reduces to maximum likelihood estimation. Example 5.4.1 is actually an example of this.

Remark 5.4.3. The posterior distribution can be used for more than just extracting its maximum! It is a probability distribution and one can read off all kinds of things from it. For example its variance can be understood as information on, how sure we can be that the "true" parameter is somewhere close to the maximum. This brings us right to the question of how much sense it actually makes to describe parameters by a probability distribution. One way to make sense of it is to think of that probability distribution as expressing our degree of certainty that the paramaters is (close to) what we think it is. There is a debate on whether this is a meaningful thing to do or not – see the next section:

5.5 Bayesian and frequentist statistics

The prior distribution is meant to model the experimenter's belief about θ. This is the subject of an ongoing debate between the so-called "Bayesian" approach and the "Frequentist" approach to statistics. Roughly, while Bayesians think that it makes sense, and is consistently possible, to measure the strength of a belief, or the degree to which an assumption is reasonable, by a number between 0 and 1 (i.e. a probability), the frequentists deny that. Frequentists insist that the likelihood is *not* a probability of anything, and that the only meaningful interpretation that can be given to probability of an event, is in terms of the frequency with which that event will occur during many repeated experiments.

For example, according to a frequentist, one could not meaningfully talk about the probability that there is surviving population of mammoths, left over from the last ice age somwhere in Siberia, because evolution is not an experiment that could be repeated. According to a Bayesian, considering all kinds of factors that are in favor or against of the survival of a mammoth one could possibly compute number expressing how probable the hypothesis is.

There are lots of grey areas there. For example, the probability of rain on a specific day, given in a weather forecast, is of course not literally based on a repeatable experiment (because we can not let the same day hapen many times), but it is actually computed by considering all previous days with similar weather conditions and taking the percentage of those days on which it later rained. A frequentist would be totally fine with that.

In part this is a philosophical debate, but in part it also has explicit consequences for concrete computations. For some examples see Chapter 37 McKay's book "Information Theory, Inference, and Learning Algorithms". For a really simple and detailed example see Björn Lindqvist's answer on this forum.

One can, however, simply employ both, frequentist and Bayesian methods without entering this debate. They have been proven successful in practice, and that may be reason enough. Machine Learning largely relies on Bayesian methods.

For further introduction and history see Lecture 1 these course notes on Bayesian Modeling and Inference. Also here is a Statistics FAQ taking a decidedly Bayesian view, that is full of interesting viewpoints and references.

5.6 Conjugate priors and updating hyperparameters

In general, computation of the posterior by hand can get fairly complicated, and, depending on the involved priors, can not even be given a closed form formula. This does not stop us from computing it numerically, which is just as fine for concrete problems. For theoretical considerations, however, it is desirable to have a closed formula. One way of achieving this, is to use a prior from an appropriate class of distributions, which is closed under Bayesian updates.

Definition 5.6.1. *Let X be a random variable with distribution $X \sim f(x \mid \theta)$ ($\theta \in \Theta$ unknown). A collection $\mathcal{C}$ of probability density functions, or probability mass functions, is called a* conjugate prior family *for the family $\{f(x \mid \theta) \mid \theta \in \Theta\}$, if, whenever one chooses a prior from $\mathcal{C}$, the posterior is also from $\mathcal{C}$.*

According to this definition, certainly the collection $\mathcal{C}$ of *all* probability density functions is a conjugate prior family, but not a very useful one. A conjugate prior family is useful, if it is itself a parametrized family, preferrably depending on very few parameters. One can then hope to compute the process of Bayesian updating entirely in terms of the parameters, getting back a result in the same class. This is especially convenient for iterating the updates when more and more data become available.

Example 5.6.2. An important family of probability distributions are the Beta distributions. This is the family of distributions on the open unit interval $(0,1)$, depending on two parameters α and β, with density functions

$$h_{\alpha,\beta}(\theta) := Beta(\alpha,\beta)(\theta) := C \cdot \theta^{\alpha-1}(1-\theta)^{\beta-1}$$

where the constant C is a normalizing constant ensuring that the integral over the unit interval of this function is 1.

Concretely $C := \left(\int_0^1 \theta^{\alpha-1}(1-\theta)^{\beta-1}\,\mathrm{d}u\right)^{-1}$, but this constant can also be expressed in terms of the Gamma function, which is a well-known function in mathematics.

The Beta distributions are often taken as distributions for a parameter expressing a probability. They form a conjugate family for the Binomial distributions: Recall that the Binomial distribution is the distribution for an $\mathbb{N}$-valued random variable X that counts the number of successes in a sequence of k trials with success probability θ. It is given by

$$P(X=x) := \binom{k}{x}\theta^{x}(1-\theta)^{k-x}.$$

If we have a sample of size n, i.e. natural numbers $x_1,\ldots,x_n$, the likelihood function is

$$\Pi_{i=1}^{n}\binom{k}{x_i}\theta^{x_i}(1-\theta)^{k-x_i}.$$

Thus if the prior is $Beta(\alpha,\beta)$, then the posterior is given by

$$\begin{aligned} f(\theta \mid x_1,\ldots,x_n) &= \frac{\Pi_{i=1}^{n}\binom{k}{x_i}\theta^{x_i}(1-\theta)^{k-x_i}\cdot C\cdot\theta^{\alpha-1}(1-\theta)^{\beta-1}}{f(x_1,\ldots,x_n)} \\ &= \left(\Pi_{i=1}^{n}\binom{k}{x_i}\right)\cdot\frac{C}{f(x_1,\ldots,x_n)}\cdot\theta^{(\alpha-1+x_1+\ldots+x_n)}(1-\theta)^{(\beta-1+nk-x_1-\ldots-x_n)} \\ &= Beta(\alpha+x_1+\ldots+x_n, \beta+nk-x_1-\ldots-x_n) \end{aligned}$$

Here the last two factors clearly are in the form required for a Beta distribution. That the first factor $\left(\Pi_{i=1}^{n}\binom{k}{x_i}\right)\cdot\frac{C}{f(x_1,\ldots,x_n)}$ really is the normalizing factor required for the Beta distribution, simply follows from the fact that the Bayesian updating procedure always results in a probability density function. An explicit calculation with the normalizing factors is therefore unnecessary, but if you want to see it, it can be found in this blog post.

Note also that the update rule can be easily interpreted: successes raise the hyperparameter α and lower the parameter β, and vice versa with failures. High α (in comparison to β) means that a high value of θ is more likely; see the animation on the Wikipedia page to get a feeling on what effects the parameters α, β have.

So if we trust the method of Bayesian updates, then the successes in our samples shift our belief towards a higher value of p, and failures towards a lower value of p, as they should.

Example 5.6.3. With almost the same computation as in Example 5.6.2, one sees that the Beta distributions also form a conjugate family for the Geometric distributions. I leave this to you as an exercise.

Definition 5.6.4. *The parameters that parametrize the conjugate family are called* hyperparameters*, to distinguish them from the parameters of the probability distribution.*

In Example 5.6.2 you can see how one can conveniently update the belief about the parameter θ each time one obtains new data: Just switch the old hyperparameters to the new ones obtained via the calculation of the posterior.

You can also see that this can be iterated: If new data arises, just take the old posterior as the new prior and update again. This makes Online Learning easy to realize.

You can also see that in this particular example it does not matter whether you take a whole chunk of new data and do an update for everything at once, or if you separate your data in several little pieces and update repeatedly – it just amounts to adding up hyperparameters. Many examples behave similarly.

Remark 5.6.5. Ideally, a conjugate family should also contain some uninformative priors – otherwise one could only use it when specific information is given which supports a member of the family. This is the case for the family of Beta distributions: The distribution $Beta(1,1)$ is the uniform distribution on the unit interval $[0,1]$, and $Beta(0,0)$ and $Beta(\frac{1}{2},\frac{1}{2})$ can also be seen as uninformative priors.

Here is a small table of conjugate families for some likelihood functions. Note that the choice of conjugate family depends on what parameter we want to estimate with Bayesian updates.

Distribution	parameter	conjugate family
Bernoulli(p)	p	Beta
Binomial(k,p)	p	Beta
Geometric(p)	p	Beta
Poisson(λ)	λ	Gamma
Multinomial($p_1, \dots, p_n$)	$p_1, \dots, p_n$	Dirichlet
univariate Normal(μ, σ^2)	μ	Normal
univariate Normal(μ, σ^2)	σ^2	Inverse Gamma
multivariate Normal($\boldsymbol{\mu}, \boldsymbol{\Sigma}$)	$\boldsymbol{\mu}$	multivariate Normal
multivariate Normal($\boldsymbol{\mu}, \boldsymbol{\Sigma}$)	$\boldsymbol{\Sigma}$	inverse Wishart
Uniform($[0, \theta]$)	θ	Pareto

A bigger table of conjugate priors for several distributions is given on the Wikipedia page on conjugate priors. It includes the rules for how to update the hyperparameters in dependence of new data. For a big list, see Fink's Compendium of conjugate priors.

5.7 Exponential families

Exponential families are certain families of probability distributions. Many standard families are exponential. For exponential families it is easy to give conjugate families and often one can employ reasoning that works for Gaussian distributions.

Definition 5.7.1. *A family of probability distributions, parametrized by an $\mathbb{R}^n$-valued parameter $\boldsymbol{\theta}$, is an* exponential family, *if the density functions, or probability mass functions, belonging to this family are of the form*

$$f(\boldsymbol{x} \mid \boldsymbol{\theta}) = h(\boldsymbol{x}) \exp\left(\boldsymbol{b}(\boldsymbol{\theta})^T \cdot \mathbf{T}(\boldsymbol{x}) - A(\boldsymbol{\theta})\right)$$

for some $\mathbb{R}^n$-valued functions $\boldsymbol{b}, \mathbf{T}$ and $\mathbb{R}$-valued functions h, A.

Equivalently, setting $w(\boldsymbol{\theta}) := \exp(-A(\boldsymbol{\theta}))$, one can write

$$f(\boldsymbol{x} \mid \boldsymbol{\theta}) = h(\boldsymbol{x}) w(\boldsymbol{\theta}) \exp\left(\boldsymbol{b}(\boldsymbol{\theta})^T \cdot \mathbf{T}(\boldsymbol{x})\right).$$

Example 5.7.2. (The Bernoulli distribution) The possible values of x here are just 0 and 1. The probability mass function is

$$\begin{aligned} f(x \mid \pi) &= \pi^x (1-\pi)^{1-x} \\ &= \exp\left\{\log\left(\frac{\pi}{1-\pi}\right) x + \log(1-\pi)\right\} \end{aligned}$$

with

$$\begin{aligned} b(\pi) &= \log\left(\frac{\pi}{1-\pi}\right) \\ \mathbf{T}(x) &= x \\ A(\pi) &= -\log(1-\pi) \\ h(x) &= 1. \end{aligned}$$

Example 5.7.3. (The Poisson distribution) Here the possible values for x are the natural numbers. The probability mass function is

$$\begin{aligned} f(x \mid \lambda) &= \frac{\lambda^x e^{-\lambda}}{x!} \\ &= \frac{1}{x!}\exp\{\log(\lambda)x - \lambda\} \end{aligned}$$

with

$$\begin{aligned} b(\lambda) &= \log\lambda \\ \mathbf{T}(x) &= x \\ A(\lambda) &= \lambda \\ h(x) &= \frac{1}{x!}. \end{aligned}$$

Example 5.7.4. (The Gaussian distribution)

$$\begin{aligned} f(x \mid \mu, \sigma^2) &= \frac{1}{\sqrt{2\pi}\sigma}\exp\left\{-\frac{1}{2\sigma^2}(x-\mu)^2\right\} \\ &= \frac{1}{\sqrt{2\pi}}\exp\left\{\frac{\mu}{\sigma^2}x - \frac{1}{2\sigma^2}x^2 - \frac{1}{2\sigma^2}\mu^2 - \log\sigma\right\} \end{aligned}$$

with

$$\begin{aligned}\boldsymbol{b}((\mu,\sigma^2)) &= \begin{pmatrix}\mu/\sigma^2\\ -1/2\sigma^2\end{pmatrix}\\ \mathbf{T}(x) &= \begin{pmatrix}x\\ x^2\end{pmatrix}\\ A((\mu,\sigma^2)) &= \frac{\mu^2}{2\sigma^2}+\log\sigma\\ h(x) &= \frac{1}{\sqrt{2\pi}}.\end{aligned}$$

Example 5.7.5. (The multinomial distribution) For an experiment having possible outcomes $1,\dots,K$ with probabilities $p_1,\dots,p_K$, one can count how many times each of the outcomes occurred. This gives rise to an $\mathbb{N}^K$-valued random variable with the multinomial distribution, with probability mass function

$$\begin{aligned}f(\mathbf{x}\mid\boldsymbol{p}) &= \frac{(\sum_{i=1}^K x_i)!}{x_1!\cdots x_K!}p_1^{x_1}\cdots p_K^{x_K}\\ &= \frac{(\sum_{i=1}^K x_i)!}{x_1!\cdots x_K!}\exp\left\{\sum_{i=1}^K x_i\log p_i\right\}\end{aligned}$$

with

$$\begin{aligned}\boldsymbol{b}(\boldsymbol{p}) &= \begin{pmatrix}\log p_1\\ \vdots\\ \log p_K\end{pmatrix}\\ \mathbf{T}(\mathbf{x}) &= \begin{pmatrix}x_1\\ \vdots\\ x_K\end{pmatrix}\\ A(\boldsymbol{p}) &= 0\\ h(\mathbf{x}) &= \frac{(\sum_{i=1}^K x_i)!}{x_1!\cdots x_K!}.\end{aligned}$$

While the above general form of exponential families is good for recognizing an exponential family, one can always bring such a family into a more restricted form, which will be convenient for the following theorem. Given an exponential family in the second (equivalent) form of Def. 5.7.1

$$f(\boldsymbol{x} \mid \boldsymbol{\theta}) = h(\boldsymbol{x}) w(\boldsymbol{\theta}) \exp\left(\boldsymbol{b}(\boldsymbol{\theta})^T \cdot \mathbf{T}(\boldsymbol{x})\right),$$

it is clear that the last factor only depends on $\boldsymbol{b}(\boldsymbol{\theta})$, and not on $\boldsymbol{\theta}$ itself. The other factor featuring $\boldsymbol{\theta}$ is $w(\boldsymbol{\theta})$, and there is only one choice for this that ensures that the integral of the whole expression is 1 (making it into a density function), namely

$$w(\boldsymbol{\theta}) = \left(\int_{\mathbb{R}^n} h(\boldsymbol{x}) \exp\left(\boldsymbol{b}(\boldsymbol{\theta})^T \cdot \mathbf{T}(\boldsymbol{x})\right) \mathrm{d}\boldsymbol{x}\right)^{-1}$$

This shows that $w(\boldsymbol{\theta})$ also only depends on $\boldsymbol{b}(\boldsymbol{\theta})$, and not on $\boldsymbol{\theta}$ itself.

Thus defining $\boldsymbol{\eta} := \boldsymbol{b}(\boldsymbol{\theta})$, $\tilde{w}(\boldsymbol{\eta}) := w(\boldsymbol{\theta})$ and $\tilde{A}(\boldsymbol{\eta}) := -\ln \tilde{w}(\boldsymbol{\eta})$, we get

$$f(\boldsymbol{x} \mid \boldsymbol{\theta}) = f(\boldsymbol{x} \mid \boldsymbol{\eta}) = h(\boldsymbol{x})\tilde{w}(\boldsymbol{\eta}) \exp\left(\boldsymbol{\eta}^T \cdot \mathbf{T}(\boldsymbol{x})\right) = h(\boldsymbol{x}) \exp\left(\boldsymbol{\eta}^T \cdot \mathbf{T}(\boldsymbol{x}) - \tilde{A}(\boldsymbol{\eta})\right)$$

What we do when we pass from $\boldsymbol{b}(\theta)$ to η is a reparametrization of our family of distributions: If $\boldsymbol{b}$ is non-injective, this even removes redundancy, if not, it simply "renames" the members of our family. This allows us to leave out the function $\boldsymbol{b}$ without loss of generality in the following theorem.

Theorem 5.7.6. *Consider the exponential family of distributions*

$$f(\boldsymbol{x} \mid \boldsymbol{\eta}) = h(\boldsymbol{x}) \exp\left(\boldsymbol{\eta}^T \cdot \mathbf{T}(\boldsymbol{x}) - A(\boldsymbol{\eta})\right).$$

The family of distributions

$$q(\boldsymbol{\eta} \mid \chi, \nu) := u(\chi, \nu) \cdot \exp\left(\boldsymbol{\eta}^{\mathrm{T}} \cdot \chi - \nu \cdot A(\boldsymbol{\eta})\right)$$

with hyperparameters $\chi \in \mathbb{R}^n$ *and* $\nu \in \mathbb{R}_{>0}$, *and* $u(\chi, \nu)$ *the unique function ensuring* $\int_{\mathbb{R}^n} q(\boldsymbol{\eta} \mid \chi, \nu) d\boldsymbol{\eta} = 1$, *is a conjugate family (and visibly also an exponential family). The posterior for a size n sample* $(x_1, x_2, \ldots, x_n)$ *is:*

$$q(\boldsymbol{\eta} \mid \chi + \sum_{i=1}^{n} \boldsymbol{T}(x_i), \nu + n)$$

Proof. The likelihood function for a size n sample $(x_1, x_2, \dots, x_n)$ is

$$f(x_1, x_2, \dots, x_n \mid \boldsymbol{\eta}) = \left(\prod_{i=1}^{n} h(x_i)\right) \exp\left(\boldsymbol{\eta}^T \left(\sum_{i=1}^{n} \boldsymbol{T}(x_i)\right) - nA(\boldsymbol{\eta})\right)$$

Taking $q(\boldsymbol{\eta} \mid \chi, \nu)$ as a prior, and denoting the denominator from the posterior formula simply by D, the posterior is thus

$$\begin{aligned}
&\frac{f(x_1, x_2, \dots, x_n \mid \boldsymbol{\eta}) q(\boldsymbol{\eta} \mid \chi, \nu)}{D} \\
&= \frac{u(\chi, \nu)}{D} \left(\prod_{i=1}^{n} h(x_i)\right) \exp\left(\boldsymbol{\eta}^T \left(\sum_{i=1}^{n} \boldsymbol{T}(x_i)\right) - nA(\boldsymbol{\eta})\right) \\
&\qquad\qquad\qquad\qquad \cdot \exp\left(\boldsymbol{\eta}^{\mathrm{T}} \cdot \chi - \nu \cdot A(\boldsymbol{\eta})\right) \\
&= \frac{u(\chi, \nu)}{D} \left(\prod_{i=1}^{n} h(x_i)\right) \exp\left(\boldsymbol{\eta}^T \left(\chi + \sum_{i=1}^{n} \boldsymbol{T}(x_i)\right) - (\nu + n)A(\boldsymbol{\eta})\right) \\
&= \tilde{u}(\chi, \nu) \exp\left(\boldsymbol{\eta}^T \left(\chi + \sum_{i=1}^{n} \boldsymbol{T}(x_i)\right) - (\nu + n)A(\boldsymbol{\eta})\right) \\
&= q(\boldsymbol{\eta} \mid \chi + \sum_{i=1}^{n} \boldsymbol{T}(x_i), \nu + n)
\end{aligned}$$

where in the line before the last we define $\tilde{u}(\chi, \nu) := \frac{u(\chi,\nu)}{D} \left(\prod_{i=1}^{n} h(x_i)\right)$. □

The form of the posterior suggests an interpretation of the hyperparameters χ and ν of the conjugate family of Thm. 5.7.6: Since the Bayesian update of the prior with the information gained from a size n sample adds n to the parameter ν, this parameter can be understood as denoting sample size. If we use a prior $q(\boldsymbol{\eta} \mid \chi, \nu)$, we can consider it as trustworthy as information obtained from ν observations. Likewise, the parameter χ can be understood as the sum of $\boldsymbol{T}$ applied to each of these hypothetical observations.

Remark 5.7.7. Theorem 5.7.6 tells us that we are in a good situation for Bayesian reasoning when our statistical model is an exponential family. But when is it justified to assume such a statistical model? In the special cases of multinomial, normal or Poisson distributions, for example, we know what kind of situations they apply to. There is, however, also a general principle that serves as justification for exponential families, and that applies very often: When we have an $\mathbb{R}^n$-valued random variable, and we have data giving us an estimate for the expectation, then the distribution of maximum entropy can be shown to fall into an exponential family – what this means will be explained in the chapter on Information Theory. Even better: While we gave the definition of exponential family without motivation, one can derive, and hence justify, this definition as the solution to an optimization problem.

There are further natural occurrences and pleasant properties of exponential families: For example graphical models give rise to exponential families, and for exponential families one can achieve the optimal value for the Cramér-Rao bound, a lower bound for the variance of a maximum likelihood estimator, but this is out of the scope of this course. The latter story is not just a good property that exponential families happen to enjoy, but one can derive the definition of exponential family in this way – see Suhov/Kelbert: Probability and Statistics by example, Section 3.7 for this.

5.8 Regression

Regression is again a kind of parameter estimation for a statistical model. In the previous sections our statistical models just consisted of parametrized families of distributions. Under the assumption that the observed data was generated by one of these distributions, we tried to estimate its parameters from the observed data.

In regression one tries to display one random variable as a function of other variables. Thus one has a slightly different notion of statistical model: Now it is a parametrized family of *functions* from which one tries to pick the one that best reflects the data. More precisely, the ingredients in a typical setup are

1. A list of random variables, called *independent variables*, or *explanatory variables*, or *regressors*, often denoted $X_1, \ldots, X_n$

2. A random variable, called *dependent variable* or *response variable*, often denoted Y

3. A parametrized family of functions $f(X_1, \ldots, X_n \mid \theta)$

4. A random variable ϵ, called the *error term*.

In practice the random variables X_i, Y should be observable, i.e. their outcomes should be obtainable as data resulting from some experiment.

The goal in regression is to display the response variable as a function of the explanatory variables, as good as possible, using one of the functions $f(-, \theta)$: One assumes that there is a relationship of the form $Y = f(X_1, \ldots, X_n \mid \theta) + \epsilon$. Then one tries to find the θ that realizes this relationship in the most plausible way.

More precisely: We already know from Theorem 4.7.9 that the random variable $E(Y \mid X_1, \ldots, X_n)$ is the best approximation of Y by a function of $X_1, \ldots, X_n$. So we are trying to find a θ such that $E(Y \mid X_1, \ldots, X_n) = f(X_1, \ldots, X_n \mid \theta)$, and the remaining variation for fixed values of the X_i then needs to be explained by the error term ϵ.

If we manage to find a θ with $E(Y \mid X_1, \ldots, X_n) = f(X_1, \ldots, X_n \mid \theta)$, then the error term ϵ has expectation zero. Ideally the family $f(-, \theta)$ should be big enough that one can find a member which makes this true.

We will get to know linear and generalized linear models as examples of regression models, among others.

Remark 5.8.1. I would have liked to start this section with a formal definition of "statistical model", but there is no universally agreed upon such definition. Usual working definitions are often too loose, allowing many non-sensical examples, or too tight, tailored to special cases and excluding some examples of statistical models used in practice. It is the subject of ongoing research and discussion, what a formal notion of "statistical model" might look like. An interesting contribution to this is the article

McCullagh: What is a statistical model? (2002).

The linked article is accompanied by lots of replies to McCullagh's proposal, illustrating how much potential for interesting debates this question has. A substantial recent contribution (from 2020) is the following article:

Patterson: The algebra and machine representation of statistical models

Apart from addressing the question itself, it also discusses the potential benefits of having a well-defined notion of statistical model, in view of the modern development of machine learning.

In these notes we will not enter this discussion, and simply consider parametrized families of probability distributions, possibly with declarations of which variables are explanatory and which are response variables. After all, that is how most statisticians work, and rather successfully so.

5.9 Linear regression

Linear regression is probably the simplest kind of regression. The corresponding statistical model is a linear model. Linear regression is used as an all purpose tool in economy.

In this section essentially we will review the search for linear models via the least squares method, that we discussed at the end of the linear algebra section, in the language of probability theory.

Recall that given a set of k samples, or measurements, each with $n+1$ features $(x_1^{(i)}, x_2^{(i)}, \ldots, x_n^{(i)}, y^{(i)})$ we were trying to find coefficients $b_1, \ldots, b_n$ allowing to express each $y^{(i)}$ as a linear combination of the $x_1^{(i)}, \ldots, x_n^{(i)}$. In other words, we wanted a solution b of $y = Ab$ where $y = (y^{(1)}, \ldots, y^{(k)})$ and $A = (x_j^{(i)})$.

Even if this was not possible simultaneously for all i, we wanted to find the best approximation, i.e. those $b_1, \ldots, b_n$ that minimize the sum of all (squared) distances $(x_1^{(i)} b_1 + \ldots + x_n^{(i)} b_n - y^{(i)})^2$.

By Theorem 1.78 the solution is given by the pseudoinverse: $\hat{b} := A^+ y$. Back then we chose to center our data first around its mean. If we do not do that, then we are instead asking for an expression of the $y^{(i)}$ as values of an *affine* (instead of linear) function of the $x_1^{(i)}, \ldots, x_n^{(i)}$, i.e. as $y^{(i)} = b_0 + x_1^{(i)} b_1 + \ldots + x_n^{(i)} b_n$. Still our theorem applies – just add a column of 1s to the matrix A.

We now want to describe the situation in terms of probability theory. A reasonable first approach to a description would be: The samples are values of random variables $X_1, \ldots, X_{n-1}, Y$, i.e. $x_j^{(i)} = X_j(\omega_i)$ and $y^{(i)} = Y(\omega_i)$, for some ω_i, $i = 1, \ldots, k$ in a probability space Ω.

We suspect that the random variable Y is a linear combination of the random variables X_j, up to some “noise”, which accounts for the fact that a precise solution is not possible:

$$Y = b_0 + X_1 b_1 + \ldots + X_n b_n + \epsilon$$

Here ϵ is supposed to be a normally distributed random variable with expectation 0, capturing the noise, i.e. the deviation from Y being really linearly dependent on the X_j.

We will now analyze, and describe by a probabilistic model, *the process of taking k samples and estimating the parameters $b_0, \ldots, b_n$ based on this.* In comparison to our first approach we will therefore pass to the probability space Ω^k, where an elementary event consists of k samples from the original Ω, and consider random variables $X_j^{(i)}$, $(1 \leq j \leq n,\ \ 1 \leq i \leq k)$ and $Y^{(1)}, \ldots, Y^{(k)}$.

As the deviation from our suspected linear relation can vary with each sample, we also introduce k error random variables $\epsilon^{(1)}, \ldots, \epsilon^{(k)}$ and postulate for each $1 \leq i \leq k$:

$$Y^{(i)} = b_0 + X_1^{(i)} b_1 + \ldots + X_n^{(i)} b_n + \epsilon^{(i)}$$

The $X_j^{(i)}$ are called the *explanatory variables* and the $Y^{(i)}$ are called the *response variables*.

We make the following assumptions:

1. $E\epsilon^{(i)} = 0$
2. $\mathrm{Var}(\epsilon^{(i)}) = \mathrm{Var}(Y^{(i)}) = \mathrm{Var}(Y^{(i)} \mid X_1^{(i)} = x_{i1}, \ldots, X_n^{(i)} = x_{in}) = \sigma^2$ for all $1 \leq i \leq k$ and some $\sigma \in \mathbb{R}$
3. $\mathrm{Cov}(\epsilon^{(i)}, \epsilon^{(j)}) = \mathrm{Cov}(Y^{(i)}, Y^{(j)}) = 0$ for all $1 \leq i, j \leq k$

Remark 5.9.1. Assumption 1. is not really an assumption about the situation - if the $E\epsilon^{(i)} \neq 0$, we can simply substitute the constant coefficient b_0 by $b_0 + E\epsilon^{(i)}$ and $\epsilon^{(i)}$ by $\epsilon^{(i)} - E\epsilon^{(i)}$. Note that assumption 1. is equivalent to $E(Y^{(i)} | X_1^{(i)} = x_{i1}, \ldots, X_n^{(i)} = x_{in}) = b_0 + b_1 x_1^{(i)} + \ldots + b_n x_n^{(i)}$.

Assumption 2. says two things: One the one hand the variances of the $\epsilon^{(i)}$ and $Y^{(i)}$ are the same for all i. On the other hand it says that the variances of the $\epsilon^{(i)}$ and $Y^{(i)}$ are not influenced by the values the $X_j^{(i)}$ – this is called *homoscedasticity*.

Assumption 3. says that the different $Y^{(i)}$ are uncorrelated.

Assumption 3., and the part of assumption 2. that says that all $Y^{(i)}$ have the same variance are very reasonable if one thinks of the intended meaning that the different $Y^{(i)}$ arise from repeated picking of samples, i.e. from repetition of the same experiment. Homoscedasticity, on the other hand, is a more substantial assumption.

Our postulate about the relation of the $Y^{(i)}$ to the other data can be expressed as $Y = \mathbf{X}b + \epsilon$, where Y, ϵ are the obvious vectors of random variables, b is a vector of numbers and $\mathbf{X}$ is the matrix consisting of the $X_j^{(i)}$. The assumptions become, in this notation:

$$\epsilon \sim \mathcal{N}(0, \sigma^2 \mathbf{I}_k), \ \mathrm{Cov}(Y) = \sigma^2 \mathbf{I}_k \text{ and } E(Y \mid (X_1|\ldots|X_n) = \mathbf{X}) \ = \ \mathbf{X}b.$$

Proposition 5.9.2. *Define $\hat{b} := \mathbf{X}^+ Y$. Then $\hat{Y} := \mathbf{X}\hat{b}$ is a linear estimator of Y minimizing the square error $\epsilon^2 = \|\hat{Y} - Y\|^2$ at each $\omega \in \Omega$.*

Proof. Of course $\hat{Y}$ is linear in the explanatory variables $\mathbf{X}$ by definition. We know from Theorem 1.78 that $\hat{b} := X^+ Y$ is the solution minimizing the squared distances $\|\hat{Y} - Y\|^2$. □

We turn towards the properties of the estimator $\hat{b}$ itself. From now on we assume that $\mathbf{X}$ has full rank $n+1$. This is usually the case, when we have enough samples, unless there is a linear dependence between the X_i. Recall that in this case we have $X^+ = (X^T X)^{-1} X^T$, and hence $X^+ X = \mathbf{I}$.

Proposition 5.9.3. *The estimator $\hat{b}$ is unbiased*

Proof. $E(\hat{b}) = E(\mathbf{X}^+ Y) = \mathbf{X}^+ E(Y) = \mathbf{X}^+ E(\mathbf{X}b + \epsilon) = \mathbf{X}^+ \mathbf{X} E b + E\epsilon = Eb$ where the last equality holds because of $X^+ X = \mathbf{I}$ and $E\epsilon = 0$. □

We can compute the covariance matrix of $\hat{b}$:

Proposition 5.9.4. $\mathrm{Cov}(\hat{b}) = \sigma^2 (\mathbf{X}^T \mathbf{X})^{-1}$

Proof. By Prop. 4.6.7 we have $\mathrm{Cov}(\mathbf{Az}) = \mathbf{A}\,\mathrm{Cov}(\mathbf{z})\mathbf{A}^T$. Hence

$$\begin{aligned} \mathrm{Cov}(\hat{b}) &= \mathrm{Cov}(\mathbf{X}^+ Y) = \mathbf{X}^+ \mathrm{Cov}(Y)(\mathbf{X}^+)^T = \mathbf{X}^+(\sigma^2 \mathbf{I}_k)(\mathbf{X}^+)^T \\ &= \sigma^2 \mathbf{X}^+ (\mathbf{X}^+)^T = \sigma^2 (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T ((\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T)^T \\ &= (\sigma^2 \mathbf{I}_k)(\mathbf{X}^T \mathbf{X})^{-1} (\mathbf{X}^T \mathbf{X})(\mathbf{X}^T \mathbf{X})^{-1} = \sigma^2 (\mathbf{X}^T \mathbf{X})^{-1} \end{aligned}$$

□

Theorem 5.9.5 (Gauss-Markov)**.** *The estimators $\hat{b}_i$ have the smallest variance among all linear unbiased estimators of b_i.*

Proof. A straightforward computation, see e.g. the proof on Wikipedia (note that A' there is supposed to mean A^T). □

An estimator that minimizes variance is called a *best estimator*. The theorem of Gauss-Markov is summarized as saying that $\hat{b}_i$ is BLUE (best linear unbiased estimator).

5.10 Generalized linear models

Let $Y, X_1, \dots, X_n$ be $\mathbb{R}^n$-valued random variables. We continue to suppose that Y depends somehow on the X_i and call Y the *response variable* and the X_i the *explanatory variables*.

One uses generalized linear models, if one assumes that Y depends on the X_i only through some linear combination $\eta = b_0 + b_1 X_1 + \ldots + b_n X_n$, to which one applies a further invertible function g:

Definition 5.10.1. *A* generalized linear model (GLM) *(for Y depending on $X_1, \dots, X_n$) consists of*

1. *a* linear predictor $\eta = b_0 + b_1 X_1 + \ldots + b_n X_n$
2. *a differentiable and invertible* link function $g\colon \mathbb{R}^n \to \mathbb{R}^n$ *such that* $\mu := E(Y \mid X) = g^{-1}(\eta)$

Remark 5.10.2. Sometimes one additionally takes as part of the datum of a generalized linear model a function giving the variance in terms of the expectation:

3. a function $V\colon \mathbb{R} \to \mathbb{R}$ such that $\mathrm{Var}(Y_i) = V(\mu_i)$, where Y_i, μ_i denote the respective ith components of the vectors Y, μ.

Generalized linear models are applicable in situations where linear models are not, e.g. when

(a) the variance is not constant as the values of X_i vary, or

(b) the range of Y is restricted (e.g. it is $\{0, 1\}$) and thus is not the range of a linear function.

In matrix notation, and without the variance function, the components of a linear model are related to each other by $\eta = \mathbf{X}b$ and $E(Y \mid X) = \mu = g^{-1}(\eta)$.

Examples 5.10.3. Some important distributions suitable for a generalized linear model:

distribution of Y	link function	inverse link function
normal	id: $Xb = \mu$	$\mu = Xb$
Poisson	ln: $Xb = \ln(\mu)$	$\mu = e^{Xb}$
Bernoulli $\{0,1\}$	logit: $Xb = \ln(\frac{\mu}{1-\mu})$	$\mu = \frac{e^{Xb}}{1+e^{Xb}} = \frac{1}{1+e^{-Xb}}$
binomial $\mathbb{N}$		
multinoulli $\{1,\ldots,k\}$		
multinomial $\mathbb{N}^k$		

Remark 5.10.4. There also are the generalized additive models (GAMs), where one just demands $g(\mathrm{E}(Y)) = b_0 + f_1(x_1) + f_2(x_2) + \cdots + f_n(x_n)$ for smooth functions $f_1, \ldots, f_n$. These are very general, but fitting is computationally very expensive.

The process of choosing the parameters of a model (here a generalized linear model) in such a way that it is a good explanation of some observed data is called *fitting the model*. A standard procedure for doing this is to choose those parameters which maximize the likelihood function. Another procedure is maximizing the a posteriori likelihood.

Usually generalized linear models are applied to distributions forming an exponential family.

Example 5.10.5 (Poisson regression)**.** Recall that an $\mathbb{N}$-valued random variable Y is Poisson distributed with parameter λ if Y has probability mass function $P(Y = y) = f(y \mid \lambda) = \frac{\lambda^y}{y!}e^{-\lambda}$. The expectation of Y is λ. A typical use case for this distribution is counting of instances of radioactive decay per minute.

Now say we believe *the logarithm of the expectation* to depend linearly on some other $\mathbb{R}^n$-valued random variable X. For example the radioactive decay might be slowed down by a lead barrier with several gates, and the entries of X measure how wide we open the gates. Then we can use the logarithm as a link function in a generalized linear model:

$$\ln E(Y \mid X = x) = \theta^T x, \qquad \text{i.e.} \qquad E(Y \mid X = x) = e^{\theta^T x}$$

for an unknown parameter vector θ. Then the Poisson distribution's mass function is

$$f(y \mid x, \theta) = \frac{\lambda^y}{y!} e^{-\lambda} = \frac{e^{y\theta^T x}}{y!} e^{-\left(e^{\theta^T x}\right)}$$

Given measurements $y_1, \ldots, y_n$ in the states $x_1, \ldots, x_n$ we have the likelihood function

$$f(\theta \mid x_1, \ldots, x_n, y_1, \ldots, y_n) = \prod_{i=1}^{n} \frac{e^{y_i \theta^T x_i}}{y_i!} e^{-\left(e^{\theta^T x_i}\right)}$$

To find the parameters θ that best explain our measurements, we could now try to maximize this function, or equivalently the log-likelihood function

$$\ell(\theta \mid x_1, \ldots, x_n, y_1, \ldots, y_n) = \sum_{i=1}^{n} \left(y\theta^T x - \ln(y_i!) - \left(e^{\theta^T x_i}\right)\right)$$

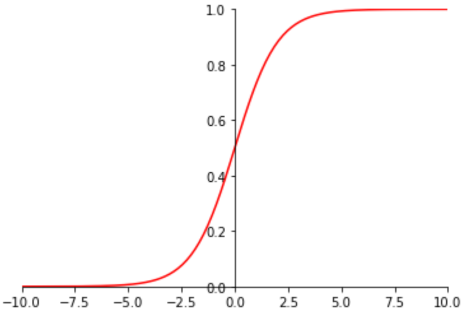
This is a concave maximization problem that one can solve numerically.

Example 5.10.6 (Logistic Regression)**.** For binary classification tasks, i.e. "Yes/No" prediction, one often uses generalized linear models. For example one tries to predict whether a patient has a certain illness based on chemical values of a blood sample. In this case one can use a general linear model with inverse link function any monotonically increasing $g^{-1} \colon \mathbb{R} \to \mathbb{R}$ that is close to 0 for low values and close to 1 for high values.

Any cumulative distribution function would do, but probably the most common is the *logistic function* or *inverse logit function* defined by

$$g^{-1}(x) := \frac{1}{1 + e^{-x}} = \frac{e^x}{1 + e^x}$$

(both of the above forms seem to appear equally often in the literature). Here is the graph of this function:



The probability mass function of a $\{0,1\}$-valued random variable Y that we are trying to understand as depending on an $\mathbb{R}^n$-valued random variable X in a logistic model with parameters $\theta \in \mathbb{R}^n$ is given by:

$$P(Y=1 \mid x,\theta) = f(y \mid x,\theta) := \frac{1}{1+e^{-\theta^T x}}$$

$$P(Y=0 \mid x,\theta) = 1 - f(y \mid x,\theta) = 1 - \frac{1}{1+e^{-\theta^T x}}$$

For observations $y_1,\ldots,y_n$ and $x_1,\ldots,x_n$ we thus get the likelihood function

$$L(\theta \mid y_1,\ldots,y_n,_1,\ldots,x_n) = \prod_{i=1}^{n} f(y_i \mid x_i,\theta)^{y_i} \cdot (1-f(y_i \mid x_i,\theta))^{1-y_i}$$

Now one can fit the parameter θ, using maximum likelihood estimation or Bayesian updating.

Logistic regression is very versatile and extremely important in Machine Learning. A single neuron, as they occur in a typical neural network, behaves like a random variable modelled by a logistic regression model: There is a linear part and then a link function, in this context also called *activation function*. See here for an elaboration of this.

In Example 5.10.6 the explanatory variables were $\mathbb{R}$-valued, but in general one can also have categorical variables or a mixture of both. The solution is here to simply see a categorical random variable with n possible values as $n-1$ random variables taking values 0 or 1 – i.e. real valued random variables – and use them as in Example 5.10.6. We have seen this for linear regression in a previous exercise.

A standard task of this kind, i.e. with categorical random variables, is to judge whether an email is spam, based on the occurrence of certain words in the subject line or the IP address of its sender occurring in a list of marked addresses: The answers here are yes/no questions, like "Is the IP address on the spam list?".

6 Stochastic Processes

A good source for the basics on Markov chains is the book Rozanov: Probability Theory: A concise course already mentioned as a source for basic probability theory. In general Markov chains are a part of many first probability theory courses, so your favourite book might treat them as well.

For probability theory in general, stochastic processes in general, Markov chains and continuous time stochastic processes a great source is: Grimmett/Stirzaker, Probability and Random processes

6.1 Stochastic processes: Terminology

Definition 6.1.1. *A* stochastic process *is a collection* $\mathcal{X} = \{X_t\}_{t\in\mathcal{T}}$ *of random variables, all taking values in the same set* $\mathcal{S}$ *(called the* state space*).*

Often $\mathcal{T}$ will be a linearly ordered set. We then can think of $\mathcal{T}$ as a set of times at which we evaluate the state of a system: The state of the system at time $t \in \mathcal{T}$ is given by X_t - this explains the word "process" in "stochastic process". For example one can think of taking blood samples of some patient every day and measuring the quantity of iron, hemoglobin, and other substances ($\mathcal{S} = \mathbb{R}^n$). Or of observing the state of the weather at a fixed place every day ($\mathcal{S} = \{sunny, cloudy, rainy\}$). Or of counting the number of of packets arriving at an internet router ($\mathcal{S} = \mathbb{N}$).

The most common examples for the index set $\mathcal{T}$ are $\mathbb{Z} = \{\ldots, -2, -1, 0, 1, 2, \ldots\}$, $\mathbb{N} = \{0, 1, 2, \ldots\}$, intervals $[0, T] \subseteq \mathbb{R}$, $[0, \infty)$ or $(-\infty, \infty)$.

If $\mathcal{T}$ is countable, one says that $\mathcal{X}$ is a *discrete-time stochastic process*, or a *time series*. When $\mathcal{T}$ is an interval (possibly infinite), one says that $\mathcal{X}$ is a *continuous-time stochastic process*.

We are interested in what we can say about "later" states given our knowledge about earlier states, or vice versa. For example, for a discrete-time stochastic process $\mathcal{X} = \{X_t\}_{t\in\mathbb{N}}$ we are interested in the conditional probabilities $P(X_{n+1} = s \mid X_n = s_n, X_{n-1} = s_{n-1}, \ldots, X_0 = s_0)$.

6.2 Markov Chains

In the previous chapters we have been talking a lot about repetitions of the same experiment, where previous instances did not influence the outcome of the next instance. These were modelled by i.i.d. random variables; in this case we would have $P(X_{n+1} = s \mid X_n = s_n, X_{n-1} = s_{n-1}, \ldots, X_0 = s_0) = P(X_{n+1} = s)$.

The next best situation is when we do not need to remember the whole history of previous states, but only need to know the present state, in order to guess the next state. This is captured by the following definition:

Definition 6.2.1. *A* Markov chain *is a discrete-time stochastic process $\mathcal{X} = \{X_t\}_{t\in\mathbb{N}}$ such that*

$$P(X_{n+1} = s \mid X_n = s_n, X_{n-1} = s_{n-1}, \ldots, X_0 = s_0) = P(X_{n+1} = s \mid X_n = s_n)$$

for all s, s_i, n. This property is called the Markov property.

The conditional probabilities $P(X_{n+1} = s \mid X_n = s_n)$ are called transition probabilities.

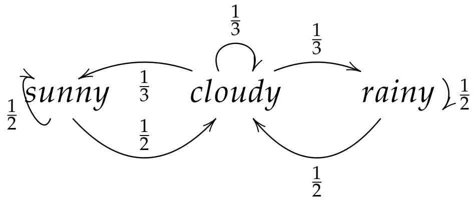
A Markov chain is called time homogeneous *if the transition probabilities $P(X_{n+1} = s \mid X_n = w)$ are independent of n, i.e. if $P(X_{n+1} = s \mid X_n = w) = P(X_1 = s \mid X_0 = w)$ for all n.*

Example 6.2.2. Consider hourly weather measurements taking the possible values $\mathcal{S} = \{sunny, cloudy, rainy\}$. Assume that

- if it is *sunny*, then it will stay *sunny* with probability $\frac{1}{2}$ and become *cloudy* with probability $\frac{1}{2}$
- if it is *cloudy*, then it will stay *cloudy* with probability $\frac{1}{3}$ become *sunny* with probability $\frac{1}{3}$, and become *rainy* with probability $\frac{1}{3}$
- if it is *rainy*, then it will stay *rainy* with probability $\frac{1}{2}$ and become *cloudy* with probability $\frac{1}{2}$

It is reasonable to postulate that past weather conditions do no longer influence the future weather, only the present weather does. Thus this defines a time homogeneous Markov chain - once we choose a starting distribution for the weather state on day zero (just from the transition probabilities we don't get a probability distribution for X_n, in particular not for X_0).

We may record the transition probabilities in a *transition diagram*



Definition 6.2.3. *Let $\mathcal{X}$ be a Markov chain with countable state space $\mathcal{S}$. The nth* transition matrix *of $\mathcal{X}$ is the $|\mathcal{S}| \times |\mathcal{S}|$-matrix $P(n) = (p_{ij}(n))$ with entries $p_{ij}(n) := P(X_{n+1} = j \mid X_n = i)$*

If $\mathcal{X}$ is time homogeneous, then this matrix is independent of n and is simply called the transition matrix of $\mathcal{X}$.

Example 6.2.4. In the weather example Ex. 6.2.2 we have a time homogeneous Markov chain with 3 states. Hence we obtain a 3×3-matrix, namely

$$\begin{pmatrix} \frac{1}{2} & \frac{1}{2} & 0 \\ \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \\ 0 & \frac{1}{2} & \frac{1}{2} \end{pmatrix}$$

Remark 6.2.5. The transition matrix of a homogeneous Markov chain has the property the entries of each row sum to 1, reflecting the fact that some state has to occur at the next time. Such a matrix is called a *stochastic matrix*.

Definition 6.2.3 does also make sense if the state space is countably infinite. In this case we obtain an infinite matrix.

Example 6.2.6. Consider a particle which moves randomly through the integers $\{\ldots, -2, -1, 0, 1, 2, \ldots\}$. If the particle is at position i, it jumps in the next step either to position $i+1$ or to position $i-1$ with probabilities p and $q = 1 - p$ repectively. Let X_n be the position of the particle after the nth step.

Then $\{X_n\}$ is a Markov chain with transition probabilities

$$p_{ij} = \begin{cases} p & \text{if } j = i+1 \\ q & \text{if } j = i-1 \\ 0 & \text{otherwise} \end{cases}$$

The transition matrix has zeros on the central diagonal, ps on the next diagonal to the right and qs on the next diagonal to the left, zeros otherwise

$$\begin{pmatrix} \ddots & \ddots & & & \\ \ddots & 0 & p & 0 & \\ & q & 0 & p & \\ & 0 & q & 0 & \ddots \\ & & & \ddots & \ddots \end{pmatrix}$$

Example 6.2.6 is an example of a so-called *random walk*, in this case a walk on the line of integers. More generally one can consider discrete random walks on $\mathbb{Z}^n$, with $2n$ possible directions to go to. Even more generally one can consider continuous time stochastic processes where the random variables can take arbitrary values in $\mathbb{R}^n$ – these are also called random walks. In a different direction, one can consider random walks on $\mathbb{N}$ or similarly truncated regions of the above. An example for this is the Gambler's Ruin random walk: Imagine a gambler repating the same game over and over, with fixed winning chance p. One can set up a Markov chain $\{X_t\}_{t\in\mathbb{N}}$ where X_t counts the money in the gambler's pocket – since the casino is unwilling to give any credit, this amount cannot go below zero. One can now for example caculate the expected time until $X_t = 0$ happens (gambler's ruin, which ends the game in practice). This may give a first hint of how more sophisticated stochastic processes play a role in finance.

Proposition 6.2.7. *[Chapman-Kolmogorov equation] Let $\{X_n\}$ be a Markov chain with countable state space $\mathcal{S}$ and $l < m < n$. Then*

$$P(X_n = j \mid X_l = i) = \sum_{k\in\mathcal{S}} P(X_n = j \mid X_m = k) P(X_m = k \mid X_l = i).$$

Proof. We have

$$\begin{aligned} P(X_n = j, X_l = i) &= \sum_{k\in\mathcal{S}} P(X_n = j, X_m = k, X_l = i) \\ &= \sum_{k\in\mathcal{S}} P(X_m = k, X_l = i) P(X_n = j \mid X_m = k, X_l = i) \\ &= \sum_{k\in\mathcal{S}} P(X_m = k, X_l = i) P(X_n = j \mid X_m = k) \end{aligned}$$

where the last step uses the Markov property. Now dividing by $P(X_l = i)$ we arrive at the claim. □

The previous proof in words: For every state there are $|\mathcal{S}|$ possibilities of getting from state i to state j in $n-l$ steps, namely by going from state i to state k in $m-l$ steps, and then from state k to state j (for any $k \in \mathcal{S}$). These are all possibilities and they are mutually exclusive, which explains the equations.

Corollary 6.2.8. *Let $\mathcal{X}$ be a Markov chain with state space $\{1,\dots,r\}$ transition matrices $P(n) = (p_{ij}(n))$. The probability of reaching state j from state i in m steps, i.e. $P(X_{n+m} = j \mid X_n = i)$, is the entry at the place (i,j) of the matrix $P(n) \cdot P(n+1) \cdot \ldots \cdot P(n+m-1)$.*

In particular, if $\mathcal{X}$ is a time homogeneous Markov chain with transition matrix P, then the said probability is the entry at the place (i,j) in the m-fold power P^m of the matrix P.

Proof. We will check the case $m = 2$. The general case follows by induction. Abbreviate $p_{ij}(n) = P(X_{n+1} = j \mid X_n = i)$. Then

$$\begin{aligned} P(X_{n+2} = j \mid X_n = i) &= \sum_{k \in \mathcal{S}} P(X_{n+2} = j \mid X_{n+1} = k) P(X_{n+1} = k \mid X_n = i) \\ &= \sum_{k=1}^{r} p_{ik}(n) p_{kj}(n+1) \end{aligned}$$

But this is exactly the entry at the place (i,j) of the matrix $P(n)P(n+1)$. □

Note the order of the matrix multiplication: The earlier times are *left* of the later times.

Corollary 6.2.9. *Powers of a stochastic matrix are stochastic matrices again*

Proof. They are transition matrices of a Markov chain.

An alternative proof, not relying on Cor. 6.2.8, goes as follows: □

Example 6.2.10. For simplicity we consider a time homogeneous Markov chain with transition matrix P. By definition, the probabilities of going to the next state from initial state i are written in the ith row of P. If we denote by e_i the row vector with a 1 in the ith place and 0s otherwise, then the matrix product $e_i \cdot P$ gives us back the ith row.

More generally, if we have state space $\mathcal{S} = \{1,\dots,n\}$ and probabilities for a starting position given by $p = (p_1,\dots,p_n)$, then the probabilities for ending up in the states $1,\dots,n$ are given by the entries of the row vector $p \cdot P$.

The fact that we are sending in our initial distribution vectors from the left fits to the order of multiplication observed in Corollary 6.2.8. Since this order of multiplication is unusual, I repeat:

The state probability vector is multiplied *from the left* with the transition matrix, to obtain the state probabilities one step later!

Remark 6.2.11. The distribution of the random variables in a time homogeneous Markov chain $(X_n)_{n\in\mathbb{N}}$ is completely determined by an initial state (i.e. the distribution of X_0) and the transition matrix P: One can reconstruct the distribution of X_{n+1} recursively via $P(X_{n+1}=s) = \sum_{s_0,\dots s_n\in\mathcal{S}} P(X_{n+1} = s \mid X_n = s_n)\cdot P(X_n = s_n \mid X_{n-1} = s_{n-1})\cdot\ldots\cdot P(X_1 = s_1 \mid X_0 = s_0)P(X_0 = s_0)$.

Therefore one often presents a Markov chain just by giving an initial state and a transition matrix.

6.3 Recurrence and Transience

One often considers the long term behaviour of Markov chains: Will a given state be eventually reached? How long will it take, if one starts at some state, to come back to it? Are there states that are attractive and others that are repellant, as the process evolves? How many times can we expect to visit a given state?

Let $\mathcal{X} = \{X_t\}_{t\in\mathbb{N}}$ be a time homogeneous Markov chain with transition matrix P. We assume that at time 0 we are in state i.

We fix the following notation: Let $p_{ij}^{(n)} := P(X_{m+n} = j \mid X_m = i)$ denote the probability of getting from state i to state j in n steps. Thus these numbers are the entries of the nth power of the transition matrix: $P^n = (p_{ij}^{(n)})$.

For the following discussion we fix a state i, so that we can suppress it from the notation subsequently. Let $u_n := p_{ii}^{(n)}$ be the probability of returning from state i to itself after n steps.

Let further v_n be the probability of returning to state i after n steps *for the first time*, i.e. $v_n = P(X_n = i, X_{n-1} \neq i, \dots, X_1 \neq i \mid X_0 = i)$.

Then, setting $v_0 := 0$ and $u_0 := 1$, we have the equation

$$u_n = u_0 v_n + u_1 v_{n-1} + \dots u_{n-1} v_1 + u_n v_0,$$

reflecting the fact that for returning to i after n steps we have the mutually exclusive possibilities of either returning to i for the first time after n steps, or returning to i for the first time after $n-1$ steps and then staying at i, or returning to i for the first time after $n-2$ steps and then coming back in 2 steps etc.

We take the numbers u_n, v_n as coefficients in power series:

$$U(z) := \sum_{n=0}^{\infty} u_n z^n, \qquad V(z) := \sum_{n=0}^{\infty} v_n z^n$$

Since all the u_n, v_n, being probabilities, are at most 1, these power series actually converge for $|z| < 1$: The geometric series $\Sigma_{n=0}^{\infty} z^n$ then provides a convergent upper bound. But we do not use these power series for representing functions, rather we use them as so-called generating functions, i.e. a slick way of storing the coefficients.

Lemma 6.3.1. *We have* $U(z) = \frac{1}{1-V(z)}$.

Proof. By the rules of multiplication of power series we have

$$\begin{aligned} U(z)V(z) &= \Big(\sum_{i=0}^{\infty} u_n z^n\Big)\Big(\sum_{j=0}^{\infty} v_n z^n\Big) \\ &= \sum_{n=0}^{\infty} \Big(\sum_{i+j=n} u_i v_j\Big) z^n \\ &= \sum_{n=0}^{\infty} \big(u_0 v_n + u_1 v_{n-1} + \ldots u_{n-1} v_1 + u_n v_0\big) z^n \\ &= \Big(\sum_{n=0}^{\infty} u_n z^n\Big) - u_0 = U(z) - 1 \end{aligned}$$

Here we have to subtract u_0 because in the case $n = 0$ the coefficient of z^0 is $u_0 v_0 = 0$. For all $n \neq 0$ the coefficient is equal to u_n, as we have seen above.

Now dividing this equation by $U(z)$, then adding $+\frac{1}{U(z)}$, adding $-V(z)$ and applying $(\ )^{-1}$ yields the claim. □

Definition 6.3.2. *The state i is called* recurrent *if one returns to i with probability* 1, *i.e. if* $\sum_{n=0}^{\infty} v_n = 1$. *Otherwise it is called* transient.

Remark 6.3.3. A drunk person always finds home, but a drunk bird may be lost forever. Except a chicken.

Theorem 6.3.4. *The state i is recurrent if and only if $\sum_{n=0}^{\infty} p_{ii}^{(n)} = \infty$.*

Proof. We go on denoting $p_{ii}^{(n)}$ by u_n. We have

$$\sum_{n=0}^{\infty} u_n = \lim_{z\to 1} U(z) = \lim_{z\to 1} \frac{1}{1-V(z)} = \frac{1}{1-\sum_{n=0}^{\infty} v_n}.$$

Clearly, the first term diverges if and only if $\sum_{n=0}^{\infty} v_n = \lim_{z\to 1} V(z) = 1$ (note that, being a probability, this term is always ≤ 1). □

Example 6.3.5. Consider a Markov chain with transition matrix

$$P = \begin{pmatrix} \frac{1}{2} & \frac{1}{4} & \frac{1}{4} \\ 0 & \frac{1}{2} & \frac{1}{2} \\ 0 & 0 & 1 \end{pmatrix}$$

Since P is upper triangular we can read off the $p_{ii}^{(n)}$ immediately: $p_{11}^{(n)} = p_{22}^{(n)} = \frac{1}{2^n}$ and $p_{33}^{(n)} = 1$. Since $\sum_{n=0}^{\infty} \frac{1}{2^n} = 2 < \infty$, the states 1 and 2 are transient. Clearly state 3 is recurrent.

Remark. For a state i, we can modify the given Markov chain (X_n) by setting the initial distribution to give state i with probability 1. That is, we can define a new Markov chain $(\tilde{X}_n)$ by setting $P(\tilde{X}_n = j) := P(X_n = j | X_0 = i)$.

Now define the random variable Z_n to be 1 if $\tilde{X}_n = i$, and 0 otherwise. Then $Z := \sum_{n=0}^{\infty} Z_n$ is the number of times that the new process passes through state i (during the entire, infinite time). Z can either be viewed as an $\mathbb{N} \cup \{\infty\}$-valued random variable, or you can instead consider the partial sums (measuring the number of visits to state i until time k), as usual with infinite series.

The expectation of Z is

$$EZ = \sum_{n=0}^{\infty} Z_n \cdot P(Z_n = 1) = \sum_{n=0}^{\infty} EZ_n = \sum_{n=0}^{\infty} P(X_n = j | X_0 = i) = \sum_{n=0}^{\infty} p_{ii}^{(n)}.$$

Thus Thm. 6.3.4 says that a state is recurrent, if and only if the expected number of times that the process passes returns to it, is infinite.

While we defined recurrence and transience via the probability of returning from a state to itself, one can also recognize transience and recurrence when starting from another state.

Proposition 6.3.6. *Let i be a state from which one can reach the state j in some number of steps. Then the state j is recurrent if and only if $\sum_{n=0}^{\infty} p_{ij}^{(n)} = \infty$, and transient if and only if $\sum_{n=0}^{\infty} p_{ij}^{(n)} < \infty$.*

Proof. Define $f_{ij}^{(n)} := P(X_n = i, X_{n-1} \neq i, \ldots, X_1 \neq i \mid X_0 = j)$ to be the probability of reaching state j for the first time after n steps, starting from state i. Thus $v_n = f_{ii}^{(n)}$ in our previous notation.

We clearly have $p_{ij}^{(n)} = \sum_{m=0}^{n} f_{ij}^{(m)} p_{jj}^{(n-m)}$, and therefore

$$\begin{aligned}
\sum_{n=1}^{\infty} p_{ij}^{(n)} &= \sum_{n=1}^{\infty} \sum_{m=0}^{n} f_{ij}^{(m)} p_{jj}^{(n-m)} \\
&= f_{ij}^{(1)} p_{jj}^{(0)} \\
&+ f_{ij}^{(1)} p_{jj}^{(1)} + f_{ij}^{(2)} p_{jj}^{(0)} \\
&+ f_{ij}^{(1)} p_{jj}^{(2)} + f_{ij}^{(2)} p_{jj}^{(1)} + f_{ij}^{(3)} p_{jj}^{(0)} \\
&+ \qquad \ldots \\
&= \left(\sum_{n=1}^{\infty} f_{ij}^{(n)} \right) \left(\sum_{m=0}^{\infty} p_{jj}^{(m)} \right)
\end{aligned}$$

Since j is reachable from i, the left factor is non-zero. Therefore, $\sum_{n=1}^{\infty} p_{ij}^{(n)}$ diverges if and only if $\sum_{m=0}^{\infty} p_{jj}^{(m)}$ diverges, which by Thm. 6.3.4 is the case if and only if j is recurrent.

Clearly, the statement about transience is equivalent to this. □

Corollary 6.3.7. *If from a state i one can reach a state j (in any number of steps), from which it is not possible go back to i, then i is transient.*

Proof. That it is impossible to go from j to i means that $p_{ji}^{(n)} = 0$ for all n. Then i is transient by Prop. 6.3.6 □

Corollary 6.3.8. *Every Markov chain with a finite set of states has at least one recurrent state.*

Proof. Suppose that every state is transient. Then for every pair of states i, j we have $\sum_{n=0}^{\infty} p_{ij}^{(n)} < \infty$: Either j is not reachable from i, then $p_{ij}^{(n)} = 0$ for all n, or j is reachable from i, then this is true by Prop. 6.3.6. Hence $\lim_{n\to\infty} p_{ij}^{(n)} = 0$.

Denote the set of states by S. For every state i and every $n \in \mathbb{N}$ we have $\sum_{j\in S} p_{ij}^{(n)} = 1$, because this is the probability of going anywhere at all in n steps. Hence $1 = \lim_{n\to\infty} 1 = \lim_{n\to\infty} \left(\sum_{j\in S} p_{ij}^{(n)}\right) = \sum_{j\in S} \lim_{n\to\infty} p_{ij}^{(n)} = \sum_{j\in S} 0 = 0$ which is a contradiction.

Here we used the finiteness of S for exchanging the limit and the sum over all states in S. □

Remark 6.3.9. By the remark after Example 6.3.5 a state is recurrent if and only if the process is expected to return to it infinitely many times. This should makes Cor. 6.3.8 very plausible, since in a Markov chain with finitely many states some state necessarily needs to be visited infinitely often.

Theorem 6.3.10. *If a state j is accessible from a recurrent state i, then i is also accessible from j, and j is also recurrent.*

Proof. Suppose the state j is accessible from the recurrent state i, in M steps. This means that $p_{ij}^{(M)} = \alpha > 0$.

Suppose that from j one cannot get to i. The probability of eventually returning from i to i, given that no return happens in the first M steps, decomposes as the probability of going to j in the first M steps ($= \alpha$) and then to i plus the probability of not going to j in the first M steps ($= 1 - \alpha$) and then going to i. The first summand is zero, since one cannot get from j to i, hence the probability is at most $1 - \alpha$, contradicting that i is recurrent.

Hence i must be accessible from j, i.e. there is an N such that $p_{ji}^{N} = \beta > 0$. Now observe that $p_{jj}^{(n+N+M)} \geq p_{ji}^{(N)} p_{ii}^{(n)} p_{ij}^{(M)} = \alpha\beta p_{ii}^{(n)}$ for every $n \in \mathbb{N}$. Therefore $\sum_{k=0}^{\infty} p_{jj}^{(k)} \geq \sum_{n=0}^{\infty} p_{ii}^{(n)} = \infty$. By Theorem 6.3.4 the state j is recurrent. □

Definition 6.3.11. *Let $\mathcal{X}$ be a time homogeneous Markov chain with transition matrix P and state space $\mathcal{S} = \{1, \ldots, n\}$. A probability distribution $\pi = (p_1, \ldots, p_n)$, giving the probabilities to start in the initial state $1, \ldots, n$, is called* invariant distribution *or* stationary *or* stable distribution, *if $\pi P = \pi$*

A stationary distribution is an eigenvector of P for the eigenvalue 1. Note that here we mean a row vector and the eigenvector property is understood via multiplication with P from the right. Of course eigenvectors are not unique but rather form an infinite eigenspace. However, if we ask for a probability distribution on the states, i.e. an eigenvector with positive entries and ℓ_1-norm 1, it often happens to become unique.

Conditions for the uniqueness and existence of an invariant initial distribution are given by the Perron-Frobenius theorem below.

Definition 6.3.12. *A Markov chain for which there exists a number n, such that from each state one can pass to each other state in exactly n steps, is called* regular.

To rephrase the condition of the theorem, note that a Markov chain is regular if and only if in some power P^n of the transition matrix P all entries are positive. This is how one often verifies the condition in practice.

Theorem 6.3.13 (Perron-Frobenius Theorem). *For a regular Markov chain with transition matrix P, the sequence of powers P^n converges to a matrix P^∞. There is then a unique stationary distribution, given by πP^∞, where π can be* any *initial distribution.*

About the proof: We will not show the convergence of the sequence P^n. For that see e.g. Norris, Markov Chains, Thm. 1.8.3.

We only check that if the limit P^∞ exists, then indeed πP^∞ is a stationary distribution. For this first note that P^n is a stochastic matrix for all n. Indeed, that a matrix P is stochastic means precisely that $Pe = e$, where e is the column vector with every entry 1. Clearly if this is true for P, then it is true for P^n, and hence it is true in the limit (since multiplication with e is a continuous map, and hence commutes with the limit).

Similarly, the sum of entries of the row vector πP is 1, since it is again a probability distribution on the states. By induction this is then true for every πP^n. Since multiplying with π from the right and summing the entries is a continuous operation, we also get that πP^∞ is a probability distribution on $\mathcal{S}$. Now it is enough to observe

$$(\pi P^\infty)P = \pi(\lim_{n\to\infty} P^n)P = \pi(\lim_{n\to\infty} P^{n+1}) = \pi P^\infty$$

(using that multiplying with P from the right is continuous), showing that πP^∞ is stationary. For the uniqueness again see the above reference.

♡

For regular Markov chains, Theorem 6.3.13 says that no matter how we start, we will approach the invariant distribution over time.

The existence of the limit in Theorem 6.3.13 is guaranteed by the condition that all entries are positive. This is for example violated for the stochastic matrix

$$A = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

The sequence A^n just jumps between A and the identity matrix and thus does not converge.

Also the uniqueness of a stable distribution relies on the positivity of the transition matrix. The matrix

$$A = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

for example has the stable distributions $(1, 0)$ and $(0, 1)$.

Remark 6.3.14. As remarked in the proof of Theorem 6.3.13, a matrix P is stochastic if and only if $Pe = e$, where e is the column vector with every entry 1. Thus a stochastic matrix has eigenvalue 1. Since a matrix and its transpose have the same eigenvalues, P^T also has eigenvalue 1. This means that there exists a row vector π with $P^T \pi^T = \pi^T$, i.e. $\pi P = \pi$. So this part of Theorem 6.3.13 is easy. It is not that easy, however, that π can be chosen with all entries positive (and thus is a probability distribution), and that there is a *unique* such eigenvector.

In general there is no power of the transition matrix with only positive entries. One can, however, always look for subsets of states from which one can not exit anymore. If one reorders the states so that all states in such a subset are next to each other, then the transition matrix becomes a block matrix. Powers of the transition matrix are again block matrices. One can then ask whether the block in question has only positive entries for some power – in this case these are recurrent and there is a stationary distribution which only positive probabilities only for these states.

Proposition 6.3.15 (formerly known as Prop. 6.3.12)**.** *Suppose that for some power of the stochastic matrix P all the entries are positive. Then in the limit matrix P^∞, all rows are equal to the stationary distribution. In particular the limits of the n-step return probabilities $\lim_{n\to\infty} p_{ii}(n)$ are the entries of the invariant distribution vector.*

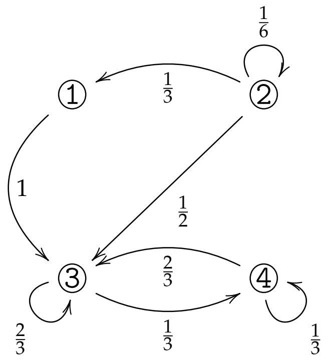
Proof. By Thm. 6.3.13, the stationary distribution can be obtained as $\pi = \tau P^{\infty}$ for any vector τ. Taking $\tau = (1,0,0,0),(0,1,0,0),(0,0,1,0),\ldots$, the expression τP^{∞} gives back the 1st, 2nd, 3rd ... row of P^{∞}, and they all must be equal to π.

On the diagonal of P^n we have the n-step return probabilities $p_{ii}(n)$. Thus, taking the limit, on the diagonal of P^{∞} we find $\lim_{n\to\infty} p_{ii}(n)$. □

Corollary 6.3.16. *Transient states have probability* 0 *in any stationary distribution.*

Proof. By Prop. 6.3.15 the probability of state i of the stationary distribution is $\lim_{n\to\infty} p_{ii}^{(n)}$ □

Example 6.3.17. Consider the Markov chain given by the following transition diagram:



We will determine which states are recurrent and which transient, and we will find all stationary distributions. We will do this in two ways, showing different possibilities of argumentation.

First, the transition matrix is

$$\begin{pmatrix} 0 & 0 & 1 & 0 \\ \frac{1}{3} & \frac{1}{6} & \frac{1}{2} & 0 \\ 0 & 0 & \frac{2}{3} & \frac{1}{3} \\ 0 & 0 & \frac{2}{3} & \frac{1}{3} \end{pmatrix}$$

First solution:

Just from looking at the diagram, it is clear that it is impossible to go from states ③ and ④ to states ① or ②, in any number of steps. Alternatively one can see that the transition matrix is a block matrix with a lower left block of zeros. Any power of this matrix will also have a lower left block of zeros, hence it is impossible in any number of steps to go from states ③ and ④ to states ① or ②, since the corresponding transition probabilities are the entries of that lower left block.

On the other hand, it is possible to go from states ① or ② to states ③ and ④. Again, this can either be seen directly from the diagram, or one can notice that in the second power of the transition matrix we have positive entries in the upper right corner. Hence by Cor. 6.3.7 states ① and ② are transient.

By Cor. 6.3.8 the Markov chain has a recurrent state, so that state must be among states ③ and ④. Since these states are mutually accessible from each other, by Thm. 6.3.10 both must be recurrent. Thus we have classified all states.

To find the stationary distribution, we can simply solve the equation that it has to satisfy. Since by Cor. 6.3.16 the transient states have probability zero in the stationary distribution, it suffices to find the probabilities for states ③ and ④. For this we consider the submatrix describing just the transition probabilities between those states (here: the lower right block). We need to solve the following equation:

$$(x,y)\begin{pmatrix} \frac{2}{3} & \frac{1}{3} \\ \frac{2}{3} & \frac{1}{3} \end{pmatrix} = (x,y)$$

This amounts to the two equations $\frac{2}{3}x + \frac{2}{3}y = x$ and $\frac{1}{3}x + \frac{1}{3}y = y$. We know that 1 is an eigenvalue of the (transpose of the) transition matrix, hence there is a solution and one of the equations is redundant. We solve e.g. the second one, obtaining $x = 2y$. The unique solution (x,y) that is a probability distribution (i.e. satisfies $x + y = 1$ and has non-negative entries) is $x = \frac{2}{3}$, $y = \frac{1}{3}$.

Second solution: Reordering the states by switching the positions of ① and ②, results of course in the same Markov chain, but in the new transition matrix

$$\begin{pmatrix} \frac{1}{6} & \frac{1}{3} & \frac{1}{2} & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & \frac{2}{3} & \frac{1}{3} \\ 0 & 0 & \frac{2}{3} & \frac{1}{3} \end{pmatrix}$$

obtained by switching the first two rows and columns. Now from the triangular block form it is immediately visible that $p_{11}^{(n)} = 0$ for all n: this is the entry 0 in the second row and second column, and it is clearly 0 in all powers of that matrix as well. Thus $\sum_{n=0}^{\infty} p_{11}^{(n)} = 0$ and by Thm. 6.3.4 state ① is transient.

Likewise, the upper left entry of the n-th power this reordered matrix is $p_{22}^{(n)} = \left(\frac{1}{6}\right)^n$. Hence $\sum_{n=0}^{\infty} p_{22}^{(n)} = \sum_{n=0}^{\infty} \left(\frac{1}{6}\right)^n < \infty$ (a geometric series $\sum_{n=0}^{\infty} a^n$ converges if $|a| < 1$), so state ② is transient.

For the other two states ③ and ④ consider the submatrix describing their transition probabilities:

$$A := \begin{pmatrix} \frac{2}{3} & \frac{1}{3} \\ \frac{2}{3} & \frac{1}{3} \end{pmatrix}$$

This matric has only positive entries, hence the sub-Markov chain given by states ③ and ④ is regular, and we are in the situation of Thm. 6.3.13. One sees that $A^2 = A$, and hence $A^n = A$ for all n, which means $A = A^{\infty}$. By Prop. 6.3.15, the rows are the stationary distribution, so we have probability $\frac{2}{3} = \lim_{n\to\infty} p_{33}^{(n)}$ for state ③ and probability $\frac{1}{3} = \lim_{n\to\infty} p_{44}^{(n)}$ for state ④.

Famous uses of Markov chains include text generation (actually Markov invented Markov chains for analyzing literature!), and Google's Page rank algorithm: The Google founders' breakthrough idea was to rank the web pages turned up in an internet search using Markov chains. For this, one sees each webpage as a state, and assigns a positive transition probability from page A to page B if on page A there is a link to page B (in the absence of better information, just give each link on a page the same probability). Then one can compute the stable state of this Markov chain and rank pages accoring to their probability in this stable state. The pages that are likelier to be visited by someone arbitrarily following links get a higher rank in this way. Of course this basic idea has been refined, but it remains a good approach.

We will encounter Markov chains as building blocks of Hidden Markov models and Markov Chain Monte Carlo methods.

6.4 Hidden Markov models

A hidden Markov model (HMM) consists of two sequences of random variables $(X_n)_{n\in\mathbb{N}}$ and $(Y_n)_{n\in\mathbb{N}}$, that have to satisfy certain conditions. It is often depicted as follows:

$$\begin{tikzcd}
X_1 \arrow[r, bend left] \arrow[d] & X_2 \arrow[r, bend left] \arrow[d] & X_3 \arrow[r, bend left] \arrow[d] & X_4 \arrow[r, bend left] \arrow[d] & \dots \\
Y_1 & Y_2 & Y_3 & Y_4 &
\end{tikzcd}$$

Do not confuse this drawing with a transition diagram as for Markov chains! Instead, this diagram summarizes the dependencies between the different random variables: X_{n+1} and Y_n depend on X_n, and if one knows X_n, then X_{n+1} and Y_n depend on no other of the random variables. The X_i are supposed to form a Markov chain, and the states of this Markov chain are called *hidden states*. The Y_i should all take values in the same set, called the set of *observable states*.

The basic idea is that with this setup one can model processes (X_n) that one can not directly observe, but which influence some observable phenomenon reflected in (Y_n). Imagine driving a car and X_n is the state of a screw in the motor at time n, either "firm", "loose" or "missing". From one point in time to the next, a firm screw goes loose with probability $\frac{1}{100}$ and otherwise stays firm, and a loose screw stays in place, but still loose, with probability $\frac{1}{23}$ and goes missing otherwise. A missing screw will never jump back into place. This decribes a Markov chain as you know it.

Now you cannot watch the motor while you are driving, but you can hear the sound it produces, either "steady humming", "rhythmic rattling" or "irregular clunking". These would be the possible values of the observable states, influenced by the state of the screw, maybe not deterministically but rather according to some probability table (say for a loose screw we might hear humming, rattling and clunking with probabilities $\frac{1}{3}, \frac{1}{2}, \frac{1}{6}$ respectively). There would be similar probabilities for the other states of the screw. Now you would like to infer from what you are hearing, whether the motor is ok, or whether you should stop and check the screws.

As the example illustrates, this setup is all about transition probabilities, which is reflected in the following formal definition of a Hidden Markov Model.

Definition 6.4.1. *A* hidden Markov model *(HMM) consists of the following ingredients:*

1. *Two sets of states, $\mathcal{S}$ (the* hidden states*) and $\mathcal{O}$ (the* the observable states*)*

2. *A probability distribution π on $\mathcal{S}$, called the* initial distribution*)*

3. *A stochastic $|\mathcal{S}| \times |\mathcal{S}|$-matrix $P = (p_{ss'})$, called the* transition matrix

4. *A stochastic $|\mathcal{S}| \times |\mathcal{O}|$-matrix $B = (b_{so})_{s\in\mathcal{S},o\in\mathcal{O}}$, called the* emission matrix

The initial distribution and the transition matrix of a hidden Markov model together determine a time homogeneous Markov chain $(X_n)_{n\in\mathbb{N}}$ of $\mathcal{S}$-valued random variables as in Remark 6.2.11. The X_n are called the *hidden variables*, or *latent variables*.

Furthermore, the emission matrix, together with the random variable X_n determines an $\mathcal{O}$-valued random variable Y_n by setting $P(Y_n = o \mid X_n = s) := b_{so}$. One can then calculate the distribution by $P(Y_n = o) = \sum_{s\in\mathcal{S}} P(Y_n = o \mid X_n = s)P(X_n = s)$.

Thus one could equivalently define a hidden Markov model to consist of two sequences of random variables, $\{X_n\}_{n\in\mathbb{N}}$ and $\{Y_n\}_{n\in\mathbb{N}}$, that are appropriately connected by the given conditional probabilities.

Example 6.4.2. Hidden Markov models are used in DNA sequencing. A DNA string is a sequence of nucleotides abbreviated A, C, G, T. It has parts which are not used for protein building, called *introns*, and parts which are used, called *exons*. While one can nowadays easily observe the sequence of nucleotides, it is not always clear which parts are introns and which are exons.

One can describe this situation by a hidden Markov model in which the observable random variables Y_n are taking values A, C, G, T and the hidden random variables X_n take values I, E (for intron and exon).

Suppose it has been observed that in the intron parts more Cs and Gs occur and in the exon parts more As occur – a simple hidden Markov model describing this state of affairs would be the one given by

$$\mathcal{S} = \{I, E\}, \qquad \mathcal{O} = \{A, C, G, T\}$$

$$\pi = \begin{pmatrix} 0.9 & 0.1 \end{pmatrix}, \qquad P = \begin{pmatrix} p_{II} & p_{IE} \\ p_{EI} & p_{EE} \end{pmatrix} = \begin{pmatrix} 0.8 & 0.2 \\ 0.1 & 0.9 \end{pmatrix}$$

$$B = \begin{pmatrix} b_{IA} & b_{IC} & b_{IG} & b_{IT} \\ b_{EA} & b_{EC} & b_{EG} & b_{ET} \end{pmatrix} = \begin{pmatrix} 0.1 & 0.3 & 0.4 & 0.2 \\ 0.5 & 0.2 & 0.2 & 0.1 \end{pmatrix}$$

The initial distribution $\pi = \begin{pmatrix} 0.9 & 0.1 \end{pmatrix}$ reflects the fact that a DNA string is much more likely to start with an unused intron part.

The relatively large probabilities on the diagonal of the transition matrix reflect the fact that quick alternations between intron and exon segments are unlikely – this results in relatively long extron, resp.intron, segments.

Finally, the emission matrix shows the different composition of nucleotids in intron, resp.exon, segments.

One can used this hidden Markov model and its associated hidden likelihood functions to try and determine which parts of a DNA sequence are introns and which are exons.

Example 6.4.3. A paleontologist wants to infer the climate in centuries long past. She only wants to know whether a year was hot (H) or cold (C). Thus the set of hidden states in the HMM that we set up is $\mathcal{S} = \{H, C\}$. The yearly climate at the era in question is believed to have fluctuated according to the transition matrix

$$P = \begin{pmatrix} p_{HH} & p_{HC} \\ p_{CH} & p_{CC} \end{pmatrix} = \begin{pmatrix} \frac{7}{10} & \frac{3}{10} \\ \frac{4}{10} & \frac{6}{10} \end{pmatrix}$$

She has some fossilized trees and can measure the width of the tree rings, to guess the climate in the years when the tree was alive. The tree ring widths come in three sizes, small (S), medium (M) and large (L), so the set of observable states is $\mathcal{O} = \{S, M, L\}$. The trees were subtropical fig trees which thrive in hot climate. Thus the emission matrix is believed to be

$$B = \begin{pmatrix} b_{HS} & b_{HM} & b_{HL} \\ b_{CS} & b_{CM} & b_{CL} \end{pmatrix} = \begin{pmatrix} \frac{1}{10} & \frac{4}{10} & \frac{5}{10} \\ \frac{7}{10} & \frac{2}{10} & \frac{1}{10} \end{pmatrix}$$

She has no clue as to whether the first year she considered was hot or cold, and assumes a fifty-fifty distribution for year zero. Thus she takes the initial distribution to be $\pi = (\frac{1}{2}, \frac{1}{2})$.

Now the tree has a small ring in year 0, and a medium ring in year 1 (yes, I know, two tree rings is a puny amount of data, but I don't want to fill pages with calculations). What were the most likely climates in years 0 and 1 to produce these rings? We want to find the $(x_0, x_1) \in \mathcal{S}^2$ maximizing $P(X_0 = x_0, X_1 = x_1 | Y_0 = S, Y_1 = M)$.

The idea is to use Bayes' formula

$$P(X_0 = x_0, X_1 = x_1 | Y_0 = S, Y_1 = M) \\ = \frac{P(Y_0 = S, Y_1 = M | X_0 = x_0, X_1 = x_1) P(X_0 = x_0, X_1 = x_1)}{P(Y_0 = S, Y_1 = M)}$$

The ingredients of the right hand side are all easily computable from the specification of the HMM, as you will see.

First we compute the conditional probabilities for the observed data, given all possible climate sequences:

$$P(Y_0 = S, Y_1 = M | X_0 = H, X_1 = H) = \frac{1}{10} \cdot \frac{4}{10} = \frac{4}{100}$$

$$P(Y_0 = S, Y_1 = M | X_0 = H, X_1 = C) = \frac{1}{10} \cdot \frac{2}{10} = \frac{2}{100}$$

$$P(Y_0 = S, Y_1 = M | X_0 = C, X_1 = H) = \frac{7}{10} \cdot \frac{4}{10} = \frac{28}{100}$$

$$P(Y_0 = S, Y_1 = M | X_0 = C, X_1 = C) = \frac{7}{10} \cdot \frac{2}{10} = \frac{14}{100}$$

Here the factors are the entries of the emission matrix, and we can just multiply them, because in the presence of data on X_0, X_1 the random variables Y_0, Y_1 are independent.

Next, we compute the probabilities of getting the possible sequences of climates:

$$P(X_0 = H, X_1 = H) = P(X_1 = H | X_0 = H) P(X_0 = H) = \frac{7}{10} \cdot \frac{5}{10} = \frac{35}{100}$$

$$P(X_0 = H, X_1 = C) = \frac{3}{10} \cdot \frac{5}{10} = \frac{15}{100}$$

$$P(X_0 = C, X_1 = H) = \frac{4}{10} \cdot \frac{5}{10} = \frac{20}{100}$$

$$P(X_0 = C, X_1 = C) = \frac{6}{10} \cdot \frac{5}{10} = \frac{30}{100}$$

Here the numbers come directly from the transition matrix, and the initial distribution (where we expressed $\frac{1}{2}$ as $\frac{5}{10}$).

If we were interested in the total probability of observing the sequence that we observed, we could easily compute it now:

$$\begin{aligned}
&P(Y_0 = S, Y_1 = M) \\
&\quad = \sum_{x_0, x_1 \in S} P(Y_0 = S, Y_1 = M | X_0 = x_0, X_1 = x_1) P(X_0 = x_0, X_1 = x_1) \\
&\quad = \frac{4}{100} \cdot \frac{35}{100} + \frac{2}{100} \cdot \frac{15}{100} + \frac{28}{100} \cdot \frac{20}{100} + \frac{14}{100} \cdot \frac{30}{100} \\
&\quad = \frac{1}{100^2} (140 + 30 + 560 + 420) = \frac{1150}{100^2}
\end{aligned}$$

For the task at hand this is not necessary though, because the maximum of the Bayes quotient is achieved wherever the enumerator achieves its maximum.

The maximum is thus obtained for the sequence $X_0 = C, X_1 = H$, which gives the enumerator $\frac{20}{100}$

$$P(X_0 = C, X_1 = H | Y_0 = S, Y_1 = M) = \frac{\frac{560}{100^2}}{\frac{1150}{100^2}} = \frac{560}{1150} = \frac{56}{115}$$

Of course in real examples one wants to deal with much longer sequences of observations, and the question is how to continue this procedure. The brute force approach of just computing all possible values and taking the maximum is not feasible. Fortunately there is a much more efficient algorithm, called the Viterbi algorithm, that will be treated in the next section.

Example 6.4.4. Another field of application of hidden Markov models is speech recognition: For example, one can set up a model where the latent variables take values in the phonemes of the language at hand, i.e., roughly, the ideal sounds that a perfectly articulate speaker of the language would emit when speaking. The observed variables describe the actual recorded sound, which may deviate from the ideal situations by mispronunciations and background noise.

There also are other applications of HMMs in language processing. Here is a famous tutorial by Rabiner.

6.4.1 The Viterbi algorithm

The basic use case for HMMs, as in the DNA example, is to infer from a sequence of observations a sequence of underlying hidden states. This can be done efficiently using the so-called *Viterbi algorithm*.

Supose we are given an HMM $(\mathcal{S}, \mathcal{O}, \pi, P, B)$ and a sequence of observed states $(o_0, \ldots, o_n)$. We want to find out which sequence $(s_0, \ldots, s_n)$ of hidden states which is most likely to have induced our observations. That is, we want to find

$$\operatorname{argmax}_{s_0,\ldots,s_n} P(X_0 = s_0, \ldots, X_n = s_n \mid Y_0 = o_0, \ldots, Y_n = o_n)$$

For events as they occur in this expression, we introduce the shorthand notations

$$\mathbf{s}_{0:k} := "X_0 = s_0, \ldots, X_k = s_k''$$
$$\mathbf{o}_{0:k} := "Y_0 = o_0, \ldots, Y_k = o_k''$$

and a sequence of states $s_0, \ldots, s_k$ will be abbreviated as $s_{0:k}$. Thus in this new notation we want to find $\operatorname{argmax}_{s_{0:n}} P(\mathbf{s}_{0:n} | \mathbf{o}_{0:n})$. We first observe that since $P(\mathbf{s}_{0:n} | \mathbf{o}_{0:n}) = \frac{P(\mathbf{s}_{0:n}, \mathbf{o}_{0:n})}{P(\mathbf{o}_{0:n})}$, and the denominator does not depend on $s_{0:n}$, we have $\operatorname{argmax}_{s_{0:n}} P(\mathbf{s}_{0:n} | \mathbf{o}_{0:n}) = \operatorname{argmax}_{s_{0:n}} P(\mathbf{s}_{0:n}, \mathbf{o}_{0:n})$.

The Viterbi algorithm computes this recursively, in each step lowering the length of the sequence. On the way one has to compute the *values* that are achieved by the argmax, i.e. of

$$\alpha_k(s_k) := \max_{s_{0:k-1}} P(\mathbf{s}_{0:k}, \mathbf{o}_{0:k}).$$

Note that the argument of the left hand side is a single hidden state s_k, and that on the right hand side we maximize over sequences $(s_0, \ldots, s_{k-1})$. Now using how the random variables X_k, Y_k depend on the other parts of the hidden Markov model, we can rewrite

$$\begin{aligned}
\alpha_k(s_k) &= \max_{s_{0:k-1}} P(\mathbf{s}_{0:k}, \mathbf{o}_{0:k}) \\
&= \max_{s_{0:k-1}} P(Y_k = o_k | X_k = s_k) P(X_k = s_k | X_{k-1} = s_{k-1}) P(\mathbf{s}_{0:k-1}, \mathbf{o}_{0:k-1}) \\
&= \max_{s_{k-1}} P(Y_k = o_k | X_k = s_k) P(X_k = s_k | X_{k-1} = s_{k-1}) \max_{s_{0:k-2}} P(\mathbf{s}_{0:k-1}, \mathbf{o}_{0:k-1}) \\
&= \max_{s_{k-1}} P(Y_k = o_k | X_k = s_k) P(X_k = s_k | X_{k-1} = s_{k-1}) \alpha_{k-1}(s_{k-1})
\end{aligned}$$

Here in the second line we are maximizing over the variables $s_0, \ldots, s_{k-1}$; in the passage to the third line, we separate this into two maximization tasks, the first two factors being terms that do not depend on the initial segment $s_0, \ldots, s_{k-2}$.

The appearance of $\alpha_{k-1}(s_{k-1})$ in the final expression shows that we indeed have a recursive task. The other factors are elements of the emission matrix, resp. the transition matrix, and thus are known quantities. The recursion ends with $\alpha_0(s_0) := P(s_0, o_0) = P(X_0 = s_0)P(Y_0 = o_0|X_0 = s_0)$, and the factors are again known quantities from the initial distribution and the emission matrix.

The recursive computation of the maximum in question proceeds by finding at each step the optimal s_k among all the $s \in \mathcal{S}$ Now to find $\operatorname{argmax}_{s_{0:n}} P(\mathbf{s}_{0:n}|\mathbf{o}_{0:n})$, we simply record this finding at each step, and we will end up with the optimal sequence.

Other tasks around Hidden Markov Models

Hidden Markov models have found other uses than that of guessing hidden states. For example, in optical character recognition (OCR) one first *trains* one hidden Markov model for each letter.

If for example the letters are scanned in vertical stripes of 2 by 20 pixels, and a typical letter is 10 stripes wide, then in such an HMM the hidden states could be the regions of the letter (1st stripe, 2nd stripe, ..., 10th stripe), and the observables could be vectors of length 20 encoding which pixels are black and which are white.

The HMM for a letter is trained using the Baum-Welch algorithm, a version of the EM-algorithm that you know from the Machine Learning lecture course.

At the time of usage, after scanning a letter, i.e. after obtaining a string of observables $(o_0, \ldots, o_1 0)$, one computes which of the HMMs has the highest probability of producing that string – that one is the guess for the scanned letter. For this computation one can again use a recursive approach as in the Viterbi algorithm, obtaining the values $P(\mathbf{s}_{0:k}, \mathbf{o}_{0:k})$ for all sequences $s_{0:k}$, and then summing up: $P(\mathbf{o}_{0:k}) := \sum_{s_{0:k}} P(\mathbf{s}_{0:k}, \mathbf{o}_{0:k})$. This is also called the Forward Algorithm.

The three tasks

1. Training an HMM

2. Computing the probability of a given observable sequence

3. Finding the most likely underlying hidden states of a given observable sequence

are often named as the main tasks asociated to HMMs.

All three tasks are very well explained in these notes by Mark Stamp, or on these slides by Andrew Moore.

Remark 6.4.5. HMMs can also be used for clustering. The basic idea is this: If you think that the observations in your sequence fall into K clusters, you could try to explain the data with a hidden Markov model with K hidden states. Fitting the model gives you an emission matrix, that can be read as giving likelihoods to belonging to a given cluster.

With a more refined setup, you can also use HMMs to infer the number of clusters from your observations.

We close by mentioning that, although in our examples the set of observable states was always finite, there are also Hidden Markov models where the observable variables have continuous distributions on an uncountable set of states. A typical example are Gaussian distributions $\mathcal{N}(\mu(s), \mathbf{\Sigma}(s))$, where the parameters $\mu(s), \mathbf{\Sigma}(s)$ now depend on the hidden state s.

6.5 Continuous time stochastic processes

This section is not relevant for the exam.

6.5.1 Continuous time Markov processes

Our definition of a stochastic process $(X_t)_{t \in T}$ allowed arbitrary index sets T, but until now we only considered $T = \mathbb{N}$. If we think of (X_t) as a process evolving in time, then $T = \mathbb{N}$ means that we jump through time, only taking snapshots at certain moments.

If one instead takes $T = [0, \infty) \subseteq \mathbb{R}$, one speaks of a *continuous time stochastic processes*. One can take other index sets, e.g. $T = \mathbb{R}^n$, but then might lose the typical temporal intuition suggested by the word *process*.

We will not study continuous time stochastic processes in any depth. But to give a hint, what changes in relation to discrete time, here is a definition of continuous time Markov process:

Definition 6.5.1. *(a) A stochastic process $(X_t)_{t\in[0,\infty)}$, where each X_t takes values in the set of states $\mathcal{S}$, is called a* Markov process, *if for any $x_1,\ldots,x_n \in \mathcal{S}$ and any sequence $t_1 < \ldots < t_n$ of times in $[0,\infty)$ we have*

$$P(X_{t_n} = x_n | X_{t_{n-1}} = x_{n-1}, \ldots X_{t_1} = x_1) = P(X_{t_n} = x_n | X_{t_{n-1}} = x_{n-1})$$

(b) The Markov process is called time homogeneous, *if for all times $s < t$ and states $i, j \in \mathcal{S}$ we have*

$$P(X_t = j | X_s = i) = P(X_{t-s} = j | X_0 = i)$$

Note that the Markov property is again only expressed referring to finitely many previous states, but now these can be anywhere on the nonnegative real line. This makes sense, given that we usually have finite knowledge, but there still can be a continuously ongoing process that we are studying.

The theory of continuous Markov processes is quite parallel to the discrete version, once one has the right mathematical setup. In the discrete case one studies the transition matrix P and its powers P^n, to assess the long term behaviour of a Markov chain. With continuous time t however, there is no obvious definition of P^t.

For simplicity let us consider time homogeneous Markov processes on a finite set of states. As a substitute for the transition matrices one defines $p_{ij}(t) := P(X_{t-s} = j | X_0 = i)$ and sets $P_t := (p_{ij}(t))$. Each P_t is a stochastic matrix. One then has the Chapman-Kolmogorov equations $P_s P_t = P_{s+t}$. We assume $P(0) = \mathbf{I}$ (the identity matrix) and $\lim_{t\to 0} P(t) = \mathbf{I}$.

Then one can show that for small $h \in [0,\infty)$ we have:

$$P(h) = \mathbf{I} + Gh + o(h)$$

for some matrix G, and higher order terms summarized in the expression $o(h)$. That is, $P(t)$ is differentiable (from the right) at 0 with differential given by a matrix $G = (g_{ij})$. It this G that plays the role of the transition matrix P from the discrete case.

Due to time homegeneity, we have the same infinitesimal change G at every time, which is reflected in the following equations of matrix valued functions:

$$\begin{aligned}\frac{\partial}{\partial t}P(t) &= \lim_{h\to 0}\frac{1}{h}(P(t+h) - P(t)) = \lim_{h\to 0}\frac{1}{h}(P(t)P(h) - P(t)) \\ &= P(t)\lim_{h\to 0}\frac{1}{h}(P(h) - I) = P(t)\frac{1}{h}(I + Gh - I) = P(t)G\end{aligned}$$

and

$$\begin{aligned}\frac{\partial}{\partial t}P(t) &= \lim_{h\to 0}\frac{1}{h}(P(h+t)-P(t)) = \lim_{h\to 0}\frac{1}{h}(P(h)P(t)-P(t)) \\ &= \lim_{h\to 0}\frac{1}{h}(P(h)-I)P(t) = \frac{1}{h}(I+Gh-I)P(t) = GP(t)\end{aligned}$$

The only solution of this differential equation is given by

$$P(t) = \exp(tG) = \sum_{k=0}^{\infty}\frac{t^k}{k!}G^k$$

(maybe you know this property of the matrix exponential function from your course on ODEs).

One thing that one studies in Markov processes is the expected time of the first jump from a given state to another state. One can show that this time is described by a random variable with exponential distribution – where the exponential distribution comes into the picture precisely because of the above involvement of the exponential function (see e.g. Grimmett/-Stirzaker, Probability and Random processes, Section 6.9, Claim (13$*$) – it's not hard).

One can turn this discovery around and alternatively *define* a Markov process to be a stochastic process whose jump time has an exponential distribution – see the "Jump chain/holding time definition" on the Wikipedia page on Markov processes.

Furthermore, one can show that the matrix G behaves like the logarithm of a stochastic matrix: The rows sum up to $0 = \log 1$, and one can show that a state is stationary (i.e. $\pi P(t) = \pi$ for all t) if and only if $\pi G = 0$. A Markov process can also alternatively be defined in terms of this matrix G and its relation to the random variables X_t – this is what the Wikipedia page on Markov processes takes as the definition.

This was just a taste of the theory of continuous time Markov processes. There also are continuous time Hidden Markov Models. In case you encounter the expression *"state space process"* in your reading: This term is overused, but often it is used as a synonym for continuous time Hidden Markov models. Some people present the notion "state space process" as denoting a linear system of differential equations – they probably mean the same story, and the link between their presentation and ours is given by the above differential equation, whose solution is the matrix exponential function.

6.5.2 Gaussian processes

See the Machine Learning lecture course.

7 Information Theory

Information theory studies the transmission, compression and encoding of data. It can be used in electrical engineering and computer science in very concrete and practical ways. Intuitively, receiving information means an increase of knowledge. Neither the notion of information nor the notion of knowledge are formally defined in information theory. Instead one makes the intuition of lack of knowledge precise by considering random variables: A random variable, with its probability distribution, is the formalization of "uncertainty" adopted in information theory. The first basic definition is that of the *entropy* of a random variable X. Roughly, this is a measure of how much more one knows, i.e. how much information one gained, after seeing the outcome of the random experiment described by X.

This formulation of information theory in terms of proability theory makes information theory useful, in return, for probability theory and Machine Learning. For example there is the notion of *mutual information* of two random variables, which provides a measure of how independent they are. And there is the notion of Kullback-Leibler divergence, which can be understood as a measure of how far two probability distributions are apart from each other, and can be used in a learning process to ensure that one does not go to far from a given distribution with a high prior preference.

A great source for this part of the course is the book Cover/Thomas, Elements of Information Theory, 2nd edition (not free), which also inspired these notes.

Another great book, closely linking Information Theory to Machine Learning, is MacKay, Information Theory, Inference and Learning Algorithms, freely available for download.

Coding is something inherently discrete, and this is the reason that Information Theory is easier explained for discrete random variables. We will treat continuous random variables only at the end of this chapter.

In all but the last section of this chapter all random variables will be discrete!

7.1 Entropy, conditional entropy, mutual information, Kullback-Leibler divergence

The *entropy* of a random variable X can be understood as a measure of how much information one gained, after seeing the outcome of the random experiment described by X. This is maybe not apparent directly from the definition, but will be so from the results about it.

Definition 7.1.1. *The* entropy *to base b of a discrete random variable* $X\colon \Omega \to \mathcal{X}$ *with probability mass function p is:*

$$H_b(X) := -\sum_{x\in\mathcal{X}} p(x)\log_b(p(x))$$

For $b = 2$ *we just write* $H(X)$*, resp.* $log(p(x))$

Conventions 7.1.2.

- $0 \cdot \log 0 := 0$
- for $b = 2$ we call the units of entropy "bits"
- for $b = e$ we call the units of entropy "nats"

Further names are "trits" for $b = 3$ and "Hartleys" for $b = 10$.

Observations 7.1.3.

1. We always have $H_b(X) \geq 0$, because $0 \leq p(x) \leq 1$ implies $\log p(x) \leq 0$.
2. $H(X) = E(-\log p(X))$, where $p(X)$ is the random variable $\Omega \to \mathcal{X}$, $\omega \mapsto P(X^{-1}(X(\omega)))$, or just $p(X)\colon \mathcal{X} \to \mathbb{R}$, $x \mapsto p(x)$.
3. The entropy only depends on the probability distribution, and accordingly we sometimes write $H(p)$.

From the definition itself it is hard to read off the intended meaning of the notion of entropy. Roughly, it should measure how much information we gain, if we get told what is the value of a random variable. More precisely, if a random experiment is performed and you knew the probabilities of the possible outcomes beforehand, entropy is supposed to tell you how much more you know after being told the outcome. This is to be understood *on average*, if you repeat the experiment many times.

Examples 7.1.4. 1. Let X be a random variable with values in $\mathcal{X} = \{1,\dots,32\}$ and uniform distribution. Then

$$H(X) = -\sum_{i=1}^{32} \frac{1}{32} \log \frac{1}{32} = -\log \frac{1}{32} = -\log 2^{-5} = 5 \text{ bits}$$

Indeed, to communicate to outcome of the random variable we have to tell which one of the 32 possible values was assumed, and this can be done using 5 bits.

2. Consider a horse race with 8 horses, called **1**, . . ., **8** and having chances of winning $\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{16}, \frac{1}{64}, \frac{1}{64}, \frac{1}{64}, \frac{1}{64}$. We have $8 = 2^3$ possible outcomes, and again we could communicate the winning horse using 3 bits.

A computation of the entropy of the random variable X denoting the winning horse yields

$$\begin{aligned} H(X) &= -\tfrac{1}{2}\log\tfrac{1}{2} - \tfrac{1}{4}\log\tfrac{1}{4} - \tfrac{1}{8}\log\tfrac{1}{8} - \tfrac{1}{16}\log\tfrac{1}{16} - \tfrac{4}{64}\log\tfrac{1}{64} \\ &= \tfrac{1}{2}\log 2^1 + \tfrac{1}{4}\log 2^2 + \tfrac{1}{8}\log 2^3 + \tfrac{1}{16}\log 2^4 + \tfrac{4}{64}\log 2^6 \\ &= \tfrac{1}{2} + \tfrac{1}{2} + \tfrac{3}{8} + \tfrac{4}{16} + \tfrac{4}{64}\cdot 6 \;=\; 2 \text{ bits} \end{aligned}$$

We can communicate the winning horse using the following code:

horse	code
1	0
2	10
3	110
4	1110
5	111100
6	111101
7	111110
8	111111

We can see this code as a random variable $Y\colon \{\mathbf{1},\dots,\mathbf{8}\} \to \{\texttt{0, 1}\}^*$ (where $\{\texttt{0, 1}\}^*$ denotes the set of finite sequences consisting of the symbols 0, 1). From this we obtain a further random variable $\ell(Y)\colon \{\mathbf{1},\dots,\mathbf{8}\} \to \mathbb{R}$ associating to the name of a horse *the length of its code*. The *average description length* of our code is the expectation of this random variable $\ell(Y)$:

$$E(\ell(Y)) = \frac{1}{2}\cdot 1 + \frac{1}{4}\cdot 2 + \frac{1}{8}\cdot 3 + \frac{1}{16}\cdot 4 + 4\cdot\frac{1}{64}\cdot 6 = 2$$

If we submit the winners of many consecutive horse races, on average we will have to transmit 2 bits per race. It is no coincidence that this is the value of the entropy:

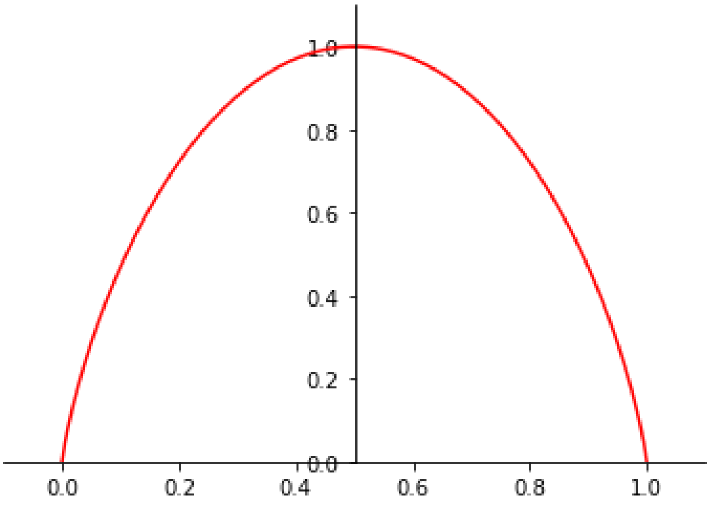
There is a theorem saying that the entropy is a lower bound for the average description length of any so-called prefix code; $H(X) \leq E(\ell(Y))$.

The above code actually attained this lower bound and we will see later on how we can achieve this systematically.

3. As a last example consider a Bernoulli random variable

$$X = \begin{cases} 1 & \text{with probability } p \\ 0 & \text{with probability } 1-p \end{cases}$$

We have $H(X) = H(p) = -p \log p - (1-p)\log(1-p)$. One can check that this is a strictly concave function of p attaining its maximum at $p = \frac{1}{2}$. The entropy is in this case $H(\frac{1}{2}) = \frac{1}{2} \cdot 1 + \frac{1}{2} \cdot 1 = 1$ bit, and indeed we can communicate which of the two results was obtained using one bit.



Graph of the function $p \mapsto H(p)$

To understand the meaning of entropy values lower than 1, think of transmissions of the outcomes of repeated experiments: If the probability of getting a 1 would be $\frac{9}{10}$, we could set up a code for which e.g.

"11" means that we obtained nine 1s in a row. Then, in the long run, we could transmit our results for less than one bit per experiment.

The next definitions combine different random variables:

Definition 7.1.5. *Let X, Y be discrete random variables taking values in sets $\mathcal{X}, \mathcal{Y}$ respectively, with joint distribution $p(x, y)$.*

(a) Joint entropy: $H(X,Y) := -\sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} p(x,y)\cdot\log p(x,y)$

(b) Conditional entropy:

$$\begin{array}{rcl} H(Y \mid X) & := & \sum_{x\in\mathcal{X}} p(x)\cdot H((Y \mid X = x)) \\ & = & -\sum_{x\in\mathcal{X}} p(x)\cdot\sum_{y\in\mathcal{Y}} \frac{p(x,y)}{p(x)}\cdot\log\frac{p(x,y)}{p(x)} \\ & = & -\sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} p(x,y)\cdot\log p(y \mid x) \end{array}$$

where in the last line we used the notation $p(y \mid x) := \frac{p(x,y)}{p(x)}$ for the conditional probability mass function.

Thus the joint entropy is simply the entropy of the random variable $X \times Y$ taking values in $\mathcal{X} \times \mathcal{Y}$. The conditional entropy $H(Y \mid X)$ can be thought of as a measure of how much information about Y we gain, if we know the outcome of X. This point of view is confirmed by the next proposition:

Proposition 7.1.6.

$$H(X,Y) = H(X) + H(Y \mid X)$$

Proof.

$$\begin{aligned} H(X,Y) &= -\sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} p(x,y)\cdot\log p(x,y) \\ &= -\sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} p(x,y)\cdot\log p(x)p(y \mid x) \\ &\overset{(1)}{=} -\sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} p(x,y)\cdot\log p(x) - \sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} p(x,y)\cdot\log p(y \mid x) \\ &\overset{(2)}{=} -\sum_{x\in\mathcal{X}} p(x)\cdot\log p(x) - \sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} p(x,y)\cdot\log p(y \mid x) \\ &= H(X) + H(Y \mid X) \end{aligned}$$

where for (1) we pulled the product out of the logarithm to become a sum, and for (2), in the first term, we summed over all $y \in \mathcal{Y}$, which turns $p(x,y)$ into the marginal distribution $p(x)$. □

So the conditional entropy is what we have to add to the entropy of X to get the entropy of Y.

Remark 7.1.7. Generalizing Prop. 7.1.6, and with an analogous proof, we have

$$\begin{aligned} H(X_1, \ldots, X_n) = H(X_1) + H(X_2 \mid X_1) \\ + H(X_3 \mid X_2, X_1) + \ldots + H(X_n \mid X_{n-1}, \ldots, X_1) \end{aligned}$$

This is called the *chain rule* for conditional entropy.

Observation 7.1.8. If Y is a function of X, say $Y = f(X)$, then $H(Y \mid X) = 0$: In this case we have, for any pair (x, y)

$$p(x,y) = \begin{cases} p(x) & \text{if } y = f(x) \\ 0 & \text{otherwise} \end{cases}$$

and in both cases we have $\frac{p(x,y)}{p(x)} \log \frac{p(x,y)}{p(x)} = 0$. Thus all the summands occurring in the definition of conditional entropy are zero.

Of course this result also makes sense intuitively: If Y is a function of X, then when we know the value of X, there is no more uncertainty about the value of Y – but that is precisely what $H(Y|X)$ measures.

Next we introduce Kullback-Leibler divergence, or KL-divergence, a concept that is known under many other names, and that is used a lot in Machine Learning

Definition 7.1.9. *(a) Let p, q be two probability mass functions on th same set $\mathcal{X}$. The* relative entropy / Kullback-Leibler divergence / KL-distance / information divergence / information discrimination *is defined as:*

$$D(p\|q) := \sum_{x \in \mathcal{X}} p(x) \log \frac{p(x)}{q(x)}$$

(b) Let X, Y be random variables with joint probability mass function $p(x, y)$ and marginal distributions $p(x), p(y)$. The mutual information *is:*

$$I(X;Y) := \sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} p(x,y) \cdot \log \frac{p(x,y)}{p(x)p(y)} = D(p(x,y)\|p(x)p(y))$$

Remark 7.1.10. The KL-divergence $D(p\|q)$ measures the inefficiency stemming from using an optimal code for the distribution q when the actual distribution is p:

Using an optimal code for the distribution p we need on average around $H(p)$ bits per transmission.

Using instead an optimal code for the distribution q, we need on average $H(p) + D(p\|q)$ bits. To get an idea of why this is true, look at the following calculation:

$$\begin{aligned} H(p) + D(p\|q) &= -\sum_{x\in\mathcal{X}} p(x)\log p(x) + \sum_{x\in\mathcal{X}} p(x)\log\frac{p(x)}{q(x)} \\ &= -\sum_{x\in\mathcal{X}} p(x)\log p(x) + \sum_{x\in\mathcal{X}} p(x)\log p(x) - \sum_{x\in\mathcal{X}} p(x)\log q(x) \\ &= -\sum_{x\in\mathcal{X}} p(x)\log q(x) \end{aligned}$$

Here for the passage to the second line we expand the fraction inside the logarithm into a difference. The final expression measures the average length of the code optimized for the distribution q: the factor $p(x)$ gives the true probability of x occurring and the factor $\log q(x)$ gives the length of the q-optimized code that indicates the outcome x.

Remark 7.1.11. Remark 7.1.10 now also offers a perspective on mutual information. Since we have $I(X;Y) = D(p(x,y)\|p(x)p(y))$, the mutual information measures how much more efficiently we can encode the results of the random variables X, Y together (i.e. optimizing the code for their joint distribution) than if we transmit them separately (i.e. optimizing codes for the individual distributions and sticking the results together).

The perspective of Remark 7.1.11 is confirmed by the following proposition.

Proposition 7.1.12.

$$I(X;Y) = H(X) - H(X \mid Y)$$

Proof.

$$\begin{aligned} I(X;Y) &= \textstyle\sum_{x,y} p(x,y)\log\frac{p(x,y)}{p(x)p(y)} = \sum_{x,y} p(x,y)\log\frac{p(x|y)}{p(x)} \\ &= \textstyle -\sum_{x,y} p(x,y)\log p(x) + \sum_{x,y} p(x,y)\log p(x \mid y) \\ &= \textstyle -\sum_{x} p(x)\log p(x) + \sum_{x,y} p(x,y)\log p(x \mid y) \\ &= H(X) - H(X \mid Y) \end{aligned}$$

Here, as we did before, we expanded the fraction in the logarithm into a difference and used to summing $p(x, y)$ over all $y \in Yc$ gives the marginal $p(x)$. □

Corollary 7.1.13. *(a)* $H(X) - H(X \mid Y) = H(Y) - H(Y \mid X)$

(b) $I(X; X) = H(X)$

Proof. (a) This follows from Prop. 7.1.12 because it is immediately visible from the definition of mutual information that $I(X; Y) = I(Y; X)$.

(b) We have $I(X; X) = H(X) - H(X \mid X)$, by Prop 7.1.12. But the second summand is zero by Obs. 7.1.8, because definitely X is a function of X. □

Theorem 7.1.14. *Let p, q be two probability mass functions on $\mathcal{X}$. Then $D(p\|q) \geq 0$, and equality holds if and only if $p = q$.*

Proof.

$$\begin{aligned} -D(p\|q) &= -\textstyle\sum_{x\in\mathcal{X}} p(x) \log \frac{p(x)}{q(x)} = \sum_{x\in\mathcal{X}} p(x) \log \frac{q(x)}{p(x)} \\ &\leq \log \left(\textstyle\sum_{x\in\mathcal{X}} p(x) \frac{q(x)}{p(x)}\right) = \log \left(\sum_{x\in\mathcal{X}} q(x)\right) = \log 1 = 0 \end{aligned}$$

Here the inequality holds because log is a *strictly concave* function.

If equality holds, then, because of the *strict* concavity, there can not occur two different values in the convex combination, i.e. $\frac{q(x)}{p(x)}$ must be the same value for all x – call this value c. Thus $q(x) = c \cdot p(x)$ for all x, and, summing over all x, we get $1 = \sum_{x\in\mathcal{X}} q(x) = c \cdot \sum_{x\in\mathcal{X}} p(x) = c \cdot 1 = c$, i.e. $c = 1$ and hence $p = q$. □

As some of the many names of the KL distance indicate, one can think of it as a measure of distance between two probability distributions. The theorem provides one of the standard properties of a distance function (metric), but others fail, e.g. KL distance is not symmetric.

The next Corollary says that mutual information is a much better measure of independence of random variables than their covariance – remember that there are dependent random variables with covariance zero.

Corollary 7.1.15. $I(X; Y) \geq 0$ *with equality if and only if X and Y are independent.*

Proof. $I(X;Y) = D(p(x,y)\|p(x)p(y)) \geq 0$ with equality if and only if $p(x,y) = p(x)p(y)$. But this latter condition means precisely that X and Y are independent. □

Theorem 7.1.17 below says that the uniform distribution has the highest entropy of all distributions. A more precise statement is given in this lemma:

Lemma 7.1.16. *Let X be a random variable taking values in $\mathcal{X}$ and let $u(x) := \frac{1}{|\mathcal{X}|}$ be the probability mass function of the uniform distribution on $\mathcal{X}$. Then $H(X) = \log|\mathcal{X}| - D(p(x)\|u(x))$.*

Proof. Let p be the probability mass function of X. Then

$$\begin{aligned} D(p(x)\|u(x)) &= \sum_{x\in\mathcal{X}} p(x)\log\frac{p(x)}{u(x)} \\ &= \sum_{x\in\mathcal{X}} p(x)\log p(x) + \sum_{x\in\mathcal{X}} p(x)\log|\mathcal{X}| \\ &= -H(X) + \log|\mathcal{X}| \end{aligned}$$

□

Theorem 7.1.17. *$H(X) \leq \log|\mathcal{X}|$ with equality if and only if X has the uniform distribution.*

Proof. Immediate from Lemma 7.1.16 together with Thm. 7.1.14 which says that KL divergence is positive, and is zero iff the two distributions coincide. □

It is now worth rereading Lemma 7.1.16 in the light of Thm. 7.1.17: It gives a new interpretation of KL distance between the uniform distribution and the given distribution as the difference between the entropy of the uniform distribution,

$$H(u) = -\sum_{x\in\mathcal{X}} u(x)\log u(x) = -\sum_{x\in\mathcal{X}} \frac{1}{|\mathcal{X}|}\log\frac{1}{|\mathcal{X}|} = \log|\mathcal{X}|,$$

(which is the maximal possible entropy) and the given distribution.

The following theorem states that conditioning reduces entropy. In other words: Additional information about some other random variable Y can not hurt for finding a good description of X.

Theorem 7.1.18. *$H(X \mid Y) \leq H(X)$, with equality if and only if X and Y are independent.*

Proof. We have $0 \leq I(X;Y) = H(X) - H(X \mid Y)$, where the inequality holds by Cor. 7.1.15 and is an equality iff X and Y are independent, and the equality comes from Prop. 7.1.12. □

Corollary 7.1.19. *$H(X_1, \ldots, X_n) \leq \sum_{i=1}^n H(X_i)$ with equality iff the X_i are independent.*

Proof. We have $H(X_1, \ldots, X_n) = \sum_{i=1}^n H(X_i | X_{i-1}, \ldots, X_1) \leq \sum_{i=1}^n H(X_i)$, where the first equality comes from the chain rule, Remark 7.1.7, and the second equality is that of the previous theorem. □

The previous Corollary says that transmitting the results of several random variables together, may require fewer, but definitely not more bits, than transmitting all the single results. This makes intuitive sense, as might use the dependencies of the random variables, and indeed that is how the proof goes.

Remark 7.1.20. Often we want to transmit *sequences* of values of random variables, e.g. the symbols of a text or the succession of sounds of a recording, and not just a single value. A sequence of random variables is a stochastic process, and the entropy of a stochastic process $\mathbb{X} = \{X_n \mid n \in \mathbb{N}\}$ is defined as $H(\mathbb{X}) := \lim_{n\to\infty} \frac{1}{n} H(X_n, \ldots, X_1)$.

An important result is that, for a stationary stochastic process (i.e. one whose conditional probabilities are invariant under time shifts) we have $H(\mathbb{X}) = \lim_{n\to\infty} H(X_n \mid X_{n-1} \ldots, X_1)$. This uses Cor. 7.1.19 and reinforces the idea that for the transmission of joint outcomes or random variables one can use their interdependence.

7.2 Codes

We now come to the problem of finding optimal codes, i.e. codes of minimal average length.

Definition 7.2.1. *Let D be a set, called the* alphabet.

(a) we write D^* for the set of strings of finite length of elements of D

(b) we write $\ell \colon D^* \to \mathbb{N}$ for the map associating to a string its length.

(c) a code *for an $\mathcal{X}$-valued random variable X is a map $C\colon \mathcal{X} \to D^*$.*

(d) the expected length *of the code C is $E(\ell(C))$*

A code is called non-singular, *if it is injective.*

A code can be extended from $\mathcal{X}$ to a map $\mathcal{X}^* \to D^*$, still denoted by C, by setting$C(x_1 \dots x_n) := C(x_1) \dots C(x_n)$.

A code is called uniquely decodable, *if this extension is injective.*

For a code to be uniquely decodable, one needs to ensure that in the sequence $C(x_1) \dots C(x_n)$ one can recognize when a string of symbols $C(x_{i-1})$ ends and the next string $C(x_i)$ starts.

It can happen that a code is uniquely decodable, but that even to determine the first symbol x_1 one needs to analyze the whole string $C(x_1) \dots C(x_n)$. When this can not happen, i.e. when we can parse the string from front to end and always know when the code of the next symbol starts, we speak of a *prefix code*, or *instantaneous code*:

Definition 7.2.2. *A code is called a* prefix code, *or an* instantaneous code, *if no code word (i.e. string of the form $C(x)$ for an $x \in Xc$) is a prefix (i.e. initial fragment) of another code word.*

One also says that such a code is "self-punctuating": If one thinks of the sequence $C(x_1) \dots C(x_n)$ as a list of codewords, then the code "knows where to put the sparating commas".

We have the following implications between properties of codes:

$$\text{prefix code} \;\Rightarrow\; \text{uniquely decodable} \;\Rightarrow\; \text{non-singular}$$

The following examples show that the reverse implications do not hold.

Example 7.2.3. We consider several codes for the alphabet {`0`, `1`}, encoding the elements of the set $\mathcal{X} := \{\mathbf{1}, \mathbf{2}, \mathbf{3}, \mathbf{4}\}$.

		non-singular	uniquely decodable	
$\mathcal{X}$	singular	but not uniq. decod.	but not prefix code	prefix code
1	0	0	10	0
2	0	010	00	10
3	0	01	11	110
4	0	10	110	111

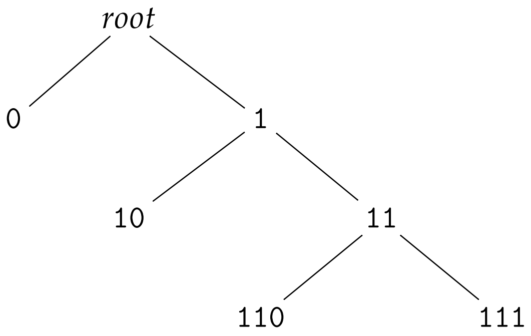
Theorem 7.2.4 (Kraft inequality). *Let $C\colon \mathcal{X} = \{1,\dots,m\} \to D^*$ be a prefix code. Denote by $l_i := \ell(C(i))$ the codeword lengths. Then we have the inequality*

$$\sum_{i=1}^{m} |D|^{-l_i} \leq 1$$

Conversely, given numbers $l_1,\dots,l_m$ satisfying this inequality, there exists a prefix code with these word lengths.

Before considering the proof, note that, as the word lengths occur as negative exponents in the sum, the Kraft inequality says that a prefix code can only be achieved with sufficiently long codewords. As a border case we can take $D = \mathcal{X} = \{1,\dots,m\}$ and encode every element of $\mathcal{X}$ by itself. This gives us word lengths $l_i = 1$ and the Kraft inequality becomes an equality.

Proof. We consider a $|D|$-ary tree (in the example we draw a binary tree). That is, we have a graph which at each node except the root has one incoming edge and $|D|$ outgoing edges. Each outgoing edge is meant to represent one choice of symbol from D. Thus a path from the root to a leaf represents a word from D^*. The tree for the prefix code of Example 7.2.3 looks as follows:



The prefix condition for a code means precisely that only leaves correspond to codewords, but no intermediate nodes.

First, suppose we are given a prefix code. Let l_{max} be the length of the longest code word. Consider the full $|D|$-ary tree of depth l_{max}: It has $|D|^{l_{max}}$ leaves. Some leaves (at least one) correspond to code words, the others correspond to descendants of code words.

A code word of length l_i sits at the level l_i of the tree and has $|D|^{l_{max}-l_i}$ leaves as descendants. Since we have a prefix code, none of these leaves can occur as a code word.

Thus starting from the full set of D-words of length $\leq l_{max}$, and deleting thos which can not occur as code words, we can eliminate $\sum_{i=1}^{m} |D|^{l_{max}-l_i}$ words (two different code words have disjoint sets of descendants, so we really can take the sum here).

Since we can not eliminate more than all leaves, we have

$$\sum_{i=1}^{m} |D|^{l_{max}-l_i} \leq |D|^{l_{max}}$$

Dividing by $|D|^{l_{max}}$ gives $\sum_{i=1}^{m} |D|^{-l_i} \leq 1$.

Conversely, given numbers $l_1, \ldots, l_m$ satisfying the Kraft inequality, we can construct a prefix code with these word lengths as follows: Start with the full $|D|$-ary tree, take the first node of depth l_1 as the code for x_1, then delete all descendants. Then take the first remaining node of depth l_2 as code for x_2, delete its descendants, and so on. The Kraft inequality ensures that there will still be nodes left until the last element x_m has been assigned a code. □

Remark 7.2.5. The Kraft inequality is not about a lower bound to the length of each individual word. It says that $\sum_{i=1}^{m} d^{-l_i} \leq 1$, and this left hand side combines the lengths of all the codewords together. Indeed, you can clearly always produce a prefix code where one codeword is `0` and all the other codewords start with `1`. This gives you a lower bound of 1 for the codeword length, and that lower bound can always be realized. But this design choice will force the other codewords to be longer, if you want a prefix code - and it is the lengths of all codewords together that will make the Kraft inequality hold.

Remark 7.2.6. Theorem 7.2.4 is also true for infinite $\mathcal{X}$ (Cover/Thomas, Thm. 5.2.2) and for uniquely decodable codes (Cover/Thomas, Thm. 5.5.1). The latter implies that we can not save word length by dropping the requirement of having a prefix code and being content with a uniquely decodable code:

Theorem 7.2.4 contains an "if and only if" statement. Given numbers $l_1, \ldots, l_m$, there exists a d-ary prefix code with word lengths $l_1, \ldots, l_m$ if and only if these numbers satisfy the Kraft inequality $\sum_{i=1}^{m} d^{-l_i} \leq 1$. The corresponding theorem for uniquely decodable codes is also true. Now if you have a uniquely decodable code, with word lengths $l_1, \ldots, l_m$, then these word lengths satisfy the Kraft inequality, and therefore there is also a prefix code with precisely the same word lengths.

Algorithm 7.2.7 (Huffman algorithm for producing a code of optimal word length)**.**

Suppose we are given a random variable X assuming values $\{1, \ldots, m\}$ with probabilities $p_1, \ldots, p_m$, we can produce an optimal D-ary code (i.e. one for which the encoding alphabet has D symbols) as follows:

We will process the data of a finite collection of real numbers together with a tree attached to each number.

Start with the collection of probabilities $p_1, \ldots, p_m$ and associate to each the tree consisting only of a root.

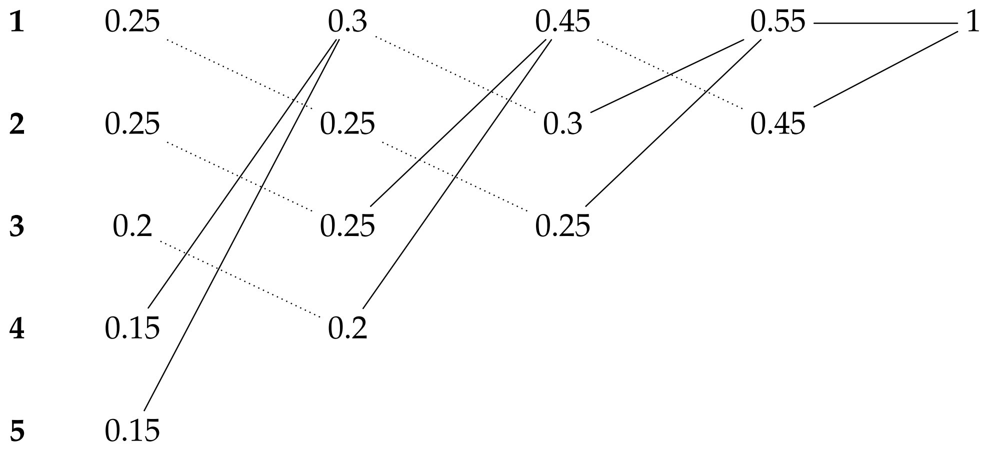
Then repeat the following steps

1. Order the probabilities by size: $p_1 \geq p_2 \geq \ldots \geq p_k$
2. Replace the collection of numbers by $p_1, \ldots, l_{k-D}, p_{k_D+1} + \ldots + p_k$
3. Join the roots of the trees of $p_{k_D+1}, \ldots, p_k$ at a new root, and associate this new tree to the number $p_{k_D+1} + \ldots + p_k$. The other numbers continue with their previously associated trees.

until we are left with only one number (which necessarily is 1).

The tree associated to the last remaining number corresponds to a code in the way we have discussed. This is called a *Huffman code*.

Example 7.2.8. We let $\mathcal{X} = \{\mathbf{1}, \dots, \mathbf{5}\}$ and construct a binary code:



Now labeling each of the solid edges with a `0` or `1`, we obtain a Huffman code; the code words are obtained by travelling from the number 1, the root, to the leftmost column and recording the labels of the edges on the way.

For example, if in the above picture we label always the upper solid edge with a `0` and the lower one with a `1`, we obtain the following Huffman code:

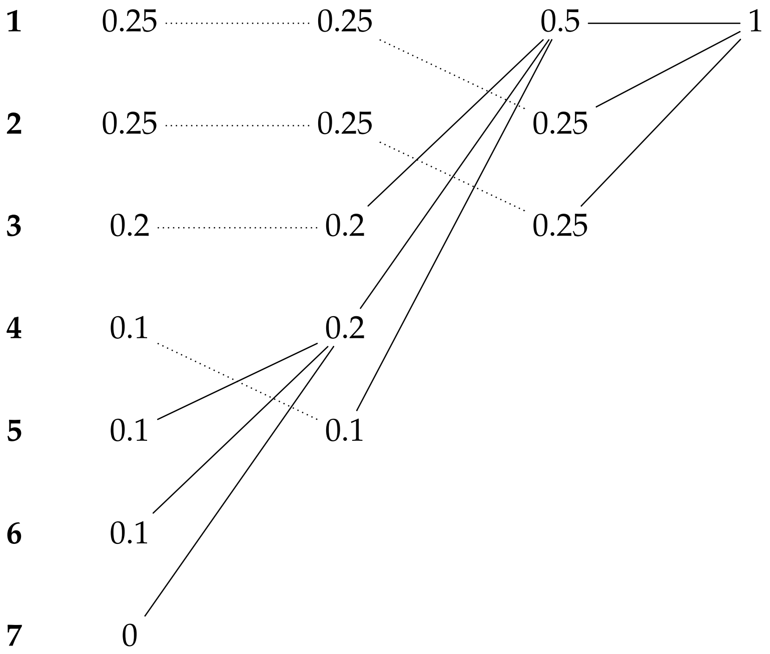
1	**2**	**3**	**4**	**5**
`01`	`10`	`11`	`000`	`001`

Remark 7.2.9. In the Huffman algorithm, each cycle reduces the list of numbers by $D - 1$, because D numbers get merged into a single one. In Example 7.2.8 we had $D = 2$, so in each step the numbers got reduced by one and we were guaranteed to be merging two numbers in the last step.

In general, to be sure to sum exactly D numbers in the last step, we need to have a number m of outcomes of $\mathcal{X}$ of the form $m = 1 + k \cdot (D - 1)$ – this will then yield a D-ary tree of depth k. If m is not of this form, we increase m to become the next bigger number of this form and we give zero probability to the additional "dummy" events that we introduced.

Example 7.2.10. Consider a random variable X with outcomes $\{\mathbf{1}, \dots, \mathbf{6}\}$, attained with probabilities $0.25, 0.25, 0.2, 0.1, 0.1, 0.1$. We use the alphabet $\{\mathtt{0}, \mathtt{1}, \mathtt{2}\}$, so $D = 3$, i.e. we aim to construct a ternary Huffman code for X.

We have six possible outcomes of X, but we need a number of the form $1 + k \cdot (D - 1) = 1 + k \cdot 2$, i.e. an uneven number. The next bigger uneven number is 7, and so we add a "dummy event" **7** attained with probability 0. The tree looks as follows:



If we label the solid edges in each column from top to bottom by `0`, `1`, `2`, we obtain the Huffman code

1	**2**	**3**	**4**	**5**	**6**
`1`	`2`	`00`	`02`	`010`	`011`

The dummy event **7** would have gotten the code `012`, but we now drop it again.

Theorem 7.2.11. *The Huffman code is optimal, i.e. it has the smallest possible expected code length.*

About the proof: For simplicity we only discuss the case $D = 2$.

Lemma: For any distribution $p_1, \dots, p_m$ there exists an optimal prefix code satisfying

1. the word lengths are ordered inversely to the probabilities: $p_i \geq p_j \Rightarrow l_i \leq l_j$
2. the two longest codewords have the same length

3. the two longest codewords only differ in the last bit

To make the lemma plausible, first note that the existence of an optimal code is clear: We can bound the lengths of code words from above and then we only have to search among finitely many codes.

Now note that if 1. is not satisfied for a pair i, j, then we can exchange the codewords, thereby lowering the expected length. For 2., observe that the code is represented by a binary tree, and if we have one branch longer than all others, we can cut it by one. Requirement 3, can be achieved by renaming.

Now we consider optimal codes for all the intermediate probability distributions occurring in the Huffman algorithm: We can pass between these intermediate distributions by *merging*, e.g. adding the smallest two numbers and e.g. passing from $\mathbf{p} := (p_1, p_2, \ldots, p_{m-2}, p_{m-1}, p_m)$ to $\mathbf{p}' := (p_1, p_2, p_{m-1} + p_m, \ldots, p_{m-2})$, and expanding, i.e. splitting the number $p_{m-1} + p_m$ into the two separate numbers p_{m-1} and p_m.

The expected code lengths are affected by these operations as follows:

If we have a code for the shorter distribution $\mathbf{p}'$, with expected code length $L(\mathbf{p}')$, then we can make it into a code for the longer distribution $\mathbf{p}$ by taking the code for the case $p_{m-1} + p_m$ and marking each of the two subcases p_{m-1} and p_m with an extra bit. The probabilities that these extra bits are used, are p_{m-1} and p_m, respectively, so the expected length of the expanded code is

$$L_{exp} = L + p_{m-1} + p_m$$

In the opposite direction, if we have an optimal code for the longer distribution $\mathbf{p}$, with expected code length $L(\mathbf{p})$, by the lemma we can assume that the least probable two cases p_{m-1} and p_m have codes which only differ by the last bit. We can delete this last bit and obtain a code for the shorter distribution $\mathbf{p}'$. The saving of the last bit occurs with probability $p_{m-1} - p_m$, so we have expected length

$$L_{merged} = L - p_{m-1} - p_m.$$

Now consider codes for the distributions $\mathbf{p}'$, $\mathbf{p}$ of *optimal lengths* $L(\mathbf{p}')$, $L(\mathbf{p})$. Adding together the above equations, we obtain

$$L_{exp}(\mathbf{p}') + L_{merged}(\mathbf{p}) = L(\mathbf{p}') + L(\mathbf{p}),$$

which implies

$$(L_{exp}(\mathbf{p}') - L(\mathbf{p}')) + (L_{merged}(\mathbf{p}) - L(\mathbf{p})) = 0.$$

Since the optimal lengths are the smallest possible, both the differences are positive, so they both must be zero.

This means that the operations of merging and expanding preserve optimality. Since the code in the last step, when we only have two cases, and code words of length 1 is clearly optimal, the Huffman code must be optimal. ♡

7.3 Information Theory and Statistics

Remember the slogan distinguishing the respective tasks of Probability theory and Statistics:

Probability theory: Given a random process, describe the emerging data.
Statistics: Given data, describe what process may have generated it.

The second of these tasks is usually drastically underdetermined. There is usually no "best", or clearly preferable, probability distribution that explains some given data. In choosing a statistical model, i.e. a parametrized family of probability distributions, we usually draw on additional knowledge or make additional assumptions, and thus narrow down the search space. Then we perform

With information theory we can shed new light on exponential families, maximum likelihood estimation and Bayesian statistics, and show that these notions and procedures arise from certain principles.

Exponential Families

Suppose we have a random variable X taking values in a set $\mathcal{X}$. For simplicity let us say that $\mathcal{X} = \{x_1, \dots, x_n\}$ is finite. Suppose further that we have some statistics (i.e. functions of the data), $T_1, \dots, T_k \colon \mathcal{X} \to \mathbb{R}$[1]. Given a number of samples from $\mathcal{X}$ we can estimate the expectations of these

[1] For example $\mathcal{X}$ could be the set of English words and T_1 could be 1 if the word has less than 8 letters and 0 otherwise, T_2 could be the number of occurrences of the letter e, etc. The distribution on $\mathcal{X}$ could be given by drawing words from this manuscript using the uniform distribution.

statistics. That is, our data gives us estimates $b_1, \dots, b_n$ for the expectations:

$$ET_1 = b_1, \dots, ET_k = b_k$$

Now our task is to find a probability distribution $p = (p_{x_1}, \dots, p_{x_n})$ on $\mathcal{X}$ which results in these expectations. There will usually be many different such distributions on $\mathcal{X}$. But now say that we assume that we know these values for the expectations and *that we know nothing else*!

If we interpret the *lack of knowledge* as maximal entropy, then we get a unique solution, namely the solution to the convex optimization problem

$$\begin{array}{ll} \text{maximize} & H(p) \\ \text{subject to} & p_{x_i} \geq 0 \\ & \sum_{x \in \mathcal{X}} p_x = 1 \\ & E_p T_i = \sum_{x \in \mathcal{X}} p_x T_i(x) = b_i \quad (\, i = 1, \dots, n) \end{array}$$

By Exercise 17 c), entropy is a concave function, and all constraints are affine, so this is really a convex optimization problem, and hence has a unique solution.

An alternative formulation would the following: We have seen in Lemma 7.1.16 that $H(p) = \log n - D(p\|u)$ where u is the uniform distribution. Since $D(p\|u) \geq 0$ for all p by Exercise 17 d), maximizing $H(p)$ is equivalent to minimizing $D(p\|u)$. Think about what this reformulation means: Among all those probability distributions that give us the required expectations we are looking for the one that is closest (in terms of the KL-distance) to the uniform distribution - and by Theorem 7.1.17 the latter has the highest possible entropy.

Instead of minimizing the distance to the distribution u of maximum entropy, we could also want to minimize the distance to some distribution q, that we see as plausible according to previous information (the typical Bayesian situation) - always while maintaining our requirement about the expectations. Then our optimization problem would be

$$\begin{array}{ll} \text{minimize} & D(p\|q) \\ \text{subject to} & p_{x_i} \geq 0 \\ & \sum_{x \in \mathcal{X}} p_x = 1 \\ & E_p T_i = \sum_{x \in \mathcal{X}} p_x T_i(x) = b_i \quad (\, i = 1, \dots, n) \end{array}$$

Let's start solving this problem. The Lagrangian has primal variables p_x ($x \in \mathcal{X}$) - the probability distribution - and dual variables λ_x ($x \in \mathcal{X}$) for inequality constraints $p_x \geq 0$, μ_0 for the equality constraint $\sum_x p_x - 1 = 0$ and $\mu_1, \dots, \mu_k$ for the equality constraints $\sum_x p_x T_i(x) - b_i = 0$ ($i = 1, \dots, k$). It is given by

$$L(p, \lambda, \mu) = \sum_x p_x \log \frac{p_x}{q_x} - \sum_x \lambda_x p_x - \mu_0 (\sum_x p_x - 1) - \sum_{i=1}^{k} \mu_i (\sum_x p_x T_i(x) - b_i)$$

The partial derivative by each primal variable p_x is

$$\frac{\partial}{\partial p_x} L(p, \lambda, \mu) = \log \frac{p_x}{q_x} + 1 - \lambda_x - \mu_0 - \sum_{i=1}^{k} \mu_i T_i(x)$$

Solving for p_x first gets us to

$$\log \frac{p_x}{q_x} = \lambda_x + \mu_0 + \sum_{i=1}^{k} \mu_i T_i(x) - 1$$

and then to

$$p_x = e^{\lambda_x + \mu_0 + \sum_{i=1}^{k} \mu_i T_i(x) - 1} \cdot q_x = q_x \cdot C \cdot e^{\sum_{i=1}^{k} \mu_i T_i(x)}$$

This is exactly the form of an exponential family, given in Def. 5.7.1. It is in the narrower form where b is the identity map, but remember that we remarked before Theorem 5.7.6 that any exponential family can be expressed in this form.

So the solution to our problem lies in an exponential family, and exponential families are exactly the families arising from minimizing KL-distance from some given distribution, while respecting certain expectations.

Maximum Likelihood Estimation

Maximum likelihood estimation can also be understood through information theory.

In the previous subsection we considered the collection of *all* probability distributions on a finite set, and wondered which one would maximize entropy under the constraints given by our data. With this we arrived at a statistical model (i.e. a parametrized family of probability distributions).

In maximum likelihood estimation instead one starts with a statistical model and then asks which member of the parametrized family best explains the given data.

Again we suppose that we have a random variable taking values in $\mathcal{X} := \{x_1, \ldots, x_m\}$, that we have a statistical model $\{p_\theta \mid \theta \in \Theta\}$ and data $D := (z_1, \ldots, z_n) \in \mathcal{X}^n$ that we suppose to have arisen from n independent draws.

From the data one can construct the so-called empirical distribution (which will usually not be part of the statistical model). It is defined by

$$p_D^{emp}(x) := \frac{1}{n} \sum_{i=1}^{n} \delta_{z_i}(x)$$

where $\delta_{z_i}(x)$ returns 1 if $x = z_i$ and 0 otherwise.

Usually the empirical distribution is not a good model: It declares impossible anything that has not yet been observed – often a case of extreme overfitting. Nevertheless, it is what best reflects the observed data.

If our statistical model was well chosen, it should reflect what observations are possible in principle and avoid the overfitting problem of the empirical distribution. So we could look for the closest (in the sense of KL-divergence) distribution to the empirical distribution that belongs to the statistical model.

In other words we want to find the θ minimizing

$$\begin{aligned}
D(p_D^{emp} \| p_\theta) &= \sum_{x \in \mathcal{X}} p_D^{emp}(x) \log \frac{p_D^{emp}(x)}{p_\theta(x)} \\
&= \sum_{x \in \mathcal{X}} p_D^{emp}(x) \log p_D^{emp}(x) - \sum_{x \in \mathcal{X}} p_D^{emp}(x) \log p_\theta(x) \\
&= -H(p_D^{emp}) - \sum_{x \in \mathcal{X}} \sum_{i=1}^{n} \delta_{z_i}(x) \log p_\theta(x) \\
&= -H(p_D^{emp}) - \sum_{i=1}^{n} \log p_\theta(z_i)
\end{aligned}$$

Therefore

$$\begin{aligned}
\operatorname{argmin}_\theta D(p_D^{emp} \| p_\theta) &= \operatorname{argmin}_\theta \left(-H(p_D^{emp}) - \sum_{i=1}^{n} \log p_\theta(z_i) \right) \\
&= \operatorname{argmin}_\theta \left(- \sum_{i=1}^{n} \log p_\theta(z_i) \right) \\
&= \operatorname{argmax}_\theta \left(\sum_{i=1}^{n} \log p_\theta(z_i) \right) \\
&= \operatorname{argmax}_\theta \left(\log \Pi_{i=1}^{n} p_\theta(z_i) \right) \\
&= \operatorname{argmax}_\theta \Pi_{i=1}^{n} p_\theta(z_i)
\end{aligned}$$

Here the step to the second line works because $-H(p_D^{emp})$ is independent of θ, and the step to the last line is by the usual argument that log is strictly monotonically increasing.

Now note that this last expression is exactly the definition of the maximum likelihood parameter: The product is the joint probability distribution of the i.i.d. random variables that we suppose to have given rise to our data.

Summary: Maximum likelihood estimation finds the member of a family of distributions that is closest to the empirical distribution of the observed data.

7.4 Differential Entropy

For each of the information theoretic notions that we have seen for discrete random variables, there is a corresponding notion for continuous random variables, obtained as a limit case by replacing sums with integrals and probability mass functions by density functions. In each case the behaviour and interpretation of the continuous notions are similar to the discrete case - except for the entropy!

Throughout this section let X be a continuous $\mathbb{R}^n$-valued random variable, with probability distribution $P(X \in A) = \int_A f(x)\, \mathrm{d}x$. We write $S := \{x \in \mathbb{R}^n \mid f(x) > 0\}$ and call this set the *support* of the density function f.

Definition 7.4.1. *The* differential entropy *of the random variable X is defined to be $h(X) := -\int_S f(x) \log f(x)\, dx$, if this integral exists.*

Clearly the differential entropy only depends on the density function, and therefore one can also write $h(f) := h(X)$.

Example 7.4.2. Consider a random variable X with uniform distribution on the interval $[0, a]$. The density function is the constant function $\frac{1}{a}$ on this interval, and so

$$h(X) := -\int_0^a \frac{1}{a} \log \frac{1}{a} \, dx = \left(x \cdot \frac{1}{a} \log a \right) |_{x=0}^{a} = \log a.$$

Note that for $a < 1$ this is negative – something that could not happen for the entropy of discrete random variables.

Example 7.4.3. Consider a normally distributed $\mathbb{R}$-valued random variable with expectation 0 and variance σ^2: $X \sim \phi(x) := \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{x^2}{2\sigma^2}}$.

$$\begin{aligned} h(\phi) &= -\int_{\mathbb{R}} \phi(x) \log(\phi(x)) \, dx = -\int_{\mathbb{R}} \phi(x) \left(-\tfrac{x^2}{2\sigma^2} - \ln \sqrt{2\pi\sigma^2} \right) dx \\ &\overset{(1)}{=} \tfrac{E(X^2)}{2\sigma^2} + \tfrac{1}{2} \ln(2\pi\sigma^2) \\ &\overset{(2)}{=} \tfrac{1}{2} + \tfrac{1}{2} \ln(2\pi\sigma^2) \\ &= \tfrac{1}{2} \ln e + \tfrac{1}{2} \ln(2\pi\sigma^2) = \tfrac{1}{2} \ln(2e\pi\sigma^2) \text{ nats} \end{aligned}$$

Here in equation (1) we used for the second summand the fact that in the term $\frac{1}{2} \ln(2\pi\sigma^2)$ there occurs no x and that $\int_{\mathbb{R}} \phi(x) dx = 1$. In equation (2) we used for the first summand that $\sigma^2 = \mathrm{Var}(X) = E(X^2) - (EX)^2 = E(X^2)$.

For a general univariate normal distribution, not necessarily with expectation 0, one gets the same result by Remark 7.4.6 below.

Discretization of continuous random variables

To better understand the meaning of differential entropy, consider the *discretization* for interval length Δ of a continuous $\mathbb{R}$-valued random variable X with density function f: We can subdivide the support of X into intervals of length Δ. Now we can define a new random variable X_Δ as follows:

In each interval $[i, i + \Delta]$, choose an x_i such that $f(x_i) \cdot \Delta = \int_i^{i+\Delta} f(x) \, \mathrm{d}x$ – this is possible by the mean value theorem from real Analysis. We define the random variable X_Δ by setting $P(X_\Delta = x_i) := f(x_i) \cdot \Delta$ and $P(X_\Delta =$

$z) = 0$ if z is none of the x_i. The function $g_\Delta(x_i) := f(x_i) \cdot \Delta$ is indeed a probability mass function because

$$\sum_{i=-\infty}^{\infty} g_\Delta(x_i) = \sum_{i=-\infty}^{\infty} \int_i^{i+\Delta} f(x)\,\mathrm{d}x = \int_{-\infty}^{\infty} f(x)\,\mathrm{d}x = 1$$

One can understand X_Δ as the random variable giving the value of X up to a certain precision. For example, if $\Delta = \frac{1}{100}$, then X_Δ gives two decimal digits of the value of X. Using the binary number system, the interval size $\Delta = \frac{1}{2^n}$ gives the value of X up to n binary digits after the point, and each digit is a 0 or a 1, and hence can be described by a bit. One says that this is the n bit approximation of X.

Theorem 7.4.4. *For a random variable X with (Riemann integrable) density function f we have*

$$\lim_{\Delta\to 0} H(X_\Delta) + \log\Delta = h(X)$$

Proof. The entropy of X_Δ is

$$\begin{aligned} H(X_\Delta) &= -\sum_{i=-\infty}^{\infty} g_\Delta(x_i)\log(g_\Delta(x_i)) \\ &= -\sum_{i=-\infty}^{\infty} f(x_i)\Delta\log(f(x_i)\Delta) \\ &= -\sum_{i=-\infty}^{\infty} f(x_i)\Delta\log(f(x_i)) - \sum_{i=-\infty}^{\infty} f(x_i)\Delta\log(\Delta) \\ &= -\sum_{i=-\infty}^{\infty} f(x_i)\Delta\log(f(x_i)) - \log(\Delta) \end{aligned}$$

Here in the 3rd line we turned the product inside the logarithm into a sum, and in the 4th line we used $\sum_{i=-\infty}^{\infty} f(x_i)\Delta = 1$. Now for $\Delta \to 0$ the first summand goes to $h(X) = \int_{-\infty}^{\infty} f(x)\log(f(x))\,\mathrm{d}x$ by definition of the Riemann integral. □

Examples 7.4.5.

Remark 7.4.6. Shifts in the values of a random variable do not affect entropy: $h(X+c) = -\int f(x-c)\log(f(x-c))\,dx = h(X)$.

The entropy of an $\mathbb{R}$-valued *discrete* random variable X does not change with rescaling, i.e. for $a \in \mathbb{R}$ we have $H(X) = H(a \cdot X)$. Indeed, aX takes different values, but still has a discrete distribution with exactly the same occurring probabilities, and that is all on which entropy depends. Intuitively, the countably many possible values are just stretched out, but do not cover more space or are assigned different weights.

For a continuous $\mathbb{R}$-valued random variable X, however, things do change. Intuitively the distribution is more spread out after multiplying with a factor $a > 1$, for example, which should lead to more uncertainty, and hence higher entropy. This is indeed the case:

Proposition 7.4.7. $h(aX) = h(X) + \log|a|$

Proof. Let f be the density function of X. By Prop. 4.6.3, the density function of $Y := aX$ is then given by $g(y) = f(\frac{y}{a}) \cdot |\frac{1}{a}|$.

$$\begin{aligned}
h(Y) &= -\int_{\mathbb{R}} g(y)\log(g(y))\,\mathrm{d}y \\
&= -\int_{\mathbb{R}} f(\frac{y}{a}) \cdot |\frac{1}{a}| \log(f(\frac{y}{a}) \cdot |\frac{1}{a}|)\,\mathrm{d}y \\
&= -\int_{\mathbb{R}} f(\frac{y}{a}) \cdot |\frac{1}{a}| \log(f(\frac{y}{a}))\,\mathrm{d}y + \int_{\mathbb{R}} f(\frac{y}{a}) \cdot |\frac{1}{a}| \log(|a|)\,\mathrm{d}y \\
&= -\int_{\mathbb{R}} f(\frac{y}{a}) \log(f(\frac{y}{a})) \cdot |\frac{1}{a}|\,\mathrm{d}y + \log(|a|) \\
&= -\int_{\mathbb{R}} f(x)\log(f(x))\,\mathrm{d}x + \log(|a|) \\
&= h(X) + \log(|a|)
\end{aligned}$$

Here in the change to the 5th line we use the change of variables formula for integrals. □

As an example, imagine that we stretch the density function of a random variable taking values in the interval $[0,1]$ by the factor $a = 4$. Intuitively we now need $\log_2 2^2 = 2$ more bits to describe in which quarter of the stretched interval our value lies; $[0,1], [1,2], [2,3]$ or $[3,4]$. The uncertainty coming from the overall shape of the density function is as before.

Remark 7.4.8. With an almost identical proof we have for $\mathbb{R}^n$-valued random variables X and a linear function g that $h(g(X)) = h(X) + \log|det(g^{-1})|$.

The following notions are defined as in the discrete case, and also satisfy analogous properties.

Definition 7.4.9. *(a) For $X_1, \dots, X_n$ $\mathbb{R}$-valued random variables with joint density function $f(x_1, \dots, x_n)$ the* joint entropy *is*

$$h(X_1, \dots X_n) := -\int f(x_1, \dots, x_n) \log(f(x_1, \dots, x_n))\, dx_1 \dots dx_n$$

(b) For X, Y $\mathbb{R}$-valued random variables with joint density function $f(x, y)$, the conditional entropy *is $h(X \mid Y) := -\int f(x, y) \log f(x \mid y)\, dxdy$ where $f(x \mid y) := \frac{f(x,y)}{f(y)}$ and $f(y) := \int f(x, y) dx$ is the marginal.*

(c) For density functions f, g we define the KL divergence *as $D(f \| g) := \int f \log \frac{f}{g}$ (setting $0 \cdot \log \frac{0}{0} := 0$)*

(d) The mutual information *of X and Y is $I(X; Y) := \int f(x, y) \log \frac{f(x,y)}{f(x)f(y)}$, where $f(x)$, $f(y)$ denote the marginals.*

Observation 7.4.10. We have $I(X; Y) = h(X) + h(Y) - h(X, Y)$, as one can see by expanding the fractions and products inside the logarithm in the definition of mutual information. We also have $I(X; Y) = D(f(x, y) \| f(x)f(y))$ directly from the definitions.

The properties of these notions are as in the discrete case, e.g. we have $D(f \| g) \geq 0$, chain rules, and the fact that the mutual information is zero if and only if the KL divergence is 0 - it is really only the entropy that behaves differently in the continuous case.

We end with a so-called *maximum entropy* result. These are statements identifying which is the distribution of highest entropy in a given class of distributions. They are important in Data Science for several reasons. For example, one is always looking for uninformative priors which do not carry any more information than what one is certain about. This lack of bias is reflected in a high entropy, as maybe has become intuitive by now.

A calculation shows that for a normally distributed random vector $X = (X_1, \dots, X_n) \sim N(\mu, K)$ with covariance matrix K we have $h(X_1, \dots, X_n) = \frac{1}{2} \log((2\pi e)^n \cdot \det K)$.

Theorem 7.4.11 (Cover/Thomas, Thm 8.6.5)**.** *Let $X = (X_1, \dots, X_n)$ be a random vector with expectation 0 and covariance matrix K. Then $h(X_1, \dots, X_n) \leq \frac{1}{2} \log((2\pi e)^n \cdot \det K)$, with equality if and only if X is normally distributed, $X \sim N(0, K)$.*

This means that among all distributions with a given covariance, the Gaussian has the maximal entropy. This fits in nicely with what we know about Gaussians so far: By the central limit theorem, Gaussians arise from averaging many independent influences – that is why one often assumes that noise over a signal has a Gaussian distribution. I think it makes some intuitive sense, that many independent influences create the least predictable kind of random data, and thus should have a high entropy.

7.5 Rate distortion theory (optional)

We end by considering the problem of describing the result of a continuous random variable $X\colon \Omega \to \mathcal{X} \subseteq \mathbb{R}^n$ by R bits ($R \in \mathbb{N}$). One can use the *Lloyd algorithm*:

Choose 2^R values in the target space $\mathbb{R}^n$. Call the set of these values $\widehat{\mathcal{X}}_0$. Then repeat the steps

1. Map each $x \in \mathcal{X}$ to the closest $\hat{x} \in \widehat{\mathcal{X}}_i$ – this creates a partition of $\mathcal{X}$ into regions (“Voronoi cells”)
2. For each cell find the point minimizing the average distance $d(x, \hat{x})$

until the average distances of step 2 are small enough.

Of course, this description of the algorithm leaves many details open. For example, for the definition of average distance in step 2. one has many choices of distance function. Very common is $d(x, \hat{x}) := \|x - \hat{x}\|^2$.

Definition 7.5.1. *(a) a* distortion function *is any function* $d\colon \mathcal{X} \times \widehat{\mathcal{X}} \to \mathbb{R}_{\geq 0}$

(b) the distortion *between finite sequences is* $d((x_1, \ldots, x_n), (\hat{x_1}, \ldots, \hat{x_n})) := \frac{1}{n}\sum_{i=1}^{n} d(x_i, \hat{x}_i)$

(c) a $(2^{nR}, n)$-rate distortion code *consists of*

- *an encoding function* $f_n\colon \mathcal{X}^n \to \{1, 2, 3, \ldots, 2^{nR}\}$ *and*
- *a decoding function* $g_n\colon \{1, 2, 3, \ldots, 2^{nR}\} \to \widehat{\mathcal{X}}^n$

(d) the distortion *associated with the distortion code* (f_n, g_n) *is:*

$$D := E\left(d\left((X_1, \ldots, X_n), g_n(f_n(X_1, \ldots, X_n))\right)\right)$$

$$= \begin{cases} \sum_{\mathbf{x}\in\mathcal{X}^n} p(\mathbf{x}) \cdot d(\mathbf{x}, g_n(f_n(\mathbf{x}))) & \textit{discrete case} \\ \int_{\mathcal{X}^n} f(x) \cdot d(x, g_n(f_n(x)))\, dx & \textit{continuous case} \end{cases}$$

Definition 7.5.2. *(a)* (R, D), *where* $R \in \mathbb{N}$ *and* $D \in \mathbb{R}_{\geq 0}$, *is called an* achievable rate distortion pair, *if there exists a sequence of* $(2^{nR}, n)$*-rate distortion codes* (f_n, g_n) *with* $\lim_{n\to\infty} E(d(\mathbf{x}, g_n(f_n(\mathbf{x})))) \leq D$.

(b) $R(D) := \inf\{R \in \mathbb{N} \mid (R, D)$ *achievable* $\}$ *is called the* rate distortion function

(c) $D(R) := \inf\{D \in \mathbb{R}_{\geq 0} \mid (R, D)$ *achievable* $\}$ *is called the* distortion rate function

The following is the main theorem of rate distortion theory:

Theorem 7.5.3 (Cover/Thomas, Thm. 10.2.1, p.306)**.**

$$R(D) = \inf\{I(X; \widehat{X}) \mid \widehat{X} \textit{ has distrib. satisfying } \sum_{(x,\hat{x})\in\mathcal{X}\times\widehat{\mathcal{X}}} p(x)p(\hat{x} \mid x)d(x, \hat{x}) \leq D\}$$

where in the above expression $p(\hat{x} \mid x)$ *is allowed to be any distribution.*

7.6 Some appearances of information theory in Data Science and Machine Learning

7.6.1 Choosing priors

One paradigm of constitutes a good prior says that it should be "uninformative", i.e. it should not carry any more information than what one is certain about. The intuitive concept of being uninformative can be made formal by asking for high entropy. This is not the only possible interpretation of what it means to be uninformative (others can be found on the Wikipedia section on uninformative priors), but one for which there are strong arguments. This view point is called the Principle of maximum entropy. Adopting this view point makes it possible to look systematically for uninformative priors - this now means solving a maximization problem.

In a different approach one chooses the prior in in such a way that the information obtained from the data is taken into account as strongly as possible. Again this intuitive idea needs to be formalized to be put into practice. The formal goal is then to find a prior such that that the Kullback-Leibler distance between prior and posterior is maximal. See this forum post for a short account and references. And see the book Robert: The Bayesian Choice: From Decision-Theoretic Foundations to Computational Implementation for a thorough explanation.

7.6.2 Independent component analysis

In Independent Component Analysis one is given a random vector and asks whether there is a base change of $\mathbb{R}^n$ that makes the component into independent random variables. A famous instance of this problem is the Cocktail Party problem: At a cocktail party where everybody is talking at once, i.e. where the soundscape consist of a mixture of independent conversations, one tries to listen to one particular conversation. The sound (i.e. the random vector representing the sound) may initially be given by volume levels in different frequency bands, and one is looking for a transformation such tht the components are the different individual conversations.

This can be made formal as a minimization problem: One wants to find a matrix such that base change with this matrix minimizes the mutual information between the components (there is a notion of mutual information of more than two random variables).

A good reading for this is Hyvärinen: New Approximations of Differential Entropy for Independent Component Analysis and Projection Pursuit. This article also addresses the question of how to actually estimate the entropy from data.

7.6.3 Variational Autoencoders

For a very rough idea on variational autoencoders, see the lecture – or not, if there was no time. Here is a pretty thorough tutorial: Kingma/Welling: An Introduction to Variational Autoencoders. And here is another, slightly shorter: Doersch: Tutorial on Variational Autoencoders. In these tutorials You will see KL-divergence pop up in crucial places, most importantly in the *evidence lower bound* (ELBO).

8 Monte Carlo Methods

The term Monte Carlo methods summarizes methods for the computation of (deterministic) numbers by random sampling. A very early example is the computation of π via Buffon's needle/noodle: Here one throws a number of needles (or noodles) on a floor with vertical lines an counts how often they cross a line. For a large enough number of needles (noodles) one can obtain arbitrarily close approximations of π with prescribed certainty.

More specifically, the idea is to compute an integral or an infinite sum by seeing it as (or expressing it via) an expectation and using the law of large numbers and the central limit theorem to approximate that expectation up to a desired precision and certainty.

Sources for this chapter:

For a short account in the context of Machine Learning, see MacKay: Information Theory, Inference and Learning Algorithms, Chapter 29.

A an accessible free source is Kroese: Monte Carlo Methods, Course notes which is a condensed version of the non-free comprehensive reference Kroese et al.: Handbook of Monte Carlo Methods

For a thorough and comprehensive reference with detailed explanations see the unfinished, freely available book Owen: Monte Carlo theory, methods and examples

The book Gilks, Richardson, Spiegelhalter: Markov Chain Monte Carlo in Practice contains descriptions of concrete practical applications of Markov Chain Monte Carlo Methods, and addresses the issues arising in practice, like convergence control. Yo may want to keep it in mind for the day when you want to use Monte Carlo Methods on your Data Science Project...

8.1 Simple Monte Carlo

Imagine you want to compute an integral, a large (possibly infinite) sum, or a probability. In each case we can express the number in question via an expectation:

- Let Y be a random variable, and A be a subset of its range. Then $P(Y \in A) = EI_A$, where I_A denotes the indicator function which gives back 1 for elements of A and 0 otherwise.

- The value of an integral $\int_a^b f(x)\,\mathrm{d}x$ can be obtained from an expectation as follows: Consider a random variable X with uniform distribution on the interval $[a,b]$, and call its density function u. Thus $u(x) = \frac{1}{b-a}$ for all $x \in [a,b]$. The we have

  $$(b-a)\int_a^b f(x)\frac{1}{b-a}\,\mathrm{d}x = (b-a)\int_a^b f(x)u(x)\,\mathrm{d}x = Ef(X)$$

  More generally we have, for any density function p on $A \subseteq \mathbb{R}^n$ and random variable X with that density, $\int_A f(x)p(x)\,\mathrm{d}x = Ef(X)$. Notice that this also includes integrals over unbounded domains $A \subseteq \mathbb{R}^n$.

- If $q\colon \mathcal{X} \to \mathbb{R}$ is a function on a countable set $\mathcal{X}$, and p is a probability mass function on $\mathcal{X}$, then for a random variable with that probability mass function we have

  $$\sum_{x\in\mathcal{X}} q(x)p(x) = Eq(X)$$

The basis of Monte Carlo methods is the fact that we can estimate expectations of random variables in a very easy and controlled way, and thereby can approximately compute the other side of the equation.

For concreteness, let's stick to the problem of computing $\int_A f(x)p(x)\,\mathrm{d}x = Ef(X)$. Consider i.i.d. random variables X_i all having the distribution of X, and set $S_N := \frac{1}{N}\sum_{i=0}^N f(X_i)$ By linearity of the expectation we have

$$ES_N = \frac{1}{N}\sum_{i=0}^N Ef(X_i) = \frac{1}{N}\sum_{i=0}^N Ef(X) = Ef(X).$$

This is true for every N. The variance of S_N on the other hand is getting lower with rising N: $\mathrm{Var}(S_N) = \mathrm{Var}(\frac{1}{N}\sum_{i=0}^N f(X_i)) = \frac{1}{N^2}\mathrm{Var}(\sum_{i=0}^N f(X_i)) =$

$\frac{1}{N^2}\sum_{i=0}^{N}\operatorname{Var}(f(X_i)) = \frac{1}{N^2}N\operatorname{Var}(f(X)) = \frac{1}{N}\operatorname{Var}(X)$ (where we used that the variance of a sum of *independent* random variables is the sum of their variances). Thus it gets more likely to be close to the expectation as we draw bigger samples. If we want to estimate how many samples we need in order to be within a given range of the true value with a given probability we can use the Chebyshev inequality:

Proposition (Chebyshev inequality, Prop. **??**)**.** For an $\mathbb{R}^n$-valued random variable Z and a $\delta \in \mathbb{R}$ we have the inequality

$$P(\|Z - EZ\| \geq \delta) \leq \frac{\operatorname{Var}(Z)}{\delta^2}$$

For our average-of-samples random variable S_N, the Chebyshev inequality yields

$$P(|S_N - Ef(X)| \geq \delta) \leq \frac{\operatorname{Var}(X)}{N\delta^2} \qquad (*)$$

If we know the variance of our random variable X, we can use this to compute the number of samples needed to be within a given range of the expectation with a given probability:

If we want to be within a distance δ of the (typically unknown) true value $Ef(X)$ with probability at least $1-\epsilon$, then we need to ensure that the inequality $P(|S_N - Ef(X)| \geq \delta) \leq \frac{\operatorname{Var}(X)}{N\delta^2} \leq \epsilon$ holds, i.e. that $N \geq \frac{\operatorname{Var}(X)}{\epsilon\delta^2}$.

In summary: We can compute N such that for N samples $x_1, \ldots, x_N$, with very high probability the number $\frac{1}{N}\sum_{i=0}^{N} f(x_i)$ is close to $Ef(X)$. More vaguely, but more succintly, the idea of Monte Carlo approximation is:

$$\int_S f(x)p(x)dx \approx \frac{1}{N}\sum_{i=0}^{N} f(x_i)$$

The right hand side is often easier to compute, but it requires that we have samples $x_1, \ldots, x_N$ which were drawn according to the probability distribution with density function $p(x)$. The remaining question is then how to obtain such samples! In the rest of this chapter, we discuss some techniques for this.

The above justification of the Monte Carlo method assumed that both the expectation $Ef(X)$ and the variance $\operatorname{Var}(X)$ are finite. Indeed it can fail if the expectation is ∞ or if the expectation is finite but the variance is ∞. For realistic examples of both cases see section 2.8 of Owen's book.

8.2 Random and pseudorandom numbers

Not covered in the exam.

For now let me just summarize this:

There are random number generators and pseudorandom number generators.

In the justification of the sampling methods above we supposed that we have access to true random numbers. This is difficult. It is really only possible through measuring physical phenomena.

This can be done through hardware random number generators, which for example are measuring optical effects taking place in a little box that can be connected to a computer.

Alternatively there are some online services transmitting the results:

- `https://www.random.org/` — uses atmospheric noise
- `https://www.fourmilab.ch/hotbits/` — uses radioactive decay
- `http://qrng.anu.edu.au/index.php` — uses quantum fluctuations of the vacuum

Other, supposedly less reliable, sources of random numbers are built into the hardware of many computers and use e.g. electric circuit noise from the mainboard, timestamps of user actions or mouse movements. On Linux systems the results are stored in a file called `/dev/random`, other operating systems have similar devices.

For most purposes one uses, however, so-called pseudorandom number generators. These are deterministic algorithms which, given a parameter called the *seed*, produce a number. The numbers coming up for different seeds ideally have many properties of random numbers. There is a series of test of randomness, and pseudorandom number generators can have wildly varying quality, meaning that they can perform better or worse in those tests. One set of tests is the Dieharder Random Number Test Suite.

Python uses the Mersenne twister algorithm, which is good enough for Monte Carlo integration, but not for secure cryptography. The aim of this section is simply to raise awareness that there are “random numbers” of different quality and that one should make sure to take a sufficiently good one for one’s intended application.

In the article Jones: Good Practice in (Pseudo) Random Number Generation for Bioinformatics Applications the author warns pronouncedly against using the in-built random number generation functions of programming languages. The Python `rand()` function is deemed ok by him, though.

Remark 8.2.1. A theoretical measure of randomness is given by Kolmogorov complexity, i.e. the length of the shortest program in a universal computer that produces a given number. Typically, a truly random number cannot be produced by a program of much shorter length, yet this is precisely what pseudorandom number generators do. This is one way in which pseudorandom numbers are different from true random numbers.

A Forum post where George Marsaglia introduced a bunch of random number generators.

8.3 The inversion method

Suppose you want to sample from a distribution on $\mathbb{R}$ with density p and cumulative distribution function $F(x) = \int_{-\infty}^{x} p(x)dx$. Suppose that p is non-zero on a single interval. Then $F(x)$ is strictly monotonically increasing, and therefore has an inverse F^{-1}.

Now to sample from our distribution, we can instead sample from the random variable U with uniform distribution on the interval $[0, 1]$ and apply the function F^{-1} to the results, i.e. consider $X := F^{-1}(U)$. We then have

$$P(X \leq x) = P(F^{-1}(U) \leq x) = P(U \leq F(x)) = F(x).$$

So our random variable X has the required distribution.

In this way one can sample from many distributions, e.g. 1-dimensional Gaussians, provided one can sample from the uniform distribution.

Downsides: 1. This only works for 1-dimensional distributions.

2. One needs to be able to calculate the inverse of the cumulative distribution function of the target distribution. This can be satisfied either by a closed formula (e.g. in the case of the exponential distribution) or by a table (e.g. in the case of the Gaussian). But this can be a problem for some distributions.

8.4 Rejection sampling

The standing assumption for this and the next section is that we have a probability distribution with density function $p(x)$, that we know to compute for a given value x, but for which is maybe hard to produce samples.

Instead of producing samples for the target distribution with density $p(x)$, which we suppose is too difficult, we choose a distribution with density $q(x)$, for which we know how to draw samples. One could for example choose q to be a uniform or a Gaussian distribution.

Rejection sampling works as follows:
First rescale q, i.e. choose a $c \in \mathbb{R}$ such that $p(x) \leq c \cdot q(x)$ for all $x \in \mathbb{R}$. Thus the area below the graph of q completely contains the area below the graph of p. Now repeat the following steps until you have enough samples:

1. Produce a sample x_i for q
2. Produce a sample y_i for the uniform distribution on the interval $[0, q(x_i)]$
3. If $y_i \leq p(x_i)$, accept the sample x_i, otherwise reject it.

Inuitively, at the places where p is much smaller than q, it is much more likely that in the second and third steps we come to reject the q-sample – this corrects the difference between q that we are sampling from and p that we actually want to sample from. At the places where p is almost as big as q it is likely that our samples get accepted, which is fine, because there is not much to correct.

8.5 Importance sampling

This method applies if we want to compute an integral by a Monte Carlo method.

Again, instead of producing samples for the target distribution with density $p(x)$ (difficult) we choose a distribution with density $q(x)$, for which we know how to draw samples.

We simply use the fact that

$$\int f(x)p(x)\,dx = \int f(x)\frac{p(x)}{q(x)}q(x)\,dx,$$

Since, by our choice of q, we know how to draw samples from q, we can now approximate the integral by inserting enough samples for the distribution with density q into the function $f(x)\frac{p(x)}{q(x)}$:

$$\int_S f(x)p(x)dx \approx \frac{1}{N}\sum_{i=0}^{N} f(x_i)\frac{p(x_i)}{q(x_i)}$$

8.6 Markov Chain Monte Carlo Methods

The idea of Markov Chain Monte Carlo Methods is to produce, instead of independent samples, i.e. i.i.d. random variables, a Markov chain of random variables which in the limit together have our target distribution.

Metropolis-Hastings algorithm

We consider a target density $p\colon \mathbb{R}^n \to \mathbb{R}_{\geq 0}$ from which we want to sample. Again, we choose an auxiliary function, $q\colon \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}_{\geq 0}$ such that for every $\tilde{x} \in \mathbb{R}^n$ the function $q(-;\tilde{x})\colon \mathbb{R}^n \to \mathbb{R}_{\geq 0}$ is a density function, i.e. integrates to 1. $q(-;\tilde{x})$ is called the *proposal distribution* at $\tilde{x}$.

Examples 8.6.1. (a) $q(x;\tilde{x}) := N(x;\tilde{x},K)$, the normal distribution with mean $\tilde{x}$ and some fixed covariance matrix K.

(b) $q(x;\tilde{x}) := \frac{1}{2}N(x;\tilde{x}-1,K) + \frac{1}{2}N(x;\tilde{x}+1,K)$

We choose a starting point x_0 as a sample from some distribution, e.g. the uniform distribution. Then repeat the following steps:

Given a sample x_n,

1. sample x' from the distribution with density $q(-;x_n)$
2. sample $d \in [0,1]$ from the uniform distribution
3. if $d \leq \frac{p(x')}{p(x_n)}\frac{q(x_n;x')}{q(x';x_n)}$ then $x_{n+1} := x'$, else $x_{n+1} := x_n$

To get a hint of what the algorithm does, consider the case when $q(x;x') = q(x';x)$ (this was the case in our examples a) and b) above). In this case the algorithm is just called the Metropolis algorithm, since this was the original case considered by Metropolis, and Hastings extended it to the general case.

In the Metropolis algorithm we have $\frac{p(x')}{p(x_n)}\frac{q(x_n;x')}{q(x';x_n)} = \frac{p(x')}{p(x_n)}$, so steps 2. and 3. can be rephrased as follows:

"If x' satisfies $p(x') \geq p(x_n)$, then accept it as the next sample, $x_{n+1} = x'$. If not, accept it with probability $\frac{p(x')}{p(x_n)}$."

Thus if in our target distribution the newly drawn sample x' is more likely than the old x_n, we take it as the next one. If not, we still consider it, but with less enthusiasm for less likely samples.

More concisely one could say that x' is accepted as the next sample with probability $\min(1, \frac{p(x')}{p(x_n)})$ (and in the non-symmetric Metropolis-Hastings case with probability $\min(1, \frac{p(x')}{p(x_n)} \frac{q(x_n;x')}{q(x';x_n)})$).

Observation 8.6.2. Note that as long as we start with an x_0 for which we have $p(x_0) > 0$, it can never happen that $p(x_{n+1}) = 0$, because then the acceptance probability would be zero. Thus we can also never divide by zero in the fraction $\frac{p(x')}{p(x_n)}$. Likewise $q(x'; x_n) > 0$ always, because we obtain x' as a sample from the distribution $q(-; x_n)$, so there is also no danger of dividing by zero in the expression $\frac{p(x')}{p(x_n)} \frac{q(x_n;x')}{q(x';x_n)}$.

The Metropolis-Hastings algorithm has some resemblance with Rejection sampling. One difference is that in each cycle we get a new sample: "Rejection" here means that we get another copy of the same sample, instead of getting nothing. So if we knew that the algorithm actually produced samples from our target distribution, there would be no issue of having to wait for a sufficient number of samples.

This, however, is not quite true. Instead it is asymptotically true: The algorithm defines a Markov chain, and if that Markov chain is ergodic, it has a stationary distribution which turns out to be our target distribution. Thus no matter where we start, in the long run the distribution of the values that we produce, will converge to the target distribution. We will also see that the samples will be approximately independent.

The Metropolis-Hastings algorithm defines a Markov chain

This is true by design, because the next output only depends on the previous output, and not on the entire history. Explicitly, define the acceptance probability by

$$A(x'; x_n) := \min(1, \frac{p(x')}{p(x_n)} \frac{q(x_n; x')}{q(x'; x_n)})$$

the Metropolis-Hastings algorithm starts by choosing some starting distribution (i.e. a distribution for X_0). Then the further random variables X_n are defined via transition probabilities (for countably many possible states)

$$P_{x',x} := P(X_{n+1} = x' | X_n = x_n) = \begin{cases} q(x'; x_n) \cdot A(x'; x_n) & \text{if } x' \neq x_n \\ q(x_n; x_n) + \sum_{x' \neq x_n} q(x'; x_n)(1 - A(x'; x_n)) & \text{else} \end{cases}$$

For proving that this Markov chain has our target distribution as a stationary distribution, we will use the following concept:

Definition 8.6.3. *Consider a Markov chain with set of states I and transition probabilities P_{ij} (i.e. the probability of going from state i to state j). A distribution $(p(i))_{i \in I}$ on I satisfies the* detailed balance condition *if $p(i)P_{ij} = p(j)P_{ji}$ for all $i, j \in I$.*

The detailed balance condition roughly says that going back and forth between states i and j are equally likely.

Lemma 8.6.4. *A distribution satisfying the detailed balance condition is stationary.*

Proof. Call the distribution $p = (p(i))_{i \in I}$. We have

$$p(i) = p(i) \sum_j P_{ij} = \sum_j p(j) P_{ji}$$

where the first equality holds because we are summing over a row of a stochastic matrix (i.e. the sum is 1) and he second by the detailed balance condition. But the right hand side is the formula for the ith entry of the product of the vector $p = (p(i))_{i \in I}$ with the matrix (P_{ij}). Altogether we get: $p = pP$. □

If you imagine the Markov chain as a network in which some substance or quality flows between the states, then a stationary distribution is a distribution such that in each state there is equal amount flowing in as is flowing out. This is a global condition considering all influxes together. The detailed balance condition, in contrast, is a stronger local condition, saying that in any given state, to each other state there is as much outgoing as is coming back from there. There are indeed Markov chains with stationary states that do not satisfy the detailed balance condition.

Proposition 8.6.5. *For the Markov chain of the Metropolis-Hastings algorithm, the target distribution p satisfies the detailed balance condition.*

Proof. Note that the acceptance probabilities $A(x';x) := \min(1, \frac{p(x')}{p(x)}\frac{q(x;x')}{q(x';x)})$ satisfy

$$\frac{A(x';x)}{A(x;x')} = \frac{p(x')}{p(x)}\frac{q(x;x')}{q(x';x)}$$

Indeed, if $p(x')q(x;x') > p(x)q(x';x)$, then $A(x';x) = 1$ and $A(x;x') = \left(\frac{p(x')}{p(x)}\frac{q(x;x')}{q(x';x)}\right)^{-1}$. Otherwise $A(x;x') = 1$ and $A(x';x) = \frac{p(x')}{p(x)}\frac{q(x;x')}{q(x';x)}$.

It follows that for $x' \neq x$ we have

$$\frac{P_{x',x}}{P_{x,x'}} = \frac{A(x';x)}{A(x;x')}\frac{q(x';x)}{q(x;x')} = \frac{p(x')}{p(x)}$$

and therefore

$$p(x')P_{x',x} = p(x)P_{x,x'}$$

□

A Markov chain satisfying the detailed balance condition is also called a *reversible Markov chain*.

Theorem 8.6.6. *If $q(x;\tilde{x}) > 0$ for all $x, \tilde{x}$, then for $n \to \infty$ the distribution of the random variable X_n defined by the Metropolis-Hastings algorithm converges to a random variable with our target distribution p.*

Proof. By Prop. 8.6.5 we have the target distribution as stationary distribution. By assumption we have a regular Markov chain, and hence by Thm. 6.3.13 there is a unique stationary distribution to which every starting distribution converges. □

Finally, we wanted *independent samples*, but we also do approximately get that, because of the form of the limiting transition matrix, as told in Prop. 6.3.15: All rows of this matrix are equal, which means that the probabilities of the next states are the same, no matter what state we start in. This means that, although consecutive samples are not independent from each other, the whole collection of (late enough) samples is as if it had been drawn independently from the stationary distribution.

In case one really needs approximately independent samples, one can only take every kth result of the Markov chain for some large enough k: The longer the gap between two random variables in the Markov chain, the closer they are to being independent (to be precise: for a Markov chain one can show $\lim_{n\to\infty} I(X_0; I_n) = 0$, and mutual information measures independence).

For more on the Metropolis-Hastings algorithm see Chib/Greenberg: Understanding the Metropolis-Hastings algorithm.

Gibbs sampling

Gibbs sampling is a way to break down the task of sampling from a high-dimensional multivariate distribution into repeatedly sampling from 1-dimensional distributions: One simply samples coordinate by coordinate.

Suppose we want to draw samples according to the probability mass function $p(x_1, \dots, x_d)$. The algorithm is as follows: Start with an arbitrary point. Given a sample $(x_1^{(n)}, \dots, x_d^{(n)})$, produce the next sample as follows:

1. Pick $j \in \{1, \dots, d\}$ according to the uniform distribution.
2. Pick a new value for the j-th coordinate, using the 1-dimensional marginal distribution $p(x_1^{(n)}, \dots, x_{j-1}^{(n)}, -, x_{j+1}^{(n)}, \dots, x_d^{(n)})$.

By construction, the next point only depends on the previous point and the no other parts of the sequence, so this process defines a Markov chain. We analyze the transition probabilities:

For two points $x = (x_1, \dots, x_d)$ and $y = (y_1, \dots, y_d)$ we write $x \sim_j y$ ("x and x are j-similar") if $x_i = y_i$ for all $i \neq j$, i.e. if x and y differ at most at the j-th coordinate.

Any two consecutive points obtained from our Markov chain above are j-similar, by construction. The transition probabilities are therefore

$$P_{xy} = \begin{cases} \frac{1}{d} \frac{p(y)}{\sum_{z \sim_j x} P_{xz}} & \text{if } x \sim_j y \\ 0 & \text{otherwise} \end{cases}$$

With this we can see that our Markov chain satisfies the detailed balance condition for the target distribution p:

$$p(x)P_{xy} = \frac{p(x)p(y)}{\sum_{z \sim_j x} P_{xz}} = \frac{p(y)p(x)}{\sum_{z \sim_j y} P_{xz}} = p(y)P_{yx}$$

here the middle equality (which only has a change in the denominator) come from the fact that a point z is j-similar to x if and only it is j-similar to y.

So we have a Markov chain that converges to the target distribution p.

The Markov chain can be pictured as a random walk in $\mathbb{R}^d$, where at every step we change the direction by a right angle (or not at all).

A possible problem: Consider a distribution on just 4 points in $\mathbb{R}^2$, namely $(0,0),(1,0),(0,1),(1,1)$ such that $p(0,0) = p(1,1) = \frac{1}{2}$ and $p(0,1) = p(1,0) = 0$. If we start in the position $(1,1)$, we can never get to the other possible value $(0,0)$, because changing one coordinate at a time we would have to pass through an impossible region. More generally, we could assign very small but non-zero probabilities to $(1,0),(0,1)$ – then we could expect to eventually get from $(1,1)$ to $(0,0)$, but the expected waiting time for this transition would be very long.

A remedy for this type of problem is to not just change one coordinate at a time but change correlated coordinates together, in blocks. This is called *block sampling*.

The final question for Gibbs sampling is then, how to sample from the marginal distributions or the block distributions. The answer is: By any of the previously discussed methods. Gibbs sampling needs to be combined with other methods and is really a vehicle to avoid the problems of high dimensionality.

Convergence

A practical question arising for all Markov chain Monte Carlo methods is how to know when the Markov Chain evolved long enough to be close to the stationary distribution. One answer is that for reversible Markov chains one can actually estimate convergence rates precisely. But another answer is that we don't really need to do that, but instead can use a technique called Coupling from the past. It is very thoroughly explained in these notes, and explained by Haskell code in this blog post.