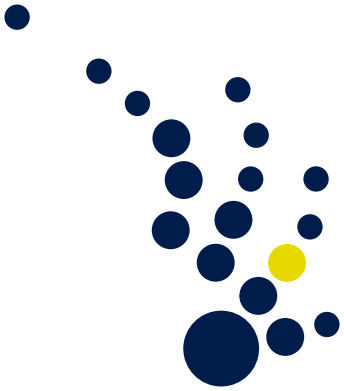


Institut für Numerische Simulation

Rheinische Friedrich-Wilhelms-Universität Bonn

Scientific Computing II

Kernel Methods and Nonlinear Dimensionality Reduction

held by:

Prof. Dr. Jochen Garcke

August 11, 2021

Contents

1. Kernel based methods

Let

$$\{(x_i, \hat{f}_i)\}_{i=1}^N$$

with $x_i \in \mathbb{R}^d$, $f_i \in \mathbb{R}$.

Aim: Find a "good" function f such that

$$f(x_i) = \hat{f}_i \quad i = 1, \ldots, N$$

To compute an f, we can make of a discrete representation of f in some basis, i.e.

$$f(x) = \sum_{j=1}^{N} c_j b_j(x).$$

For interpolation, one can solve this via

$$BC = \hat{F},$$

where $B_{kj} = b_j(x_k)$, $j, k = 1, \ldots, N$, $C = (c_1, \ldots, c_N)^T$, and $\hat{F} = (\hat{f}_1, \ldots, \hat{f}_N)^T$. If B is a nonsingular matrix we have a solution. It turns out that so-called kernel functions that are centered at the locations x_i are a good choice:

$$b_j(x) = k(x_j, x),$$

which gives

$$f(x) = \sum_{j=1}^{N} c_j k(x_j, x).$$

We will also consider approximation instead of interpolation $f(x_i) \approx \hat{f}_i$. This is in particular relevant in machine learning, where one usually assumes, and actually has, noise and measurement errors in the given data.

1.1. Kernels

The Gaussian kernel is the prime example of a kernel:

$$k(x, y) \coloneqq \exp\left(-\alpha \|x - y\|_2^2\right)$$

for all $x, y \in \mathbb{R}^d$, where α is a scaling parameter.

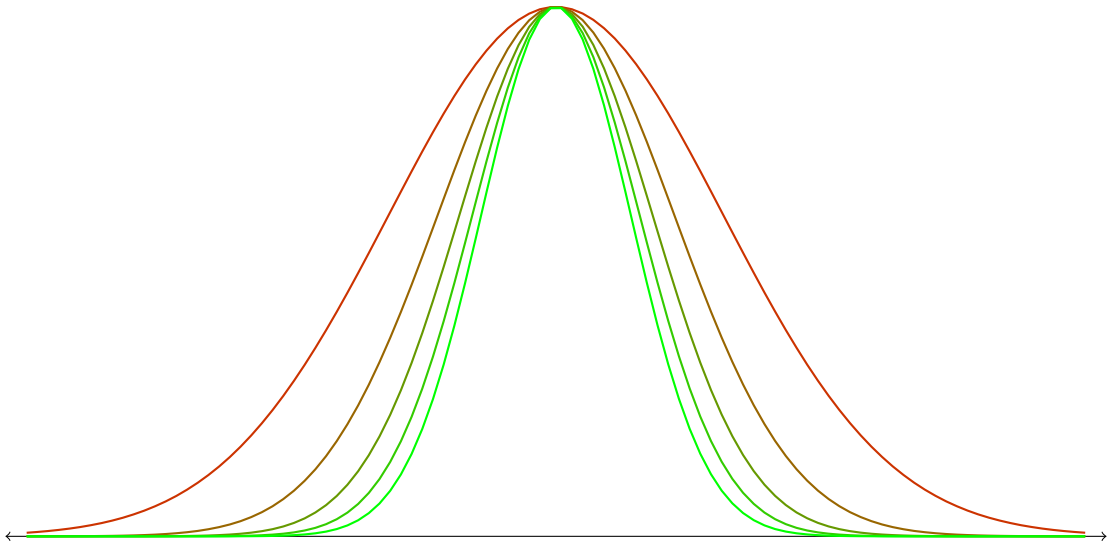


Figure 1.1.: Gaussian kernel for various α values

Definition 1.1. Let Ω be an arbitrary nonempty set. A function $k : \Omega \times \Omega \to \mathbb{R}$ is called a *kernel* on Ω. We call k a *symmetric kernel* if

$$k(x, y) = k(y, x)$$

for all $x, y \in \Omega$.

The Gaussian kernel can be written as

$$\begin{aligned} k(x, y) &= \exp(-\alpha \underbrace{\|x - y\|_2^2}_{r \coloneqq \|x-y\|_2}) \\ &= \exp(-\alpha r^2) \\ &= \phi(r) \\ &= \phi(\|x - y\|_2) \end{aligned}$$

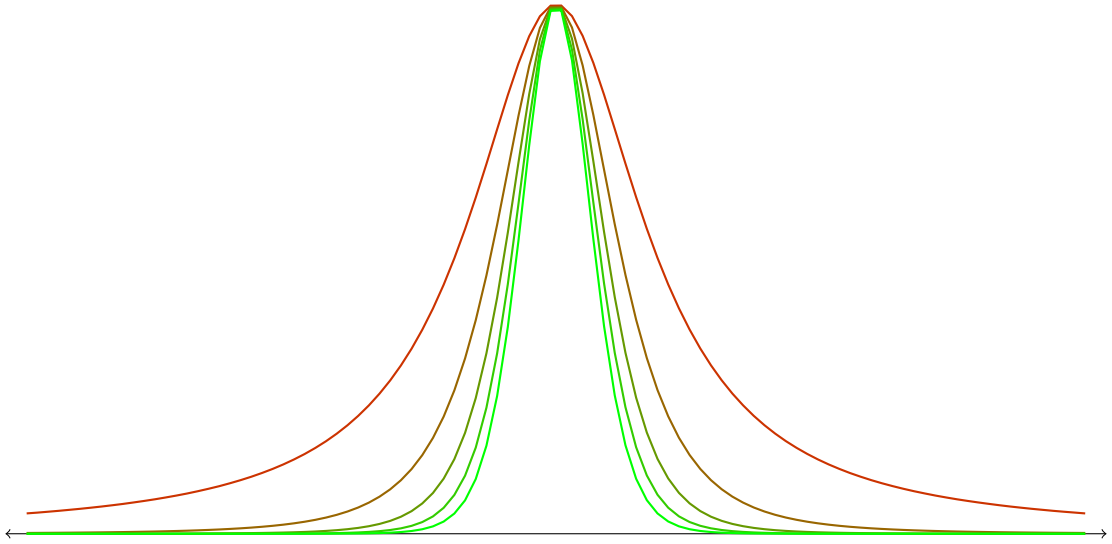


Figure 1.2.: Inverse multiquadratics for various β values

Radial basis functions

Definition 1.2. A function $\Phi : \mathbb{R}^d \to \mathbb{R}$ is said to be radial if there exists a function $\phi : [0, \infty] \to \mathbb{R}$ such that

$$\Phi(x) = \Phi(\|x\|_2)$$

for all $x \in \mathbb{R}^d$. Such a function is traditionally called *radial basis function (RBF)*.

Other common RBFs are

- inverse multiquadratics $\phi(r) = (1 + \alpha r^2)^\beta$, $\beta < 0$
- multiquadratics, $\phi(r) = (1 + \alpha r^2)^\beta$, $\beta > 0$
- powers: $\phi(r) = r^\beta$, $\beta \notin 2\mathbb{Z}$.

  These belong to the

- polyharmonic family of kernels:

  $$\phi(r) = r^\beta \log(|r|), \quad \beta \in 2\mathbb{Z}.$$

  Special case:

  $$\phi(r) = r^2 \log(|r|).$$

  This is the so-called thin-plate spline. It relates to the partial differential equation that describes the bending of thin plates.

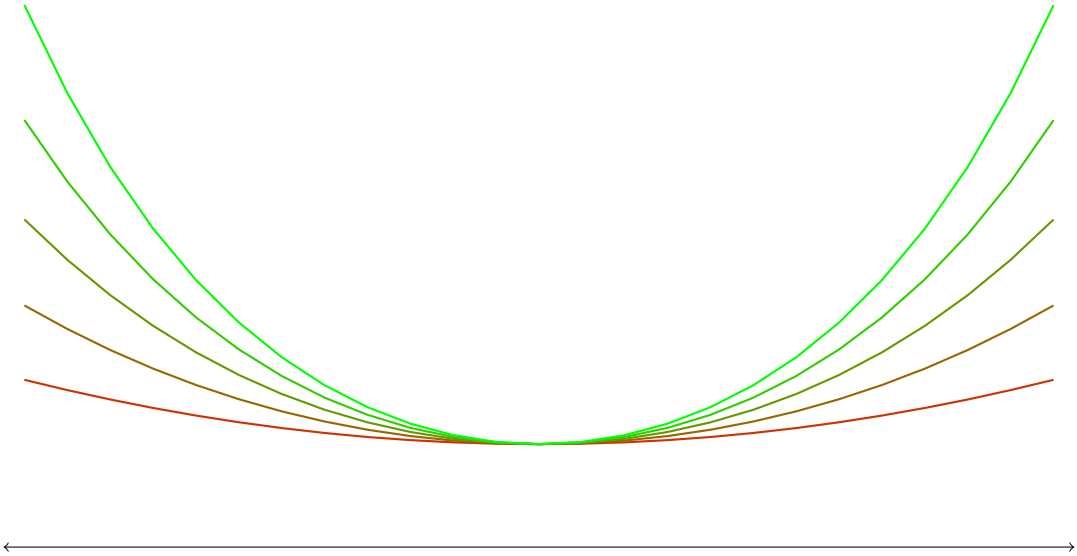


Figure 1.3.: Multiquadratics for various β values

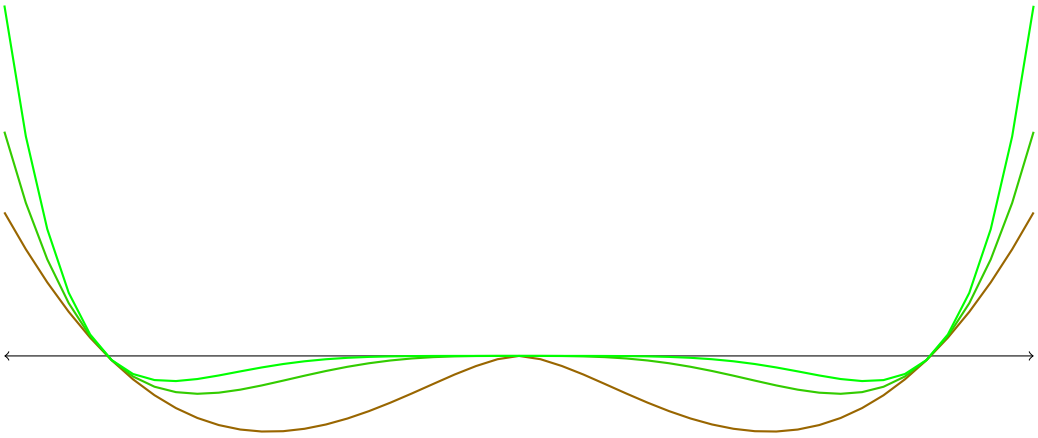


Figure 1.4.: Polyharmonic kernel for various β values

- In contrast to the other kernels that live over $\mathbb{R}$, Wendland's kernel is a compactly supported kernel:
$$\phi_{a,1}(r) = (1-r)_+^{(a+1)}(1+(a+1)r)$$
with the cut-off function
$$(x)_+ \coloneqq \begin{cases} x, & x \geq 0 \\ 0, & x < 0 \end{cases}$$

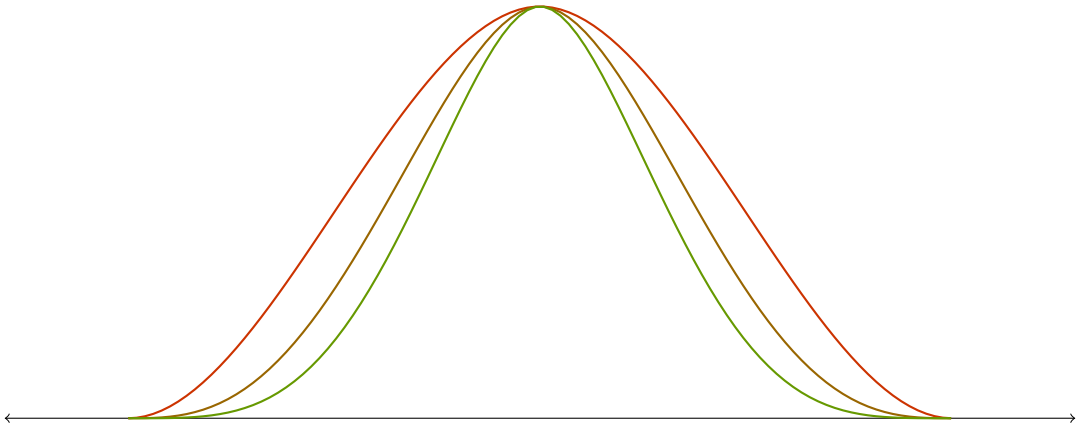


Figure 1.5.: Wendland's C^2-kernel for various a

Remark. Kernels can always be restricted to subsets without losing essential properties. This easily allows kernels on embedded manifolds, e.g. the sphere.

Remark. We will see that a kernel k on Ω defines a function $k(x,\cdot)$ for all fixed $x \in \Omega$. The space
$$\mathcal{K}_0 \coloneqq \operatorname{span}\{k(x,\cdot) \mid x \in \Omega\}$$
can be used as a so-called trial space in meshless methods for solving PDEs.

Kernels in Machine Learning

In machine learning the data $x \in \Omega$ can be quite diverse and without (much) structure on first glace. For example consider images, text documents, customers, graphs, ...

Here, one views the kernel as a similarity measure, i.e.
$$k : \Omega \times \Omega \to \mathbb{R}$$
returns a number $k(x,y)$ describing the similarity of two patterns x and y.

If in $\mathbb{R}^d$, we can work with the standard scalar product

$$\langle x, y \rangle = \sum_{i=1}^{d} x_i \cdot y_i,$$

where x and y are very dissimilar if they are orthogonal and x and y are very similar if they point in the same direction. As a reminder, if normalized the scalar product computes the cosine of the angle between x and y.

To work with general data, we first need to represent it in a Hilbert space $\mathcal{F}$, the so called *feature space*. One considers the (application dependent) feature map

$$\Phi : \Omega \to \mathcal{F}.$$

The map Φ describes each $x \in \Omega$ by a collection of features which are characteristic for a x and capture the essentials of elements of Ω. Since we are now in a Hilbert space we can work with linear techniques in $\mathcal{F}$. In particular we can use the scalar product in $\mathcal{F}$ of two elements of Ω represented by their features:

$$\langle \Phi(x), \Phi(y) \rangle_{\mathcal{F}} =: k(x, y) \quad \text{for all } x, y \in \Omega$$

and define a kernel k that way.

Note that given a kernel, neither the feature map nor the feature space are unique. Let

$$\Omega \coloneqq \mathbb{R}, \quad k(x, y) \coloneqq x \cdot y \text{ for all } x, y \in \mathbb{R}.$$

First, a feature map is the identity map on $\mathbb{R}$, with the feature space $\mathcal{F} = \mathbb{R}$.

But, the map $\Phi : \Omega \to \mathbb{R}^2$ defined by

$$\Phi(x) \coloneqq (x/\sqrt{2}, x/\sqrt{2}) \text{ for all } x \in \Omega$$

is also a feature map given the same k since

$$\langle \Phi(x), \Phi(y) \rangle_{\mathbb{R}^2} = \frac{x}{\sqrt{2}} \frac{y}{\sqrt{2}} + \frac{x}{\sqrt{2}} \frac{y}{\sqrt{2}} = x \cdot y = k(x, y) \text{ for all } x, y \in \Omega.$$

Such a construction can be made for any arbitrary kernel, therefore every kernel has many different feature spaces.

As a simple example for a "different" kind of kernel, we consider a collection of documents. We represent each document as a bag of words and describe a bag as a vector in a space in which each dimension is associated with a term from the set of words, i.e. the dictionary. The feature map is

$$\Phi(t) \coloneqq (\mathrm{wf}(w_1, t), \mathrm{wf}(w_2, t), \ldots, \mathrm{wf}(w_d, t)) \in \mathbb{R}^d$$

where $\mathrm{wf}(w_i, t)$ is the frequency of the word w_i in the document t. A simple kernel is the vector space kernel

$$K(t_1, t_2) \coloneqq \langle \Phi(t_1), \Phi(t_2) \rangle = \sum_{j=1}^{d} \mathrm{wf}(w_j, t_1) \cdot \mathrm{wf}(w_j, t_2).$$

Natural extensions to this kernel take e.g. word order, relevance or semantics into account, which can be achieved by using matrices in the scalar product:

$$\begin{aligned} K(t_1, t_2) &\coloneqq \langle S\Phi(t_1), S\Phi(t_2) \rangle \\ &= \Phi^T(t_1) S^T S \Phi(t_2). \end{aligned}$$

Another non-Euclidean data object are graphs, where the class of random walk kernels can be defined [Vis+10]. These are based on the idea that given a pair of graphs, one performs random walks on both and counts the number of matching walks. With $\tilde{A}_\times$ the adjacency matrix of the direct product graph of the two involved graphs, one defines:

$$k(G, H) \coloneqq \sum_{j=1}^{N_G} \sum_{k=1}^{N_H} \sum_{k=0}^{\infty} \lambda_k \left[\tilde{A}_\times^k \right]_{j,k}.$$

More generally, one can define a random walk graph kernel k as

$$k(G, H) \coloneqq \sum_{k=0}^{\infty} \lambda_k q_\times^T W_\times^k p_\times,$$

where $W_\times$ is the weight matrix of the direct product graph, $q_\times^T$ is the stopping probability on the direct product graph, and $p_\times$ is the initial probability distribution on the direct product graph. For a current survery of graph kernels see [KJM20].

An example application for graph kernels are organic molecules, which can be represented as graphs and where one aims to predict electronic ground-state properties [Fab+17].

Mercer kernels

More generally, one can consider kernels of the Hilbert-Schmidt or Mercer form

$$k(x, y) = \sum_{i \in I} \lambda_i \varphi_i(x) \varphi_i(y) \text{ for all } x, y \in \Omega,$$

with certain functions $\varphi : \Omega \to \mathbb{R}, i \in I$, certain positive weights $\lambda_i, i \in I$, and an index set I such that the summability condition

$$k(x, x) = \sum_{i \in I} \lambda_i \varphi_i(x)^2 < \infty \tag{1}$$

holds for all $x \in \Omega$.

Remark. Such kernels arise in machine learning if the functions φ_i each describe a feature of x and the feature space is the weighted ℓ_2-space:

$$\ell_{2,I,\lambda} \coloneqq \left\{ \{\xi_i\}_{i\in I} : \sum_{i\in I} \lambda_i \xi^2 < \infty \right\}$$

of sequences with indices in I.

These expansion also occurs when kernels generating positive integral operators are expanded into eigenfunctions on Ω. Such kernels can be viewed as arising from generalized convolutions.

Generally, kernels have there major application fields: convolutions, trial spaces, and covariances. We are mainly concerned with the last two.

1.1.1. Properties of Kernels

We consider an arbitrary set $X = \{x_1, \ldots, x_N\}$ of N distinct elements of Ω. We can from linear combinations

$$f(x) = \sum_{j=1}^{N} a_j k(x_j, x), \quad x \in \Omega$$

of translates of the kernel.

This is a very convenient technique to generate functions on an otherwise unstructured set Ω. Note that the coefficients might not be unique, we do not assume that the $k(x_j, x)$ are linearly independent.

For X we construct the symmetric $N \times N$ *kernel matrix*

$$K \coloneqq K_{X,X} \coloneqq (k(x_j, x_k))_{1\le j,k\le N}$$

and obtain the interpolation problem

$$\hat{f}_k = f(x_k) = \sum_{j=1}^{N} a_j k(x_j, x_k).$$

In matrix form:

$$K_{X,X}\alpha = \hat{F}.$$

In general it is difficult to determine if such a linear equation system can be solved, but for kernels we can see it via positive semi-definiteness.

Definition 1.3. A *kernel* on $\Omega \times \Omega$ is *symmetric and positive semidefinite*, if all kernel matrices for all finite point sets of distinct elements of Ω are symmetric and positive semidefinite.

Theorem 1.4.

1. *Kernels arising from feature maps, i.e.*

$$k(x, y) = \langle \Phi(x), \Phi(y) \rangle_{\mathcal{F}}$$

are positive semidefinite.

2. *Hilbert-Schmidt or Mercer kernels*

$$k(x, y) = \sum_{i \in I} \lambda_i \varphi_i(x) \varphi_i(y) \text{ for all } x, y \in \Omega$$

are positive semidefinite.

Proof. The first statement is obvious since such kernels result in kernel matrices that are Gramian matrices, and these are always positive semi-definite. For the second statement we consider the quadratic form corresponding to the kernel matrix and we write for all $a \in \mathbb{R}^N$:

$$\begin{aligned} a^T K a &= \sum_{j,k=1}^{N} a_j a_k k(x_j, x_k) \\ &= \sum_{j,k=1}^{N} a_j a_k \sum_{i \in I} \lambda_i \varphi_i(x_j) \varphi_i(x_k) \\ &= \sum_{i \in I} \lambda_i \sum_{j=1}^{N} a_j \varphi_i(x_j) \sum_{j=1}^{N} a_k \varphi_i(x_k) \\ &= \sum_{i \in I} \lambda_i \left(\sum_{j=1}^{N} a_j \varphi_i(x_j) \right)^2 \geq 0 \end{aligned}$$

■

Theorem 1.5. *Let K be a symmetric positive semi-definite kernel on Ω. Then*

1. *$k(x,x) \geq 0$ for all $x \in \Omega$.*
2. *$k(x,y)^2 \leq k(x,x) \cdot k(y,y)$ for all $x, y \in \Omega$.*
3. *$2k(x,y)^2 \leq k(x,x)^2 + k(y,y)^2$ for all $x, y \in \Omega$.*
4. *Any finite linear combination of positive semidefinite kernels with nonnegative coefficients gives a positive semidefinite kernel. If one of these kernels is positive definite, and its coefficient is positive, then the combination of kernels is positive definite.*
5. *The product of two positive semidefinite kernels is positive semidefinite.*
6. *The product of two positive definite kernels is positive definite.*

Proof. 1. This follows with the point set $\{x\} \subset \Omega$ in Definition 1.3.

2. Consider the kernel matrix for the set $\{x, y\}$. The determinant of such a positive semi-definite symmetric matrix is non-negative, therefore $k(x,x) \cdot k(y,y) - k(x,y)^2 \geq 0$.

3. With the inequality
$$2ab \leq a^2 + b^2 \quad \text{for } a, b \in \mathbb{R}_0^+$$
this follows from Item 2.

4. This property is easy to see, just expand the sum $x^T K x$.

5. The property follows from Lemma 1.6.

6. The property follows from the proof of Lemma 1.6. One can repeat the same proof with strict inequalities and some more linear algebra. ∎

Lemma 1.6 (Schur Product Lemma). *For two matrices A, B the matrix C with elements*
$$C_{ij} = A_{ij} B_{ij}$$
is called Schur product *or* Hadamard product. *The Schur product of two positive semidefinite matrices is positive semidefinite.*

Proof. We can decompose a positive semidefinite matrix into
$$A = S^T D S$$

with S an orthogonal matrix and $D = \operatorname{diag}(\lambda_1, \ldots, \lambda_N)$ a diagonal matrix with $\lambda_i \geq 0$ the eigenvalues of A. For all $q \in \mathbb{R}^N$, we look at

$$\begin{aligned} q^T C q = \sum_{j,k=1}^{N} q_j q_k \overbrace{a_{jk} b_{jk}}^{c_{jk}} &= \sum_{j,k=1}^{N} q_j q_k b_{jk} \sum_{m=1}^{N} \lambda_m s_{jm} s_{km} \\ &= \sum_{m=1}^{N} \lambda_m \sum_{j,k=1}^{N} \underbrace{q_j s_{jm}}_{p_{jm}} \underbrace{q_k s_{km}}_{p_{km}} b_{jk} \\ &= \sum_{m=1}^{N} \lambda_m \underbrace{\sum_{j,k=1}^{N} p_{jm} p_{km} b_{jk}}_{\geq 0, \text{ since } B \text{ is positive semidefinite}} \geq 0. \end{aligned}$$

This however is exactly what we had to prove. ∎

Note that we consider only positive definiteness for symmetric matrices. The above also holds if one of the matrices is not symmetric, but positive definite.

Our overall aim is to go from kernels to a reproducing kernel Hilbert space (RKHS). Therefore we will define "candidate" spaces and a bilinear form in the way we would expect them.

As before, let K be a symmetric positive semidefinite kernel on Ω. We define the linear space

$$H \coloneqq \operatorname{span}\{k(x, \cdot) \mid x \in \Omega\}$$

of all finite linear combinations of translates of the kernel.

In the same way, we define the linear space

$$L \coloneqq \operatorname{span}\{\delta_x \mid x \in \Omega,\ \delta_x : H \to \mathbb{R}\}$$

of all finite linear combinations of point evaluation functionals acting on functions in H,

$$\delta_x : H \to \mathbb{R}, f \mapsto f(x).$$

In particular, we explicitly restrict the action to functions in H. We can write all elements from L and H as

$$\lambda_{a,X} \coloneqq \sum_{j=1}^{N} a_j \delta_{x_j}$$

$$f_{a,X}(x) \coloneqq \lambda_{a,X}^{y} k(x, y) = \sum_{j=1}^{N} a_j k(x, x_j)$$

with some $a \in \mathbb{R}^N$, $X = \{x_1, \ldots, x_N\} \subset \Omega$, where X is any arbitrary finite subset of Ω. Unfortunately, from $f_{a,X}(\cdot) = 0$ on $\lambda_{a,X}(\cdot) = 0$ it does not follow that $a = 0$, we need to observe that, but it will pose no problem.

We now define a bilinear form on L:

$$\begin{aligned} \langle \lambda_{a,X}, \lambda_{b,Y} \rangle_L &\coloneqq \sum_{j=1}^{M} \sum_{k=1}^{N} a_j b_k k(x_j, y_k) \\ &= \lambda_{a,X}^x \lambda_{b,Y}^y k(x, y) \\ &= \lambda_{a,X} \left(f_{b,Y} \right). \end{aligned} \tag{2}$$

This is well-defined, since it is independent of the specific representation, i.e. a, b, due to the functionals and their actions in the second line. We observe that the bilinear form is positive semi-definite since the kernel matrix has this property. Further, we have

$$\begin{aligned} \left| \lambda_{a,X} \left(f_{b,Y} \right) \right| &= \left| \langle \lambda_{a,X}, \lambda_{b,Y} \rangle_L \right| \\ &\leq \left\| \lambda_{a,X} \right\|_L \left\| \lambda_{a,Y} \right\|_L \end{aligned} \tag{3}$$

where it may be just a seminorm, not a norm.

Somewhat surprisingly, the bilinear form is actually positive definite, even for K positive semidefinite.

Theorem 1.7. *If k is a symmetric positive semidefinite kernel on Ω, then the bilinear form $\langle \cdot, \cdot \rangle_L$ from Eq. (2) is positive definite on the space L of functionals defined on function on Ω. Thus L is a pre-Hilbert space of functions on Ω.*

Proof. Assume that

$$0 = \langle \lambda_{a,X}, \lambda_{a,X} \rangle_L = \sum_{j,k=1}^{N} a_j a_k k(x_j, x_k) = \lambda_{a,X} \lambda_{a,X} k(x, y) = \lambda_{a,X} \left(f_{a,X} \right)$$

for $a \in \mathbb{R}^N$, and $X \subset \Omega$ Then by the equality Eq. (3) we have $\lambda_{a,X} = 0$ as a functional on H. Here we use that the functionals in L are restricted to functions in H. Note that we do not get $a = 0$, neither do we need that. ∎

Theorem 1.8. *The mapping $R : \lambda_{a,X} \mapsto f_{a,X} \coloneqq \lambda_{a,X} \left(k(\cdot, y) \right)$ is linear and bijective from L onto H. Thus*

$$\begin{aligned} \langle f_{a,X}, f_{b,Y} \rangle_H &\coloneqq \langle \lambda_{a,X}, \lambda_{b,Y} \rangle_L \\ &= \langle R \left(\lambda_{a,X} \right), R \left(\lambda_{b,Y} \right) \rangle_H \end{aligned}$$

is an inner product on H. R acts as the Riesz map.

PROOF. Linearity is obvious. If a $f_{b,Y} = R(\lambda_{b,Y}) \in H$ vanishes, the definition of $\langle \cdot, \cdot \rangle_L$ implies that $\lambda_{b,Y}$ is orthogonal to all of L.. Due to Theorem 1.7 it is zero. Thus we have bijectivity. The Riesz property is present in the definition of $\langle \cdot, \cdot \rangle_L$, since

$$\begin{aligned} \lambda_{a,X}(f_{b,Y}) &= \langle \lambda_{a,X}, \lambda_{b,Y} \rangle_L \\ &= \langle f_{a,X}, f_{b,Y} \rangle_H \\ &= \langle R(\lambda_{a,X}), f_{b,Y} \rangle_H \end{aligned}$$

■

When we specialize the bilinear form $\langle \cdot, \cdot \rangle_L$ to $\lambda_{1,x}$, i.e. one point $x \in \Omega$, with $a = 1$, we observe:

$$\begin{aligned} f_{b,Y}(x) = \delta_x(f_{b,Y}) &= \lambda_{1,x}(f_{b,Y}) \\ &= \langle \lambda_{1,x}, \lambda_{b,Y} \rangle_L \\ &= \langle R(\lambda_{1,x}), R(\lambda_{b,Y}) \rangle_H \\ &= \langle R(\lambda_{1,x}), f_{b,Y} \rangle_H \\ &= \langle k(x, \cdot), f_{b,Y} \rangle_H . \end{aligned}$$

In other words, for all $f \in H, x \in \Omega$ we have

$$f(x) = \delta_x(f) = \langle f, R(\lambda_{1,x}) \rangle_H = \langle f, k(x, \cdot) \rangle_H, \tag{4}$$

which is the very useful so-called *reproduction equation* to obtain for values of functions from the inner product. Furthermore, for $f = \lambda_{1,y} = \lambda_{b,Y}$ it follows

$$k(x, y) = \langle k(x, \cdot), k(y, \cdot) \rangle_H .$$

We can now observe for all $f \in H, x \in \Omega$:

$$|\delta_x(f)| = |f(x)| = |\langle f, k(x, \cdot) \rangle_H| \leq \|f\|_H \|k(x, \cdot)\|_H = \|f\|_H \sqrt{k(x, x)},$$

and

$$k(x, y) = \langle k(x, \cdot), k(y, \cdot) \rangle_H = \langle \delta_x, \delta_y \rangle_L \text{ for all } x, y \in \Omega.$$

Furthermore, we have the identity

$$\begin{aligned} \|\delta_x - \delta_y\|_L^2 &= \|\delta_x\|_L^2 - 2\langle \delta_x, \delta_y \rangle_L + \|\delta_y\|_L^2 \\ &= k(x, x) - 2k(x, y) + k(y, y) \text{ for all } x, y \in \Omega. \end{aligned}$$

With that, we can derive a notion of distance on Ω via

$$d_k(x, y) := \operatorname{dist}(x, y) := \|\delta_x - \delta_y\|_L^2 = \sqrt{k(x, x) - 2k(x, y) + k(y, y)} \text{ for } x, y \in \Omega. \tag{5}$$

We see now

$$|f(x) - f(y)| \leq \|f\|_H \|\delta_x - \delta_y\|_L = \|f\|_H \operatorname{dist}(x, y) \text{ for all } x, y \in \Omega, f \in H,$$

so all functions in H are continuous in this special distance.

Let us remember what we do know for H, it is an inner product space of functions on Ω with the inner product $\langle \cdot, \cdot \rangle_H$, as long as k is a symmetric positive semi-definite kernel on Ω. We now can use classical Hilbert space arguments and go to the closure $\mathcal{H}$ of H under $\langle \cdot, \cdot \rangle_H$. This is similar to the transition from rational numbers to real numbers. The closure $\mathcal{H}$ is, at first, an abstract space defined by equivalence classes of Cauchy sequences in H, but it is complete space and therefore we have a Hilbert space. Furthermore, each continuous map from H to a Banach space Y extends uniquely to the closure.

Theorem 1.9. *Each symmetric positive definite kernel k on a set Ω is the* reproducing kernel *of a Hilbert space called the* native space $\mathcal{H} = \mathcal{N}_k$ *of the kernel. This Hilbert space is unique and it is a space of functions on Ω. The kernel k is a reproducing kernel of $\mathcal{N}_k$ fulfilling*

$$\langle f, k(x, \cdot) \rangle_H = f(x) \quad \text{for all } x \in \Omega, f \in \mathcal{N}_K. \tag{6}$$

Proof. The existence of the native space follows from standard Hilbert space arguments, see e.g. chapter 11 from the lecture notes of Schaback [Sch11].

In the reproduction equation (4)

$$f(x) = \langle f, k(x, \cdot) \rangle_H,$$

both sides continuously depend on $f \in H$. Therefore the equation carries over to the closure, i.e. the native space, which proves the reproduction formula (6) in the theorem. The equation explains how an abstract element f of the native space can be interpreted as a function, namely the left hand side $\langle f, k(x, \cdot) \rangle_H$ defines the right hand side $f(x)$.

If k is a reproducing kernel in a possibly different Hilbert space T with an analogous reproduction equation, we observe

$$\langle k(x, \cdot), k(y, \cdot) \rangle_H = k(x, y) = \langle k(x, \cdot), k(y, \cdot) \rangle_T.$$

This shows that the inner products of T and $\mathcal{N}_k$ coincide on H. Since T is a Hilbert space, it must contain the closure $\mathcal{N}_K$ of H as a closed subspace. If T were to be larger than $\mathcal{N}_k$, there must be a non-zero element $f \in T$ that is orthogonal to $\mathcal{N}_k$ and in particular orthogonal to H. We observe

$$f(x) = \langle f, k(x, \cdot) \rangle_T = 0 \quad \text{for all } x \in \Omega,$$

which is a contradiction to $f \neq 0$. ∎

Note that usually the Hilbert space closure of an inner product space is much "larger" than the pre-Hilbert space.

Let us look again at the point evaluation functionals

$$\delta_x : \mathcal{N}_k \to \mathbb{R}, f \mapsto f(x) \text{ for all } f \in \mathcal{N}_k, x \in \Omega.$$

The dual space $\mathcal{N}_k^*$ of the native space $\mathcal{N}_k$ is again a Hilbert space, which is isometrically isomorphic to $\mathcal{N}_k$ via the Riesz map

$$\begin{aligned} R &: \mathcal{N}_k^* \to \mathcal{N}_k \\ \lambda(f) &= \langle f, R(\lambda)\rangle_{\mathcal{N}_k} \text{ for all } f \in \mathcal{N}_k, \lambda \in \mathcal{N}_k^* \\ \langle \lambda, \mu\rangle_{\mathcal{N}_k} &= \langle R(\lambda), R(\mu)\rangle_{\mathcal{N}_k} \text{ for all } \lambda, \mu \in \mathcal{N}_k^*. \end{aligned}$$

The reproduction equation tells us that

$$\delta_x(f) = \langle f, k(x, \cdot)\rangle_{\mathcal{N}_k} \text{ for all } f \in \mathcal{N}_k, x \in \Omega \tag{7}$$

and we directly observe that $k(x, \cdot)$ is the Riesz representer $R(\delta_x)$ of δ_x in $\mathcal{N}_k$, which gives

$$\langle \delta_x, \delta_y\rangle_{\mathcal{N}_k^*} = \langle R(\delta_x), R(\delta_y)\rangle_{\mathcal{N}_k} = \langle k(x, \cdot), k(y, \cdot)\rangle_{\mathcal{N}_k} = k(x, y) \text{ for all } x, y \in \Omega.$$

So what we observed for the pre-Hilbert space H holds for the native Hilbert space as well.

Same goes for

$$\|\delta_x(f)\|_{\mathcal{N}_k^*} = \|k(x, \cdot)\|_{\mathcal{N}_k} = \sqrt{k(x, x)} \text{ for all } x \in \Omega$$

and we have the extended reproduction property

$$\lambda(f) = \langle f, \lambda^x k(x, \cdot)\rangle_{\mathcal{N}_k} \text{ for all } f \in \mathcal{N}_k, \lambda \in \mathcal{N}_k^*,$$

so that $\lambda^x k(x, \cdot)$ is the Riesz representer of λ.

1.1.2. Reproducing Kernel Hilbert Space

Definition 1.10 (Reproducing Kernel Hilbert Space). A Hilbert space $\mathcal{H}$ of functions on a set Ω with inner product $\langle \cdot, \cdot \rangle_{\mathcal{N}_k}$ is called a *reproducing kernel Hilbert space (RKHS)*, if there is a kernel function

$$k : \Omega \to \mathbb{R}$$

with

$$k(x, \cdot) \in \mathcal{H}$$

for all $x \in \Omega$ and the reproduction property

$$f(x) = \langle f, k(x, \cdot)\rangle_{\mathcal{H}} \text{ for all } x \in \Omega, f \in \mathcal{H}. \tag{8}$$

This directly implies

$$k(y,x) = \langle k(y,\cdot), k(x,\cdot)\rangle_{\mathcal{H}} = \langle k(x,\cdot), k(x,\cdot)\rangle_{\mathcal{H}} = k(x,y) \text{ for all } x,y \in \Omega.$$

This alone gives us positive semi-definiteness. To see that, take any $X = \{x_1, \dots, x_N\} \subset \Omega$ and $a \in \mathbb{R}^N$:

$$\begin{aligned}\sum_{j,k=1}^{N} a_j a_k k(x_j, x_k) &= \sum_{j,k=1}^{N} a_j a_k \langle k(x_j,\cdot), k(x_k,\cdot)\rangle_{\mathcal{H}} \\ &= \left\langle \sum_{j=1}^{N} a_j k(x_j,\cdot), \sum_{k=1}^{N} a_k k(x_k,\cdot) \right\rangle_{\mathcal{H}} \\ &= \left\| \sum_{j=1}^{N} a_j k(x_j,\cdot) \right\|_{\mathcal{H}}^2 \geq 0\end{aligned}$$

Now we aim for theorem 9 in the other direction.

Theorem 1.11. *Each Hilbert space $\mathcal{H}$ of real valued functions on some set Ω with continuous point evaluation functionals*

$$\delta_x : \mathcal{H} \to \mathbb{R}, f \mapsto f(x) \text{ for all } f \in \mathcal{H}, x \in \Omega.$$

is a reproducing kernel Hilbert space (RKHS) with a positive definite kernel $k : \Omega \times \Omega \to \mathbb{R}$. The kernel is uniquely defined by providing the Riesz representer of the point evaluation functionals as in Eq. (7).

Proof. Under the given hypothesis, there must be a Riesz representer of δ_x. By the definition of the Riesz map it takes the form $k(x,\cdot) \in \mathcal{H}$ satisfying the reproduction equation. In other words, any such Hilbert space has a symmetric positive semi-definite reproducing kernel. The final statement follows from Theorem 1.9, because both the native space and $\mathcal{H}$ are Hilbert spaces that contain all $k(x,\cdot)$. ∎

Now to collect some properties

Theorem 1.12. *If a Hilbert (sub-)space of functions on Ω has a finite orthonormal basis (ONB) $v_1, \dots, v_N$ the reproducing kernel is*

$$k_N(x, \cdot) = \sum_{j=1}^{N} v_j(x) v_j(\cdot) \quad \text{for all } x \in \Omega.$$

In case of a subspace we have

$$\sum_{j=1}^{N} |v_j(x)|^2 = k_N(x, x) \leq k(x, x) \quad \text{for all } x \in \Omega.$$

PROOF. The kernel, no matter what it is, must have a representation in the orthonormal basis as

$$\begin{aligned} k_N(x, \cdot) &= \sum_{j=1}^{N} \langle k_N(x, \cdot), v_j \rangle_{\mathcal{H}} v_j(\cdot) \\ &\overset{\text{Eq. (8)}}{=} \sum_{j=1}^{N} v_j(x) v_j(\cdot). \end{aligned}$$

For the subspace consider

$$\begin{aligned} k_N(x, x) &= \langle k_N(x, \cdot), k_N(x, \cdot) \rangle_{\mathcal{H}} \\ &= \langle k_N(x, \cdot), k(x, \cdot) \rangle_{\mathcal{H}}. \end{aligned}$$

With Cauchy-Schwarz we get then

$$k_N(x, x) \leq \sqrt{k_N(x, x)} \sqrt{k(x, x)} \quad \text{for all } x \in \Omega.$$

■

Note that for increasing N the functions v_N must get small. Which is a bit surprising, since their normalization in the ONB is independent of N. But consider the case where the Hilbert space norm includes derivatives, which has an effect on the normalization, namely that basis functions with sharp spikes tend to be small in the function values.

This observation gives a hint that not all ONB would give an RKHS. Clearly, those where the expansion in $\mathcal{H}$ does only converge in the Hilbert space norm but not pointwise would not have continuous point evaluation functionals.

1.1.3. Mercer kernels

But, we saw the kernels of Mercer form

$$k(x, y) = \sum_{i \in I} \lambda_i \varphi_i(x) \varphi_i(y) \quad \text{for all } x, y \in \Omega, \tag{9}$$

with the summability condition Eq. (1)

$$k(x,x) = \sum_{i\in I} \lambda_i \varphi_i(x)^2 < \infty. \tag{10}$$

Then we can observe in the expansion

$$\begin{aligned} |f(x)| &= \left| \sum_{i\in I} \langle f, \varphi_i\rangle_{\mathcal{H}} \varphi_i(x) \right| \\ &\leq \sum_{i\in I} |\langle f, \varphi_i\rangle_{\mathcal{H}}||\varphi_i(x)| \\ &= \sum_{i\in I} \frac{|\langle f, \varphi_i\rangle_{\mathcal{H}}|}{\sqrt{\lambda_i}} |\varphi_i(x)|\sqrt{\lambda_i} \\ &\leq \sqrt{\sum_{i\in I} \frac{\langle f, \varphi_i\rangle_{\mathcal{H}}^2}{\lambda_i}} \sqrt{\sum_{i\in I} \varphi_i^2(x)\lambda_i}. \end{aligned}$$

We have boundedness of the point evaluation in the subspace

$$\mathcal{H}_\lambda := \left\{ f \in \mathcal{H} \,\middle|\, \|f\|_\lambda^2 := \sum_{i\in I} \frac{\langle f, \varphi_i\rangle_{\mathcal{H}}^2}{\lambda_i} < \infty \right\}.$$

This space has a norm that arises from the inner product

$$\langle f, g\rangle_\lambda := \sum_{i\in I} \frac{\langle f, \varphi_i\rangle_{\mathcal{H}} \langle g, \varphi_i\rangle_{\mathcal{H}}}{\lambda_i} \quad \text{for all } f, g \in \mathcal{H}_\lambda.$$

We now define the Mercer kernel according to Eq. (9) and check if all $f_x := k(x,\cdot)$ are in $\mathcal{H}_\lambda$. We observe for the expansion coefficients

$$\langle f_x, \varphi_i\rangle_{\mathcal{H}} = \lambda_i \varphi_i(x)$$

and get $f_x \in \mathcal{H}_\lambda$ due to the summability condition

$$\sum_{i\in I} \frac{\langle f_x, \varphi_i\rangle_{\mathcal{H}}^2}{\lambda_i} = \sum_{i\in I} \lambda_i \varphi_i(x)^2 < \infty.$$

Further, each $f \in \mathcal{H}_\lambda$ satisfies the reproduction equation

$$\begin{aligned} \langle f, k(x_i,\cdot)\rangle_\lambda &= \sum_{i\in I} \frac{\langle f, \varphi_i\rangle_{\mathcal{H}} \langle k(x,\cdot), \varphi_i\rangle_{\mathcal{H}}}{\lambda_i} \\ &= \sum_{i\in I} \frac{\langle f, \varphi_i\rangle_{\mathcal{H}} \lambda_i \varphi_i}{\lambda_i} \\ &= f(\lambda) \end{aligned}$$

for all $x \in \Omega$. The Mercer kernel is therefore reproducing in spH_λ.

This proves

Theorem 1.13. *If a Hilbert space of functions on Ω has a countable orthonormal basis $\{\varphi_i\}_{i\in I}$, each summability property of the form Eq. (10) leads to a reproducing Mercer kernel Eq. (9) for a suitable subspace of functions with continuous point evaluation.*

We add without proof that spaces such as $\mathcal{H}_\lambda$ are always complete because they are isometrically isomorphic to some Hilbert space of weighted sequences, see e.g. chapter 11 from the lecture notes of Schaback [Sch11].

Corollary 1.14. *The spaces $\mathcal{H}_\lambda$ defined above are the native spaces for the corresponding Mercer kernels.*

Example. Consider trigonometric polynomials, i.e. the orthonormal functions

$$\frac{1}{\sqrt{2}}, \cos(nx), \sin(nx), \quad n \in \mathbb{N}$$

and the inner product

$$\langle f, h\rangle_H = \frac{1}{\pi}\int_{-\pi}^{\pi} f(t)g(t)dt$$

giving the space of 2π-periodic square integrable functions.

We write $I := (0, 0 \cup (\mathbb{N}, 0) \cup (0, \mathbb{N}))$ and

$$\phi_i(x) = \begin{cases} \frac{1}{\sqrt{2}} & i = (0,0) \\ \cos(nx) & i = (n,0), n \geq 1 \\ \sin(nx) & i = (0,n), n \geq 1 \end{cases}.$$

Since all functions are uniformly bounded, the summability condition does hold when the weights are summable. Fixing some $m \geq 1$ we define

$$\lambda_i(x) = \begin{cases} 1 & i = (0,0) \\ n^{-2m} & \text{otherwise} \end{cases}$$

and get the Mercer kernel:

$$\begin{aligned} k_{2m}(x,y) &:= \frac{1}{\sqrt{2}} + \sum_{n=1}^{\infty} n^{-2m}(\cos(nx)\cos(ny) + \sin(nx)\sin(ny)) \\ &= \frac{1}{\sqrt{2}} + \sum_{n=1}^{\infty} n^{-2m}\cos(n(x-y)). \end{aligned}$$

These are kernels for the Sobolev space of 2π-periodic functions.

As already mentioned, Mercer kernels arise from integral operators.

Definition 1.15. Let $k : \Omega \times \Omega \to \mathbb{R}$ be continuous, Ω be a compact domain, ν be a Borel measure and $L_2^\nu(\Omega)$ be the Hilbert space of square integrable functions on Ω. We define the integral operator $T_k : L_2^\nu(\Omega) \to L_2^\nu(\Omega)$ by

$$(T_k f)(\cdot) = \int_\Omega k(x, \cdot) f(x) \, \mathrm{d}\nu$$

We call k the kernel of T_k.

T_k is often called a Hilbert-Schmidt-Integraloperator.

One knows from functional analysis, that for a Mercer kernel the corresponding T_k is a self-adjoint, positive semidefinite, compact operator and the spectral theorem applies, i.e. a complete orthonormal system of eigenfunctions of T_k exists for $L_2^\nu(\Omega)$ and the eigenvalues λ_i of T_k are nonnegative with $\lambda_i \to 0$ for $i \to \infty$. In other words, one can show that

$$k(x, y) = \sum_{i=1}^{\infty} \lambda_i \phi_i(x) \phi_i(y)$$

for positive semidefinite operators T_k.

Theorem 1.16 (Mercer's theorem). *Let Ω be a compact domain, ν a Borel measure on Ω, and $k : \Omega \times \Omega$ symmetric and continuous. The corresponding integral operator T_k shall be positive semidefinite, i.e.*

$$\int_{\Omega \times \Omega} k(x, y) f(x) f(y) \, \mathrm{d}\nu \geq 0 \quad \text{for all } f \in L_2^\nu(\Omega).$$

Let λ_i be the i-th eigenvalue of T_k and $\{\phi_i\}_{i \geq 1}$ the corresponding and normalized eigenfunctions. It then holds

$$k(x, y) = \sum_{i=1}^{\infty} \lambda_i \cdot \phi_i(x) \phi_i(y) \quad \text{for all } x, y \in \Omega,$$

where the convergence is absolute (for each $x, y \in \Omega \times \Omega$) and uniform (on $\Omega \times \Omega$).

In particular the eigenvalues are absolutely summable.

Proof. A proof is given in [Hoc89] for $\Omega = [0, 1]$ and the Lebesgue measure, but the proof is valid for this more general situation. ∎

Remark. Continuity of the kernel is essential, otherwise e.g. $k(x, x)$ could have any value without relation to the eigenvalues.

We give here a simplified version of the Mercer's theorem, which fits for the machine

learning setting, where the proof is made without too much functional analysis.

Theorem 1.17. *Let $\Omega \subset \mathbb{R}^d$ be compact and k be a continuous and symmetric kernel. Let the corresponding integral operator T_k be positive semidefinite:*

$$\int_{\Omega\times\Omega} k(x,y)f(x)f(y)\,\mathrm{d}x\,\mathrm{d}y \geq 0 \quad \textit{for all } f \in L_2(\Omega).$$

Then we can expand $k(x,y)$ in a uniformly convergent series (on $\Omega\times\Omega$) in terms of functions ϕ_j, satisfying $\langle\phi_i,\phi_j\rangle = \delta_{ij}$, as

$$k(x,y) = \sum_{i=1}^{\infty} \phi_i(x)\phi_i(y).$$

Furthermore, the series

$$\sum_{i=1}^{\infty} \|\phi_i\|_{L_2}^2$$

is convergent.

Proof. Assume there is a finite set $\{x_1,\ldots,x_N\}$ so that the corresponding kernel matrix is not positive semidefinite. Let q be such that

$$\sum_{i,j=1}^{N} q_i q_j k(x_i,x_j) = \varepsilon < 0.$$

Now let

$$f_\sigma(x) = \sum_{i=1}^{d} q_i \frac{1}{(2\pi\sigma)^{\frac{d}{2}}} \exp\left(-\frac{\|x-x_i\|^2}{2\sigma}\right).$$

We have $f_\sigma \in L_2$, and additionally

$$\lim_{\sigma\to 0}\int_{\Omega\times\Omega} k(x,y)f_\sigma(x)f_\sigma(y)\,\mathrm{d}x\,\mathrm{d}y = \varepsilon.$$

But then for some $\sigma > 0$ the integral will be less than 0, which contradicts the positivity of the integral operator T_k. Therefore, we have a Mercer kernel.

Now consider the native space $\mathcal{N}_k$ of k. For continuous k and $\Omega \subset \mathbb{R}^d$, one can show that $\mathcal{N}_k$ is separable and in particular there exists a countable orthonormal basis of $\mathcal{N}_k$ called ϕ_i, $i = 1,\ldots$. The technical proof of the separability is being omitted here. Then we have an expansion in the orthonormal basis for $k(x,\cdot)$, namely

$$\begin{aligned} k(x,y) &= \sum_{i=1}^{\infty} \langle k(x,\cdot),\phi_i(\cdot)\rangle\phi_i(y) \\ &= \sum_{i=1}^{\infty} \phi_i(x)\phi_i(y) \end{aligned}$$

with the required properties. Finally, we observe

$$\begin{aligned}\lim_{n\to\infty}\sum_{i=1}^{n}\|\phi_i\|_{L_2}^2 &= \lim_{n\to\infty}\sum_{i=1}^{n}\int_\Omega \phi_i(x)\phi_i(x)\,\mathrm{d}x\\ &= \lim_{n\to\infty}\int_\Omega\sum_{i=1}^{n}\phi_i(x)\phi_i(x)\,\mathrm{d}x\\ &= \int_\Omega\lim_{n\to\infty}\sum_{i=1}^{n}\phi_i(x)\phi_i(x)\,\mathrm{d}x<\infty,\end{aligned}$$

where we use the compactness of Ω. ∎

Therefore, we can write

$$K(x,y)=\sum_{i=1}^{\infty}\phi_i(x)\phi_i(y)=\langle\Phi(x),\Phi(y)\rangle_{\ell_2}.$$

Or, any Hilbert space $\mathcal{H}$ with a countable orthonormal basis allows such a feature map since it is isomorphic to ℓ_2.

1.2. Function Approximation

Kernels for subspaces

We aim to work with a finite number of trial functions $k(x,\cdot)$ or data points $\{x_i\}_{i=1}^{\infty}$ and represent functions in $\mathcal{H}$ by functions given as finite linear combinations.

Let us fix a non-empty subset $X\subseteq\Omega$ and look at the closed subspace of finite linear combination

$$\mathcal{H}_X\coloneqq\overline{\operatorname{span}\{k(x,\cdot)\mid x\in X\}}\subseteq\mathcal{H}.$$

Generally, any close subspace $\mathcal{H}_0$ of $\mathcal{H}$ is a Hilbert space of functions on Ω with its own reproducing kernel k_0. With the projector $\Pi_0:\mathcal{H}\to\mathcal{H}_0$ we get

Theorem 1.18. *Let $\mathcal{H}_0$ be a closed subspace of $\mathcal{H}$ with reproducing kernel k_0 and let $\Pi_0:\mathcal{H}\to\mathcal{H}_0$ be the projector onto $\mathcal{H}_0$. The subspace kernel is*

$$k_0(x,\cdot)=\Pi_0(k(x,\cdot))$$

for all $x\in\Omega$. The reproducing kernel for the orthogonal complement $\mathcal{H}_0^{\perp}$ is $k-k_0$.

PROOF. The identity on $\mathcal{H}$ can be decomposed into the orthogonal projections

$$I = \Pi_0 + (I - \Pi_0) = \Pi_0 + \Pi_0^\perp.$$

Thus:

$$f(x) = (\Pi_0 f)(x) + \left(\Pi_0^\perp f\right)(x).$$

Inserting into Eq. (6) gives:

$$\begin{aligned} f(x) &= \langle f, K(x,\cdot)\rangle_{\mathcal{H}} \\ &= \langle \Pi_0 f + \Pi_0^\perp f, \Pi_0 k(x,\cdot) + \Pi_0^\perp k(x,\cdot)\rangle_{\mathcal{H}} \\ &= \langle \Pi_0 f, \Pi_0 k(x,\cdot)\rangle_{\mathcal{H}} + \langle \Pi_0^\perp f, \Pi_0^\perp k(x,\cdot)\rangle_{\mathcal{H}}. \end{aligned}$$

With $f \in \mathcal{H}_0$ or $f \in \mathcal{H}_0^\perp$ both results follow. ∎

Remark. Orthogonal space decompositions correspond to additive kernel decompositions using the appropriate projectors.

Now back to the finite point set X.

Theorem 1.19. *Let $X \subset \Omega$ be non-empty. For the closed subspace*

$$\mathcal{H}_X \coloneqq \overline{\operatorname{span}\{k(x,\cdot) \mid x \in X\}}$$

it holds that

$$\mathcal{H}_X^\perp = \{f \mid f \in \mathcal{H}, f(X) = \{0\}\}.$$

PROOF. If $f(X) = \{0\}$, then $f \in \mathcal{H}_X^\perp$ by Eq. (6), and the converse holds analogously. ∎

Now take the projector Π_X from $\mathcal{H}$ to $\mathcal{H}_X$ and denote

$$f_X \coloneqq \Pi_X(f).$$

Standard results from Hilbert space theory give us

Theorem 1.20. *Each function $f \in \mathcal{H}$ has an orthogonal decomposition*

$$f = f_X + f_{X^\perp}$$

with $f_X \in \mathcal{H}_X$ and $f_{X^\perp} \in \mathcal{H}_{X^\perp}$. In particular, each $f \in \mathcal{H}$ has an interpolant $f_X \in \mathcal{H}_X$ recovering the values of f on X. Additionally,

$$\|f - f_X\|_{\mathcal{H}} = \inf_{g \in \mathcal{H}_X} \|f - g\|_{\mathcal{H}} \tag{11}$$

and

$$\|f_X\|_{\mathcal{H}} = \inf_{\substack{f(x)=g(x)\ \forall x \in X \\ g \in \mathcal{H}}} \|g\|_{\mathcal{H}} = \inf_{v \in \mathcal{H}_X^\perp} \|f - v\|_{\mathcal{H}} \tag{12}$$

due to the orthogonality of the decomposition.

The results Eq. (11) and Eq. (12) are two important optimality principles, which we state again separately.

Corollary 1.21. *The interpolant $f_X \in \mathcal{H}_X$ to a function f on X is at the same time the best approximation to f from all functions in $\mathcal{H}_X$.*

Corollary 1.22. *The interpolant $f_X \in \mathcal{H}_X$ to a function f on X minimizes the norm under all interpolants from the full space $\mathcal{H}$.*

With $f_\emptyset = 0$, $f_\emptyset^\perp = f$, and $\mathcal{H}_\emptyset = \{0\}$ and $\mathcal{H}_\emptyset^\perp = \mathcal{H}$ for completeness, we easily see

Corollary 1.23. *For all sets $X \subseteq Y \subseteq \Omega$ and all $f \in \mathcal{H}$ we have*

$$\|f_X\|_{\mathcal{H}} \leq \|f_Y\|_{\mathcal{H}} \leq \|f\|_{\mathcal{H}}$$

and

$$\|f\|_{\mathcal{H}} \geq \|f - f_X\|_{\mathcal{H}} \geq \|f - f_Y\|_{\mathcal{H}}.$$

Power function

The power function will allow us the analyze errors and stability.

We consider now only $f = k(x, \cdot)$ for a fixed $x \in \Omega$.

Definition 1.24. The function

$$P_X(x) := \|k(x,\cdot) - k_X(x,\cdot)\|_{\mathcal{H}} \quad x \in \Omega$$

is called *power function* with respect to the set X and the kernel k.

A different definition goes with the error functional

$$\epsilon_{x,X} f \mapsto f(x) - (\Pi_X(f))(x).$$

which is in $\mathcal{H}^\star$ and the power function is defined as

$$P_X(x) := \|\epsilon_{x,X}\|_{\mathcal{H}^\star} \quad \text{for all } x \in \Omega.$$

Theorem 1.25. *The two definitions for the power function are equivalent. Furthermore, the power function has the properties*

1. $P_X(x) = 0$ *for all* $x \in X$,
2. $P_0(x)^2 = k(x,x)$ *for all* $x \in \Omega$,
3. $P_\Omega(x) = 0$ *for all* $x \in \Omega$,
4. $0 = P_\Omega \le P_Y(x) \le P_X(x) \le P_0(x)$ *for all* $x \in \Omega$, $X \subseteq Y \subseteq \Omega$,
5. $P_X(x) = \inf_{g \in \mathcal{H}_X} \|k(x,\cdot) - g\|_{\mathcal{H}}$,
6. $P_X(x) = \sup_{\substack{f \in \mathcal{H}, \|f\|_{\mathcal{H}} \le 1 \\ f(X) = \{0\}}} |f(x)|$ *for all* $x \in \Omega$,

and finally the important error bound

$$\begin{aligned} |f(x) - f_X(x)| &= \left| f_X^\perp(x) \right| \\ &\le P_X(x) \left\| f_X^\perp \right\|_{\mathcal{H}} \\ &= P_X(x) \|f - f_X\|_{\mathcal{H}} \\ &\le P_X(x) \|f\|_{\mathcal{H}} \end{aligned}$$

for all $x \in \Omega, f \in \mathcal{H}$.

Proof. Due to $\langle \epsilon_{x,X}, \epsilon_{x,X} \rangle_{\mathcal{H}^*} = \langle R(\epsilon_{x,X}), R(\epsilon_{x,X}) \rangle_{\mathcal{H}}$ we have to show that the Riesz representer of $\delta_x \circ \Pi_X$ is $k(x,\cdot)_X$. We see that it follows from the representer properties

and various orthogonalities

$$\begin{aligned}\langle f, R(\delta_x \circ \Pi_X)\rangle_{\mathcal{H}} &= \delta_x \circ \Pi_X(f)\\ &= f_X(x)\\ &= \langle f_X, k(x,\cdot)\rangle_{\mathcal{H}}\\ &= \langle f_X, k_X(x,\cdot) + k_{X^\perp}(x,\cdot)\rangle_{\mathcal{H}}\\ &= \langle f_X, k_X(x,\cdot)\rangle_{\mathcal{H}}\\ &= \langle f - f_{X^\perp}, k_X(x,\cdot)\rangle_{\mathcal{H}}\\ &= \langle f, k_X(x,\cdot)\rangle_{\mathcal{H}}.\end{aligned}$$

The first five listed properties are easily derived from Definition 1.24 and the previous results.

With the error representation

$$\begin{aligned}f(x) - f_X(x) &= f_{X^\perp}(x)\\ &= \langle f_{X^\perp}, k(x,\cdot)\rangle_{\mathcal{H}}\\ &= \langle f_{X^\perp}, k(x,\cdot) - k_X(x,\cdot)\rangle_{\mathcal{H}}\end{aligned}$$

the error bound follows.

For the sixth property, we see from the first inequality of the error bound that

$$P_X(x) \geq \sup_{\|f_{X^\perp}\|\leq 1} |f_{X^\perp}(x)|$$

and equality must hold for the representation of $\epsilon_{x,X}$. ∎

Remark. Consider the subspace

$$\mathcal{H}_X^\star \coloneqq \overline{\operatorname{span}\{\delta_x \mid x \in X\}}$$

of the dual space. Then the fifth property can be equivalently be given as

$$P_X(x) = \inf_{\lambda \in \mathcal{H}_X^\star} \|\delta_x - \lambda\|_{\mathcal{H}^\star} \quad \text{for all } x \in \Omega. \tag{13}$$

It indicates how well the point evaluation functional δ_x can be approximated by arbitrary linear combinations of the point evaluation functionals $\delta_{x'}$ for points $x' \in X$.

Interpolate on finite sets

We now restrict ourselves to finite sets $X = \{x_1, \ldots, x_N\} \subseteq \Omega$. For each $f \in \mathcal{H}$ we can write

$$f_X(\cdot) = \sum_{j=1}^{N} \alpha_j k(x_j, \cdot)$$

with $\alpha_j \in \mathbb{R}$.

We know that f_X interpolates f on X, therefore we get a (non-unique in general) solution $\alpha \in \mathbb{R}^N$ of the interpolation problem

$$\sum_{j=1}^{N} a_j k(x_j, x_k) = \hat{f}_k, \quad 1 \leq k \leq N,$$

with $f_k = f(x_k)$.

The coefficients α_j might not be unique since we do not assume that the kernels $k(x_j, \cdot)$ are linearly independent.

With $f(x) = k(x, \cdot)$ we get that for every $x \in \Omega$

$$k(x, x_k) = \sum_{j=1}^{N} u_j(x) k(x_j, x_k), \quad 1 \leq k \leq N.$$

has a solution $u_j(x)$ as a function on Ω.

Additionally, it holds for all $x, z \in \Omega$

$$k_X(x, z) = \sum_{j=1}^{N} u_j(x) k(x_j, z), \tag{14}$$

i.e. the interpolant of $k(x, \cdot)$ on X.

In case of non-singularity of the kernel matrix we have

$$u_j(x_k) = \delta_{jk},$$

and we have a Lagrange basis.

Together

Theorem 1.26. *For every $x \in \Omega$*

$$k(x, x_k) = \sum_{j=1}^{N} u_j(x) k(x_j, x_k), \quad 1 \leq k \leq N. \tag{15}$$

has a solution $u_j(x)$ as a function on Ω.

It holds the generalization of the Lagrange formulation of interpolation

$$f_X(x) = \sum_{j=1}^{N} u_j(x) f(x_j). \tag{16}$$

This is also called quasi-interpolation.

Proof.

$$\begin{aligned} f_X(\cdot) &= \sum_{k=1}^{N} \alpha_k k(x_k, \cdot) \\ &\overset{\text{Eq. (14)}}{=} \sum_{k=1}^{N} \alpha_k \sum_{j=1}^{N} u_j(\cdot) k(x_j, x_k) \\ &= \sum_{j=1}^{N} u_j(\cdot) \sum_{k=1}^{N} \alpha_k k(x_i, x_k) \\ &= \sum_{j=1}^{N} u_j(\cdot) f(x_j). \end{aligned}$$

This proves what was to be shown. ∎

Back to $f = k(x, \cdot)$ for a fixed $x \in \Omega$ we get with Eq. (14) the following.

Theorem 1.27. *The power function has the explicit representation*

$$\begin{aligned} P_X^2(x) &= k(x,x) - 2\sum_{j=1}^{N} u_j(x) k(x, x_j) + \sum_{j=1}^{N}\sum_{k=1}^{N} u_j(x) u_k(x) k(x_j, x_k) \\ &= k(x,x) - k_X(x, \cdot)(x) \end{aligned}$$

PROOF.

$$\begin{aligned} P_X^2(x) &= \langle k(x,\cdot) - k(x,\cdot)_X, k(x,\cdot) - k(x,\cdot)_X \rangle_{\mathcal{H}} \\ &\overset{\text{Eq. (14)}}{=} k(x,x) - 2\langle k(x,\cdot), \sum_{j=1} u_j(x) k(x_j,\cdot)\rangle_{\mathcal{H}} + \sum_{j=1}^{N}\sum_{k=1}^{N} u_j(x)u_k(x)k(x_j,x_k) \\ &= k(x,x) - 2\sum_{j=1}^{N} u_j(x)k(x,x_j) + \sum_{j=1}^{N}\sum_{k=1}^{N} u_j(x)u_k(x)k(x_j,x_k) \end{aligned}$$

Now as we can use Eq. (15)

$$k(x,x_k) = \sum_{j=1}^{N} u_j(x)k(x_j,x_k)$$

in the third term, it cancels out once with the second term. Using Eq. (14) once more we identify the remaining terms with

$$k(x,x) - k_X(x,\cdot)(x).$$

■

Best linear estimation

Now come to a third optimality property. We will see in what kind of sense it is the best linear predictor/estimator.

As just seen

$$f_X(x) = \sum_{j=1}^{N} u_j(x) f(x_j)$$

is the interpolant on the set X.

Now let us consider completely arbitrary estimation formulas

$$(x,f) \mapsto \sum_{j=1}^{N} v_j(x) f(x_j)$$

with no assumption on $v_j(x)$. These representations are linear in f. For fixed x we get the error functional

$$f \mapsto f(x) - \sum_{j=1}^{N} v_j(x) f(x_j) = \left(\delta_x - \sum_{j=1}^{N} v_j(x)\delta_{x_j}\right)(f).$$

We want to have the estimation to be optimal over all $f \in \mathcal{H}$, therefore we chose the $v_j(x)$ to minimize

$$V_{X,v}(x) \coloneqq \left\| \delta_x - \sum_{j=1}^{N} v_j(x)\delta_{x_j} \right\|_{\mathcal{H}^\star}.$$

By the equivalent formulation of the error function Eq. (13), as remarked after **??**, we know the solution, namely the functions u_j and the optimal error in the worst case sense is described by the power function.

Theorem 1.28. *In the above worse case sense, kernel based interpolation yields the best linear predictor of unknown function values $f(x)$ from known functions values $f(x_j)$ at points x_j, $1 \leq j \leq N$.*

This is in particular relevant when the kernel comes from a covariance, i.e.

$$k(s,t) \coloneqq \operatorname{Cov}(X_s, X_t),$$

where for every $t \in \Omega$ we have a random variable X_t with finite second moments. Therefore, two data inputs from Ω are very similar if they are closely correlated, if they have very similar features, or the feature map kernel $k(x,y) = \langle \Phi(x), \Phi(y) \rangle$ has large positive value.

Now consider random variables X_t with mean zero and bounded variance. In this case the numerical estimation technique we introduced so far is called (simple) Kriging and $V_{X,v}^2$ can be seen to be the variance of the prediction error.

We define the error of the general linear predictor at x by

$$\epsilon_{x,X,v} \coloneqq X_x - \sum_{j=1}^{N} v_j(x) X_{x_j}.$$

and we aim to minimize the variance of the prediction error. It has zero mean and the variance is

$$\begin{aligned}\mathbb{E}\left(\epsilon_{x,X,v}^2\right) = \operatorname{Cov}(X_x, X_x) - 2\sum_{j=1}^{N} v_j(x) \operatorname{Cov}(X_x, X_{x_j}) \\ + \sum_{j=1}^{N}\sum_{k=1}^{N} v_j(x) v_k(x) \operatorname{Cov}(X_{x_j}, X_{x_k})\end{aligned}$$

Because

$$\operatorname{Cov}(X, X) = k(x,x)$$

and

$$k(x,y) = \langle \delta_x, \delta_y \rangle_{\mathcal{H}}$$

this yields:

$$\mathbb{E}\left(\epsilon_{x,X,v}^2\right) = V_{X,v}^2.$$

(Simple) Kriging is the best linear unbiased estimator under suitable assumptions.

Power function and stability

A kind of uncertainty principle holds:

It is impossible to make the power function and the condition of the kernel matrix small at the same time.

We express now the power function via the kernel matrix to analyse this effect. Besides the set $X = \{x_1, \ldots, x_N\}$ we now take another point $x_0 := x$ and we set $u_0(\cdot) := -1$. We define

$$A = k(x_i, x_k)_{0 \le i,j \le N}$$

and

$$u := (u_0(x), u_1(x), \ldots, u_N(x))^T.$$

Now look at the quadratic form (and remember $k(x_0, x_0) := k(x, x)$):

$$\begin{aligned} u^T A u &= \sum_{j=0}^{N}\sum_{k=0}^{N} u_j(x)u_k(x)k(x_j, x_k) \\ &= k(x,x) - 2\sum_{j=1}^{N} u_j(x)k(x, x_j) + \sum_{j=1}^{N}\sum_{k=1}^{N} u_j(x)u_k(x)k(x_j, x_k) \\ &\overset{\text{Theorem 1.27}}{=} P_X^2(x). \end{aligned}$$

A is symmetric and positive semidefinite, therefore has $N+1$ nonnegative real eigenvalues $\lambda_0 \ge \lambda_1 \ge \ldots \ge \lambda_N \ge 0$ and we obtain

$$P_X^2(x) \ge \lambda_N \left(1 + \sum_{j=1}^{N} u_j(x)^2\right) \ge \lambda_N,$$

where we can use

$$\lambda_N \|u\|_2^2 \le u^T A u \le \lambda_0 \|u\|_2^2.$$

We eliminate the special role of the point x and obtain the following:

Theorem 1.29. *The kernel matrix for N points $x_1, \ldots, x_N$ forming a set X has a smallest eigenvalue λ bounded from above by*

$$\lambda \le \min_{1 \le j \le N} P_{X \setminus \{x_j\}}(x_j).$$

This gives us information about the condition of the kernel matrix. In situations where the power function is still small after one point is left out, the kernel matrix must be ill-conditioned.

Consider from Theorem 1.25:

$$|f(x) - f_X(x)| \leq P_X(x)\|f\|_{\mathcal{H}},$$

which splits the error into two independent factors for f and X. Both depend on k, one measures by the norm of f the smoothness of the function and the other measures the quality of the point set. The optimality property of the power function is

$$P_X(x) = \inf_{\lambda \in \mathcal{H}_X^\star} \|\delta_x - \lambda\|_{\mathcal{H}^\star} \quad \text{for all } x \in \Omega$$

and it allows upper bounds of the above error.

We want to bound the data dependent part, where we know $f - f_X$ is zero on X.

Assume for now, that any directional derivative of both f and f_X is bounded by some constant C. Then we can write

$$|f(x) - f_X(x)| \leq |f(x_j) - f_X(x_j)| + 2C\|x - x_j\|_2$$

if the line connecting x and $x_j \in X$ is in Ω and if we integrate the directional derivatives along the line. Using the following definition this observation results in a first simple bound.

Definition 1.30. The *fill distance* of a set of points $X \subseteq \Omega$ for a bounded domain Ω is defined to be

$$h_{X,\Omega} = \sup_{x \in \Omega} \min_{1 \leq j \leq N} \|x - x_j\|_2.$$

In words this definition can be interpreted as

- any point $x \in \Omega$ has a point $x_j \in X$ not farther away then $h_{X,\Omega}$,
- $h_{X,\Omega}$ denotes the radius of the largest ball that is completely in Ω and does not contain a data location,
- $h_{X,\Omega}$ describes the size of the largest data-free hole Ω.

If Ω is convex, we get the simple error bound

$$\|f - f_X\|_{\infty,\Omega} \leq 2Ch_{X,\Omega}$$

where we still need to have control over C, e.g. in terms of a norm for f.

More generally, one can use bounds such as

$$\|f\|_{\infty,\Omega} \leq F(h_{X,\Omega})|f|_{\mathcal{W}} + C\|f\|_{\infty,X}$$

with $F(h) \to 0$ for $h \to 0$ and f in some function space $\mathcal{W}$ with a semi-norm $|f|_{\mathcal{W}}$. This is a form of a so-called *sampling inequality*.

When f and f_X are in $\mathcal{W}$ we can write

$$\begin{aligned}\|f - f_X\|_{\infty,\Omega} &\leq F(h_{X,\Omega})|f - f_X|_{\mathcal{W}} + C \underbrace{\|f - f_X\|_{\infty,X}}_{=0 \text{ for interpolation}} \\ &\leq F(h_{X,\Omega})(|f|_{\mathcal{W}} + |f_X|_{\mathcal{W}})\end{aligned}$$

$|f_X|_{\mathcal{W}}$ will depend on X and thus on $h_{X,\Omega}$, but usually one has a stability bound such as

$$|f_X|_{\mathcal{W}} \leq C|f|_{\mathcal{W}},$$

with C independent of X, e.g. similar to Corollary 1.23. Then

$$\|f - f_X\|_{\infty,\Omega} \leq (1 + C)F(h_{X,\Omega})|f|_{\mathcal{W}}.$$

This works also for general function spaces, it is not restricted to our native Hilbert spaces.

Other simple bounds can be obtained from Theorem 1.25, where we had

$$P_X(x) = \sup_{\substack{f \in \mathcal{H}, \|f\|_{\mathcal{H}} \leq 1 \\ f(X)=\{0\}}} f(x).$$

If for two kernels k_1, k_2 with native spaces $\mathcal{H}_1, \mathcal{H}_2$, we have an inclusion

$$f \in \mathcal{H}_1, \|f\|_{\mathcal{H}_1} \leq 1 \implies f \in \mathcal{H}_2, \|f\|_{\mathcal{H}_2} \leq 1,$$

then

$$P_{X,k_1}(x) \leq P_{X,k_2}(x) \quad \text{for all } x \in \Omega.$$

Roughly speaking, larger native spaces in the sense of unit ball inclusion lead to larger power functions.

Other simple upper bound are based on the optimality properties. We essentially will be using some coefficients $u \in \mathbb{R}^N$ instead of the optimal $u^\star$ from Theorems 1.26 and 1.27. Therefore

$$P_X^2(x) \leq k(x,x) - 2\sum_{j=1}^N u_j(x)k(x_j,x) + \sum_{j,k=1}^N u_j(x)u_k(x)k(x_j,x_k). \tag{17}$$

In the simplest case we use a nearest neighbor construction. Assume that for each $x \in \Omega$ we pick a single $x_{i(x)} \in X$ and define:

$$u_j(x) := \begin{cases} 1 & j = i(x) \\ 0 & \text{else} \end{cases}.$$

Then

$$\begin{aligned} P_X^2(x) &\leq k(x,x) - 2k(x_{i(x)}, x) + k(x_{i(x)}, x_{i(x)}) \\ &= d_k\left(x, x_{i(x)}\right)^2 \end{aligned}$$

with the distance $d(\cdot,\cdot)$ defined as in Eq. (5)

$$\begin{aligned} d_k &: \Omega \times \Omega \to [0,\infty], \\ d_k(x,y) &:= \sqrt{k(x,x) + k(y,y) - 2k(x,y)}. \end{aligned}$$

Therefore, one should pick $x_{i(x)} \in X$ closest to x in that distance.

Theorem 1.31. *If K is positive semidefinite, the power function on nonempty sets X of interpolation points satisfies*

$$P_X(x) \leq \min_{x_j \in X} d_k(x, x_j)$$

with the distance defined in Eq. (5).

For this results the smoothness of K or the structure on Ω did not play a role.

If we are in $\mathbb{R}^d$, we can use barycentric coordinates, i.e. x lies in a simplex with vertices consisting of the $d+1$ nearest x_i. That way one can locally recover linear functions. With some further properties of barycentric coordinates it holds for twice differentiable functions f

$$P_X(x) \leq C\varepsilon(x)^2,$$

where $\varepsilon(x)$ is the diameter of the simplex.

In general we want some quantity $E(x,h)$ that is small if x is surrounded by enough well-placed points of X. The general idea is to provide local error bounds, which are similar enough so that a similar global error bound follows from the local ones under suitable assumptions.

Let us consider what we would need for a good bound on the power function. Assume we can prove

$$P_X(x) \leq CE(x,h)$$

for all data sets X with fill distance at most h. This implies

$$|f(x) - \sum_{j=1}^{N} u_j^*(x) f(x_j)| \leq CE(x,h)\|f\|_{\mathcal{H}}, \quad \forall f \in \mathcal{H}, x \in \Omega$$

for the Lagrange-type basis u_j^* associated to the kernel k and data set X.

Now simplify Eq. (17) by introducing the error operator

$$E_x^y(f(y)) := f(x) - \sum_{j=1}^{N} u_j(x) f(x_j)$$

to get

$$\begin{aligned}
P_X^2(x) &\leq k(x,x) - 2\sum_{j=1}^{N} u_j(x)k(x_j,x) + \sum_{j,k=1}^{N} u_j(x)u_k(x)k(x_j,x_k) && (18)\\
&= k(x,x) - \sum_{j=1}^{N} u_j(x)k(x_j,x) + \sum_{j=1}^{N} u_j(x)\left(\sum_{k=1}^{N} u_k(x)k(x_j,x_k) - k(x_j,x)\right) && (19)\\
&= E_x^z k(z,x) - \sum_{j=1}^{N} u_j(x) E_x^z k(z,x_j) && (20)\\
&= E_x^y E_x^z k(z,x). && (21)
\end{aligned}$$

One technique is to use a bound of the form

$$|E_x^y(f(y))| := |f(x) - \sum_{j=1}^{N} u_j(x) f(x_j)| \leq \epsilon_{x,k}(h)\|Lf\| \tag{22}$$

with some linear differential operator L with values on some normed space. We then can bound the power function by

$$\begin{aligned}
P_X^2(x) &\leq |E_x^y E_x^z k(z,x)| && (23)\\
&\leq \epsilon_{x,k}(h)\|L^y E_x^z k(y,z)\| && (24)\\
&\leq \epsilon_{x,k}^2(h)\|L^y\|L^z k(y,z)\|\| && (25)
\end{aligned}$$

assuming the final expression makes sense.

Elementary univariate Case Consider a compact interval $\Omega = [a,b]$ and a finite subset $X = \{x_1, \ldots, x_N\} \subset \Omega$. Fix $x \in [a,b]$ and select a "local" subset

$$X_x := \{x_j \in X | j \in N(x) \subset \{1, \ldots, N\}$$

of points of X that are "sufficiently many" and "well-placed".

We fix $k \in \mathbb{N}$ and work locally with polynomials of order at most k. The simplest idea would be to pick the k closest neighbors to x within X and to perform local Lagrange interpolation by some polynomial p_X of order at most k at these points. We can take the error formula for interpolation in Newton form assigns

$$f(y) - p_x(y) = [y, X_x]f \prod_{x_j \in X_x} (y - x_j) \quad \text{for all } y \in [a, b]$$

where $[y, X_x]f$ os tje divided difference on the points of $X_x \cup \{y\}$ applied to f. If we assume f to be continuously k-times differentiable. we get the local error bound

$$|f(y) - p_x(y)| \leq \frac{\|f^{(k)}\|_{\infty,[a,b]}}{k!} \prod_{x_j \in X_x} |x - x_j|.$$

This is of the form Eq. (22), if we use the fact that

1. the 1st NN to x is at distance of at most h
2. the 2nd NN to x is at distance of at most $3h$
3. the 3rd NN to x is at distance of at most $5h$
4. ...
5. the k-th NN to x is distance of at most $(2k-1)h$

and thus

$$\prod_{x_j \in X_x} |x - x_j| \leq h^k \frac{(2k)!}{2^k k!}$$

leading to

$$|E_x^y f(y)| \leq h^k \frac{(2k)!}{2^k (k!)^2} \|f^{(k)}\|_{\infty,[a,b]}$$

Why do we look at polynomials ?

Consider the univariate case, with $f \in C^k$. At a point $x_0 \in \mathbb{R}$ we have the Tayolor polynomial

$$p(x) = \sum_{j=0}^{k-1} \frac{f^{(j)}(x_0)}{j!} (x - x_0)^j$$

and we have for $|x - x_0| \leq h$ the local approximation error

$$|f(x) - p(x)| = \frac{|f^{(k)}(\zeta)|}{k!} |x - x_0|^k \leq C h^k$$

with ζ between x and x_0. This local approximation roder is inherited by every approximation process that recovers polynomials at least locally.

Now we use a univariate kernel k that has k continuous and independent derivatives with respect to both variables. Then we can use (25) to get

$$P_x^2(x) \leq \left(h^k \frac{(2k)!}{2^k (k!)^2}\right)^2 \sup_{a\leq z\leq b} \sup_{a\leq y\leq b} \left|\frac{\partial^k}{\partial z^k}\frac{\partial^k}{\partial y^k} k(z,y)\right|$$

which gives

Theorem 1.32. *Assume a positive semi-definite kernel k on $[a,b]\times[a,b]$ that is k-times continuously and independently differentiable with respect to both arguments. Then, with the constant*

$$C_k = \frac{(2k)!}{2^k (k!)^2} \sqrt{\sup_{a\leq z\leq b} \sup_{a\leq y\leq b} \frac{\partial^k}{\partial z^k}\frac{\partial^k}{\partial y^k} k(z,y)}$$

for every point set $X \subset [a,b]$ consisting of at least k points and with fill distance at most h, the power function can be bounded as

$$P_X(x) \leq C_k h^k \quad \text{for all } x \in \Omega.$$

As seen for one dimension based on Taylor, if we have local polynomials reproduction, we can use polynomial approximation properties to get general approximation bounds.

Definition 1.33. A compact domain $\Omega \in \mathbb{R}^d$ allows *uniformly stable local polynomial reproduction* of order $l \geq 1$, if there are positive constants h_0, c_1, and c_2 such that for all finite sets $X = \{x_1, \ldots, x_N\} \subseteq \Omega$ with fill distance $h_{X,\Omega} \leq h_0$ there are scalars $u_1(x), \ldots, u_N(x)$ such that

1. $\sum_{j=1}^N u_j(x) p(x_j) = p(x)$ for all polynomials $p \in \mathcal{P}_\ell^d, x \in \Omega$,
2. $\sum_{j=1}^N |u_j(x)| \leq c_1$,
3. $u_j(x) = 0$ if $\|x - x_j\|_2 > c_2 h_{X,\Omega}$.

Observe Item 1 is the polynomial precision, Item 2 is needed to control the growth of the error estimates, where the left hand side is known as the Lebesgue constant at x, and Item 3 shows that the scheme is local.

We now aim for an error estimate in terms of the fill distance. For that we focus on positive definite kernels from now on.

It will be useful to view the power function as a function of the coefficients of f_X. With that in mind we can observe:

Theorem 1.34. *Let $\Omega \subseteq \mathbb{R}^d$ and let $k : \Omega \times \Omega \to \mathbb{R}$ be a positive definite kernel on $\mathbb{R}^d$. Let X be a set of N distinct points on Ω and define the quadratic form $Q : \mathbb{R}^N \to \mathbb{R}$ for any $x \in \Omega$ (see also Theorem 1.27):*

$$Q(u) := k(x,x) - 2\sum_{j=1}^{N} u_j k(x,x_j) + \sum_{i=1}^{N}\sum_{j=1}^{N} u_i u_j k(x_i,x_j).$$

The minimum of $Q(u)$ is given for the vector $u^\star(x)$ from Theorem 1.26:

$$Q(u^\star(x)) \le Q(u) \quad \text{for all } u \in \mathbb{R}^N.$$

Proof. With $b = [k(x_1,\cdot),\dots,k(x_N,\cdot)]^T$ and $A_{i,j} = k(x_i,x_j)$, $i,j = 1,\dots,N$ we have

$$Q(u) = k(x,x) - 2b^T(x)u + u^T A u.$$

The minimum of this quadratic form is the solution of the lnear equation system

$$Au = b(x),$$

which is fulfilled by $u = u^\star(x)$. ∎

For the following analysis we need the local existence of suitable coefficients vectors u for local polynomial reconstruction, which do exists for domains fulfilling the following condition.

Definition 1.35. A region $\Omega \subseteq \mathbb{R}^d$ satisfies an *interior cone condition* if there exists an angle $\Theta \in (0, \frac{\pi}{2})$ and a radius r such that for every $x \in \Omega$ there exists a unit vector $\xi(x)$ such that the cone

$$C = \left\{ x + \lambda y \;\middle|\; y \in \mathbb{R}^d, \|y\|_2 = 1, y^T\xi(x) \ge \cos\Theta, \lambda \in [0,r] \right\}$$

is contained in Ω.

Note that an Ω which satisfies this condition contains balls of a controllable radius, see [Wen05].

Theorem 1.36. *Assume $\Omega \in \mathbb{R}^d$ is bounded and satisfies an interior cone condition with agle $\Theta \in (0, \frac{\pi}{2})$ and radius r. Fix the nonnegative integer ℓ. Then Ω allows uniformly stable local polynomial reproduction with positive constants h_0, c_1 and c_2 depending only on l, Θ, r.*

Figure 1.6.: For the left domain the interior cone condition holds on the whole boundary, for the right not in the point p, where the only possible interior cone is a line.

Proof. See [Wen05]. ∎

We now will use multi-index notation and a multivariate Taylor expansion. For $\beta = (\beta_1, \dots, \beta_d)^T \in \mathbb{N}_d^0$ we define the differential operator

$$D^\beta = \frac{\partial^{|\beta|}}{(\partial_{x_1})^{\beta_1} \dots (\partial_{x_d})^{\beta_d}}.$$

The notation $D^\beta_{(2)} k(x, \cdot)$ indicates that the operator is applied to $k(x, \cdot)$ viewed as a function of the second variable. The multivariate Taylor expansion of $k(x, \cdot)$ centered at x is

$$k(x, y) = \sum_{|\beta| < 2k} \frac{D^\beta_{(2)} k(x, x)}{\beta!} (y - x)^\beta + R(x, y)$$

with remainder

$$R(x, y) = \sum_{|\beta| = 2k} \frac{D^\beta_{(2)} k(x, \xi_{x,y})}{\beta!} (y - x)^\beta,$$

where $\xi_{x,y}$ lies on the line connecting x and y.

With that we can formulate an error estimate in terms of the fill distance.

Theorem 1.37. *Assume $\Omega \subseteq \mathbb{R}^d$ is bounded, and satisfies an interior cone condition. Suppose $K \in C^{2k}(\Omega \times \Omega)$ is symmetric positive definite. Denote by f_X the interpolant to $f \in \mathcal{N}_k(\Omega)$ on the set X. Then there exists positive constants h_0 and C (independent of x, f and K) such that*

$$|f(x) - f_X(x)| \leq C h_{X,\Omega}^k \sqrt{C_k(x)} \|f\|_{\mathcal{N}_k(\Omega)}$$

provided that $h_{X,\Omega} \leq C h_0$. Here

$$C_k(x) := \max_{|\beta|=2k} \max_{x,y \in \Omega \cap B(x, c_2 h_{X,\Omega})} \left| D_{(2)}^\beta k(x,y) \right|.$$

Proof. We know from Theorem 1.25

$$|f(x) - f_X(x)| \leq P_X(x) \|f\|_{\mathcal{N}_k(\Omega)}.$$

We aim for a bound for the power function in terms of the fill distance

$$P_X(x) \leq C h_{X,\Omega}^k \sqrt{C_k(x)}.$$

So far we know

$$P_X(x)^2 = Q(u^\star(x))$$

and that $Q(u)$ is minimized by $u = u^\star(x)$, i.e. any other coefficient vector will result in an upper bound using the Theorem 1.34. With $\tilde{u}(x)$ from Theorem 1.36 we have polynomial precision of degree $\ell \geq 2k-1$. For the $\tilde{u}(x)$ we see (hereafter abbreviating by writing $\tilde{u}$ for $\tilde{u}(x)$)

$$\begin{aligned} P_X(x)^2 &\leq Q(\tilde{u}) \\ &= k(x,x) - 2\sum_{j=1}^N \tilde{u}_j k(x, x_j) - 2\sum_{i=1}^N \sum_{j=1}^N \tilde{u}_i \tilde{u}_j k(x_i, x_j) \end{aligned}$$

where many $\tilde{u}_j$ will be zero. We now apply the Taylor expansion centered at x to $k(x,\cdot)$ and centered at x_i to $k(x_i,\cdot)$ and evaluate both at x_j:

$$\begin{aligned} Q(\tilde{u}) = k(x,x) &- 2\sum_j \tilde{u}_j \left[\sum_{|\beta|<2k} \frac{D_{(2)}^\beta k(x,x)}{\beta!} (x_j - x)^\beta + R(x, x_j) \right] \\ &+ \sum_i \sum_j \tilde{u}_i \tilde{u}_j \left[\sum_{|\beta|<2k} \frac{D_{(2)}^\beta k(x_i,x_i)}{\beta!} (x_j - x_i)^\beta + R(x_i, x_j) \right] \end{aligned}$$

With the notation $\sum_i$ and $\sum_j$ we point out that we only sum over the indices where $\tilde{u}_i$ and $\tilde{u}_j$ are respectively nonzero.

Next we identify $p(z) = (z-x)^\beta$ so that $p(x) = 0$ unless $\beta = 0$. With the polynomial precision property of $\tilde{u}$ this simplifies $Q(\tilde{u})$ to

$$\begin{aligned} Q(\tilde{u}) = k(x,x) - 2k(x,x) - 2\sum_j \tilde{u}_j R(x,x_j) \\ + \sum_i \tilde{u}_i \sum_{|\beta|<2k} \frac{D^\beta_{(2)} k(x_i,x_i)}{\beta!}(x-x_i)^\beta + \sum_i \sum_j \tilde{u}_i \tilde{u}_j R(x_i,x_j). \end{aligned}$$

Now we apply Taylor again and observe that when evaluating at x

$$\sum_{|\beta|<2k} \frac{D^\beta_{(2)} k(x_i,x_i)}{\beta!}(x-x_i)^\beta = k(x_i,x) - R(x_i,x).$$

Inserting this into the above gives

$$\begin{aligned} Q(\tilde{u}) = -k(x,x) - \sum_j \tilde{u}_j \left[2R(x,x_j) - \sum_i \tilde{u}_i R(x_i,x_j) \right] \\ + \sum_i \tilde{u}_i \left(k(x_i,x) - R(x_i,x) \right). \end{aligned}$$

Using Taylor once more shows:

$$\begin{aligned} k(x_i,x) &= k(x,x_i) \\ &= \sum_{|\beta|<2k} \frac{D^\beta_{(2)} k(x,x)}{\beta!}(x_i-x)^\beta + R(x,x_i). \end{aligned}$$

Inserting this with observing

$$\sum_i \tilde{u}_i k(x_i,x) = k(x,x) + \sum_i \tilde{u}_i R(x,x_i)$$

as above using the polynomial precision property:

$$Q(\tilde{u}) = -\sum_j \tilde{u}_j \left[R(x,x_j) + R(x_j,x) - \sum_i \tilde{u}_i R(x_i,x_j) \right].$$

The polynomial reproduction Definition 1.33 gives $\sum_j |\tilde{u}_j| \le c_1$.

We know for $\tilde{u}_j \neq 0$ that $\|x - x_j\|_2 \le c_2 h_{X,\Omega}$ and also get $\|x_i - x_j\|_2 \le 2c_2 h_{X,\Omega}$ if $\tilde{u}_i \neq 0$, $\tilde{u}_j \neq 0$. Therefore all three remainder terms can be bounded by an expression $C h^{2k}_{X,\Omega} C_k(x)$, where the interior cone condition ensures that the ball remains inside. Combining these and taking the square root gives the bound for the power function. ∎

The theorem says, that interpolation with a C^{2s} smooth kernel k has approximation order s, if f is in the corresponding native space. For infinitely smooth positive definite

kernels such as the Gaussian or the (generalized) inverse multiquadrics the approximation order is arbitrarily high.

This is still a generic estimate, the factor $C_k(x)$ depends on k, and for many kernel functions it is possible to get additional powers of h out of C_k.

1.2.1. Generalized Interpolation

So far we dealt with point evaluation functionals, but we want to also consider more general functionals. In particular

- derivation $\lambda(f) = \frac{\partial f}{\partial x_j}(\underline{z})$,
- integration $\lambda(f) = \int_\Omega f(z)\,\mathrm{d}z$.

We consider a subset $\Lambda \subseteq \mathcal{H}^\star$ of the dual that generalizes the role of the point set X and the associated point evaluation functionals δ_x. One can say that we now consider interpolation using the data $\lambda(f)$ for all $\lambda \in \Lambda$. The construction from functional analysis that we have seen earlier carries over do this setting of closures of subsets in the dual that are more specific than just point evaluation functionals.

The goal is to treat a *Dirichlet boundary value problem*

$$\begin{aligned} Lu &= f \qquad \text{in } \Omega \subseteq \mathbb{R}^d \\ u &= g \qquad \text{on } \Gamma = \partial\Omega, \end{aligned} \tag{26}$$

where L is a linear differential operator.

Collocation is a general approach which treats the problem in a strong sense and discretizes it by

$$\begin{aligned} Lu\left(\underline{x}_j^\Omega\right) &= f\left(\underline{x}_j^\Omega\right), \qquad \underline{x}_j^\Omega \in \Omega, 1 \le j \le N^\Omega \\ u\left(\underline{x}_j^\Gamma\right) &= g\left(\underline{x}_j^\Gamma\right), \qquad \underline{x}_j^\Gamma \in \Gamma, 1 \le j \le N^\Gamma. \end{aligned} \tag{27}$$

with $N = N^\Omega + N^\Gamma$.

The exact solution of Eq. (26) will surely satisfy Eq. (27), but there are many functions which satisfy Eq. (27). We will consider a finite dimensional space U with at least N dimensions. The question arises if Eq. (27) is solvable for $u \in U$.

Interpolation of general functionals $\lambda_1, \ldots, \lambda_N$ by a span of functions $u_1, \ldots, u_N$ is difficult without additional properties. In particular we need additional properties to get a generalized kernel matrix which is nonsingular.

We now proceed to study *Hermite-interpolation*, where also derivatives are used for interpolation. Here, we assume to have data

$$\{(\underline{x}_i, \lambda_i f)\}, \quad \underline{x}_i \in \mathbb{R}^d \text{ with } \Lambda = \{\lambda_1, \dots, \lambda_N\}$$

and a linearly independent set of continuous linear functionals. We aim for an interpolant

$$u(\underline{x}) = \sum_{i=1}^{N} u_i \lambda_i^{(1)} k(\underline{x}, \underline{x}), \quad \underline{x} \in \mathbb{R}^d$$

that satisfies

$$\lambda_i u = \lambda_i f \quad i = 1, \dots, N.$$

Here $\lambda^{(1)}$ indicates that the functional acts on the first argument of K.

The resulting linear system has a matrix with entries

$$A_{ij} = \lambda_i^{(2)} \lambda_j^{(1)} k(x_j, x_i), \quad i, j = 1, \dots, N.$$

It can be shown that A is nonsingular for positive semi-definite kernels, but also for more general conditionally positive semidefinite (cpsd) kernels, see [Wu92; Wen05].

Assume the λ_j are of the form

$$\lambda_j = \delta_{\underline{x}_j} \circ D^{\alpha(j)}$$

and $\alpha(j) \neq \alpha(k)$ if $\underline{x}_j = \underline{x}_k$ for $k \neq j$ the λ_j are linearly independent on the native space of a positive definite kernel, see [Wen05].

One advantage of Hermite interpolation is that less data locations are required for a certain predictive accuracy. For example, if a data location corresponds to running an expensive numerical simulation with a specific choice of d simulation parameters, and the simulation procedure can deliver both, a function value and a derivative value, then using the gradient data is far more efficient than generating further d function values.

Example. For illustration we now denote the center of the RBFs by ξ_j and the data locations by x_j, although these are the same locations.

Given are

$$\{(\underline{x}_j, f(\underline{x}_j))\}_{j=1}^{p} \text{ and } \left\{ \left(\underline{x}_j, \frac{\partial f}{\partial x}(\underline{x}_j) \right) \right\}_{j=p+1}^{N},$$

with $\underline{x} = (x, y) \in \mathbb{R}^2$. Thus

$$\lambda_j = \begin{cases} \delta_{\underline{x}_j} & j = 1, \dots, p \\ \delta_{\underline{x}_j} \circ \frac{\partial}{\partial x} & j = p+1, \dots, N. \end{cases}$$

With $k(\underline{x}_j, \underline{x}_k) = \varphi(\|\underline{x}_j - \underline{x}_k\|)$ we get,

$$\begin{aligned} u(\underline{x}) &= \sum_{j=1}^{N} a_j \lambda_j^{(1)} k(\cdot, \underline{x}) \\ &= \sum_{j=1}^{P} a_j k(\underline{\xi}_j, \underline{x}) + \sum_{j=P+1}^{N} a_j \frac{\partial}{\partial \xi} k(\underline{\xi}_j, \underline{x}) \\ &= \sum_{j=1}^{P} a_j k(\underline{x}_i, \underline{x}) - \sum_{j=P+1}^{N} a_j \frac{\partial}{\partial x} k(\underline{\xi}_j, \underline{x}) \end{aligned}$$

The system matrix after inserting u into $\lambda_j u = \lambda_j f$ is

$$A = \begin{bmatrix} K & K_\xi \\ K_X & K_{XX} \end{bmatrix}$$

with

$$\begin{aligned} K_{jk} = k(\underline{\xi}_k, \underline{x}_j) &= \varphi(\|\underline{\xi}_k - \underline{x}_j\|), && j,k = 1,\dots,p \\ K_{\xi,jk} = \frac{\partial \varphi}{\partial \xi}(\|\underline{\xi}_k - \underline{x}_j\|) &= -\frac{\partial \varphi}{\partial x}(\|\underline{\xi}_k - \underline{x}_j\|), && j = 1,\dots,p, k = p+1,\dots,N \\ K_{X,jk} = \frac{\partial \varphi}{\partial x}(\|\underline{\xi}_k - \underline{x}_j\|), && j = p+1,\dots,N, k = 1,\dots,p \\ K_{XX,jk} = -\frac{\partial^2 \varphi}{\partial^2 x}(\|\underline{\xi}_k - \underline{x}_j\|), && j,k = p+1,\dots,N \end{aligned}$$

Observe here, that the partial derivative of φ with respect to x will always contain a linear factor in x. The sign for the derivative depends switches if we switch between $\underline{\xi}_j$ and $\underline{x}_j$. In case the ξ_j and x_j are the same, we have that the entries of K_ξ and K_X^T correspond, since the sign change due to the differentiation at the other position is cancelled by the interchange of the roles of x and ξ. Therefore we later use $L^{(1)}, L^{(2)}$ when considering differential operators.

In view of what we had earlier, instead of

$$k(x,y) = \langle \delta_x, \delta_y \rangle_{\mathcal{H}^\star}.$$

we have now in some way

$$k^\star(\lambda, \mu) = \langle \lambda, \mu \rangle_{\mathcal{H}^\star} \quad \text{for all } \lambda, \mu \in \mathcal{H}^\star,$$

which gives

$$k(x,y) = k^\star(\delta_x, \delta_y),$$

i.e. for point evaluation as before.

One can now redo most of what we did earlier using $\Omega = \mathcal{H}^\star$ replacing points x and y by functionals λ and μ while $k^\star$ replaces k. Note, that this also allows us to work in Hilbert spaces without continuous point evaluation where one uses weak methods, e.g. Sobolev spaces.

Theorems 1.19 and 1.20 and Corollaries 1.21 to 1.23 again follow. We cannot use point evaluation functionals to measure the error and instead use a functional $\mu \in \mathcal{H}^\star$

$$\mu(f - f_\Lambda) = (\mu - \mu \circ \Pi_\Lambda) f$$

so we get the generalized power function

$$P_\Lambda(\mu) = \|\mu - \mu \circ \Pi_\Lambda\|_{\mathcal{H}^\star} \quad \text{for all } \mu \in \mathcal{H}^\star,$$

and obtain Theorem 1.25 with the generalized power function.

Now we describe a kernel-based collocation method for the partial differential equation

$$\begin{aligned} Lu &= f && \text{in } \Omega \subseteq \mathbb{R}^d \\ u &= g && \text{on } \Gamma = \partial\Omega \end{aligned}$$

We use an expansion for u as

$$u(x) = \sum_{i=1}^{p} u_i k(x_i, x) + \sum_{i=p+1}^{N} u_i L^{(1)}(k(x_i, x)) \tag{28}$$

where p is the number of boundary points and $N - p$ the number of interior points denoted by I, accordingly. We now split the collocation set X in to a set of boundary points B and the set I of interior points. Similar to the Hermite interpolation before we get a block matrix

$$A = \begin{pmatrix} K & L^{(1)}(K) \\ L^{(2)}(K) & L^{(2)}L^{(1)}(K) \end{pmatrix}, \quad \text{and} \quad Au = \begin{pmatrix} g \\ f \end{pmatrix}$$

with g being of size p and f of size $N - p$, accordingly, and

$$\begin{aligned} K_{ij} &= k(x_i, x_j) && x_i, x_j \in B \\ L^{(1)}(K)_{ij} &= L^{(1)}\left(K\left(x_i, \tilde{x}_j\right)\right) && x_i \in B, \tilde{x}_j \in I \\ L^{(2)}(K)_{ij} &= L^{(2)}\left(K\left(\tilde{x}_i, x_j\right)\right) && \tilde{x}_i \in I, x_j \in B \\ L^{(2)}L^{(1)}(K)_{ij} &= L^{(2)}\left(L^{(1)}\left(K\left(\tilde{x}_i, \tilde{x}_j\right)\right)\right) && \tilde{x}_i, \tilde{x}_j \in I \end{aligned}$$

The matrix is of the same type as the Hermite interpolation matrix and nonsingular if the $\delta_{x_i}, \delta_{x_i} \circ L$ are linearly independent.

For this symmetric collocation method the approximation result carries over.

Theorem 1.38. *Let $\Omega \subseteq \mathbb{R}^d$ be a polygonal and open domain. Furthermore, let $L \neq 0$ be a second order elliptic differential operator with coefficients in $C^{2(k-2)}\left(\bar{\Omega}\right)$ that either vanishes on $\partial\Omega$ or has no zero there. Suppose that $K \in C^{2k}(\mathbb{R}^d \times \mathbb{R}^d)$ is a positive definite kernel. Assume further that*

$$\begin{aligned} Lu &= f \qquad \text{in } \Omega \subseteq \mathbb{R}^d \\ u &= g \qquad \text{on } \Gamma = \partial\Omega \end{aligned}$$

has a unique solution $u \in \mathcal{N}_k(\Omega)$ for given $f \in C(\Omega)$ and $g \in C(\partial\Omega)$. Moreover, let $\hat{u}$ be the approximation in the form Eq. (28). Then

$$\begin{aligned} \|u - \hat{u}\|_{L_\infty(\Omega)} &\leq C h_{I,\Omega}^{k-2} \|u\|_{\mathcal{N}_k(\Omega)} \\ \|u - \hat{u}\|_{L_\infty(\partial\Omega)} &\leq C h_{B,\Omega}^{k} \|u\|_{\mathcal{N}_k(\Omega)} \end{aligned}$$

for small enough $h_{I,\Omega}$, $h_{B,\Omega}$.

Proof (Sketch). The result for the interior essentially uses the approximation Theorem 1.37 for $|Lu - L\hat{u}|$, and uses that "LLK" is positive definite for positive definite K.

Since $\partial\Omega$ does not satisfy an interior cone condition, we cannot use Theorem 1.37 directly for the result on the boundary. As Ω was demanded to be polygonal, $\partial\Omega$ is locally a hyperplane which can be mapped to $\mathbb{R}^{d-1}$ where the image satisfies an interior cone condition. Using further results connecting K and LLK, their native spaces, and their power functions one can show the result for each surface, of which there are only finitely many.

For full (somewhat messy) details, see [Wen05]. ∎

From the result, one should aim to have a finer discretization in the interior than on the boundary by balancing $h_{I,\Omega}^{k-2} \approx h_{B,\Omega}^{k}$.

We need high regularity assumptions for the good approximation rates, but often $u \in C^{2k}(\Omega)$ cannot be assumed and here finite element methods are the better approach. But in particular in higher dimensions and with decent smoothness these meshless methods have their potential.

In practice, a non symmetric collocation approach is more commonly used, called Kansa's method. That approach is simpler to implement since only one application of L is needed. But it can be shown that this approach can fail, albeit only known for specifically constructed situations.

In short, we use

$$u(x) = \sum_{i=1}^{N} u_i k(x_i, x)$$

and get

$$A = \begin{pmatrix} K \\ LK \end{pmatrix}, \quad Au = \begin{pmatrix} g \\ f \end{pmatrix}$$

with

$$\begin{aligned} K_{ij} &= k(x_i, x_j), & x_i \in B, x_k \in X \\ (LK)_{ij} &= L^{(1)} k(x_i, x_j), & x_i \in I, x_k \in X \end{aligned}$$

This gives again a $N \times N$-matrix, but non-symmetric. Besides needing less derivatives and therefore less smoothness, the non-symmetric approach can be easier applied to a partial differential equation (PDE) with non-constant coefficients or nonlinear problems.

1.2.2. Conditionally Positive Semi-Definite Kernels

Definition 1.39. We call a set of points $X = \{x_1, \ldots, x_N\} \subset \mathbb{R}^d$ *m-unisolvent* if the only polynomial of total degree at most m interpolating zero data on X is the zero polynomial.

Definition 1.40. A symmetric kernel $K : \Omega \times \Omega \to \mathbb{R}$ is called *conditionally positive semidefinite of order m*, if

$$\sum_{i,j=1}^{N} a_i a_j K(x_i, x_j) \geq 0$$

for any $(m-1)$-unisolvent set of points $x_1, \ldots, x_N \in \Omega \subseteq \mathbb{R}^d$ and $a_i \in \mathbb{R}^N$ satisfying the moment conditions

$$\sum_{i=1}^{N} a_i p(x_i) = 0 \tag{29}$$

for any polynomial p of total degree at most $m-1$.

Conditionally positive definite of order m is defined accordingly.

Corollary 1.41. *A kernel that is cpsd of order m is also cpsd of order $\ell \geq m$. In particular, a positive semi-definite kernel is cpsd of any order.*

Thus, usually one gives the minimal order for a function.

Example. Multiquadratics in the form of

$$k(x,y) = \Phi(\|x-y\|) = \Phi(r) = (-1)^{\lceil \beta \rceil}(1+r^2)^{\beta}, \quad 0 < \beta \notin \mathbb{N}$$

are cpsd of order $m = \lceil \beta \rceil$.

Thin-plate-splines of the form

$$\Phi(r) = (-1)^{\beta+1} r^{2\beta} \log r, \quad \beta \in \mathbb{N}$$

are cpsd of order $m = \beta + 1$. The classical thin-plate-spline with $\beta = 1$ is cpsd of order 2.

For radial functions $K(x,y) = \Phi(\|x-y\|)$ we can use simpler criteria, which are based on monotone functions.

Definition 1.42. A function $\varphi : [0,\infty) \to \mathbb{R}$ that is in $C^\infty((0,\infty))$ and satisfies

$$(-1)^\ell \varphi^{(\ell)}(r) \geq 0, \quad r > 0, \ell = 0, 1, 2, \ldots$$

is called *completely monotone* on $(0,\infty)$.

If in addition $\varphi \in C([0,\infty))$ it is called completely monotone on $[0,\infty)$

Example. Examples of completely monotone functions are

- $\varphi(r) = \varepsilon$
- $\varphi(r) = e^{-\varepsilon r}$ with $\varepsilon \geq 0$. This is due to

$$(-1)^\ell \varphi^{(\ell)}(r) = \varepsilon^\ell e^{-\varepsilon r} \geq 0$$

Theorem 1.43. *Let $\varphi \in C([0,\infty)) \cap C^\infty((0,\infty))$. Then the kernel $k(x,y) = \Phi(\|x-y\|)$ with $\Phi(r) = \varphi(r^2)$ is cpsd of order m if and only if $(-1)^m \varphi^{(m)}$ is completely monotone on $(0,\infty)$.*

Proof. [Mic86; Wen05].

Sufficient is based on the observation that completely monotone functions are Laplace transforms of non-negative and finite Borel measures, i.e.

$$\phi(r) = \mathcal{L}\nu(r) = \int_0^\infty e^{-rt} d\nu(t).$$

With Taylor on $\phi_\epsilon(r) = \phi(r+\epsilon)$ the result follows.

Necessary is based on a result from Schoenberg that a function ϕ is completely monotone on $[0, \infty)$ if and only in $\Phi(r) \coloneqq \phi(r^2)$ is positive semi-definite, followed by a proof by induction.

Example (revisited). 1. For the multiquadratics

$$\varphi(r) = (-1)^{\lceil \beta \rceil}(1+r)^\beta, \quad 0 < \beta \notin \mathbb{N}$$

it follows

$$\varphi^{(\ell)}(r) = (-1)^{\lceil \beta \rceil}\beta(\beta-1)\cdot \ldots \cdot (\beta-\ell+1)(1+r)^{\beta-\ell},$$

so that

$$(-1)^{\lceil \beta \rceil}\varphi^{(\lceil \beta \rceil)}(r) = \beta(\beta-1)\cdot \ldots \cdot (\beta-\lceil \beta \rceil+1)(1+r)^{\beta-\lceil \beta \rceil} \geq 0$$

and φ is therefore completely monotone and the corresponding kernel is cpsd of order $\lceil \beta \rceil$.

2. The thin-plate-splines can also be seen to be completely monotone and therefore cpsd of order $\beta + 1$.

One can show a stronger version of the theorem.

Corollary 1.44. *Suppose that the function φ of Theorem 1.43 is not a polynomial of degree at most m. Then $\varphi(r^2)$ is conditionally positive definite of order m.*

Interpolation with cpsd kernels

We aim for

$$s(x) \coloneqq S_{X,a,b}(x) \coloneqq \sum_{i=1}^{N} a_i k(x_i, x) + \sum_{m=1}^{M} b_m p_m(x) \quad \text{for all } x \in \Omega$$

with $a \in \mathbb{R}^N$, $b \in \mathbb{R}^M$ with the moment condition Eq. (29) for a.

We use the space $\mathcal{P}^d_{m-1}$ of polynomials of total degree less or equal to $m-1$ in d variables and polynomials $p_1, \ldots, p_M$ from a basis of size

$$M = \binom{m-1+d}{m-1}$$

for $\mathcal{P}^d_{m-1}$.

We want to interpolate $(x_i, f(x_i))$ on an $(m-1)$-unisolvent point set X. This results in the $(N+M) \times (N+M)$ linear system

$$\begin{aligned} s(x_k) = \sum_{i=1}^{N} a_i k(x_i, x_k) + \sum_{i=1}^{M} b_m p_m(x_k) &= f_k \qquad 1 \le k \le N \\ \sum_{c=1}^{N} a_i p_m(x_i) + 0 &= 0 \qquad 1 \le m \le N \end{aligned}$$

In matrix form, this yields:

$$\begin{bmatrix} K & P \\ P^T & 0 \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} f \\ 0 \end{bmatrix}$$

Theorem 1.45. *If the set X is $(m-1)$-unisolvent and K is conditionally positive definite (not semidefinite) we can solve the system*

$$\begin{bmatrix} K & P \\ P^T & 0 \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} f \\ 0 \end{bmatrix}$$

uniquely.

Proof. We assume a, b is a solution of the homogeneous linear system, i.e. $f_k = 0$, $1 \le k \le N$ and show that $a = 0, b = 0$ is the only possible solution. Multiply the first line by a^T:

$$a^T K a + a^T P b = 0$$

We know from the bottom line $P^T a = 0$, therefore $a^T P = 0^T$. Thus, we conclude

$$a^T K a = 0.$$

Since we have a conditionally positive definite K of order m, $P^T a = 0$ and we have a $(m-1)$-unisolvent set, this only holds for $a = 0$. The unisolvency of the X also gives the linear independence of the columns of P and so it follows from

$$Pb = Ka + Pb = 0$$

that $b = 0$. ∎

Taking a linear algebra view, the kernel matrix of a cpsd kernel is positive semi-definite on the space of vectors "orthogonal" to d-variate polynomials of degree at most $m-1$.

Consider now cpsd of order one. Then k is cpsd on a subspace of dimension $N-1$.

Theorem 1.46 (Courant-Fisher). *Let A be a real symmetric $N \times N$-matrix with eigenvalues $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_N$. Then*

$$\lambda_k = \max_{\dim V = k} \min_{\substack{x \in V, \\ \|x\|=1}} x^T A x$$

and

$$\lambda_k = \min_{\dim V = N-k+1} \max_{\substack{x \in V, \\ \|x\|=1}} x^T A x.$$

We get from Theorem 1.46 that at least $N-1$ eigenvalues of a kernel matrix for a cpd-kernel are positive. With an additional assumption we even get:

Theorem 1.47. *An $N \times N$ kernel matrix K that is cpd of order 1 and has a non-positive trace possesses one negative and $N-1$ positive eigenvalues.*

Proof. Let $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_N$ denote the eigenvalues of K. From Theorem 1.46 we get

$$\lambda_{N-1} = \max_{\dim V = N-1} \min_{\substack{x \in V, \\ \|x\|=1}} x^T K x \geq \min_{c: \sum c_k = 0, \atop \|c\|=1} c^T K c \geq 0,$$

so that K hast at least $N-1$ positive eigenvalues. Since $tr(K) = \sum_{k=1}^{N} \lambda_k \leq 0$, K also must have one negative eigenvalue. ∎

We now can use Theorem 1.47 to conclude that we can use RBF that are cpd order one without appending the constant term to solve the interpolation problem.

Theorem 1.48. *Suppose Φ is cpd of order one and that $\Phi(0) \leq 0$. Then for a set X of distinct points, the matrix K with $K_{jk} = \Phi(\|x_j - x_k\|)$ has $N-1$ positive and one negative eigenvalue. Is it therefore non-singular.*

Proof.

$$tr(K) = N\Phi(0) \leq 0$$

For the learning case we did relate kernels to scalar products in the the feature space. Now we consider distance measures from norms in the feature space. We assume that $\|\Phi(x) - \Phi(y)\|^2$ is stemming from the positive (semi)definiteness of the kernel, i.e. with the kernel

$$k(x, y) = \langle \Phi(x), \Phi(y) \rangle$$

we can write the norm as

$$\|\Phi(x) - \Phi(y)\|^2 = k(x, x) + k(y, y) - 2k(x, y).$$

Consider now translations of the data $x \mapsto x - t$: $\|x-y\|^2$ is translation invariant, while $\langle x, y\rangle$ is not. After a short calculation, we observe the connection

$$\langle (x-t), (y-t)\rangle = \frac{1}{2}\left(-\|x-y\|^2 + \|x-t\|^2 + \|t-y\|^2\right) \tag{30}$$

and

$$\sum_{i,j} a_i a_j \langle (x_i - t), (x_j - t)\rangle = \left\| \sum_i a_i (x_i - t) \right\|^2 \geq 0.$$

Therefore we still have a positive semidefinite kernel. In other words for any choice of t we get a similarity measure Eq. (30) associated with the dissimilarity measure $\|x-y\|$.

A natural extension is to consider other nonlinear dissimilarity measures.

Lemma 1.49. *Let $t \in \Omega$ and k be a symmetric kernel on $\Omega \times \Omega$. Then*

$$\tilde{k}(x,y) \coloneqq \frac{1}{2}\left(k(x,y) - k(x,t) - k(t,y) + k(t,t)\right)$$

is positive semidefinite if and only if K is cpsd.

If $k(t,t) \leq 0$ then

$$\hat{k}(x,y) \coloneqq \frac{1}{2}\left(k(x,y) - k(x,t) - k(t,y)\right)$$

is positive semidefinite if and only if k is cpsd.

Proof. see exercise ∎

This does generalize Eq. (30), i.e. the negative squared distance is cpsd. Here $\sum_{i=1}^N a_i = 0$ implies

$$\begin{aligned} -\sum_{i,j} a_i a_j \|x_i - x_j\|^2 &= -\sum_i a_i \underbrace{\sum_j a_j \|x_j\|^2}_{\text{const}} - \sum_j a_j \sum_i a_i \|x_i\|^2 + 2\sum_{i,j}\langle x_i, x_j\rangle \\ &= 0 + 0 + 2\left\|\sum_i a_i x_i\right\|^2 \geq 0 \end{aligned}$$

Sometimes a definition for negative kernels with a smaller or equal to zero in the quadratic form, but again the condition $\sum_{i=1}^N a_i = 0$ is used, e.g. this holds for the minus of cpsd kernels of order one.

Actually all

$$k(x,y) = -\|x-y\|^\beta, \quad 0 < \beta < 2$$

are cpsd due to

Proposition 1.50. *If $k : \Omega \times \Omega \to]-\infty, 0]$ is cpsd, then so are $-(-k)^\alpha$ with $0 < \alpha < 1$ and $-\log(1 - K)$.*

Proof. [BCR84] ∎

Observe that sums of cpsd kernels are also cpsd and any constant $b \in \mathbb{R}$ is cpsd. So any $k + b$ for k cpsd is also cpsd.

Taking the feature space view once more, we can construct a Hilbert space representation of a cpsd k from the corresponding positive semidefinite (psd) $\tilde{k}$. For $\tilde{k}$ we have a feature map $\Phi : \Omega \to \mathcal{F}$, i.e.

$$\langle \Phi(x), \Phi(y) \rangle_{\mathcal{F}} = \tilde{K}(x, y).$$

Therefore

$$\begin{aligned} \|\Phi(x) - \Phi(y)\|_{\mathcal{F}}^2 &= \langle \Phi(x) - \Phi(y), \Phi(x) - \Phi(y) \rangle \\ &= \tilde{k}(x, x) + \tilde{k}(y, y) - 2\tilde{k}(x, y) \end{aligned}$$

Inserting into the result from Lemma 1.49 gives for fixed $t \in \Omega$:

$$\|\Phi(x) - \Phi(y)\|^2 = -k(x, y) + \frac{1}{2}\left(k(x, x) + k(y, y)\right).$$

We have just shown:

Theorem 1.51 (Hilbert space representation of a cpsd kernel). *Let k be a cpsd kernel on Ω. Then there exists a Hilbert space $\mathcal{F}$ and a mapping $\Phi : \Omega \to \mathcal{F}$ such that*

$$k(x, y) = -\|\Phi(x) - \Phi(y)\|^2 + \frac{1}{2}\left(k(x, x) + k(y, y)\right).$$

If $k(x, x) = 0$ for all $x \in \Omega$ we have

$$k(x, y) = -\|\Phi(x) - \Phi(y)\|^2$$

and $\sqrt{-k(x, y)}$ is a semi-metric, and a metric if $k(x, y) \neq 0$ for $x \neq y$.

One sees that cpsd kernels are a natural choice in case a translation should not affect the outcome of the kernel.

Remembering quasi-interpolation from Theorem 1.26, we would like to have something similar for cpsd kernels. For that one needs in some sense to redo the construction of a native space. To give an idea of the path let us look at the inner product. Like for the psd case before we fix Ω, m, K and define the set

$$M := \left\{ (a, X) \;\middle|\; X \subseteq \Omega,\ (m-1)\text{-unisolvent}, |X| = N, a \in \mathbb{R}^N, P_X^T a = 0 \right\}$$

of vector/set pairs that satisfy the moment condition

$$P_X^T a = 0, \text{ with } P_X^T = (P_j(x_k)).$$

We assume that Ω has at least one $(m-1)$-unisolvent set. We define as before

$$H := \left\{ \lambda_{a,X}^y K(x,y) \;\middle|\; (a,X) \in M \right\}$$

and the space of functionals

$$L := \left\{ f \mapsto \lambda_{a,X}(f) := \sum_{i=1}^{N} a_j f(x_j) \;\middle|\; (a,X) \in M, f \in H \right\}.$$

L is a linear space, e.g. adding two functionals vanishing on $\mathcal{P}_{m-1}^d$ will result in a functional vanishing on $\mathcal{P}_{m-1}^d$. This holds accordingly for H. We now define a bilinear form on L as we did before Theorem 1.7, where the moment condition additionally holds. Theorems 1.7 and 1.8 carry over. We cannot use functionals $\delta_x = \lambda_{1,x}$ for providing point evaluation, since they are not necessarily in L. Nonetheless, we do have the Riesz map:

$$\begin{aligned} R : L &\to H, \\ R(\lambda_{a,X})(y) &= \lambda_{a,X}^y k(y,x) =: f_{a,X}(y) \end{aligned}$$

and the identities

$$\langle \lambda_{a,X}, \lambda_{b,Y} \rangle_L = \langle f_{b,Y}, f_{a,X} \rangle_H = \lambda_{a,X}(f_{b,Y})$$

for all $(a,X),(b,Y) \in M$.

Theorem 1.52. *The sum of the spaces $\mathcal{P}_{m-1}^d + H$ is a direct sum if the kernel k is cpsd of order m.*

Proof. Consider $p \in \mathcal{P}_{m-1}^d$, and a functional $\lambda_{b,Y} \in L$ with $p(x) = \lambda_{b,Y}^y k(x,y)$ for all $x \in \Omega$. Then $\lambda_{a,X}(p) = 0 = \langle \lambda_{a,X}, \lambda_{b,Y} \rangle_L$ for all $\lambda_{a,X} \in L$, in particular for $\lambda_{b,Y}$. Thus $\lambda_{b,Y} = 0$ as a functional on H, but $b = 0$ holds only in the case of definiteness. With the linearity of the Riesz map, one can conclude that $f_{b,Y}$ is zero in the general case and therefore also p. ∎

This space $\mathcal{P}_{m-1}^d + H$ can be used as a pre-native space for a cpsd kernel k. One can follow a similar path to completion like for psd kernels, where the right understanding of the addition of $P + H$ in that process makes the derivation more involved. For details see [Wen05].

As a last observation, there is a way to transition from a cpsd kernel of order m to a psd kernel.

Fix a $(m-1)$-unisolvent set Z of size N_Z. Every $p \in \mathcal{P}^d_{m-1}$ can be reproduced by a Lagrange basis $p_1, \dots, p_{N_Z}$ with

$$p_j(z_k) = \delta_{jk}, \quad 1 \le j,k \le N_Z, \quad \text{i.e.}$$

$$p(x) = \sum_{k=1}^{N_Z} p(z_k) p_k(x) =: (\Pi_z(p))(x) \quad \text{for all } x \in \Omega, p \in \mathcal{P}^d_{m-1}$$

after changing to the Lagrange basis. This defines a linear projector Π_z onto $\mathcal{P}^d_{m-1}$ that extends to general functions f on Ω as

$$(\Pi_z(f)))(x) = \sum_{k=1}^{N_Z} f(z_k) p_k(x) \quad \text{for all } x \in \Omega, f : \Omega \to \mathbb{R}.$$

This implies that the functionals

$$\mu_x = \delta_x - \sum_{k=1}^{N_Z} p_k(x) \delta_{z_k}$$

satisfy the moment conditions. We now define the reduced kernel

$$\begin{aligned} \tilde{k}(x,y) &:= \langle \mu_x, \mu_y \rangle \\ &= \mu_y^{(1)} \mu_x^{(2)} k(\cdot,\cdot) \\ &= k(x,y) - \sum_{k=1}^{N_Z} p_k(x) k(z_k, y) \\ &\quad - \sum_{k=1}^{N_Z} p_k(y) k(x, z_k) + \sum_{j,k=1}^{N_Z} p_j(x) p_k(y) k(z_j, z_k) \end{aligned}$$

for all $x, y \in \Omega$.

Theorem 1.53. *The reduced kernel is symmetric and psd on Ω. It vanishes, if one of the arguments is in Z. If k is cpsd of order m, then $\tilde{k}$ is psd on $\Omega \setminus Z$. Quadratic forms with moment conditions will be the same for k and $\tilde{k}$.*

The native space of a cpsd kernel of order m coincides as a space of functions with $\tilde{\mathcal{H}} := \mathcal{P}^d_{m-1} + \tilde{\mathcal{H}}$, where $\tilde{\mathcal{H}}$ is the native space for the reduced kernel $\tilde{k}$.

1.3. Kernel methods for prediction

We will now stray away from pure interpolation. So far we assumed data in the form of $f(x) = y$, but with data stemming from measurements, we will have errors of the form

$f(x) = y + \varepsilon$. If our interpolation matrix was ill-conditioned, this would pose a problem. Due to this, we want to allow some margin of error. Or we might consider a classification problem, where the labels are ± 1, or $\{0, 1\}$, but we still want to compute a continuous function suitable for prediction of the class label, which for example gives the probability of belonging to a class. There interpolation does not work.

So, one might want to value errors differently than just using the plain L_2-error. Therefore, we define

Definition 1.54. Let (Ω, Σ) be a measurable space and $Y \subset \mathbb{R}$ be a closed subset. Denote by $(x, y, f(x)) \in \Omega \times Y \times \mathbb{R}$ the triplet consisting of a pattern x, an observation y, and a prediction $f(x)$. A function $\ell : \Omega \times Y \times \mathbb{R} \to [0, \infty)$ is called a *loss function* if it is measurable and $\ell(x, y, y) = 0$ holds for all $x \in \Omega$, $y \in Y$.

We consider at first real values y and we want $y - f(x)$ to be small. The most popular choice for a loss function is the squared loss:

$$\ell_2(x, y, f(x)) = (f(x) - y)^2 = \tilde{\ell}(\underbrace{f(x) - y}_{\zeta}).$$

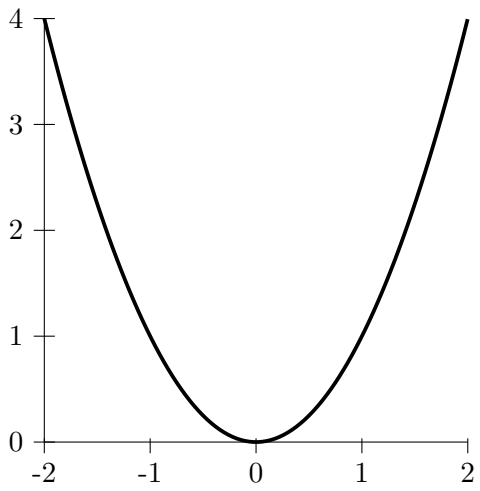


Figure 1.7.: Squared loss l_2

Furthermore, one can use the ℓ_1-loss:

$$\tilde{\ell}(\zeta) = |\zeta|.$$

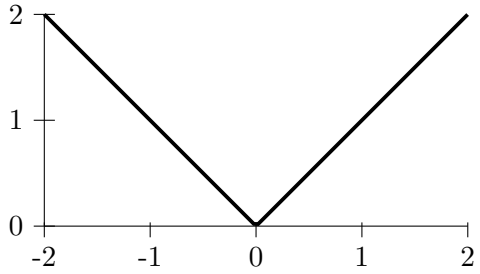


Figure 1.8.: ℓ_1-loss

While the squared loss ℓ_2 relates to the mean, the ℓ_1-loss relates to the median of values y.

For robust estimation the so called Huber's loss can also be useful, it penalizes larger errors only linear and is more robust to outliers.

$$\tilde{\ell_H}(\zeta) = \begin{cases} \frac{1}{2}(\zeta)^2 & \text{if } |\zeta| \leq \sigma \\ \sigma|\zeta| - \frac{1}{2}\sigma^2 & \text{otherwise} \end{cases}.$$

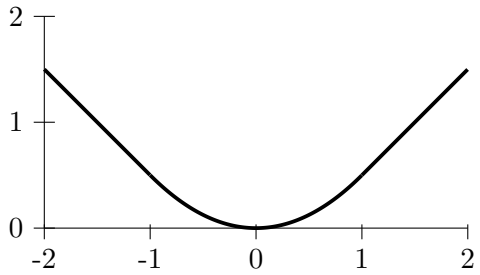


Figure 1.9.: Huber's loss ℓ_H for $\sigma = 1$.

Another possibility is to use the ε-sensitive loss. Here one tolerates errors inside the interval $[\zeta - \varepsilon, \zeta - \varepsilon]$. Formally:

$$\tilde{\ell_\varepsilon}(\zeta) = \max(|\zeta| - \varepsilon, 0) =: |\zeta|_\varepsilon.$$

Let us now consider $Y = \{-1, 1\}$ or $Y = \{0, 1\}$, i.e. the classification case.

The simplest loss is the misclassification error:

$$\ell(x, y, f(x)) = \begin{cases} 0 & \text{if } y = f(x) \\ 1 & \text{otherwise} \end{cases}$$

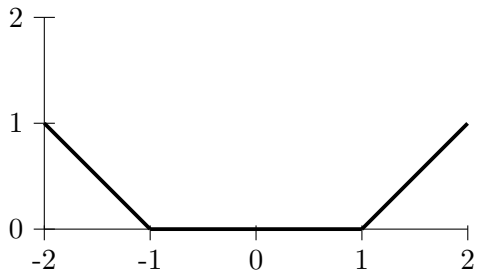


Figure 1.10.: ε-sensitive loss ℓ_ε for $\varepsilon = 1$.

or for $Y = \{-1, 1\}$ we might only care about the sign of the function for prediction

$$\ell(x, y, f(x)) = \begin{cases} 0 & \text{if } y = \operatorname{sgn} f(x) \\ 1 & \text{otherwise} \end{cases}.$$

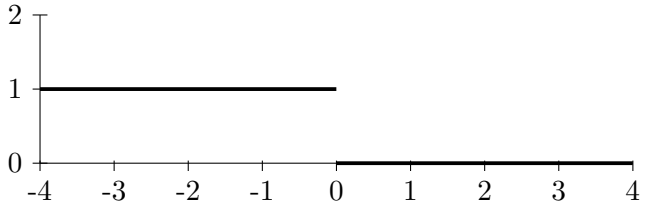


Figure 1.11.: Binary loss over $y \cdot f(x)$.

A plausible choice would be the weighted misclassification loss, in case the different classes have different importance, e.g. in health applications the difference between false positive and false negative predictions,

$$\ell(x, y, f(x)) = \begin{cases} 0 & \text{if } y = \operatorname{sgn} f(x) \\ \tilde{\ell}(y) & \text{otherwise} \end{cases}.$$

There's also the soft margin loss:

$$\ell(x, y, f(x)) = \max(1 - yf(x), 0) = \begin{cases} 0 & \text{if } yf(x) \geq 1 \\ 1 - yf(x) & \text{otherwise} \end{cases}.$$

This loss is used for support vector machines, which is one of the most common classification procedure.

Finally, the logistic loss is often used, it "assigns" a probabilistic meaning to $f(x)$:

$$\ell(x, y, f(x)) = \ell_1 \left(1 + \exp\left(-yf(x)\right)\right).$$

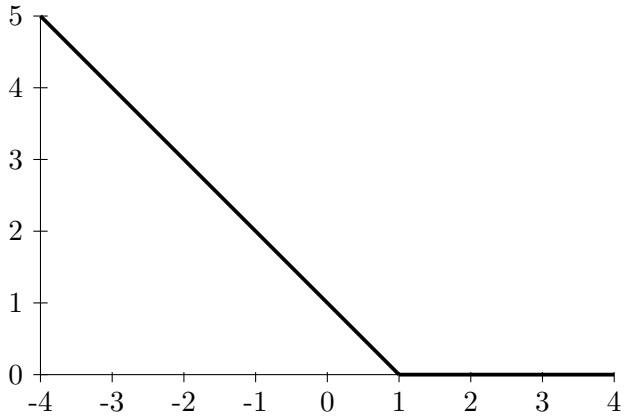


Figure 1.12.: soft margin loss plotted for an x-axis of $yf(x)$

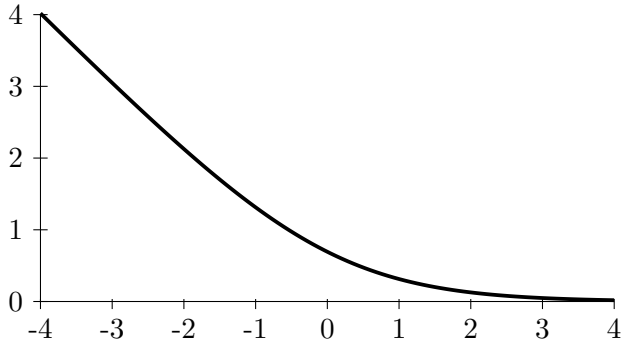


Figure 1.13.: logistic loss

Definition 1.55. Let ℓ be a loss function and $\mathcal{P}$ be a probability measure on $\Omega \times Y$. Then, for a measurable function $f : \Omega \to \mathbb{R}$, the *expected ℓ-risk* is defined by

$$\begin{aligned}\mathcal{R}_{\ell,\mathcal{P}}(f) &\coloneqq \int_{\Omega\times Y} \ell(x, y, f(x)) \,\mathrm{d}\mathcal{P}(x, y) \\ &= \int_{\Omega}\int_{Y} \ell(x, y, f(x)) \,\mathrm{d}\mathcal{P}\left(y \mid x\right) \,\mathrm{d}\mathcal{P}_x(x).\end{aligned}$$

Of course, we do not know $\mathcal{P}(x, y)$, otherwise we could determine all we need from it, we only have (training) data.

For given $D \coloneqq \{(x_i, y_i)\}_{i=1}^N$, $x_i \in \Omega$, $y_i \in Y$, we can define the empirical measure

$$\mathcal{P}_{\mathrm{emp}}(x, y) = \frac{1}{N}\sum_{i=1}^{N} \delta_{x_i,y_i}$$

by using the Dirac measure.

Definition 1.56. The *empirical ℓ-risk* of a function $f : \Omega \to \mathbb{R}$ is defined as

$$\begin{aligned}\mathcal{R}_{\ell,\mathrm{emp}}(f) &= \int_{\Omega\times Y} \ell\left(x, y, f(x)\right) \mathrm{d}\mathcal{P}_{\mathrm{emp}}(x, y) \\ &= \frac{1}{N}\sum_{i=1}^{N} \ell\left(x_i, y_i, f(x_i)\right).\end{aligned}$$

This quantity we can compute for given data, and therefore minimize it. Furthermore, recalling the law of large numbers, if the data is independent, identically distributed (i.i.d.) sampled from $\mathcal{P}$ the risk $\mathcal{R}_{\ell,\mathrm{emp}}$ will be close to $\mathcal{R}_{\ell,\mathcal{P}}$ with high probability.

But just minimizing the empirical risk can lead to numerical instabilities and lead to bad generalization performace, i.e. predictions on unseen data, which are not useful. For inverse problems one uses *Tikhonov-regularization* to restrict the class of admissible functions, this approach will be used here as well. In particular, we will use function space regularization, where one penalizes with a suitable function norm. A relevant alternative is to penalize on the coefficients α if given f as

$$f = \sum_{i=1}^{N} \alpha_j K(x_j, \cdot).$$

For example to obtain sparsity one can use $\|\alpha\|_1$, for such a sparsity approach there is a wide reach of literature available. Note for example, that in the "pure" support vector machines setting, one will expect that only a "small" number of the coefficients α are nonzero.

In the following, we will assume that $\mathcal{R}_{\ell,\text{emp}}(f)$ is continuous in f. The regularization can be seen as restricting the class of possible minimizers to some compact set $\mathcal{H}$. For a continuous function on a compact set $\mathcal{H}$, we can apply the operator inversion lemma to get that the inverse map from the minimum of $\mathcal{R}_{\ell,\text{emp}} : \mathcal{H} \to \mathbb{R}$ to its minimizer $\hat{f}$ is continuous and the optimization problem is well-posed.

Directly minimizing $\mathcal{R}_{\ell,\text{emp}}$ in $\mathcal{H}$ is typically a difficult constrained optimization problem. Instead we add a regularization term to the empirical loss:

$$\mathcal{R}_{\ell,\text{reg}}(f) \coloneqq \mathcal{R}_{\ell,\text{emp}}(f) + \lambda \tilde{s}(f).$$

Here, the λ is the regularization parameter, which balances the empirical error and the smoothness or simplicity enforced by the regularization term $\tilde{s}(f)$.

Theorem 1.57 (Representer Theorem). *Let $s : [0, \infty) \to \mathbb{R}$ be a strictly monotone increasing function, $\lambda > 0$, Ω be a set, $\mathcal{H}$ a RKHS over Ω, and let $\ell : \Omega \times Y \times \mathbb{R}$ be a loss function. Then, for the data $D \coloneqq \{(x_i, y_i)\}_{i=1}^N$, $x_i \in \Omega$, $y_i \in Y$, each minimizer $f \in \mathcal{H}$ of the regularized empirical risk*

$$\mathcal{R}_{\ell,\text{reg}}(f) = \frac{1}{N} \sum_{i=1}^{N} \ell(x_i, y_i, f(x_i)) + \lambda s\left(\|f\|_{\mathcal{H}}\right) \tag{31}$$

admits a representation

$$f(x) = \sum_{i=1}^{N} \alpha_i K(x_i, x),$$

or $f \in \mathcal{H}_X$, $X = \{x_1, \ldots, x_N\}$.

In other words, although we are solving a minimization problem in an infinite dimensional space $\mathcal{H}$ for a finite number of samples, the solution lies in the span of the N kernel functions. For separable classification problems, many of the coefficients α_i are even zero.

Proof. Without loss of generality, we assume

$$s\left(\|f\|_{\mathcal{H}}\right) = \bar{s}\left(\|f\|_{\mathcal{H}}^2\right).$$

We decompose any $f \in \mathcal{H}$ into $f_X \in \mathcal{H}_X$ and $f_{X^\perp} \in \mathcal{H}_X^\perp$ after Theorem 1.20. Then,

$$f = \sum_{i=1}^{N} \alpha_i K(x_i, \cdot) + f_{X^\perp}, \quad \alpha_i \in \mathbb{R}.$$

We know

$$\langle f_{X^\perp}, K(x_i, \cdot)\rangle_{\mathcal{H}} = 0 \quad \text{for all } i = 1, \ldots, N$$

with Eq. (4) this yields:

$$\begin{aligned} f(x_j) = \langle f(\cdot), K(x_j, \cdot)\rangle &= \sum_{i=1}^{N} \alpha_i K(x_i, x_j) + \langle f_{X^\perp}, K(x_j, \cdot)\rangle_{\mathcal{H}} \\ &= \sum_{i=1}^{N} \alpha_i K(x_i, x_j). \end{aligned}$$

Furthermore, for all $f_{X^\perp}$,

$$\begin{aligned} s(\|f\|_{\mathcal{H}}) &= \bar{s}\left(\|f_X\|_{\mathcal{H}}^2 + \|f_{X^\perp}\|_{\mathcal{H}}^2\right) \\ &\geq \bar{s}\left(\|f_X\|_{\mathcal{H}}^2\right) \end{aligned}$$

Therefore, Eq. (31) is for any fixed $\alpha \in \mathbb{R}^N$ minimal if $f_{X^\perp} = 0$. This also has to hold for minimizing f. ∎

Remark. If both the loss function and the regularization s are convex, one has a unique minimum.

Remark. One can build knowledge into this setup by considering

$$\bar{f} = f + h, \quad f \in \mathcal{H}, h \in \operatorname{span}\{\psi_p\}$$

where ψ_p can be specific functions having interpretability. Then one obtains a semiparametric representation

$$\bar{f}(x) = \sum_{i=1}^{N} \alpha_i K(x, x_i) + \sum_{p=1}^{M} \beta_p \psi_p(x).$$

We have for cpsd kernels used the concept of unisolvency of order m in view of polynomials, this can be futher generalized beyond polynomials for any finite-dimensional space of functions and an unisolvency in view of this function space.

One numerical approach to be derived from this is regularized least squares regression, also known as ridge regression. We consider

$$\mathcal{R}_{\ell_2,\mathrm{reg}}(f) = \frac{1}{N}\sum_{i=1}^{N}(f(x_i) - y_i)^2 + \frac{\lambda}{2}\|f\|_{\mathcal{H}}^2, \tag{32}$$

where

$$f(x) = \sum_{j=1}^{N} \alpha_j k(x_j, \cdot).$$

Inserting gives

$$\frac{1}{N}\sum_{j=1}^{N}\left(\sum_{k=1}^{N}\alpha_j k(x_k,x_j)-y_j\right)^2+\frac{\lambda}{2}\sum_{j,k=1}^{N}\alpha_j\alpha_k k(x_j,x_k).$$

Derivation with respect to α_k results in

$$\frac{2}{N}\sum_{j=1}^{N}\left(\sum_{k=1}^{N}\alpha_j k(x_k,x_j)-y_j\right)k(x_k,x_j)+\frac{\lambda}{2}\sum_{j=1}^{N}\alpha_j k(x_j,x_k).$$

Altogether for all α_k

$$\begin{aligned}&\frac{2}{N}K(K\alpha-Y)+\frac{\lambda}{2}K\alpha=0\\ \Rightarrow\quad &K(K+\lambda N I_d)\alpha=KY\\ \Rightarrow\quad &(K+\lambda N I_d)\alpha=Y\end{aligned}$$

Remark. We see how we numerically regularize the kernel matrix by the addition of the scaled identity, which improves of the condition of the kernel matrix. The stronger the condition "improves", the "worse" the results on the given data.

Remark. We will see that in a stochastic view, while modelling

$$y=f(x)+\epsilon,$$

with ϵ i.i.d. samples of Gaussian noise with variance σ^2, one gets to

$$(K+\sigma^2 I_d)\alpha=Y,$$

with K from a covariance function k. In other words, the regularization parameter can be connected to the noise level of the data.

There are several numerical approaches to solve $(K+\lambda N I_d)\alpha=Y$. We consider not too many data points, so that K can be stored in memory. One can now do an eigenvalue decomposition (EVD) or a Cholesky-decomposition of K, with a bit of care in view of its condition, or of $K+\lambda N I_d$ for any λ and use the decomposition to solve the linear equation system.

For example use the eigenvalue decomposition

$$K=V\Gamma V^T,$$

with an orthonormal matrix V and $\Gamma=\operatorname{diag}(x_i)_{i=1}^N$, where x_i are the eigenvalues of K. Then:

$$\begin{aligned}G\coloneqq(\lambda I+K)^{-1}&=(\lambda I+V\Gamma V^T)^{-1}\\ &=V\left(\lambda I+\Gamma\right)^{-1}V^T.\end{aligned}$$

Having once performed the costly eigenvalue decomposition, we can solve for several λ's using this equation:

$$\alpha^\lambda = V\left(\lambda I + \Gamma\right)^{-1} V^T y.$$

or the Cholesky-decomposition

$$LL^T = \lambda I_d + K,$$

with L a lower triangular matrix, and compute

$$Lz = y$$

followed by

$$L^T \alpha = z.$$

Remark. For larger datasets there are approximations needed, e.g. a pivoted Cholesky-decomposition that stops early, a hierarchical matrix decomposition, in particular in lower dimension, or other decomposition also exploiting parallelism.

Another approach is to represent the function on a subset of data, often called inducing points, but still minimize on all.

With all these approximations, iterative solver such as (conjugate or stochastic) gradient are often used.

Another view comes by looking at the regularization differently, which will give us also an observation on what the scalar product $\langle \cdot, \cdot \rangle_{\mathcal{H}}$ does. As a reminder, on $L_2(\Omega)$ we have

$$\langle f, g \rangle_{L_2(\Omega)} = \int_\Omega fg \, \mathrm{d}x.$$

We are aiming for

$$\langle f, g \rangle_{\mathcal{H}} = \langle Sf, Sg \rangle_{L_2} = \int_\Omega Sf(x) Sg(x) \, \mathrm{d}x.$$

The transformation S picks those parts of f that should be regularized. Now think of S as extracting derivations and we see that it is S for smoothness.

Definition 1.58. A *regularization operator* S is defined as a linear map from the space of functions $\{f \mid f : \Omega \to \mathbb{R}\}$ into a space $\mathcal{D}$ equipped with a scalar product. The regularization term $s(f)$ takes the form

$$s(f) \coloneqq \langle Sf, Sf \rangle_{\mathcal{D}}$$

or sometimes

$$s(f) \coloneqq \frac{1}{2} \langle Sf, Sf \rangle_{\mathcal{D}}.$$

Remark. Since we can always define $\tilde{S} \coloneqq (S^\star S)^{\frac{1}{2}}$ and

$$\langle f, S^\star S f \rangle_{\mathcal{L}} = \langle Sf, Sf \rangle_{\mathcal{D}},$$

we can assume S is a positive semidefinite regularization operator.

To better understand the relation between regularization and kernels, we need to get back to Mercer's theorem in the general form of Theorem 1.16.

We go over to weighted inner products with a positive weight function σ:

$$\langle f, g \rangle = \int_\Omega f(x) g(x) \sigma(x) \, \mathrm{d}x.$$

The eigenvalue problem of the integral operator T_K is

$$\int_\Omega k(x, z) \phi(x) \sigma(x) \, \mathrm{d}x = \langle k(x, z), \phi(x) \rangle = \lambda \phi(z).$$

The represents a homogeneous Fredholm integral equation of the 2nd kind, which is not obvious to solve for λ, ϕ. An idea is now to go from a "difficult" integral equation to an "easier" differential equation which should be easier to solve. Here we use, that Green's function play a central role in the analytic solution view of differential equations.

Definition 1.59 (Green's Kernel). Given a linear (ordinary of partial) differential operator $\mathcal{L}$ on $\Omega \subset \mathbb{R}^d$, the Greens kernel g of $\mathcal{L}$ is defined as the solution of

$$\mathcal{L} g(x, z) = \delta(x - z), \quad z \in \Omega \text{ fixed.}$$

The Green's kernel is not uniquely defined this way, one needs to add linear homogeneous boundary conditions or decay conditions, e.g.

$$g(x, z)|_{x \in \partial\Omega} = 0$$

or

$$\lim_{\|x\| \to \inf} = 0.$$

Solutions to

$$(Lu)(x) = f(x) \quad \text{on } \Omega \subset \mathbb{R}^d$$

with a linear and elliptic operator L and some appropriate conditions can be written with Green's kernel g

$$u(x) = \int_\Omega f(z) g(x, z) \, \mathrm{d}z,$$

where for g it holds

$$(Lg)(x,z) = \delta(x-z).$$

We can regard the integral operator

$$Gf(x) = \int_\Omega g(x,z)f(z)dz$$

as the inverse of the differential operator $\mathcal{L}$, i.e.

$$\mathcal{L}u = f \Leftrightarrow u = Gf$$

Example (Brownian bridge kernel). Let $\Omega = [0,1]$ and

$$\begin{aligned} g(x,z) &= \min(x,z) - xz \\ &= \begin{cases} x(1-z) & x \le z \\ z(1-x) & x \ge z \end{cases}. \end{aligned}$$

This kernel may be obtained by observing properties of the Green's function for

$$-\frac{\mathrm{d}^2}{\mathrm{d}x^2} g(x,z) = \delta(x-z)$$

with boundary conditions

$$g(0,z) = g(1,z) = 0.$$

Remark. Whenever $\mathcal{L}$ is a self-adjoint differential operator, the corresponding Green's kernel g is symmetric and the integral operator G is self-adjoint.

Theorem 1.60. *For every RKHS $\mathcal{H}$ with reproducing kernel K, there exists a corresponding regularization operator $S : \mathcal{H} \to \mathcal{D}$, such that for all $f \in \mathcal{H}$,*

$$f(x) = \langle Sk(x,\cdot), Sf(\cdot)\rangle_{\mathcal{D}} \tag{33}$$

and in particular

$$\langle Sk(x,\cdot), Sk(z,\cdot)\rangle_{\mathcal{D}} = k(x,z).$$

Likewise, for every regularization operator $S : \mathcal{F} \to \mathcal{D}$, where $\mathcal{F}$ is some function space equipped with a scalar product and with a corresponding Green's function for $S^\star S$, there exists a corresponding RKHS $\mathcal{H}$, with reproducing kernel k, such that both equations are satisfied.

PROOF. For the first direction, we consider $S = \mathrm{Id}$ and $\mathcal{D} = \mathcal{H}$. This construction fulfills all wanted properties.

Now we start with a function G_x which fulfills

$$f(x) = \langle S^\star S G_x, f\rangle_{\mathcal{L}} \quad \text{for all } f \in S^\star S\mathcal{F}.$$

From functional analysis one knows that such a function exists as the Green's function for the operator $S^\star S$ and natural conditions.

$$\begin{aligned} f(x) &= \langle \underbrace{S^\star S}_{L} G_x, f\rangle_{\mathcal{F}} \\ &= \langle \mathcal{L} G_x, f\rangle \\ &= \langle f, \delta(z-x)\rangle \\ &= f(x). \end{aligned}$$

We have the reproduction property Eq. (33) on the set $S^\star S\mathcal{F}$ using the properties of the adjoint:

$$\langle S^\star S G_x, f\rangle_{\mathcal{F}} = \langle S G_x, Sf\rangle_{\mathcal{D}}.$$

With $f = G_z$ it follows

$$\begin{aligned} G_z(x) &= \langle SG_x, SG_z\rangle_{\mathcal{D}} \\ &= \langle SG_z, SG_x\rangle_{\mathcal{D}} \\ &= G_x(z), \end{aligned}$$

i.e. G is symmetric in this sense and we write

$$k(x,z) = G_z(x),$$

with

$$g(x,z) = \langle SG_x, SG_z\rangle_{\mathcal{D}}$$

we notice that $x \mapsto SG_x$ is actually a feature map. Since kernels arising from feature maps result in kernel matrix that are Gram matrices, we get the K is positive semi-definite. It can be seen that the corresponding RKHS is the closure of

$$\left\{ f \in S^\star S\mathcal{F} \;\middle|\; \|Sf\|_{\mathcal{D}}^2 \leq \infty \right\}.$$

■

$\mathcal{D}$ is a RKHS with inner product $\langle S\cdot, S\cdot\rangle_{\mathcal{D}} = \langle \cdot, \cdot\rangle_{\mathcal{H}}$.

Fixing the regularization operator thereby determines the set of functions one can use for solving the regularized empirical risk problem, neglecting the null space of the regularization operator, if it exists.

We now want to connect

- integral operator eigenvalue problem for Mercer series representation of a positive semidefinite kernel

- related eigenvalue problem for a differential operator

To simplify the situation, we consider the free-space/fullspace Green's function without boundary conditions. We use g as a kernel k, i.e. $(\mathcal{L}k)(x,z) = \delta(x-z)$. Now applying $\mathcal{L}$ to the integral equation gives

$$\begin{aligned} & \underbrace{\mathcal{L}\int_\Omega k(x,z)\phi(x)\sigma(x)\,\mathrm{d}x}_{\int_\Omega \mathcal{L}k(x,z)\phi(x)\sigma(x)\,\mathrm{d}x} = \mathcal{L}\gamma\phi(z) \\ \iff \quad & \underbrace{\int_\Omega \delta(x-z)\phi(x)\,\mathrm{d}x}_{\phi(z)\sigma(z)} = \gamma L\phi(z) \\ \implies \quad & \mathcal{L}\phi(z) = \frac{1}{\gamma}\phi(z)\sigma(z) \end{aligned}$$

where for simplicity we assume that $\mathcal{L}$ has no eigenvalue 0. This shows that $\mathcal{L}$ has eigenvalues which are the reciprocals of the eigenvalues of T_K, while the corresponding eigenfunctions are the same, taking the weight function into account.

Example (revisited). We have

$$\int_\Omega k(x,z)\phi(x)\sigma(x)\,\mathrm{d}x = \gamma\phi(z)$$

with $\sigma \equiv 1$, $k(x,z) = \min(x,z) - xz$ on $\Omega = [0,1]$. This gives for the integral eigenvalue problem

$$\int_0^z x\phi(x)\,\mathrm{d}x + \int_z^1 z\phi(x)\,\mathrm{d}x - \int_0^1 xz\phi(x)\,\mathrm{d}x = \gamma\phi(z).$$

Now apply $\mathcal{L} = -\frac{\mathrm{d}^2}{\mathrm{d}z^2}$ to this and using elementary differentiation steps to obtain

$$\begin{aligned} & \frac{\mathrm{d}}{\mathrm{d}z}\left\{z\phi(z) - \int_1^z \phi(x)\,\mathrm{d}x - z\phi(x) - \int_0^1 x\phi(x)\,\mathrm{d}x\right\} = \gamma\phi''(z) \\ \iff \quad & -\frac{1}{\gamma}\phi(z) = \phi''(z) \end{aligned}$$

Remark. From functional analysis one can derive, that for a Green's kernel that is positive semidefinite, Mercer's theorem applies, namely we obtain a representation by the eigenvalues and eigenfunctions. One example would be Sturm-Liouville differential operators.

Note that not for all differential operators $\mathcal{L}$ a Green's kernel will exist, e.g. if $\mathcal{L}$ has a nontrivial null space of functions which also satisfy the boundary conditions.

Theorem 1.61. *Given a regularization operator S with an expansion of $S^\star S$ into a discrete normalized eigendecomposition with γ_i, ϕ_i, $i = 1, \dots$ and a kernel with*

$$k(x,z) \coloneqq \sum_{i,\gamma_i \neq 0} \frac{d_i}{\gamma_i} \phi_i(x)\phi_i(z),$$

where $d_i \in \{0,1\}$ for all i, and $\sum_{i=1}^{\infty} \frac{d_i}{\gamma_i}$ is convergent, then K satisfies from Theorem 1.60

$$\langle Sk(x,\cdot), Sk(z,\cdot)\rangle_{\mathcal{D}} = k(x,z) = \langle k(x,\cdot), k(z,\cdot)\rangle_{\mathcal{H}}.$$

Moreover, the corresponding RKHS is given by

$$\operatorname{span}\{\phi_i \mid d_i = 1, i \in \mathbb{N}\}.$$

Proof. We use $f = k(z,\cdot)$ in Theorem 1.60.

$$\begin{aligned}
\langle k(x,\cdot), S^\star S k(z,\cdot)\rangle &= \left\langle \sum_i \frac{d_i}{\gamma_i}\phi_i(x)\phi_i(\cdot), S^\star S\left(\sum_i \frac{d_i}{\gamma_i}\phi_i(z)\phi_i(\cdot)\right)\right\rangle \\
&= \sum_{i,j} \frac{d_i}{\gamma_i}\frac{d_j}{\gamma_j}\phi_i(x)\phi_j(z)\langle \phi_i(\cdot), \underbrace{S^\star S\phi_j(\cdot)}_{\gamma_j\phi_j}\rangle \\
&\overset{\text{orthonormal } \phi_i}{=} \sum_i \frac{d_i}{\gamma_i^2}\gamma_i \cdot \phi_i(x)\phi_i(z) \\
&= k(x,z)
\end{aligned}$$

From the construction of the k follows the statement about the span. ∎

Remark. This shows that from a regularization operator several kernels can be obtained, since the d_i can be chosen and one restricts thereby for a subspace of the eigenfunction decomposition. The difference is in the null space of the corresponding T_K.

So a 1-1 correspondence between kernels and regularization operators is only on the image of the integral operator T_k acting on $\mathcal{H}$.

As we have seen, translation invariant kernels are an important class of kernels, $k_x(z) = k(x,z) = k(x-z)$. For show kernels one can more easily find corresponding regularization operators using Fourier transforms, and vice versa.

We consider the Fourier transform of f over $\mathbb{R}^d$

$$F[f](\omega) \coloneqq (2\pi)^{-\frac{d}{2}} \int_{\mathbb{R}^d} f(x)\exp(-i\langle x,\omega\rangle)dx.$$

The inverse Fourier transform is given by

$$F^{-1}[f](\omega) \coloneqq (2\pi)^{-\frac{d}{2}} \int_{\mathbb{R}^d} F[f](\omega)\exp(i\langle x,\omega\rangle)d\omega$$

We now specifically consider regularization operators S, where $S^\star S$ is diagonalizable in the Fourier basis, i.e. the operator can be written as multiplication in Fourier space.

Denote by $\nu(\omega)$ a non-negative, symmetric function on $\mathbb{R}^d$, i.e. $\nu(\omega) = \nu(-\omega) \geq 0$ with $\nu(\omega) \to 0$ for $\|\omega\| \to \infty$. Denote by Ω its support.

We define a regularization operator by

$$\langle Sf, Sg \rangle := (2\pi)^{\frac{d}{2}} \int_\Omega \frac{\overline{F[f](\omega)} F[g](\omega)}{\nu(\omega)} d\omega.$$

In view of $\mathcal{R}_{\ell,\text{reg}}$ this is kept reasonably small. Small values of $\nu(\omega)$ correspond to a strong dumping of the corresponding frequencies. This is desirable for large ω, i.e. high frequency components that correspond to rapid changes in f.

Consider

$$g(x,z) = (2\pi)^{-\frac{d}{2}} \int_{\mathbb{R}^d} \exp(i\langle x - z, \omega \rangle) \nu(\omega) dx.$$

Let f have the support of its Fourier transform contained in Ω. We see

$$\begin{aligned}
\langle Sg(x,\cdot), Sf \rangle &= (2\pi)^{\frac{d}{2}} \int_\Omega \frac{\overline{F[g(x,\cdot)](\omega)} F[f](\omega)}{\nu(\omega)} d\omega \\
&= (2\pi)^{-\frac{d}{2}} \int_\Omega \frac{\overline{\nu(\omega)} \exp(i\langle x, \omega \rangle F[f](\omega)}{\nu(\omega)} d\omega \\
&= (2\pi)^{-\frac{d}{2}} \int_\Omega \exp(i\langle x, \omega \rangle F[f](\omega) d\omega.
\end{aligned}$$

We thereby have the formula from Theorem 1.60.

This is a special case of Bochner's theorem, which states that the Fourier transform of a non-negative Borel measure is positive semi-definite.

As an example consider the Gaussian kernel.

$$k(x,z) = \exp\left(-\frac{\|x - z\|^2}{2\sigma^2}\right).$$

The Fourier transform is given by

$$F[k](\omega) = \nu(\omega) = |\sigma| \exp\left(-\frac{\sigma^2 \omega^2}{2}\right).$$

The wider k is in pattern space, the more peaked its Fourier transform becomes.

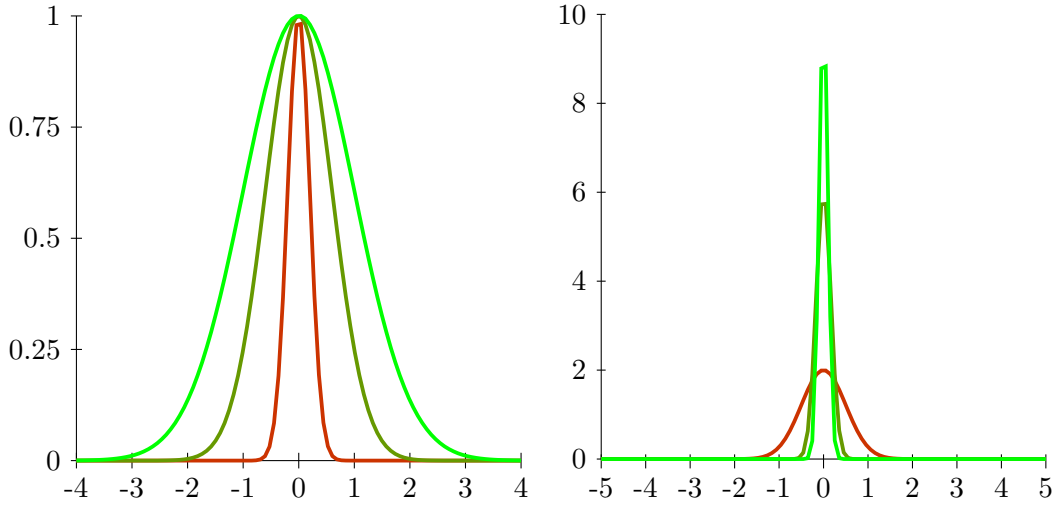


Figure 1.14.: Gaussian kernel and its Fourier transform

The transform tells us that the contribution of high frequency components is relatively small, since $\nu(\omega)$ decays rapidly.

Now consider $\|Sf\|^2$. It can be given in terms of pseudo-differential operators. It can be derived that for the Gaussian kernel it becomes

$$\|Sf\|^2 = \int_{\mathbb{R}^d} \sum_j \frac{\sigma^{2j}}{j! 2^j} (O^j f(x))^2 dx,$$

where $O^{2n} = \Delta^n$ and $O^{2n+1} = \nabla\Delta^n$.

TODO: citation

Model Selection In order to evaluate the performance of a predictive model, in the simplest setting, we split the available data D into training data D_T and the validation data D_V. The empirical risk on D_V is then used to measure the predictive performance (or generalization) performance.

Otherwise, one usually uses k-fold cross-validation (CV). We split D into k disjoint

Figure: Idea of k-fold cross-validation

subsets. One is used as D_V, the other $k-1$ as D_T, solve and evaluate, repeat k times. The average over the k empirical risks is the performance measure. By this procedure, more data is used for training, and all cases appear as validation data. k is typically between 3 and 10, while $k = N$ is called leave-one-out cross-validation.

Definition 1.62. Let $\kappa : \{1, \dots, N\} \mapsto \{1, \dots, k\}$ be an indexing function that indicates the partition to which observation j is allocated by the randomization. Denote by $f_{-k}(x)$ the function that is computed with the k-th part of the data removed. Then the cross-validation estimate of the prediction error is

$$CV(f) = \frac{1}{N} \sum_{j=1}^{N} l(x_j, y_j, f_{-\kappa(j)}(x_j)).$$

In case of hyperparameters, this is done for a set of values, and the model / hyperparameter with the lowest empirical risk is selected. Often this is done by a grid search of the parameters, or in an optimization procedure. For a fair performance statement, the empirical risk on a third, before unseen data set, should be given.

For more on model selection see chapter 7 of [HTF09].

Gaussian Process Regression

Definition 1.63. A *Gaussian process* is a collection of random variables, any finite number of which have a joint Gaussian distribution.

We have for a real process $f(x)$ the mean $m(x)$ as

$$m(x) = \mathbb{E}[f(x)]$$

and the covariance $k(x, z)$ as

$$k(x, z) = \mathbb{E}[(f(x) - m(x))(f(z) - m(z))].$$

A Gaussian process is complexity specified by its mean function and its covariance function:

$$f(x) \sim GP(m(x), k(x, z))$$

Usually one takes the mean function to be zero for notational simplicity.

Our running example for the covariance is the squared exponential / Gaussian kernel

$$\mathrm{Cov}(f(x_p), f(x_q)) = K(x_p, x_q) = \exp\left(-\frac{1}{2w^2}\|x_p - x_q\|^2\right).$$

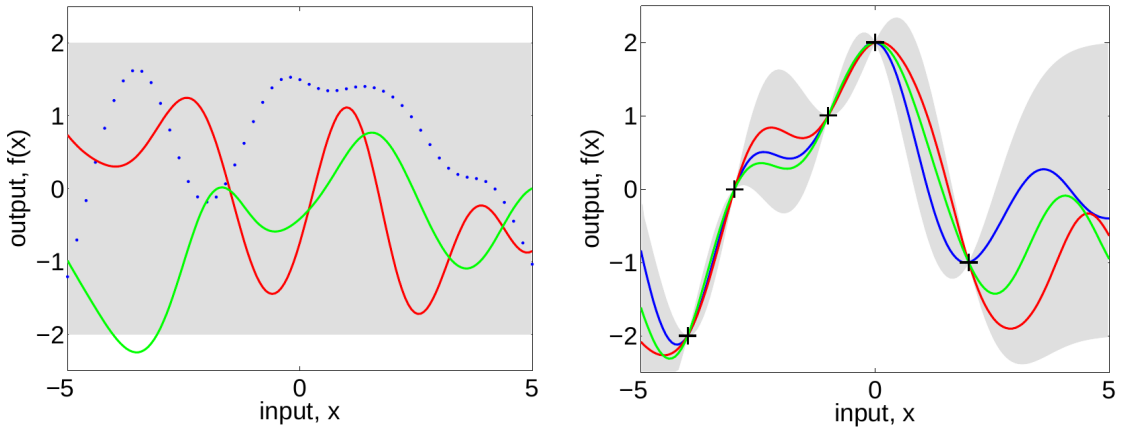


Figure 1.16.: Left: Three functions drawn at random from a GP prior. Right: Three random functions from the posterior, i.e. the prior conditioned on the five noise free observations. In both plots we see the plus and minus two times standard deviation for each input. Image from [RW06]

Note that the covariance of the outputs is written as a function of the inputs. Further, w is the length scale of the process.

The specification of the covariance function implies a distribution over functions, see Fig. 1.16.

Now, we consider noisy data with additive i.i.d. Gaussian noise ε with variance σ_N^2, i.e.

$$y = f(x) + \varepsilon.$$

The prior on the noisy observations becomes

$$\mathrm{Cov}(y_p, y_q) = k(x_p, x_q) + \sigma_N^2 \partial_{pq}$$

or

$$\mathrm{Cov}(Y) = k(X, X) + \sigma_N^2 I,$$

where $Y = [y_1, \ldots, y_N]^T$ and X is a collection of x_i.

The joint distribution of the observed target values y and the function values at the evaluation points f_e under the prior can be seen to be

$$\begin{pmatrix} y \\ f_\star \end{pmatrix} \sim \mathcal{N}\left(0, \begin{pmatrix} K(X,X) + \sigma_N^2 I & K(X, X_e) \\ K(X_e, X) & K(X_e, X_e) \end{pmatrix}\right)$$

To get the posterior over functions, we need to restrict this joint distribution to contain only those functions that "agree" with the observed data. In probabilistic terms, this

conditioning the joint distribution on the observations and we get:

$$f_\star|X,Y,X_\star \sim \mathcal{N}(k(X_\star,X)[k(X,X)+\sigma_N^2 I]^{-1}Y, k(X_\star,X_\star)-k(X_\star,X)[k(X,X)+\sigma_N^2 I]^{-1}k(X,X_\star))$$

Going over to one evaluation x_l we have

$$f(x_l) = k(X,x_l)^T[k(X,X)+\sigma_N^2 I]^{-1}y = \sum_{i=1}^{N}\alpha_i k(x_i,x_l).$$

with

$$k(X,x_l) = (k(x_1,x_l),\ldots,k(x_N,x_e)).$$

And we have

$$\mathrm{Var}(f) = k(x_l,x_l) - k(X,x_l)^T\left[k(X,X)+\sigma_N^2 I\right]^{-1}k(X,x_l).$$

The variance does not (explicitly) depend on the observed targets, but only on the inputs. Since the estimated noise level and e.g. the shape parameters of k depend on the outputs, the predicted variance depends on them at least implicitly.

Consider now the marginal likelihood, or evidence,

$$\mathcal{P}(y \mid X) = \int \mathcal{P}(y \mid f, X)\mathcal{P}(f \mid X)\,\mathrm{d}f,$$

where marginal likelihood refers to the marginalization over the function values f.

Under the Gaussian process view the prior is Gaussian $f \mid X \sim N(0, k(X,X)+\sigma_N^2 I)$, and one can obtain for the log marginal likelihood

$$\log \mathcal{P}(y \mid X) = -\frac{1}{2}Y^T\left(K(X,X)+\sigma_n^2 I\right)^{-1}Y - \frac{1}{2}\log\left|K(X,X)+\sigma_n^2 I\right| - \frac{n}{2}\log 2\pi.$$

In the Gaussian process context one can use Bayes model selection for the hyperparameters. One, roughly speaking, looks at the probability of the data y, given the model and the hyperparameters, expressed by the marginal likelihood:

$$\mathcal{P}(y \mid X, \Theta).$$

Rewriting:

$$\log \mathcal{P}(y \mid X, \Theta) = -\frac{1}{2}y^T K_y^{-1} y - \frac{1}{2}\log|K_y| - \frac{n}{2}\log 2\pi,$$

where $K_y = K + \sigma_n^2 I$ and Θ are the parameters of the covariance function.

The first term

$$-\frac{1}{2}y^T K_y^{-1} y$$

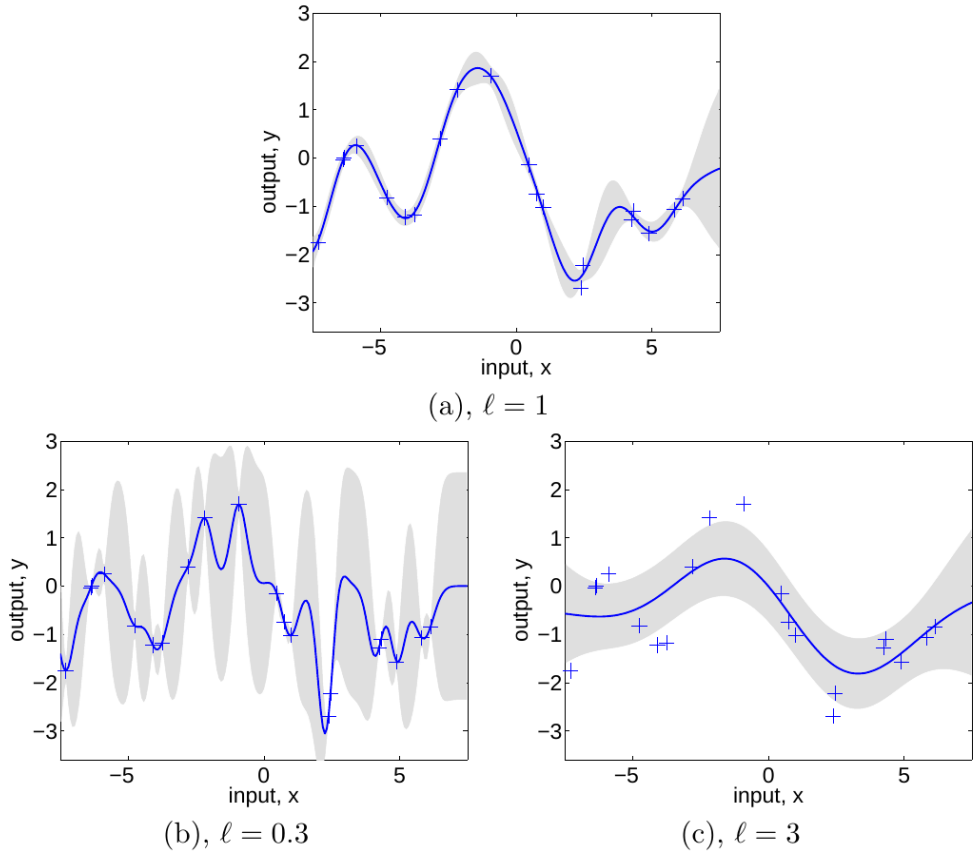


Figure 1.17.: Three GPs with different length scales and noise levels on the same data. Image from [RW06]

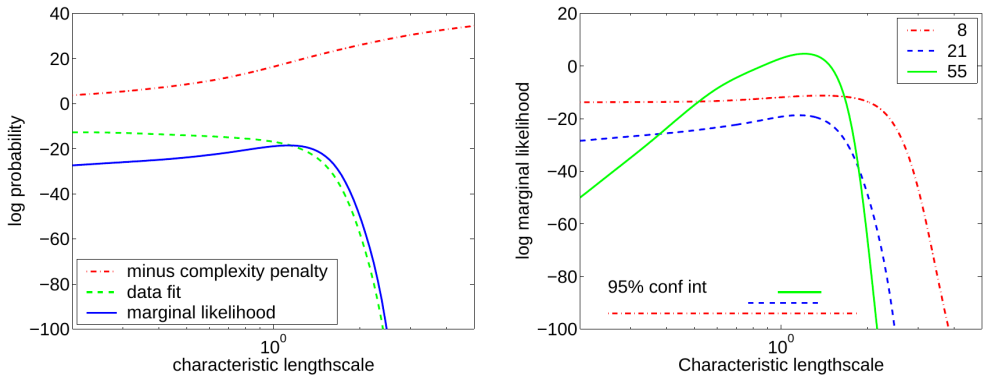


Figure 1.18.: Left: Decomposition of the log marginal likelihood. Right: log marginal likelihood as a function of the lenght-scale for different sizes of the training data. Image from [RW06]

measures the data fit. On the other hand, the second term

$$-\frac{1}{2}\log|K_y|$$

is the complexity penalty depending on the covariance function. The last term,

$$-\frac{n}{2}\log 2\pi$$

is a normalizing constant that can be ignored for optimization purposes. One observes that, see Fig. 1.19,

- the model gets less complex with growing length scale, therefore the negative complexity penalty increases.
- the data fit decreases with the length scale, since the model becomes less and less flexible.
- with more data, the log marginal likelihood gets typically more peaked.

In order to set the hyperparameters one maximizes the log likelihood. The derivative of it can be seen to be

$$\begin{aligned}\frac{\partial}{\partial\Theta_j}\log\mathcal{P}(y\mid X,\Theta) &= \frac{1}{2}y^T K y^{-1}\frac{\partial K_y}{\partial\Theta_j}K_y^{-1}y - \frac{1}{2}\operatorname{tr}\left(K_y^{-1}\frac{\partial K_y}{\partial\Theta_j}\right)\\ &= \frac{1}{2}\operatorname{tr}\left(\left(\alpha\alpha^T - K_y^{-1}\right)\frac{\partial K_y}{\partial\Theta_j}\right)\end{aligned}$$

with $\alpha = K_y^{-1}y$.

For the Gaussian kernel/squared exponential kernel one has two hyperparameters, the noise level σ and the length scale w. Although one sometimes writes the kernel as

$$\sigma_f^2\exp\left(-\frac{1}{2w^2}\|x_q - x_q\|^2\right) + \sigma_N^2\delta_{pq},$$

where one calls

- σ_f^2 the signal variance
- σ_N^2 the noise variance

it is from the optimization view two hyperparameters.

See [RW06] for more on Gaussian process regression.

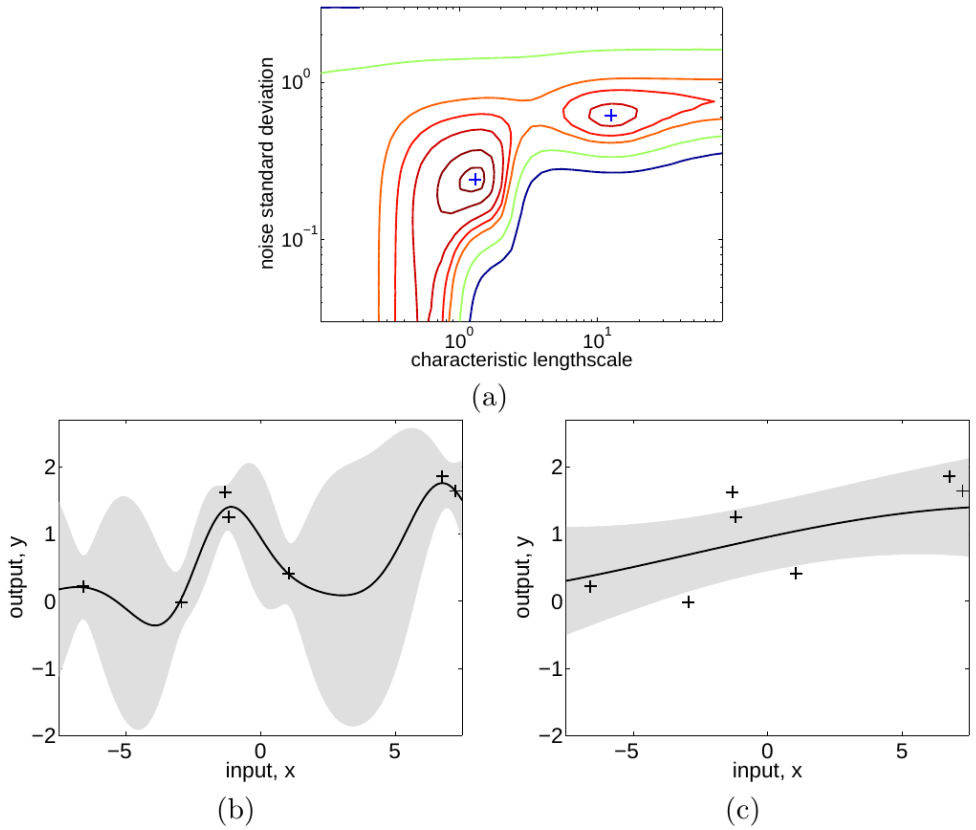


Figure 1.19.: Marginal likelihood as a function of length scale and noise level, and the two local optima, where the global optimum has low noise and a short length-scale. Image from [RW06]

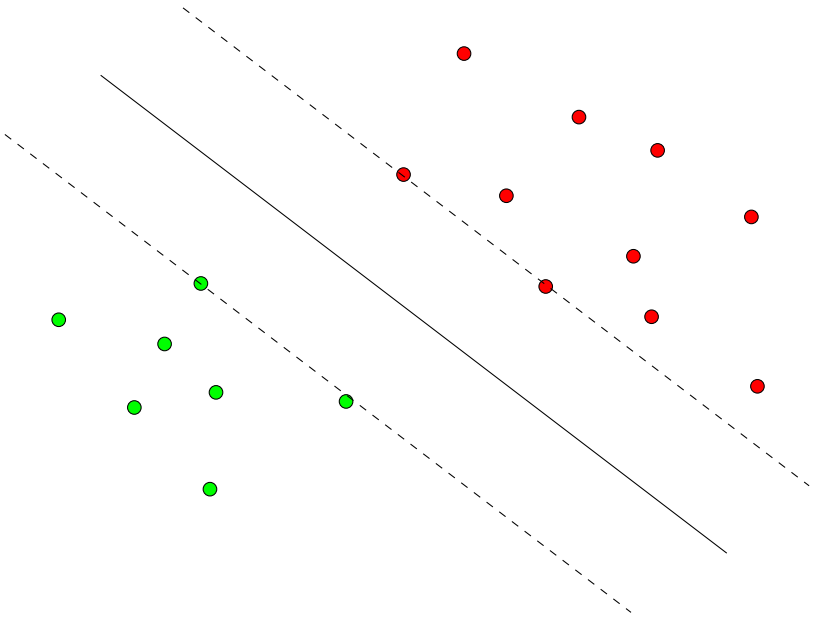
Support Vector Machine

We now consider classification, i.e. $y \in \{-1, 1\}$. We already saw use the hinge loss $\ell_n(y, f(x)) = \max\{0, 1 - yf(x)\}$, which coupled with regularization gives the support vector machine. This learning algorithm has a geometric interpretation, which we will now study.

As before, let $D = \{(x_i, y_i)\}_{i=1,\dots,N}$ with $x_i \in \mathbb{R}^d$. For a so called maximal margin approach one assumes that $w \in \mathbb{R}^d$ with $\|w\|_2 = 1$ and $b \in \mathbb{R}$ exist such that

$$\begin{aligned} \langle w, x_i \rangle + b > 0 \qquad & \text{for all } i \text{ with } y_i = 1 \\ \langle w, x_i \rangle + b < 0 \qquad & \text{for all } i \text{ with } y_i = -1. \end{aligned}$$

The hyperplane defined by (w, b) perfectly separates the training data D. Under all separating hyperplanes for the data one looks for the one with a maximal margin, i.e. maximal distance to the points in D, denoted by (w_D, b_D).



One then sets

$$f_D(x) \coloneqq \operatorname{sgn}(\langle w_D, x \rangle + b_D).$$

as the classifier. Since the sign does not change, one can easily scale, i.e. $(\kappa w_D, \kappa b_D)$, $\kappa > 0$ gives the same sign. Therefore one can also look for $w^\star \in \mathbb{R}^d$ which observes a lower bound on the margin and has minimal norm:

$$\begin{aligned} & \min_{w \in \mathbb{R}^d, b \in \mathbb{R}} \langle w, w \rangle \\ \text{subject to} \qquad & y_i(\langle w, x_i \rangle + b) \geq 1, \ i = 1, \dots, d \end{aligned}$$

One can easily see that

$$w_D = \frac{w^\star}{\|w^\star\|_2} \quad \text{and} \quad b_D = \frac{b^\star}{\|w^\star\|_2}.$$

Note that if we have two support vectors x_1, x_2 with

$$\begin{aligned} \langle w^\star, x_1 \rangle + b^\star &= 1 \\ \langle w^\star, x_2 \rangle + b^\star &= -1, \end{aligned}$$

i.e. points directly on the margin, this gives us

$$\left\langle \frac{w^\star}{\|w^\star\|_2}, (x_1 - x_2) \right\rangle = \frac{2}{\|w^\star\|_2}.$$

Consequentially, the maximal margin hyperplane is completely determined by those x_i which are on the margin. Such are called support vectors, hence the name support vector machine for this construction.

There are two obvious shortcomings:

1. a linear separation may not be possible or suitable for the data set.

2. in the presence of noise, we may need to misclassify some training points to avoid over-fitting.

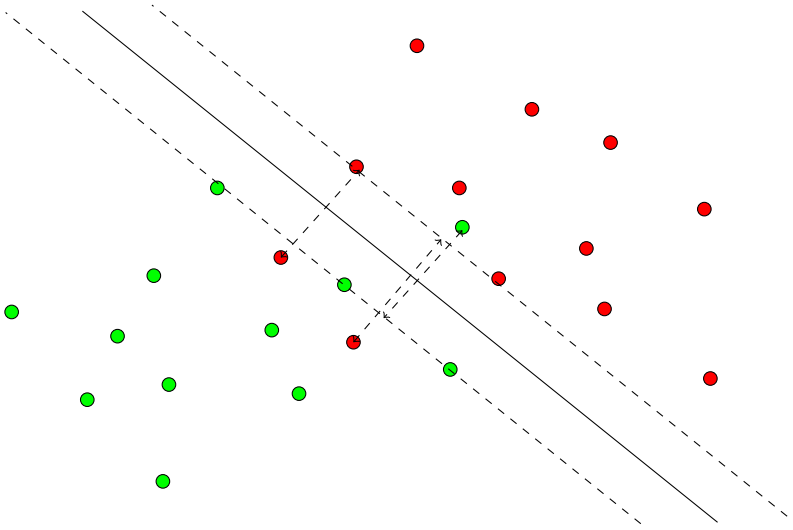
In order to address the first issue, we map x_i into a Hilbert space H by a feature map $\Phi : X \to H$.[1] One then aims for a linear separation in the feature space, i.e. consider $\{(\Phi(x_i), y_i)\}_{i=1,\dots,N}$ with the above procedure. This approach is often called the hard margin support vector machine (SVM). It is possible to show that for kernels with certain natural properties and data sets without contradictions, i.e. no (x_i, y_i), (x_j, y_j) with $x_i = x_j$, $y_i \neq y_j$, a perfect separation in the feature space is possible, see e.g. [SC08, Section 4.6] for details. We will later on see that we only require some support vectors on the margin to represent a solution, at least in the separable situation.

The second problem was addressed by the (soft margin) SVM. The idea is to relax the constraints

$$y_i(\langle w, x_i \rangle + b) \geq 1 - \zeta_i$$

by adding slack variables $\zeta_i \geq 0$. If the slack variables would be too large, the constraints would be easily fulfilled, so one penalizes on their sizes.

[1] We do not write $\mathcal{H}$ for the Hilbert space here as there is going to be a small difference between H and $\mathcal{H}$



Together with the feature map, this gives the following optimization problem:

$$\begin{aligned} & \min \frac{1}{2}\langle w, w\rangle + C\sum_{i=1}^{N}\zeta_i, && w \in H, b \in \mathbb{R}, \zeta \in \mathbb{R}^N \\ \text{subject to} \quad & y_i(\langle w, \Phi(x_i)\rangle + b) \geq 1 - \zeta_i, && \zeta_i \geq 0, i = 1, \ldots, N, \end{aligned}$$

where C is a hyperparameter balancing the two terms. Observe that the objective is convex, while the side constraints are linear. In a geometric sense, the slack parameters ζ_i control how far a value may be "on the wrong side of the margin".

Now we will connect this optimization problem to the regularized loss function formulation. We observe for the constraint

$$\zeta_i \geq 1 - y_i(\langle w, \Phi(x_i)\rangle + b)$$

with $\zeta_i \geq 0$ this gives

$$\begin{aligned} \zeta_i &\geq \max\{0, 1 - y_i(\langle w, \Phi(x_i)\rangle + b)\} \\ &= \ell_h(y_i, \langle w, \Phi(x_i)\rangle + b). \end{aligned}$$

The objective function is minimal in ζ_i when this inequality becomes an equality.

For $(w, b) \in H \times \mathbb{R}$ consider

$$\begin{aligned} f_{(w,b)} &: X \to \mathbb{R} \\ f_{(w,b)}(x) &= \langle w, \Phi(x)\rangle + b \end{aligned}$$

Multiplying the objective by $2\lambda = \frac{1}{NC}$ gives

$$\min_{(w,b)} \lambda\langle w, w\rangle + \frac{1}{N}\sum_{i=1}^{N}\ell_h\left(y_i, f_{(w,b)}(x_i)\right).$$

For suitable kernels we can write this in the RKHS setting

$$\langle w, w \rangle = \|f\|_{\mathcal{H}},$$

where

$$\|f\|_{\mathcal{H}} = \inf\{\|w\|_H : w \in H \text{ with } f = \langle w, \Phi(x) \rangle\}.$$

Modulo the offset term b the geometrically derived approach is equivalent to the RKHS view, we have

$$\inf_{(f,b) \in \mathcal{H} \times \mathbb{R}} \lambda \|f\|_{\mathcal{H}}^2 + \frac{1}{N} \sum_{i=1}^{N} \ell_h(y_i, f(x_i) + b).$$

The offset b makes a real difference, so in general the decision functions are different. For the linear setting, i.e. the identity map $\mathbb{R}^d \to \mathbb{R}^d$, the offset b has a clear advantage since it treats translated data. For many feature maps, e.g. Gaussian kernel, the offset has neither known theoretical nor empirical advantages. In the next part we do *not* consider b.

From the representer Theorem 1.57, we have

$$f_a(x) = \sum_{j=1}^{N} a_j K(x_j, x),$$

i.e. a representation with N kernel functions. But how can we solve the optimization problem?

We see that

$$\|f_a\|_{\mathcal{H}}^2 = \sum_{i=1}^{N} \sum_{j=1}^{N} a_i a_j K(x_i, x_j) = a^T K a$$

and obtain

$$\min_{a \in \mathbb{R}^N} \frac{1}{N} \sum_{i=1}^{N} \ell_h \Bigg(y_i, \underbrace{\sum_{j=1}^{N} a_j K(x_j, x_i)}_{f_a(x)} \Bigg) + \lambda a^T K a.$$

As seen, this is a finite dimensional convex optimization problem. Since ℓ_h is nonnegative, one can solve instead

$$\begin{aligned} &\min_{a, \zeta \in \mathbb{R}^N} \frac{1}{N} \sum_{i=1}^{N} \zeta_i + \lambda a^T K a \\ \text{subject to} \quad &\zeta_i \geq \ell_h(y_i, f_a(x_i)), \quad i = 1, \ldots, N \end{aligned}$$

By going back to $C = \frac{1}{2N\lambda}$ we obtain the alternative formulation:

$$\begin{aligned} &\min_{a, \zeta \in \mathbb{R}^N} C \sum_{i=1}^{N} \zeta_i + \frac{1}{2} \|f_a\|_{\mathcal{H}}^2 \\ \text{subject to} \quad &\zeta_i \geq 0, \zeta_i \geq 1 - y_i f_a(x_i), \quad i = 1, \ldots, N \end{aligned}$$

Here we can now use results for convex optimization problems with constraints. After some derivations one obtains

$$\max_{\beta} \sum_{i=1}^{N} \beta_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \beta_i y_i \beta_j y_j K(x_i, x_j) \qquad \text{with } \beta \in [0, C]^N.$$

Therefore the problem with the hinge loss is a quadratic optimization problem with box constraints, which is easier to solve than a convex problem with constraints.

See [SS02; SC08] for more on support vector machines.

2. Dimensionality reduction

We consider unsupervised learning, where we have a set Y of data $y_i \in \mathbb{R}^d$, $i = 1, \ldots, N$ without labels. We assume that there is a lower dimensional representation for each y_i by a $x_i \in \mathbb{R}^p$, $p < d$, which describes the data in a suitable fashion. We call d the *extrinsic dimension* and p the *intrinsic dimension*.

Some example data

1. *Handwritten Digits*
   A digit is represented by a $k \times k$ matrix of gray values (average color value of the respective pixel in the grid), i.e. a vector in $\mathbb{R}^{k \cdot k}$, see Fig. 2.1a. A data set is a matrix A in $\mathbb{R}^{k \cdot k \times N}$, where the columns of A are a subspace of $\mathbb{R}^{k \cdot k}$. Now consider only images of the digit 3, we aim for a basis $\left\{u_i^{(3)}\right\}_{i=1}^{p}$ of this subspace. Any new image b of a digit gets represented in this basis and obtain the error

   $$b - \sum_{i=1}^{p} x_i u_i^{(3)}.$$

   If this error is small, one would assume the digit is a 3, otherwise it isn't.

2. *Sensor Arrays*
   Sensor arrays are a set of identical sensors, e.g.

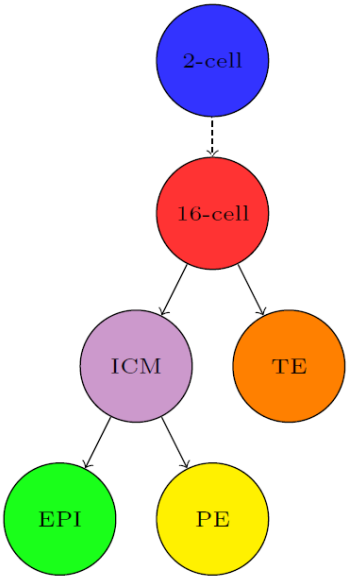
   - sensor antennas in radio telescopes
   - earth measurements in seismography or weather
   - several electrodes measure time series for electrocardiography (ECG) or electroencephalography (EEG)
   - frequency measurements over time

   The matrix is of size #sensors $\times$ #measurements.

3. *Numerical Simulations*
   Numerical simulations are also a data source. In the research & development (R&D)-

process, the engineer performs several simulations with different input parameters [Boh+13; IG19]. The data is of size #size of the simulation × #simulations.

4. *Single Cell Genomics*
   Measuring gene levels in single cells has become feasible in recent years [GKQ16]. For example, one is interested in the differentiation process of a stem cell to a specific cell type. Two stem cells of the same type can differentiate into two different cells, e.g. two different muscle cells.



   Measurements are gene levels for several genes per cells, these are measured over time. Linear and nonlinear dimensionality reduction methods can provide structure in the data [WRY16]. In this application domain, one has to also take care of (structurally) missing data.

For now assume we assume that the data is organised in a matrix.

There are several goals for dimensionality reduction.

- An analysis goal is the extraction of knowledge, e.g. by finding (sub)structures in the data.
- Dimensionality reduction can be used to reduce computational complexity or to compress data for storage reduction.
- For certain tasks, it might be enough to find "important" attributes, or attribute interactions, also called feature engineering.

Curse of Dimensionality When working in high dimensions, one has to consider the so called *curse of dimensionality* which has several forms and aspects.

Consider a grid over $[0,1]^d$ with spacing $\frac{1}{10}$, in 3 dimensions we have 10^3, or in 20 it would give 10^{20} points. Therefore, if one wants to represent a function or optimize over it, using a Cartesian grid, the computation effort scales exponentially in d, the number of dimensions. One runs into the curse of dimensionality in the form of complexity. Note that under certain assumptions, one can (drastically) reduce the complexity, e.g. sparse grids, low rank tensor decompositions, compressed sensing, Quasi-Monte Carlo, and others. Specifics will depend on the balancing approximation properties and computational complexity.

Viewed the other way around, the amount of data is generally restricted, therefore high-dimensional spaces described by data are inherently sparse. This is called the *empty space phenomenon*, which is related to the concentration of measure effect. Therefore it is very reasonable to assume e.g. lower intrinsic dimensionality, or other structure in data.

Some oddities that can be observed in high dimensions are for example:

- diagonals in hypercubes are "orthogonal" to any axis
- the ratio between the volume of a sphere and the volume of an enclosing hypercube strays to zero for $d \to \infty$,

see e.g. [LV07] for more on this.

Properties What properties are relevant for a method analyzing high dimensional data by dimensionality reduction?

1. One wants to estimate the *intrinsic dimensionality*. This can be the number of *latent variables* or the *degrees of freedom*.

2. One wants to embed the data to reduce the dimensionality. An embedding allows a compact representation and makes further processing easier. One typically assumes that the data lies on a p-dimensional manifold and aims for an embedding in a space with dimensionality close to p. It is necessary to characterize and measure the structure of the manifold. Properties could be curvature, connectivity or local relationships. Furthermore, an embedding establishes a bijective mapping between the $y_i \in \mathbb{R}^d$ and their counterparts $x_i \in \mathbb{R}^p$. The mapping might allow an out-of-sample extension, i.e. find for new $y \in \mathbb{R}^d$ an $x \in \mathbb{R}^p$, or vice versa. The way from x to y is also called generative.

3. One wants to embed for latent variable separation. Here, additional constraints are imposed on the desired low dimensional representation. A typical assumption is that the latent variables are (statistically) independent from each other. Most methods for latent variable separation need another method for dimensionality reduction or some preprocessing. Often such additional constraints impose rather simple models.

Missing Image

2.1. Linear Dimensionality Reduction

2.1.1. Principal Components Analysis

The principal component analysis (PCA) method is one of the oldest, well known, and likely best data analysis methods for dimensionality reduction. It was developed (or discovered) several times, mainly to mention are:

- Pearson (1901) in biological applications, further extended by Hotelling (1933) in psychometrics
- In the framework of stochastic processes, it was discovered independently in 1946 by Karhunen, and was later generalized by Loève in 1948, and is known as the Karhunen-Loève transform in this field

It is also known under several other names

- proper orthogonal decomposition (POD) in engineering
- empirical orthogonal functions (EOF) in meteorology
- etc.

From the underlying linear algebra we will see that it is closely related to the singular value decomposition and the Schmidt-Eckart-Young-theorem.

We now assume to have a data set $\{y_i\}_{i=1}^N$ which are samples of a random variable $\mathcal{Y} \in \mathbb{R}^d$. We assume $\mathcal{Y}$ stems from p unknown latent variables $\mathcal{X} \in \mathbb{R}^p$ by a linear transformation W, i.e. $\mathcal{Y} = W\mathcal{X}$. Moreover, we assume that $\mathcal{Y}$ is mean-centered, i.e. $\mathbb{E}(\mathcal{Y}) = 0$. This is no real restriction since otherwise we just subtract the mean from $\mathcal{Y}$. For W we assume it is an axis change, i.e. the columns $w_i \in W$ are orthogonal to each other and have unit

norm, i.e. $W^T W = I_p$. We organize the data in a matrix $Y = [y_1, \ldots, y_N]$. Using the pseudo-inverse of W we have

$$W^\dagger = (W^T W)^{-1} W^T = W^T.$$

We can write $x_i = W^\dagger y_i$. The reconstruction error becomes

$$\mathbb{E}\left(\|\mathcal{Y} - W(W^T\mathcal{Y})\|_2^2\right),$$

where $W^T W = I_p$ per assumption, but $WW^T \neq I_d$ in general.

$$\begin{aligned}\mathbb{E}\left(\|\mathcal{Y} - WW^T\mathcal{Y}\|^2\right) &= \mathbb{E}\Big(\mathcal{Y}^T\mathcal{Y} - 2\mathcal{Y}^T WW^T\mathcal{Y} + \mathcal{Y}^T W \underbrace{W^T W}_{I_p} W^T\mathcal{Y}\Big) \\ &= \mathbb{E}\left(\mathcal{Y}^T\mathcal{Y} - \mathcal{Y}^T WW^T\mathcal{Y}\right)\end{aligned}$$

We approximate the W-dependent part using the samples and compute the empirical mean

$$\begin{aligned}\mathbb{E}\left(\mathcal{Y}^T WW^T\mathcal{Y}\right) &\approx \frac{1}{N}\sum_{i=1}^{N} y_i^T WW^T y_i \\ &= \frac{1}{N}\operatorname{tr}\left(Y^T WW^T Y\right) \\ &\overset{\text{trace is cyclic}}{=} \frac{1}{N}\operatorname{tr}\left(W^T YY^T W\right).\end{aligned}$$

Adding the constraint $W^T W = I_p$ we get the Lagrangian

$$\mathcal{L} = \operatorname{tr}(W^T YY^T W) + \operatorname{tr}((I_p - W^T W)\Lambda),$$

where $\Lambda = \Lambda^T \in \mathbb{R}^{p\times p}$. The condition for an extrema is

$$YY^T W = W\Lambda \implies \Lambda = W^T YY^T W \tag{34}$$

and the objective function reduces to $\operatorname{tr}(\Lambda)$.

We can rotate W and have the same reconstruction error, e.g. we use $W' = WR$ giving $\Lambda' = R\Lambda R^T$. $\Lambda = \Lambda^T$ is diagonalizable with orthogonal matrices, so we can choose R such that Λ' is a diagonal matrix. Without loss of generality, Λ is diagonal. From Eq. (34) it follows that the columns of W must be p eigenvectors of YY^T with the corresponding eigenvalues as the diagonal of Λ. Since we maximize $\operatorname{tr}(\Lambda)$ we get the p largest eigenvalues of YY^T and the corresponding eigenvectors.

One can connect the eigenvectors of YY^T with the top left singular vectors U of the singular value decomposition (SVD) (Definition A.1) of Y, where the eigenvalues are the

squared singular values of Y. So we take the first p columns of U for W, i.e. $W = UI_{d\times p}$. Furthermore, as U is orthonormal

$$X = W^T Y = W^T U \Sigma V^T = I_{p\times d}\Sigma V^T.$$

Using $\Lambda = W^T U \Sigma^2 U W = \Sigma^2$ we obtain for the optimal least squares error

$$\operatorname{tr}\left(\Sigma^2\right) - \operatorname{tr}\left(I_{p\times d}\Sigma^2\right) = \sum_{i=p+1}^{d} \sigma_i^2.$$

We have shown

Theorem 2.1. *Let $Y = [y_1, \ldots, y_N] \in \mathbb{R}^{d\times N}$ be a matrix of zero mean data points. Denote the SVD of Y by $U\Sigma V^T$. Then for given $p < d$ the minimizer W for the reconstruction problem*

$$\min_W \sum_{i=1}^{N} \|y_i - WW^T y_i\|_2^2, \text{ such that } W^T W = I_p \tag{35}$$

is given by $W = [u_1, \ldots, u_p]$. The lower dimensional embedding is given by

$$X = I_{p\times d}\Sigma V^T = I_{p\times d} U^T Y.$$

For the reconstruction error one obtains

$$\sum_{i=p+1}^{d} \sigma^2.$$

A slightly different, but not really different, view on this by a projection, i.e. we aim for

$$y = \sum_{i=1}^{p} x_i w_i, \qquad \text{with } y, w_i \in \mathbb{R}^d, w_i^T w_j = \delta_{ij},$$

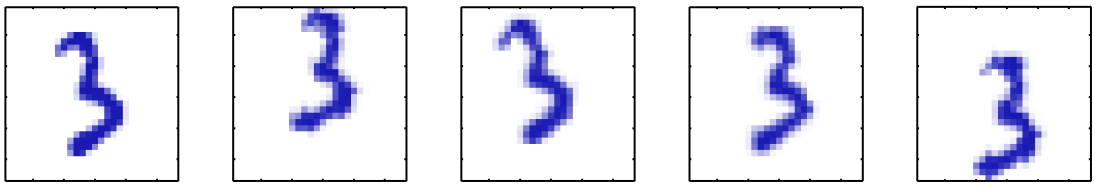
which is a projection into the linear space of dimension p spanned by the w_i. The x_i are determined by

$$\begin{aligned} x_i &= \langle y, w_i\rangle \\ y &= \sum_{i=1}^{p} \langle y, w_i\rangle w_i \end{aligned}$$

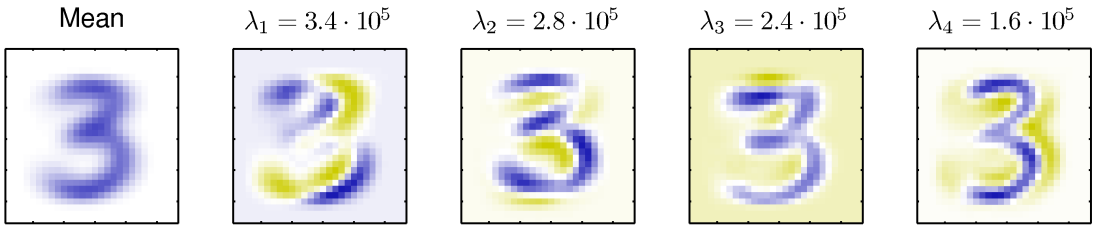
Projection is here looking for the best linear fit with the smallest L_2-error, this results in the same problem as before.

Another view is with approximation with rank constraints, i.e. we look for

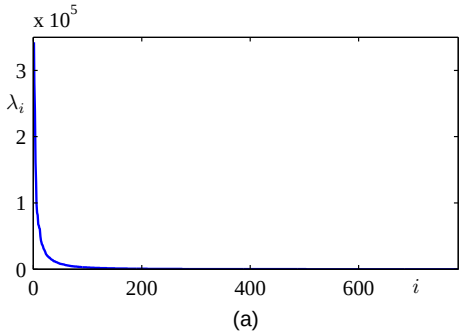
$$\min_A \|Y - A\|_F^2, \text{ such that } \operatorname{rank} A = p.$$



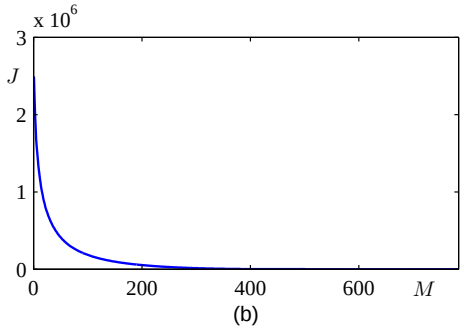
(a) Original image and 4 images obtained by random displacement and rotation.



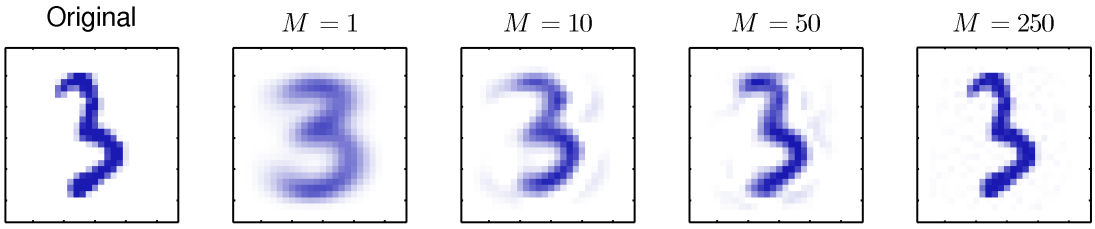
(b) Shown is the mean for digit 3 along with first four PCA components with their eigenvalue.



(c) Plot of eigenvalue spectrum.



(d) Sum of the discarded eigenvalues.



(e) Original image and its reconstruction using different number of PCA components.

Figure 2.1.: PCA applied on images of the digit 3. The synthetic data set is obtained by taking one image of a 3 (size 28×28) and generating more images by using random displacement and rotation. Images taken from [Bis06].

This is the Schmidt-Eckart-Young Theorem A.2 for the SVD, i.e.

$$A = U_p \Sigma_p V_p^T,$$

where the matrices consist of the top p singular vectors/values of Y. In other words, the truncated SVD is the best rank-p approximation of Y under the Frobenius norm.

From the statistical perspective another derivation is common. Let us first define the principal components.

Definition 2.2. Given a zero mean multivariate random variable $Y \in \mathbb{R}^d$. The p *principal components* of Y are defined as the p uncorrelated linear components of y:

$$x_i = w_i^T y \in \mathbb{R}, \quad w_i \in \mathbb{R}^d, i = 1, \ldots, p$$

such that the variance of x_i is maximized subject to $w_i^T w_i = 1$ and $\mathrm{Var}(x_1) \geq \mathrm{Var}(x_2) \geq \ldots \geq \mathrm{Var}(x_p) \geq 0$.

Theorem 2.3. *Assume that the rank of the covariance matrix $\mathbb{E}(YY^T)$ is larger than p. Then the first p principal components of a zero-mean multivariate random variable Y, denoted by x_i, $i = 1, \ldots, p$ are given by*

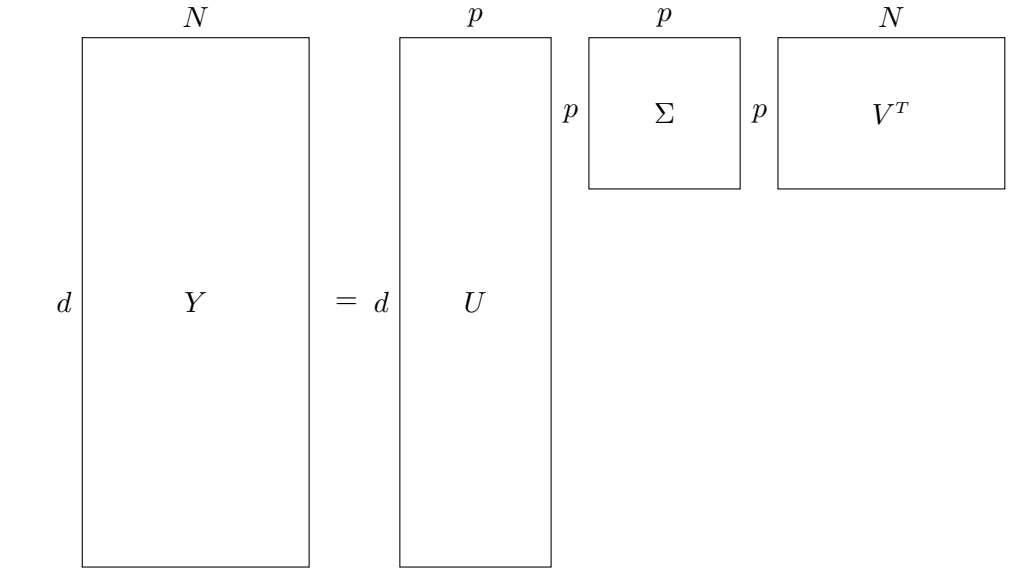
$$x_i = w_i^T y,$$

where $\{w_i\}_{i=1}^p$ are the p orthonormal eigenvectors of the covariance matrix $\mathbb{E}(YY^T)$ associated with its p largest eigenvalues $\{\lambda_i\}_{i=1}^p$. Moreover, $\lambda_i = \mathrm{Var}(x_i)$.

Proof. see exercises ∎

In the proof of Theorem 2.1 we have already seen that the columns of W are the top p eigenvectors of YY^T, so both generate the same result. Fig. 2.2 gives us two representations of Y, where we have

$$Y = \sum_{i=1}^{p} \sigma_i \underbrace{u_i v_i^T}_{\text{outer product}}.$$



or

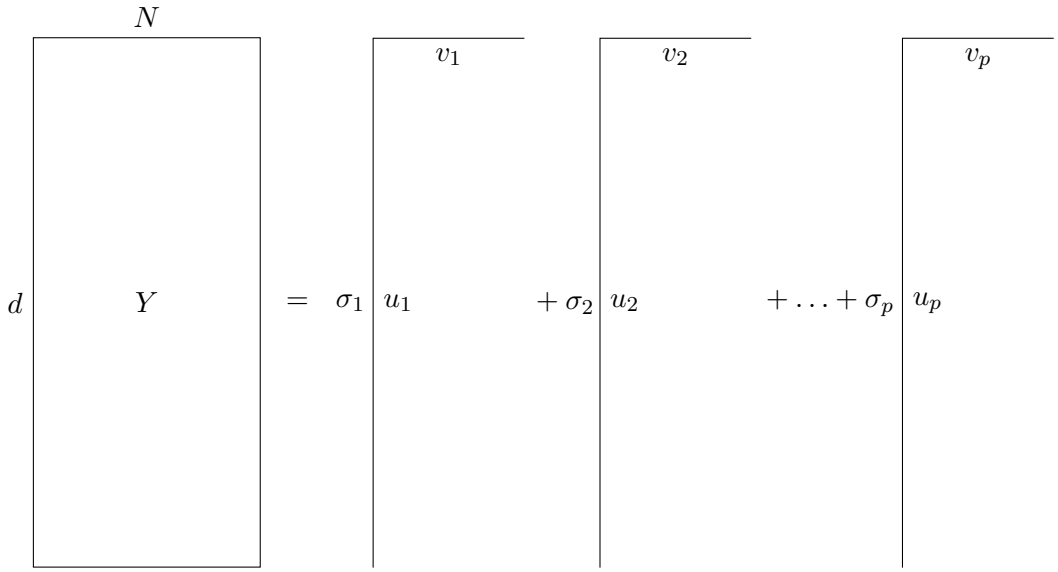


Figure 2.2.: Two representations of Y

In the view of the three properties of dimensionality reduction approaches we have:

1. Ideally we have $\text{rank}(YY^T) = p$, but in practice we have noise. Use the eigenvalue/ singular value decay for the selection of p. One might for example attempt to capture, say, 95% of the variance by
$$\frac{\sum_{i=1}^{p} \lambda_i}{\sum_{i=1}^{d} \lambda_i} \geq 0.95$$
or one uses the threshold on the λ_i, e.g.
$$\lambda_i \leq 0.01\sigma_y^2 = 0.01 \sum_{i=1}^{d} \lambda_i.$$

2. The embedding is just $X = I_{p\times d} U^T Y$.

3. For the latent variables, we do have W is orthonormal, i.e. a rotation and assume a Gaussian distribution over the latent variables.

Note that if x is a Gaussian random variable, the eigenvectors of the empirical estimate of
$$\mathbb{E}(YY^T) = \frac{1}{N} YY^T$$
are asymptotically consistent unbiased estimates for the corresponding eigenvectors of $\mathbb{E}(YY^T)$. See [Jol02] for details. Furthermore, [LV07] discusses this topic.

2.1.2. Multidimensional Scaling

We now consider another criteria for dimensionality reduction, we aim for embeddings which approximately preserve distances:
$$d^d(y_1, y_2) \approx d^p(x_1, x_2).$$
To describe and analyze this approach, we first need some definitions.

Definition 2.4. An $N \times N$ symmetric matrix D is called *Euclidean distance matrix (EDM)* if there exists an integer $d > 0$ and a vector set $Y = \{y_i\}_{i=1}^N$, $y_i \in \mathbb{R}^d$, such that
$$D_{ij} = d_E^2(y_i, y_j),\ i, j = 1, \dots N,$$
where d_E is the Euclidean distance. The vector set Y is called a *configuration* of D. We write $D \in \mathbb{EDM}$.

Later we will use the following generalization of an EDM.

Definition 2.5. An $N \times N$ symmetric matrix D with nonnegative entries d_{ij} is called *distance matrix* if $d_{ii} = 0$ for all $1 \leq i \leq N$ and

$$\sqrt{d_{ij}} \leq \sqrt{d_{ik}} + \sqrt{d_{kj}} \quad \text{for } 1 \leq i, j \leq N.$$

We write $D \in \mathbb{DM}$.

Obviously, $\mathbb{EDM} \subseteq \mathbb{DM}$.

Note the following relation between the Euclidean distance and the scalar product, see also Eq. (5),

$$d_E^2(y_i, y_j) = \underbrace{\langle y_i, y_i \rangle}_{G_{ii}} - 2 \underbrace{\langle y_i, y_j \rangle}_{G_{ij}} + \underbrace{\langle y_j, y_j \rangle}_{G_{jj}}.$$

The other matrix we look at is the Gram matrix G, where

$$G_{ij} = (Y^T Y)_{ij} = \langle y_i, y_j \rangle.$$

Like in PCA we aim for centered data, which we can achieve with the *centering matrix*

$$H = I - \frac{1}{N} \mathbb{1}_N,$$

where $\mathbb{1}_N = 1_N 1_N^T$ is the matrix of all ones. With

$$Y^c = Y - \left(\frac{1}{N} Y 1_N \right) 1_N^T = YH,$$

we get the centered data Y^c since we subtract the mean. For the corresponding *centered Gram matrix* G^c we get

$$G^c = (Y^c)^T Y^c = H^T Y^T Y H = HGH.$$

Theorem 2.6. *For the Euclidean distance matrix D and the centered Gram matrix G^c of a data set Y it holds*

$$G^c = -\frac{1}{2} HDH$$

Proof. straightforward calculation ∎

Lemma 2.7. *Assume that the matrix $D = \left[d_{ij}^2 \right]_{i,j=1}^N \in \mathbb{EDM}$ and let $G^c = -\frac{1}{2} HDH$. If the rank of G^c is r, then there is a r-dimensional centered configuration $Y = \{y_1, \ldots, y_N\} \in \mathbb{R}^r$ such that $d_E(y_i, y_j) = d_{ij}$.*

PROOF. Since $D \in \mathbb{EDM}$ there exists $z = \{z_1, \dots, z_N\} \subset \mathbb{R}^d$ such that $d_{ij}^2 = d_E^2(z_i, z_j)$. G^c is the centered Gram matrix of that data set and therefore positive semidefinite. The rank is r, therefore we have $G^c = Y^T Y$ with a centered $r \times N$ data matrix Y. The centered data satisfies $d_{ij}^2 = d_E^2(y_i, y_j)$. ∎

We call r the intrinsic configuration dimension and the centered configuration Y is the *exact configuration* of D.

classical multidimensional scaling (CMDS) Instead of the exact configuration we now look for lower dimensional data sets $X \subset \mathbb{R}^p$, $p \ll r$, which approximately preserve the distance. Formally,

$$X = \underset{X \in \mathbb{R}^{p \times N}}{\arg\min} \sum_{i=1}^{N} \left| d_{ij}^2 - d_E^2(x_i, x_j) \right|, \tag{36}$$

such that $X = T(Y)$, with T an orthogonal projection from $\mathbb{R}^r$ to a p-dimensional subspace $S_p \subset \mathbb{R}^r$, and Y is an exact configuration.

Lemma 2.8. *Let $Z \subset \mathbb{R}^r$ be a given data set with $D_Z = [d_E^2(z_i, z_j)]_{i,j=1}^N$ and let $G_Z^c = -\frac{1}{2} H D_Z H$. Then*

$$\operatorname{tr}(G_Z^c) = \frac{1}{2N} \sum_{i,j=1}^{N} d_E^2(z_i, z_j).$$

PROOF. Straightforward calculation, i.e. write out $G_Z^c = -\frac{1}{2} H D_Z H$ and look at the diagonal. ∎

Lemma 2.9. *Let $D_Z = [d_E(z_i, z_j)]_{i,j=1}^N$, $ZH =: \hat{Z} = [\hat{z}_1, \dots, \hat{z}_N]$. Then*

$$\|\hat{Z}\|_F = \frac{1}{\sqrt{2N}} \|D_Z\|_F.$$

PROOF. With $\|D_Z\|_F^2 = \sum_{i,j=1}^N d_{ij}^2$ and

$$\|\hat{Z}\|_F^2 = \operatorname{tr}\left(\hat{Z}^T \hat{Z}\right) = \operatorname{tr}(G_Z^c),$$

the result follows with Lemma 2.8. ∎

Theorem 2.10. *Let $Y \subset \mathbb{R}^r$ be the exact configuration of $D \in \mathbb{EDM}$. The SVD of Y is given by $Y = U\Sigma V^T$, $\Sigma = \operatorname{diag}(\sigma_1, \dots, \sigma_r)$. For a given $p \leq r$ let $U_p = [u_1, \dots, u_p]$. Then*

$$X = U_p^T Y$$

is a solution of the CMDS minimization problem Eq. (36) with an error of

$$2N \cdot \sum_{i=p+1}^{r} \sigma_i^2.$$

Proof. Let S_p be a p-dimensional subspace of $\mathbb{R}^r$ and B an $r \times p$ orthogonal matrix, whose columns form an orthonormal basis of S_p. We have $T(y) = BB^T y$ for $y \in \mathbb{R}^r$. We observe using $B^T B = I$ (i.e. $\|Bx\| = \|x\|$) that

$$d_E(Ty_i, Ty_j) = \|T(y_i - y_j)\| = \|B^T(y_i - y_j)\| = d_E(B^T y_i, B^T y_j).$$

With $\|y_i - y_j\| \geq \|T(y_i - y_j)\|$ we get for the objective function

$$\begin{aligned}\sum_{i,j=1}^{N} d_E^2(y_i, y_j) - d_E^2(B^T y_i, B^T y_j) = \sum_{i,j=1}^{N} &\langle y_i - y_j, y_i - y_j\rangle \\ &+ \langle B^T(y_i - y_j), B^T(y_i - y_j)\rangle \\ &- 2\langle BB^T(y_i - y_j), y_i - y_j\rangle.\end{aligned}$$

For this transformation we used $\langle B^T y, B^T y\rangle = \langle BB^T y, y\rangle$ and performed a zero addition.

$$\begin{aligned}&= \sum_{i,j=1}^{N} \|(I - BB^T)(y_i - y_j)\|^2 \\ &= \|D_Z\|_F^2\end{aligned}$$

with $Z = (I - BB^T)Y$.

Now, Lemma 2.9 gives

$$\|D_Z\|_F^2 = 2N\|ZH\|_F^2$$

Since Y is centered, therefore Z is too, and we get

$$= 2N\|Z\|_F^2.$$

Therefore we have to solve

$$\underset{B \in \mathbb{R}^{r\times p} \text{ with } B^T B = I_d}{\arg\min} \|Y - BB^T Y\|_F$$

We have to find a matrix of rank r which minimizes this expression, we can use the Schmidt-Eckart-Young Theorem A.2 for the SVD. The best one is $U_p\Sigma_p V_p^T$ so setting $B = U_p$ gives

$$U_p U_p^T U_p \Sigma_p V_p^T = U_p \Sigma_p V_p^T$$

and $X = U_p^T Y$. The error estimate follows from the truncated SVD error estimate and Lemma 2.9. ∎

Note that CMDS and PCA give for centered data the same result. Observe, that we aimed to use distances, but have Y in the theorem, i.e. in case we only know the distances we cannot proceed. But we can use the Gram matrix instead. We take $Y^T Y$ and observe with the PCA ansatz $Y = WX$:

$$G^c = Y^T Y = (WX)^T WX = X^T W^T WX = X^T X.$$

Take the eigenvalue decomposition of

$$G^c = V\Lambda V^T = (V\Lambda^{\frac{1}{2}})(\Lambda^{\frac{1}{2}} V^T) = (\Lambda^{\frac{1}{2}} V^T)^T (\Lambda^{\frac{1}{2}} V^T).$$

Taking the top eigenvalues gives

$$X_{CMDS} = I_{p\times N}\Lambda^{\frac{1}{2}} V^T.$$

For centered data Y we use $Y = U\Sigma V^T$ and get for the PCA

$$\begin{aligned} X_{PCA} &= I_{p\times d} U^T Y \\ &= I_{p\times d} U^T U\Sigma V^T = I_{p\times d}\Sigma V^T \\ &= I_{p\times N}(\Sigma^T\Sigma)^{\frac{1}{2}} V^T \\ &= I_{p\times N}\Lambda^{\frac{1}{2}} V^T = X_{CMDS}. \end{aligned}$$

Here, we used that the right singular vectors of Y are the eigenvectors of $Y^T Y$ with $\sqrt{\sigma_i}$ the corresponding eigenvalues.

Therefore, one has the options of

SVD of Y	$(d \times N)$	(reconstruct Y)
EVD of YY^T	$(d \times d)$	(maximal variance)
EVD of $Y^T Y$	$(N \times N)$	(preserving similarity)

Depending on d and N one should choose the computationally cheapest option. SVD is the more robust algorithm, but the eigenvalue decomposition is often cheaper to compute.

CMDS Algorithm
Given: Euclidean distance matrix D, embedding dimension p
Output: embedding X in p dimensions

$G = -\frac{1}{2}HDH$
$[V_p, \Lambda_p] = \mathrm{EVD}(G, p)$
return $\Lambda_p^{\frac{1}{2}} V^T$

Generalizations of CMDS consist in

- using weights in $\sum_{i,j=1}^{N} w_{ij}(d_{ij} - d_E(x_i, x_j))^2$. One way is to use $w_{ij} = \frac{1}{d_{ij}}$, which gives Sammon's nonlinear mapping.
- So far we considered metric multidimensional scaling (MDS). In non-metric MDS ordinal information or proximity measures are used instead of d_{ij}.

See [BG05] for more on metric and non-metric MDS.

2.2. Nonlinear dimensionality reduction

The main idea of nonlinear dimensionality reduction is to consider manifolds instead of just linear subspaces. In particular Riemannian manifolds (M, g) and the corresponding inner product on T_pM, see Appendix B for the underlying basic differential geometry. We are in particular interested in the geodesic distance, the length of the shortest curve on M connecting two points $x, y \in M$, which we denote by $d_M(x, y)$. We aim to preserve geodesic distances, but we do know neither M nor d_M. To approximate the geodesic distance given a data set $Y \subset M$ we arrange the data using an undirected neighbourhood graph $[Y, E]$ obtained in a suitable way, see Appendix C. Using $[Y, E]$ we compute the graph distance as an approximation of d_M.

Definition 2.11. Given a graph $[Y, E]$ for a data set $Y \subset \mathbb{R}^d$, such that $(y_i, y_j) \in E$ if and only if y_i and y_j are "adjacent". We define the *graph distance* d_G between two points $y_i, y_j \in Y$ by

1. If $(y_i, y_j) \in E$ then
$$d_G(y_i, y_j) = d_E(y_i, y_j)$$
.

2. If $(y_i, y_j) \notin E$ let
$$\Gamma := \{\gamma \mid \gamma = (\gamma_0, \ldots, \gamma_{s+1}), \gamma_i \in Y, \gamma_0 = y_i, \gamma_{s+1} = y_j\}.$$
Then
$$d_G(y_i, y_j) := \min_{\gamma \in \Gamma} \sum_{\gamma_i \in \gamma, \gamma_i \neq \gamma_{s+1}} d_E(\gamma_i, \gamma_{i+1}).$$

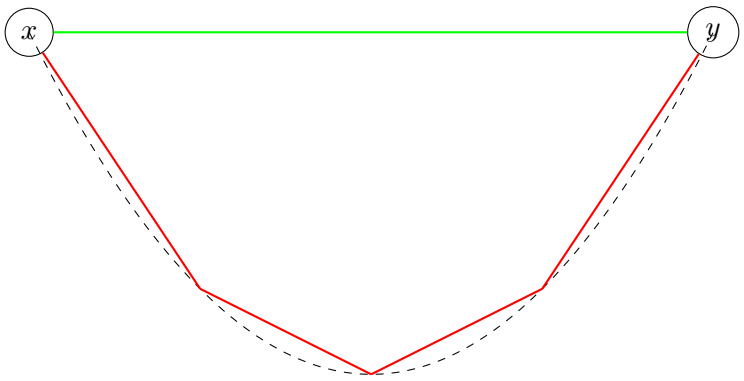


Figure 2.3.: The distance functions d_E shown in green and d_G shown in red between two points x and y. The dashed line symbolizes the geodesic distance d_M.

2.2.1. Isomap

With that, we can formulate the *Isomap* approach introduced by [TSL00].

We assume $Y \subset M \subseteq \mathbb{R}^d$ and an isometric mapping $f : M \to \mathbb{R}^p$ exists, $f(y) = x$ for $y \in M$,
$$d_E(f(y_i), f(y_j)) = d_M(y_i, y_j) \quad y_i, y_j \in M.$$
Assuming Y is sampled densely enough from M we expect that
$$d_G(y_i, y_j) \approx d_M(y_i, y_j),$$

i.e. the graph distance matrix $D_G = \left[d_G^2(y_i, y_j)\right]_{i,j=1}^N$ is a good approximation to the geodesic distance matrix D_M. So we aim for a $D \in \mathbb{EDM}$ with a configuration $X \subset \mathbb{R}^p$ such that

$$D_G \approx D = \left[d_E^2(x_i, x_j)\right]_{i,j=1}^N .$$

Following the CMDS approach, we double center D_G, obtain $G^c = -\frac{1}{2} H D_G H$ and compute the EVD of G^c.

ISOMAP ALGORITHM
Given: data set Y
Output: data set embedding X in p dimensions

Build a neighbourhood graph $[Y, E]$ using k-Nearest Neighbour (k-NN),
ε-neighbourhood or other suitable procedures giving undirected graphs.
$D_{ij} = d_G^2(y_i, y_j)$, $i, j = 1, \ldots, N$ using Dijkstra's algorithm.
$G = -\frac{1}{2} HDH$
$[V_p, \Lambda_p] = \mathrm{EVD}(G, p)$
return $\Lambda_p^{\frac{1}{2}} V_p^T$

Assuming $D_G \in \mathbb{EDM}$ we can invoke Theorem 2.10. We get

$$\sum_{i,j=1}^N \left| d_G^2(y_i, y_j) - d_E^2(x_i, x_j) \right| \leq 2N \sum_{\ell=p+1}^N \lambda_\ell .$$

Remark. When the data Y is not sampled densely enough from M, D_G might not be a good approximation of D_M. In particular D_G might not be an EDM and G might not be positive semidefinite. Computationally one could apply a constant shift technique to make it positive semidefinite, but in a certain sense, this is done implicitly, as we will see later.

We will now analyse the approximation property of the graph distance in regard to the the intrinsic geodesic distance. For the analysis we will use r-ball graphs, i.e.

$$(y_i, y_j) \in E \text{ if and only if } d_E(y_i, y_j) \leq r.$$

Further, we use the Hausdorff-distance between Y and M:

$$\varepsilon = H(M|Y) := \sup_{y \in M} \min_{y_i \in Y} d_E(y - y_i).$$

Theorem 2.12. *Consider $M \subset \mathbb{R}^d$ compact and a sample $Y = y_1, \dots, y_N \subset M$ and let $\varepsilon = H(M|Y)$. For $r > 0$ form the corresponding r-ball graph. When $\varepsilon \leq r/4$, we have for any $x, z \in Y$*

$$d_g(x, y) \leq \left(1 + \frac{4\varepsilon}{r}\right) d_M(x, z).$$

Proof. For $d_E(x, z) \leq r$ we have $(x, z) \in E$ and so

$$d_G(x, z) = d_E(x, z) \leq d_M(x, z).$$

Now, $d_E(x, z) > r$. Let $a = d_M(x, z)$ and let $\gamma : [0, a] \to M$ be a a parameterized by arc length such that $\gamma(0) = x$ and $\gamma(a) = z$. Let $\hat{y}_j = \gamma(ja/s)$ for $j = 0, \dots, s$, where $s := 2a/r \geq 2$, where $\hat{y}_0 = x$ and $\hat{y}_s = z$. Let $y_{i_j} = \arg\min_{y \in Y} d_E(y, \hat{y}_j)$ be the closest point to $\hat{y}_j$ among the sampled data Y. Clearly $y_{i_0} = x$, $y_{i_s} = z$ and $\max_j d_E(y_{i_j}, \hat{y}_j) \leq \varepsilon$. For any $j \in \{0, \dots, s-1\}$

$$\begin{aligned} d_E(y_{i_j}, y_{i_{j+1}}) &\leq d_E(y_{i_j}, \hat{y}_j) + d_E(\hat{y}_j, \hat{y}_{j+1}) + d_E(\hat{y}_{j+1}, y_{i_{j+1}}) \\ &\leq \varepsilon + d_M(\hat{y}_j, \hat{y}_{j+1}) + \varepsilon \\ &= a/s + 2\varepsilon \leq r/2 + 2\epsilon \leq r \end{aligned}$$

Therefore, $(y_{i_0}, \dots, y_{i_j})$ forms a path in the r-ball graph. With that, and using $\hat{y}_0 = y_{i_0} = x$, $\hat{y}_s = y_{i_s} = z$,

$$\begin{aligned} d_G(x, z) &\leq \sum_{j=0}^{s-1} d_E(y_{i_j}, y_{i_{j+1}}) \\ &\leq d_M(\hat{y}_0, \hat{y}_1) + \varepsilon + \sum_{j=1}^{s-2} (d_M(\hat{y}_j, \hat{y}_{j+1}) + 2\varepsilon) + d_M(\hat{y}_{s-1}, \hat{y}_s) + \varepsilon \\ &= d_M(x, z) + 2(s-1)\varepsilon \\ &\leq \left(1 + \frac{4\varepsilon}{r}\right) d_M(x, z), \end{aligned}$$

where we use that $s - 1 \leq 2a/r$ with $a = d_M(x, z)$. ∎

Remark. It is possible to tighten the bound in the special case where M is convex. In that case, a refinement of the arguments above leads to an error term of $\left(\frac{\varepsilon}{r}\right)^2$ [AJP18].

To show a result in the other direction, we need a general concept of *curvature*. Assuming a curve γ is twice differentiable at t, its curvature at t is defined as

$$\operatorname{curv}(\gamma, t) := \frac{\|\dot{\gamma}(t)\| \wedge \|\ddot{\gamma}(t)\|}{\|\dot{\gamma}(t)\|}.$$

Now, γ has a curvature bounded by κ if and only if $\sup_t \operatorname{curv}(\gamma, t) \leq \kappa$.

Lemma 2.13. *Let $\gamma : [0, a] \to \mathbb{R}^d$ be a unit-speed curve with curvature bounded by κ. Then*

$$d_E(\gamma(s), \gamma(t)) \geq \frac{2}{\kappa} \sin\left(\kappa \frac{|t-s|}{2}\right)$$

for all $s, t \in [0, a]$ such that $|t - s| \leq \pi/\kappa$.

Proof. The full, somewhat technical, proof is in [Ber+00]. Following [AL19], we show a slightly weaker bound. For that, let c denote a unit-speed parametrization of a circle of radius $1/\kappa$. From the classical work of [Dub57], we obtain for $|s - a| \leq \pi/\kappa$

$$\langle \dot{\gamma}(s), \dot{\gamma}(u) \rangle \geq \langle \dot{c}(s), \dot{c}(u) \rangle.$$

This leads to, using $\|\dot{\gamma}(s)\| = 1$ in the first line,

$$\begin{aligned} \|\gamma(t) = \gamma(s)\| &\geq \langle \dot{\gamma}(s), \gamma(t) - \gamma(s) \\ &= \int_s^t \langle \dot{\gamma}(s), \dot{\gamma}(u) \rangle du \\ &\geq \int_s^t \langle \dot{c}(s), \dot{c}(u) \rangle du \\ &= \langle \dot{c}(s), c(t) - c(s) \rangle \\ &= \frac{1}{\kappa} \sin(\kappa(t-s)) \end{aligned}$$

when $0 \leq t - s \leq \pi/\kappa$. ∎

Properties of M We now assume

1. $M \subset \mathbb{R}^d$ is a compact and connected C^2-manifold
2. $M \subset \mathbb{R}^d$ has empty or C^2-boundary

In particular, shortest paths on M have curvature bounded by some κ, see [AL19] for generalized properties.

Lemma 2.14. *Suppose $M \subset \mathbb{R}^d$ has the above two properties. Then, for any $x, z \in M$ such that $d_M(x,z) \leq \pi/\kappa$*

$$d_M(x,z) \max\left(\frac{2}{\pi}, 1 - \frac{\kappa^2}{24} d_M(x,z)^2\right) \leq d_E(x,z) \leq d_M(x,z) \tag{37}$$

Moreover, there is $\tau > 0$ depending on M, such that for all $x, z \in M$ with $d_E(x,z) \leq \tau$ it holds

$$d_E(x,z) \leq d_M(x,z) \leq d_E(x,z) \min\left(\frac{\pi}{2}, 1 + c_0 \kappa^2 d_E(x,z)^2\right), \tag{38}$$

where c_0 is a constant that can be taken to be $c_0 = \pi^2/50$.

Remark. τ can be specified in terms of the so-called *reach* of M. The reach of a subset A in some Euclidean space is the supremum over $t \geq 0$ such that for any point x at distance at most t from A, there is a unique point among those belonging to A that is closest to x, i.e., it is a unique nearest point property. When A is a C^2-manifold, its reach is known to be bound by its radius of curvature from below [Fed59].

Theorem 2.15. *Suppose $M \subset \mathbb{R}^d$ has the above two properties. Consider a sample $Y = \{y_1, \ldots, y_N\} \subset M$. For $r > 0$, form the corresponding r-ball graph. Let c_0 and τ be defined per Lemma 2.14. When $r \leq \tau$ and $\kappa r \leq 1/3$ we have*

$$d_M(x,z) \leq (1 + c_0 \kappa^2 r^2) d_G(z,z) \quad \forall x, z \in Y.$$

Proof. Fix $x, z \in Y$. Let $x = y_{i_0}, y_{i_1}, \ldots, y_{i_s} = z$ define a shortest path in the graph joining x and z, so that

$$d_M(x,z) = \sum_{j=0}^{s-1} \Delta_j, \qquad \text{where } \Delta_j = d_E(y_{i_j}, y_{i_{j+1}}).$$

Define $a = d_M(x,z)$ and $a_j = d_M(y_{i_j}, y_{i_{j+1}})$ for $j = 0, \ldots, s-1$. Since $\Delta_j \leq r \leq \tau$, by Lemma 2.14 we get

$$\Delta_j \min\left(\frac{\pi}{2}, 1 + c_0 \kappa^2 \Delta_j^2\right) \geq a_j.$$

■

By assumption, $\kappa r \leq 1/3$, and this can be seen to imply $1 + c_0 \kappa^2 r^2 \leq \pi/2$, which then implies that $a_j \leq \Delta_j + c_0 \kappa^2 \Delta_j^3$.

We thus have

$$\begin{aligned} a \leq \sum_{j=0}^{s-1} a_j &\leq \sum_{j=0}^{s-1} \left(\Delta_j + c_0 \kappa^2 \Delta_j^3\right) \\ &\leq \sum_{j=0}^{s-1} \Delta_j \left(1 + c_0 \kappa^2 r^2\right) \\ &= (1 + c_0 \kappa^2 r^2) d_G(x, z). \end{aligned}$$

Remark. Therefore we have bounded d_M and d_G in both directions. We can achieve for some $\xi < 1$:

$$1 - \xi \leq \frac{d_g(y_i, y_j)}{d_M(y_i, y_j)} \leq 1 + \xi. \tag{39}$$

Remark. One can show that with high probability the conditions are fulfilled if there is a sufficiently high density of points. This holds for both r-ball graphs and k-NN graph constructions [Ber+00].

In practice though, shortcuts can be a problem. Consider a noisy Swiss roll, which is only sparely sampled: In this situation k-NN and r-ball graphs can connect to the next layer and thus the approximation could take a shortcut between the layers of the spiral, whereas the geodesic distance would have to traverse the spiral yielding a much longer path.

2.2.2. Perturbation Analysis

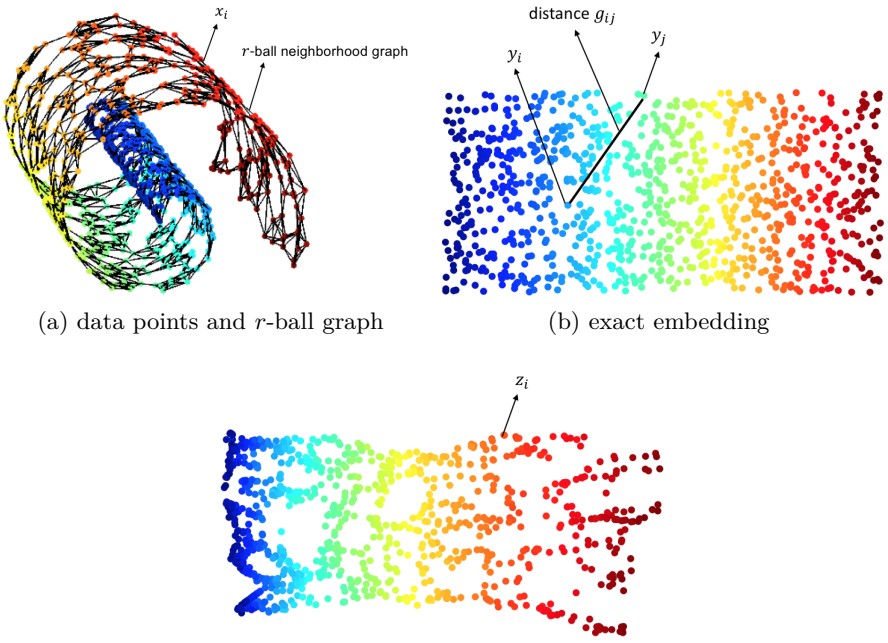
Theorem 2.16. *Consider two tall matrices X and Y in $\mathbb{R}^{N \times d}$, with X having full rank. Set $\epsilon^2 = \|YY^\top - XX^\top\|_p$. If $\|X^\ddagger\|\epsilon \leq \frac{1}{\sqrt{2}}$, then*

$$\min_{Q \in \mathcal{O}} \|Y - XQ\|_p \leq (1 + \sqrt{2}) \|X^\ddagger\| \epsilon^2. \tag{40}$$

Proof. See [AJP20]. ∎

Corollary 2.17. *Let $D, \tilde{D} \in \mathbb{R}^{N \times N}$ denote two Euclidean distance matrices, with D corresponding to a centered, exact configuration $Y \in \mathbb{R}^{N \times d}$. Set $\varepsilon^2 = \frac{1}{2}\|H(\tilde{D} - D)H\|_p$. If it holds that $\|Y^\ddagger\|\varepsilon \leq \frac{1}{\sqrt{2}}$, then MDS with input distance matrix $\tilde{D}$ and dimension d returns a centered configuration $Z \in \mathbb{R}^{m \times d}$ satisfying*

$$\min_{Q \in \mathcal{O}} \|Z - YQ\|_p \leq (1 + \sqrt{2}) \|Y^\ddagger\| \varepsilon^2. \tag{41}$$



(a) data points and r-ball graph

(b) exact embedding

(c) Returned locations by Isomap

Figure 2.4.: Schematic representation of exact locations $y_i \in \mathbb{R}^d$, data points $x_i \in M$, returned locations by Isomap $z_i \in \mathbb{R}^d$. Note that $g_{ij} = \|y_i - y_j\|$ is the geodesic distance between x_i and x_j because $\{y_i\}_{i=1}^n$ is an exact isometric embedding of data points $\{x_i\}_{i=1}^n$. Also the distances γ_{ij} are computed as shortest path distances between x_i and x_j on the r-ball neighborhood graph.

PROOF. We have

$$\|\tilde{D}^c - D^c = \frac{1}{2}\|H(D^c - D^c)H\|_p = \varepsilon^2.$$

Due to the connection to the Gram matrix we know that $\tilde{D}^c$ and D^c are are positive semi-definite and of rank at most d. Since Y is of rank d, we have rank d for D^c. Observe $D^c = Y \cdot Y^T$ and $\tilde{D}^c = Z \cdot Z^T$, the results follows from Theorem 2.16. ∎

Note that $\varepsilon^2 \leq \frac{1}{2}d^{2/p}\|\tilde{D} - D\|_p$, after using that $\|H\|_p = (d-1)^{1/p}$ since H has one zero eigenvalue and $d-1$ eigenvalues equal to one.

For a centered point set $y_1, \ldots, y_N \in \mathbb{R}^d$, stored in the matrix $Y = [y_1 \cdots y_N]^\top \in \mathbb{R}^{N \times d}$, we define its radius as the largest standard deviation along any direction in space (therefore corresponding to the square root of the top eigenvalue of the covariance matrix). We denote this by $\rho(Y)$ and note that

$$\rho(Y) = \sigma_1(Y)/\sqrt{N}. \tag{42}$$

We define its half-width as the smallest standard deviation along any direction in space (therefore corresponding to the square root of the bottom eigenvalue of the covariance matrix). We denote this by $\omega(Y)$ and note that it is strictly positive if and only if the point set spans the whole space, in which case

$$\omega(Y) = \sigma_d(Y)/\sqrt{N}. \tag{43}$$

Note that we know that the half-width quantifies the best affine approximation to the point set:

$$\omega(Y)^2 = \min_{\mathcal{L}} \frac{1}{N} \sum_{i\in[m]} \|y_i - P_{\mathcal{L}} y_i\|^2, \tag{44}$$

where the minimum is over all affine hyperplanes $\mathcal{L}$, and for a subspace $\mathcal{L}$, $P_{\mathcal{L}}$ denotes the orthogonal projection onto $\mathcal{L}$.

Further, we call $\rho(Y)/\omega(Y) = \|Y\|\|Y^\ddagger\|$ the aspect ratio of the point set.

Corollary 2.18. *Consider a centered point set $y_1, \dots, y_N \in \mathbb{R}^d$ with radius ρ, and with half-width ω, and with pairwise dissimilarities $d_{ij} = \|y_i - y_j\|^2$. Consider another set of numbers $\{\lambda_{ij}\}$ and set $\eta^4 = \frac{1}{N^2}\sum_{i,j}(\lambda_{ij} - d_{ij})^2$. If $\eta/\omega \le \frac{1}{\sqrt{2}}$, then MDS with inputs $\{\lambda_{ij}\}$ and dimension d returns a point set $z_1 \cdots z_N \in \mathbb{R}^d$ satisfying*

$$\min_{Q\in\mathcal{O}} \left(\frac{1}{N}\sum_{i}^{N} \|z_i - Qy_i\|^2\right)^{1/2} \le \frac{\sqrt{d}(\rho/\omega + 2)}{\omega}\eta^2 \le \frac{3\sqrt{d}\rho\,\eta^2}{\omega^2}. \tag{45}$$

Proof. See [AJP20]. ∎

Corollary 2.19. *Let $x_1, \dots, x_N \in \mathbb{R}^d$ denote a possible (exact and centered) embedding of the data points $y_1, \dots, y_N \in M$, preserving the geodesic distance d_M of the y_i. Assume for some $\xi < 1$:*

$$1 - \xi \le \frac{d_g(y_i, y_j)}{d_M(y_i, y_j)} \le 1 + \xi. \tag{46}$$

Let ρ and ω denote the max-radius and half-width of the embedded points, respectively. If $\xi \le \frac{1}{24}(\rho/\omega)^{-2}$, then Isomap returns $z_1, \dots, z_N \in \mathbb{R}^d$ satisfying

$$\min_{Q\in\mathcal{O}} \left(\frac{1}{n}\sum_{i\in[n]} \|z_i - Qx_i\|^2\right)^{1/2} \le \frac{36\sqrt{d}\rho^3}{\omega^2}\xi. \tag{47}$$

PROOF. Follows from Corollary 2.18 using Eq. (46) and

$$\begin{aligned}\eta^2 &\leq \max_{j,k} |d_g^2(y_j, y_k) - d_M^2(y_j, y_k)| \\ &\leq \max_{j,k} (2\xi + \xi^2)\, d_M^2(y_j, y_k) \\ &\leq (2\xi + \xi^2)(2\rho)^2 \leq 12\rho^2\xi.\end{aligned}$$

η fulfills the condition of Corollary 2.18, which we apply and simplify afterwards to obtain the result. ∎

2.2.3. Nonlinear PCA and Kernel MDS

With Isomap we have seen one generalization of CMDS, where we used a different distance in the original space. Another generalization stems from a kernel view, i.e. a replacing scalar products by so-called kernels. Here we follow the original derivation from [SSM98], which is based on a normal PCA via a detour into higher dimensions using a nonlinear function, i.e. a feature map.

We take the covariance view with its empirical estimate

$$C \coloneqq \frac{1}{N} Y Y^T = \frac{1}{N} \sum_{i=1}^{N} y_i y_i^T.$$

Using a *feature map*, $p \gg d$:

$$\begin{aligned}\phi : \mathbb{R}^d &\to \mathbb{R}^p =: \mathcal{F}, \\ y &\mapsto \phi(y)\end{aligned}$$

we go into into the (higher dimensional) feature space $\mathcal{F}$, which is a Hilbert space. Note that one can also generalize to infinite dimensional Hilbert spaces.

For simplicity we assume to have centered data in the feature space, i.e.

$$\sum_{i=1}^{N} \phi(y_i) = 0.$$

Now, we perform a PCA in $\mathcal{F}$ by using the empirical mean

$$C_\phi \coloneqq \frac{1}{N} \sum_{i=1}^{N} \phi(y_i)\phi(y_i)^T = \frac{1}{N} \underbrace{\phi(Y)}_{=:\Phi} \phi(Y)^T = \frac{1}{N} \Phi\Phi^T,$$

and compute the eigenvectors $u_i \in \mathbb{R}^p$ of the covariance matrix C_ϕ:

$$C_\phi u_i = \lambda_i u_i \quad i = 1, 2, \ldots, p$$

and by Theorem 2.3 the *nonlinear principal components* are

$$x_i = u_i^T \Phi.$$

We call this *nonlinear PCA*.

Whereas one searches for the best linear subspace with PCA, nonlinear PCA (or feature space PCA) transforms nonlinear data using a feature map ϕ and then searches for the best linear subspace in the feature space $\mathcal{F}$. Note that while the contour lines of constant projections on the principal components are straight lines orthogonal to the linear subspace, but the corresponding contour lines for the source data don't have to be straight lines.

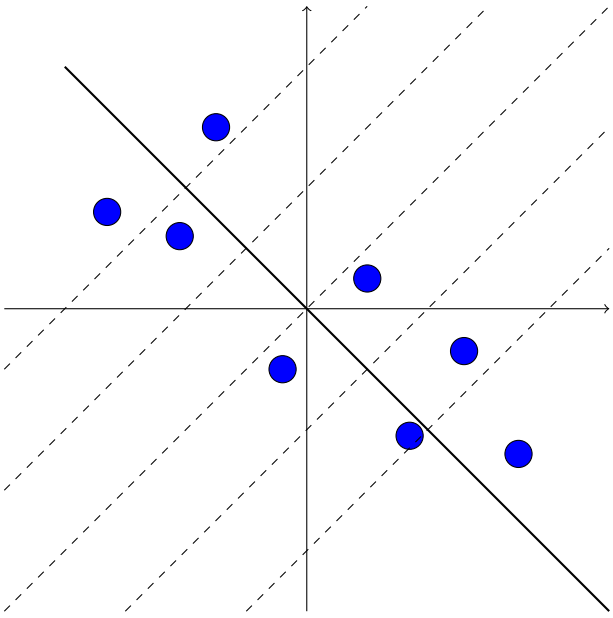


Figure 2.5.: Classical linear PCA

In some practical situations, good candidates for ϕ can be found from the nature of the problem. Generally ϕ is not known beforehand and difficult to obtain.

We now use the connection between PCA to MDS here, in particular if p is large. So, we compute the EVD of $\Phi^T\Phi$ which has the same eigenvalues as $\Phi\Phi^T$. This implies that the number of nonzero eigenvalues is $\min(p, N)$. Here, we observe that every eigenvector $u \in \mathbb{R}^p$ of $\Phi\Phi^T$ associated with a nonzero eigenvalue is in the span of Φ:

$$\Phi\Phi^T u = \lambda u \quad \Leftrightarrow \quad u = \Phi\left(\lambda^{-1}\Phi^T u\right) \in \text{range}(\Phi), \lambda \neq 0.$$

Now, let $v = \lambda^{-1}\Phi^T u \in \mathbb{R}^N$ and let u be normalized, we have

$$\|v\|^2 = \lambda^{-2} u^T \underbrace{\Phi\Phi^T u}_{\lambda u} = \lambda^{-1} \underbrace{\|u\|^2}_{=1, \text{ since normalized}} = \lambda^{-1}$$

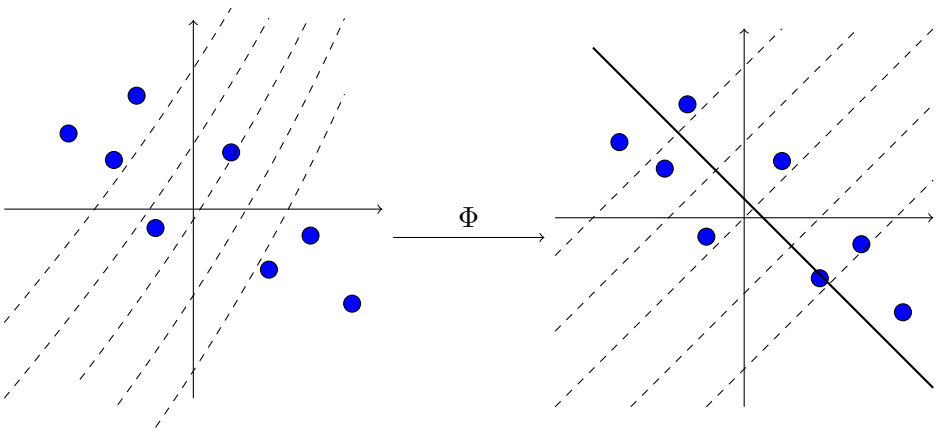


Figure 2.6.: Non-linear PCA

and

$$\Phi^T\Phi v = \lambda^{-1}\Phi^T \underbrace{\Phi\Phi^T u}_{\lambda u} = \Phi^T u = \lambda v.$$

v is an eigenvector of $\Phi^T\Phi$ with the same eigenvalue. Once we compute v_i with an EVD of $\Phi^T\Phi$ as $\hat{V}\Lambda\hat{V}^T$, where the $\hat{v}_i$ are normalized to 1, we have to normalize

$$v_i = \lambda^{-\frac{1}{2}}\hat{v}_i,$$

so that $\|v_i\|^2 = \lambda^{-1}$. We obtain u_i in the feature space as

$$u_i = \Phi v_i = \lambda^{-\frac{1}{2}}\Phi\hat{v}_i$$

and compute the nonlinear principal components, using the freshly derived relation $U^T = V^T\Phi^T$

$$X = V^T\Phi^T\Phi = \Lambda^{-\frac{1}{2}}\hat{V}^T\Phi^T\Phi = \Lambda^{-\frac{1}{2}}\hat{V}^T\hat{V}\Lambda\hat{V}^T = \Lambda^{\frac{1}{2}}\hat{V}^T.$$

As before, one can calculate x_i by

$$x_i = V^T\Phi^T\phi(y_i).$$

We now kernelize this, i.e. instead of the Gram matrix $\Phi^T\Phi$ we use the kernel matrix

$$K_{ij} = \langle\phi(y_i), \phi(y_j)\rangle_{\mathcal{F}} =: K(y_i, y_j).$$

We now can start with a kernel to begin with.

A projection of a point y with image $\phi(y)$ now goes accordingly.

$$\begin{aligned} x &= V^T \Phi^T \phi(y) \\ &= \sum_{i=1}^{N} v_i \langle \phi(y_i), \phi(y) \rangle_{\mathcal{F}} \\ &= \sum_{i=1}^{N} v_i K(y_i, y) \end{aligned}$$

This kernelized form we call kernel MDS, usually kernel PCA is used in the literature. Using kernels, the nonlinear PCA is performed only implicitly, i.e., ϕ is not used directly, instead only the kernel is employed. But, for a given kernel there is unique feature map, so kernelizing is loosing interpretation power in the sense of the feature space.

If we want to consider non-centered data we need to center it

$$\tilde{\phi}(y_i) \coloneqq \phi(y_i) - \frac{1}{N} \sum_{i=1}^{N} \phi(y_i)$$

and then do

$$\tilde{K}_{ij} = \langle \tilde{\phi}(y_i), \tilde{\phi}(y_i) \rangle.$$

We can perform the centering as we did before Theorem 2.6:

$$\tilde{\Phi}(Y) = \Phi^c(Y) = \Phi(Y) - \left(\frac{1}{N} \Phi(Y) \mathbb{1} \right) \mathbb{1}^T = \Phi(Y) H.$$

The action of H goes through the scalar product so

$$\tilde{K} = K^c = HKH.$$

Remark. Considering cpsd kernels, see Appendix **??**, we can connect this to distance matrices

$$d_{CS} = k(y_i, y_i) - 2k(y_i, y_j) + k(y_j, y_j)$$

or in matrix form

$$D = \operatorname{diag}(K) \mathbb{1}^T - 2K + \mathbb{1} \operatorname{diag}(K)^T.$$

2.2.4. Maximum variance unfolding

Kernel MDS using "just some" kernel is at first actually increasing the dimension. For example the Gaussian kernel results in a matrix with full rank N as seen in Chapter 1. In order to fully reconstruct the distance we need $N - 1$ eigenvalues of the centered kernel matrix, but also the decay of the eigenvalues is in this form not that strong.

Maximum variance unfolding (MVU) The idea behind maximum variance unfolding is to learn a kernel matrix, which does have good and useful properties. What are these?

1. The matrix should be symmetric positive semidefinite, so that it is a matrix of scalar products. Such matrices form a cone, so one has a convex domain to optimize over. One aims for $K \in \mathbb{EDM}$.

2. We want centered data, so we look for a centered matrix $K = HKH$.

3. We want to preserve distances. For maximum variance unfolding the motivation is to enforce this only locally
$$d_E(\phi(y_i), \phi(y_j)) = d_E(y_i, y_j)$$
for y_i, y_j which are nearby, e.g. if $(y_i, y_j) \in E$ in a k-NN or ε-neighbourhood graph built using the Euclidean distance.

4. We maximize the variance in the feature space by maximizing the pairwise squared distances for $y_i, y_j \in Y$ with $(y_i, y_j) \notin E$, i.e. data points further apart. From Lemma 2.8 we know that
$$\frac{1}{2N} \sum_{i,j=1}^{N} d_E^2(z_i, z_j) = \operatorname{tr}(G_Z^c)\,.$$
So we maximize the trace of $G_Z^c = K$.

Definition 2.20. Given a data set Y and a neighbourhood graph $[Y, E]$. The solution of
$$\max_{K \in \text{symmetric positive semidefinite}} \operatorname{tr}(K) \text{ s.t. } \sum_{i,j=1}^{N} K_{ij} = 0$$
and if $(y_i, y_j) \in E$, we have
$$d_E^2(y_i, y_j) = K_{ii} - 2K_{ij} + K_{jj}$$
is called the *maximum variance unfolding (MVU)* kernel matrix K.

Remark. One can formulate this problem as preserving the local geometry for each point y_i by $d_E(x_j, x_k) \approx d_E(y_j, y_k)$, where y_j, y_k are in the neighbourhood of y_i, e.g. $(y_i, y_j) \in E$, or $(y_i, y_k) \in E$.

As seen, we can write the MVU problem as
$$\begin{aligned} &\max_{D \in \mathbb{EDM}} \sum_{i,j=1}^{N} D_{ij} \\ &\text{s.t.} \quad D_{ij} = d_E^2(y_i, y_j) \quad \text{if } (y_i, y_j) \in E \end{aligned}$$

This allows us to connect the MVU problem to distance matrix approximations.

Theorem 2.21. *Let $C \subset \mathbb{DM}$ and $[Y, E]$ be a weighted graph with weights d_w. If the graph is connected, the following constrained optimization problems are equivalent:*

i)

$$\max_{D \in C} \sum_{i,j=1}^{N} D_{ij}$$

such that

$$D_{ij} = d_w^2(y_i, y_j) \text{ if } (y_i, y_j) \in E$$

ii)

$$\min_{D \in C} |||D - D^G|||_1 \coloneqq \min_{D \in C} \sum_{i,j=1}^{N} \left| D_{ij} - D_{ij}^G \right|$$

such that

$$D_{ij} = d_w^2(y_i, y_j) \text{ if } (y_i, y_j) \in E,$$

where $D_{ij}^G = d_G^2(y_i, y_j)$ is the squared graph distance matrix for the edge weights d_w.

Proof. Let $D \in C$, then for all $1 \leq i, j \leq N$ and paths γ in $[Y, E]$ connecting $y_i = \gamma_0$ and $y_j = \gamma_{s+1}$ the triangle inequality implies

$$\begin{aligned} \sqrt{D_{ij}} &\leq \sum_{k=0}^{s} \sqrt{D_{\gamma_k \gamma_{k+1}}} \\ &= \sum_{k=0}^{s} \sqrt{d_w^2(\gamma_k, \gamma_{k+1})} \\ &= \|\gamma\| =: \ell. \end{aligned}$$

In particular, this holds for the shortest path between y_i and y_j, i.e. $D_{ij} \leq d_G^2(y_i, y_j)$ for all $1 \leq i, j \leq N$. This gives

$$\begin{aligned} \sum_{i,j=1}^{N} D_{ij} - \underbrace{\sum_{i,j=1}^{N} D_{ij}^G}_{=\text{constant}} &= \sum_{i,j=1}^{N} \left(D_{ij} - D_{ij}^G \right) \\ &= - \sum_{i,j=1}^{N} \left| D_{ij} - D_{ij}^G \right| \\ &= - |||\, D - D^G \,|||_1 \,. \end{aligned}$$

Adding a constant to the objective does not affect contour lines, so the result follows. ∎

Corollary 2.22. *Let $[Y, E]$ be a weighted graph with weights d_w. If $[Y, E]$ is connected, then D^G is the unique solution of*

$$\begin{aligned} &\max_{D \in \mathbb{EDM}} \sum_{i,j=1}^{N} D_{ij} \\ &\quad s.t. \quad D_{ij} = d_w^2(y_i, y_j) \end{aligned}$$

This shows, that the shortest path problem on a graph shows is equivalent to a (non-Euclidean) variant of MVU.

Corollary 2.23. *Let $[Y, E]$ be a connected, weighted graph with weights $d_E(y_i, y_j)$. Then*

1. *If $D_G \in \mathbb{EDM}$, then D_G is the unique solution of the MVU problem from Definition 2.20.*

2. *The problem from Definition 2.20 is equivalent to*

$$\begin{aligned} &\min_{D \in \mathbb{EDM}} |||D - D_G|||_1 \\ &\quad s.t. \quad D_{ij} = d_E^2(y_i, y_j) \text{ if } (y_i, y_j) \in E \end{aligned}$$

In general $D_G \notin \mathbb{EDM}$, even if $D_M \in \mathbb{EDM}$ for all data samples. If D_G is arbitrarily close to D_M in the sense of **??** it still does no imply $D_G \in \mathbb{EDM}$. Since for the purpose of MDS we want an EDM approximation, the employed EDM-constraint is useful. Thereby we can consider MVU as a regularized shortest path problem.

Note that MVU was originally motivated by preserving local distances only, while maximizing the variance in the feature space for non-locally connected data points. But we see, that global distances in the form of graph distances as an approximation to geodesic distances are used implicitly.

Remark. Note that one can also put Isomap in this framework, it is implicitly using the best $\mathbb{EDM}$ to D_G in the sense of

$$\min_{D \in \mathbb{EDM}} \|H(D - D_G)H\|_F.$$

We now want to show an asymptotic result for MVU, for which first we need a technical lemma and some notation.

We denote for a set S by $\mathbb{R}^S$ the set of real-valued functions on S. The restriction of a function $f \in \mathbb{R}^S$ to a subset $\tilde{S}$ of S is denoted by $f|_{\tilde{S}}$.

Lemma 2.24. *Let S be a set, $\tilde{S} \subset S$, $C \subset \mathbb{R}^{\tilde{S}}$, $f \in \mathbb{R}^S$ and $\tilde{f} \in \mathbb{R}^{\tilde{S}}$. Let $\|\cdot\|$ be a norm on $\mathbb{R}^{\tilde{S}}$ and $c, \varepsilon \geq 0$. If*

$$\|\tilde{f} - f|_{\tilde{S}}\| \leq c\varepsilon \tag{48}$$

and

$$(1-\varepsilon) f|_{\tilde{S}} \in C \tag{49}$$

then

$$\left\|\hat{f} - f|_{\tilde{S}}\right\| \leq (2c + \|f|_{\tilde{S}}\|)\,\varepsilon \quad \forall \hat{f} \in \operatorname*{arg\,min}_{\bar{f} \in C} \|\bar{f} - \tilde{f}\|$$

Proof.

$$\begin{aligned}
\|\hat{f} - f|_{\tilde{S}}\| &\leq \|\hat{f} - \tilde{f}\| + \|\tilde{f} - f|_{\tilde{S}}\| \\
&\overset{\text{Eq. (49)}}{\leq} \|(1+\varepsilon) f|_{\tilde{S}} - \tilde{f}\| + \|\tilde{f} - f|_{\tilde{S}}\| \\
&\leq \varepsilon \|f|_{\tilde{S}}\| + 2\|\tilde{f} - f|_{\tilde{S}}\| \\
&\overset{\text{Eq. (48)}}{\leq} (\|f|_{\tilde{S}}\| + 2c)\varepsilon
\end{aligned}$$

This proves what was to be shown. ∎

Theorem 2.25. *Let $[Y, E]$ be a given connected graph, $Y \subset M$ where M is a convex and compact manifold. For the graph distance matrix $D_G = [d_G^2(y_i, y_j)]_{i,j=1}^N$ we assume it holds for some $\varepsilon > 0$ that*

$$(1-\varepsilon) d_M^2(y_i, y_j) \leq d_G^2(y_i, y_j) \leq (1+\varepsilon) d_M^2(y_i, y_j).$$

Then any solution D of the MVU problem from Definition 2.20 satisfies (with $D_M = [d_M^2(y_i, y_j)]_{i,j=1}^N$)

$$|||D - D_M|||_1 \leq 3\,|||\,D_M|||_1 \leq 3(n\,\mathrm{diam}(M))^2 \varepsilon,$$

where $\mathrm{diam}(M) = \sup_{x,y \in M} d_M(x, y)$.

Proof. We use Lemma 2.24 with

- $S := M \times M$, $\tilde{S} := Y \times Y$,
- $C := \left\{ \tilde{f} \in \mathbb{R}^{\tilde{S}} \;\middle|\; \left[\tilde{f}(y_i, y_j)\right]_{i,j=1}^N \in \mathbb{EDM} \right\}$,
- $f := d_M^2(\cdot, \cdot)$, $\tilde{f}(y_i, y_j) = d_G^2(y_i, y_j)$,
- $\|x\| = |||[x(y_i, y_j)]_{i,j=1}^N|||_1$,

- $c \coloneqq |||D_M|||_1$.

For the condition Eq. (48) of Lemma 2.24:

$$\begin{aligned}\left\|\tilde{f} - f|_{\tilde{S}}\right\| &= |||D_G - D_M|||_1 \\ &= \sum_{i,j=1}^{N} \left|d_G^2(y_i, y_j) - d_M^2(y_i, y_j)\right| \\ &\overset{\text{per assumption}}{\leq} \sum_{i,j=1}^{N} \varepsilon d_M^2(y_i, y_j) \\ &= \varepsilon \underbrace{|||D_M|||_1}_{c}\end{aligned}$$

For the condition Eq. (49) we observe $D_M \in \mathbb{EDM}$. Since $\mathbb{EDM}$ is a cone, it follows that $(1-\varepsilon)D_M \in \mathbb{EDM}$ so Eq. (49) holds. Lemma 2.24 then implies

$$\left\|\hat{f} - f|_{\tilde{S}}\right\| \leq (2c + \|f|_{\tilde{S}}\|)\,\varepsilon = 3\,|||\,D_M\,|||\,\varepsilon \quad \forall \hat{f} \in \arg\min_{\bar{f} \in C} \left\|\bar{f} - \tilde{f}\right\|$$

Therefore

$$|||D - D_M|||_1 \leq 3\,|||\,D_M\,|||_1\,\varepsilon$$

Additionally, we observe

$$\begin{aligned}|||D_M|||_1 &= \sum_{i,j=1}^{N} d_M^2(y_i, y_j) \\ &\leq n^2 \max_{1 \leq i,j \leq N} d_M^2(y_i, y_j) \\ &\leq (n\,\mathrm{diam}(M))^2.\end{aligned}$$

This proves the theorem. ∎

In the context of the perturbation analysis, one can write this as

$$|||\,|D - D_M|\,|||_1 \leq 3\rho^2 N^2 \xi, \tag{50}$$

with ξ small enough.

Corollary 2.26. *In the same context as in Corollary 2.19, if instead*

$$\xi \leq (12\sqrt{3})^{-1}(\rho/\omega)^{-2},$$

then Maximum Variance Unfolding returns $z_1, \ldots, z_N \in \mathbb{R}^d$ satisfying

$$\min_{Q \in \mathcal{O}} \left(\frac{1}{N}\sum_{i=1}^{N} \|z_i - Qy_i\|^2\right)^{1/2} \leq \frac{18\sqrt{d}\rho^3}{\omega^2}\xi. \tag{51}$$

PROOF. As in Corollary 2.19 we have

$$\eta^2 \leq \max_{j,k} |d_g^2(y_j, y_k) - d_M^2(y_j, y_k)| \leq 12\rho^2\xi.$$

Together with Eq. (50) and using Hölder's inequality we get

$$\|D - D_M\|_2 \leq \||\,|D - D_M|\,\|_\infty^{1/2}\ \||\,|D - D_M|\,\|_1^{1/2} \leq \sqrt{12}\rho\sqrt{\xi}\sqrt{3}\rho N\sqrt{\xi} = 6N\rho^2\xi.$$

The conditions of Corollary 2.18 are met under the assumed bound on ξ, from which the results follows. ∎

MVU ALGORITHM
Given: data set Y (can be modified to work with distance matrices)
Output: data set embedding in p dimensions

build a neighbourhood graph $[Y, E]$
use semidefinite programming to solve the MVU problem from Definition 2.20 to obtain K
$[V_p, \Lambda_p] = \mathrm{EVD}(K, p)$
return $\Lambda^{\frac{1}{2}} V_p^T$

For $P, Q \in \mathbb{R}^{N\times N}$ symmetric, a *semidefinite programming* (SDP) problem would be

$$\begin{aligned} \min_{Q \text{ positive semidefinite matrix}} \quad & \langle P, Q\rangle = \sum_{i,j=1}^{N} p_{ij}q_{ij} \\ \text{s.t.} \quad & \langle C_i, Q\rangle = b_i, \end{aligned}$$

where $\mathbf{C}_i$, $1 \leq i \leq N$, be N symmetric matrices and $\mathbf{b} = [b_1, \ldots, b_N]^{\mathrm{T}} \in \mathbb{R}^N$. Semidefinite programming is concerned with solving such semidefinite problems, Appendix D. There exist good solvers for these types of problems, if the problem size is not too large, i.e. in our setup a couple of thousand data points.

In MVU we have

$$\begin{aligned} P &= -I \\ C_i &= C_{k,j} \\ b_i &= D_{k,j}, \end{aligned}$$

where the double index (k, j) is 1-1 mapped to the single index i, and the binary matrix C_i has only non-vanishing entries at (k, k), (j, j), (k, j) and (j, k) if $(y_j, y_k) \in E$. Under the MVU setting the SDP has a non-empty feasible set, i.e. has a solution, since the centering Gram matrix is in the feasible set.

2.2.5. Spectral Clustering

with W a weight matrix for the graph, i.e.

$$W_{ij} = 0 \qquad \text{if } (y_i, y_j) \notin E,$$
$$W_{ij} \geq 0 \qquad \text{if } (y_i, y_j) \in E,$$

Let us now have a close look at neighbourhood graphs in general. We consider $[Y, E]$ as an undirected graph, where $(y_i, y_j) \in E$ if the two data points are "nearby". The weights we now organize in the weighted adjacency matrix:

$$W = [w_{jk}]_{j,k=1}^{N},$$

where so far we used a distance measure, i.e. in Isomap, Kernel MDS or MVU. In the following we focus on a similarity measure instead as edge weigths. One extreme choice for the weight matrix is the adjacency matrix, i.e. entries 1 or 0. We will see that a somewhat natural choice is the Gaussian kernel, which in this context is also often called the *heat kernel*,

$$W_{ij} = \begin{cases} \exp\left(\frac{-\|y_i - y_j\|_2^2}{t}\right) & \text{for } (y_i, y_j) \in E, \\ 0 & \text{otherwise.} \end{cases}$$

The degree of a vertex $y_j \in Y$ is

$$D_{jj} = \sum_{k=1}^{N} W_{jk},$$

and the degree matrix is $D = \operatorname{diag}(d_1, d_N)$.

With that we define $L := D - W$ the *unnormalized graph Laplacian*. Observe that self-edges, e.g. diagonal elements of W, do not change L.

Theorem 2.27. *The matrix $L = D - W$ satisfies the following properties:*

1. *For any $f \in \mathbb{R}^N$ it holds*

$$f^T L f = \frac{1}{2} \sum_{i,j=1}^{N} w_{ij}(f_i - f_j)^2$$

2. *L is symmetric positive semi-definite*

3. *The smallest eigenvalue of L is 0, the corresponding eigenvector is the constant one vector $\mathbb{1}$*

4. *L has N non-negative, real-valued eigenvalues $0 = \lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_N$*

PROOF.
1) we have

$$\begin{aligned} f^T L f = f^T D f - f^T W f &= \sum_{i=1}^{N} f_i^2 d_i - \sum_{i,j=1}^{N} f_i f_j w_{ij} \\ &= \frac{1}{2}\left(\sum_{i=1}^{N} f_i^2 d_i - 2 \sum_{i,j=1}^{N} f_i f_j w_{ij} + \sum_{j=1}^{N} f_j^2 d_j \right) \\ &= \frac{1}{2} \sum_{i,j=1}^{N} w_{ij} (f_i - f_j)^2 \end{aligned}$$

2) D, W are symmetric, therefore L as well, positive semi-definite follows from 1) 3) obvious from the definition of D via W 4) follows from 1) to 3) ∎

Theorem 2.28. *Let $[Y, E]$ be an undirected graph with nonnegative weights. Then the multiplicity k of the eigenvalue 0 of L equals the number of connected components $A_1, \ldots, A_k$ in the graph. For L the eigenspace of eigenvalue 0 is spanned by the indicator vectors $1_{A_1}, \ldots, 1_{A_k}$ of these components.*

PROOF. See e.g. [Lux07]. ∎

Motivated by the theorem one can use graph Laplacians for clustering. Clustering aims to segment data into subsets, the so called clusters. Data objects within each cluster should be more closely related to one another than objects assigned to different clusters. The basic and very frequently used clustering algorithm is k-means.

k-MEANS ALGORITHM
Given: data set Y, number of clusters k
Output: k clusters which segment the data

pick randomly k points as cluster centers
do
  For each data point determine closest center and assign to it.
  For each cluster determine new cluster center by the coordinate-wise average of all data points assigned to a cluster.
while *assignments do change*

Spectral clustering is then an approach that uses k-means on the eigendecomposition.

SPECTRAL CLUSTERING ALGORITHM
Given: L or L_{rw}, number of clusters k
Output: clusters $A_1, \ldots, A_k$ of indices.

Compute the first k smallest eigenvectors $u_1, \ldots, u_k$ of L or L_{rw}
$U = [u_1, \ldots, u_k]$
for $i = 1, \ldots, N$ **do**
  $y_i =$ i-th row of U /* $y_i \in \mathbb{R}^k$ */
end
Cluster the points $\{y_i\}_{i=1,\ldots,N}$ with the k-means algorithm.
return *the result of the k-means algorithm*

An explanation of *spectral clustering* is based on random walks on the similarity graph. The transition probability p_{ij} of jumping in one step from the vertex y_i to the vertex y_j is proportional to the (edge) weights w_{ij} and is given by

$$p_{ij} := \frac{w_{ij}}{d_{ii}}.$$

The transition matrix $P = [p_{ij}]_{i,j=1}^N$ of the random walk is thus defined by

$$P = D^{-1}W.$$

If the graph is connected and non-bipartite, the random walk always possesses a unique stationary distribution $\pi = [\pi_1, \ldots, \pi_N]^T$, where

$$\pi_i = \frac{d_{ii}}{\text{Vol}(Y)},$$

with $\text{Vol}(Y) = \sum d_{ii}$.

We introduce a normalized graph Laplacian called the called random walk graph Laplacian:

$$L_{\text{rw}} = D^{-1}L = I - D^{-1}W = I - P$$

and observe that

$$L_{\text{rw}}u = \lambda u \iff Pu = (1-\lambda)u.$$

and

$$L_{\text{rw}}u = \lambda u \iff Lu = \lambda Du$$

Theorem 2.28 also hold for L_{rw} in the same way.

One can express many properties of a graph $[Y, E]$ with P, see e.g. [Lov93].

Note that a different normalized graph Laplacian is the symmetric one

$$L_{\text{sym}} = D^{-1/2} L D^{-1/2} = I - D^{-1/2} W D^{-1/2}.$$

One can easily see

$$L_{\text{rw}} u = \lambda u \iff L_{\text{sym}} D^{-\frac{1}{2}} u = \lambda D^{-\frac{1}{2}} u$$

It can be seen, see [Lux07] for reasoning and references, that the unnormalized graph Laplacian can lead to "bad" behavior, e.g. no convergence or completely unreliable results. Therefore one usually prefers the normalized graph Laplacian.

Clustering can be viewed as finding a graph partitioning such that the edges between different groups have very low weights and edges within a group have large weights. Or in probability, it is likely that the random walk stays inside a cluster, but unlikely that a random walk moves between clusters.

There are different ways to measure the quality of a graph partitioning. We use the so-called normalized cut

$$N_{\text{cut}}(A_1, \ldots, A_k) \coloneqq \frac{1}{2} \sum_{i=1}^{k} \frac{W(A_i, \bar{A}_i)}{\text{Vol}(A_i)},$$

where

$$W(A, B) = \sum_{\substack{i \in A \\ j \in B}} W_{ij}$$

and $\bar{A}$ is the complement of A.

Theorem 2.29. *Let $[Y, E]$ be a connected and non-bipartite graph. Assume that we run the random walk $(X_t)_{t \in \mathbb{N}}$ starting with X_0 in the stationary distribution π. For disjoint subsets $A, B \subset Y$ we denote*

$$P(B \mid A) \coloneqq P(x_1 \in B \mid x_0 \in A).$$

Then

$$N_{\text{cut}}(A, \bar{A}) = \frac{1}{2} \Big(P(\bar{A} \mid A) + P(A \mid \bar{A}) \Big).$$

PROOF.

$$\begin{aligned} P(x_0 \in A, x_1 \in B) &= \sum_{\substack{i \in A \\ j \in B}} P(x_0 = i, x_1 = j) \\ &= \sum_{\substack{i \in A \\ j \in B}} \pi_i p_{ij} \\ &= \sum_{\substack{i \in A \\ j \in B}} \frac{d_{ii}}{\mathrm{Vol}(Y)} \frac{W_{ij}}{d_{ii}} \\ &= \frac{1}{\mathrm{Vol}(Y)} \sum_{\substack{i \in A \\ j \in B}} W_{ij}. \end{aligned}$$

With this we obtain

$$\begin{aligned} P(x_1 \in B \mid x_0 \in A) &= \frac{P(x_0 \in A, x_1 \in B)}{P(x_0 \in A)} \\ &= \left(\frac{1}{\mathrm{Vol}(Y)} \sum_{\substack{i \in A \\ j \in B}} W_{ij} \right) \left(\frac{\mathrm{Vol}(A)}{\mathrm{Vol}(Y)} \right)^{-1} \\ &= \frac{\sum_{\substack{i \in A \\ j \in B}} W_{ij}}{\mathrm{Vol}(A)} \end{aligned}$$

With the definition of N_{cut} we see

$$\begin{aligned} N_{\mathrm{cut}}(A, \bar{A}) &= \frac{1}{2} \left(\frac{W(A, \bar{A})}{\mathrm{Vol}(A)} + \frac{W(\bar{A}, A)}{\mathrm{Vol}(\bar{A})} \right) \\ &= \frac{1}{2} \left(P(\bar{A} \mid A) + P(A \mid \bar{A}) \right). \end{aligned}$$

This was exactly the claim of the theorem. ∎

This tells us, that when minimizing N_{cut} we look for a cut through the graph, such that a random walk seldom transitions from A to $\bar{A}$ and vice versa. Spectral clustering can be seen as an approximation to the graph partitioning problem, see e.g. [Lux07].

Now, one can rewrite the problem of minimizing N_{cut} as (here for $k = 2$)

$$\begin{aligned} \min_A f^T L f \text{ subject to } f_i &= \begin{cases} \sqrt{\frac{\mathrm{Vol}(\bar{A})}{\mathrm{Vol}(A)}} & \text{if } y_i \in A \\ \sqrt{\frac{\mathrm{Vol}(A)}{\mathrm{Vol}(\bar{A})}} & \text{if } y_i \in \bar{A} \end{cases}, \\ Df &\perp \mathbb{1} \\ f^T D f &= \mathrm{Vol}(Y) \end{aligned}$$

This leads to a relaxed optimization problem with $f_i \in \mathbb{R}$:

$$\min_{f \in \mathbb{R}^N} f^T L f \text{ subject to } Df \perp \mathbb{1}, f^T D f = \mathrm{Vol}(Y).$$

Substituting $g := D^{1/2} f$ gives

$$\min_{g \in \mathbb{R}^N} g^T D^{-1/2} L D^{-1/2} g \text{ subject to } g \perp D^{1/2}\mathbb{1}, \|g\|^2 = \mathrm{Vol}(Y).$$

Noticing here L_{sym} and that $D^{1/2}\mathbb{1}$ is the first eigenvalue of it, and $\mathrm{Vol}(Y)$ is constant. This is in the form of the Rayleigh-Ritz theorem for eigenvalues. Therefore g is given by the second eigenvector of L_{sym} and with re-substituting $f = D^{-1/2} g$ we see that f is the second eigenvector of L_{rw} or L. This can be generalized to $k > 2$ clusters using the trace.

Laplacian Eigenmaps Now, consider a p-dim embedding X of Y, where the w_{jk} are the similarities for the data Y. Looking at Theorem 2.27 we aim to minimize

$$\frac{1}{2} \sum_{i,j=1}^{N} w_{ij} \|x_i - x_j\|^2.$$

If y_j, y_k are nearby their similarity weight w_{jk} is large, therefore $\|x_j - x_k\|$ should be kept small. For w_{jk} small, i.e. y_j, y_k dissimilar, one does not care much about the distance of the corresponding x_j, x_k.

Here a side constraint becomes $X^T D X = I$ and in the end one can see to solve for the first p smallest eigenvalues

$$LX = DX\Lambda$$

ignoring the one with eigenvalue 0.

This gives the approach called Laplacian Eigenmaps [BN03] for

$$w_{jk} = \begin{cases} \exp\left(-\frac{\|y_j - y_k\|_2^2}{4t}\right) & \text{if } (y_j, y_k) \in E \\ 0 & \text{otherwise} \end{cases}$$

and solving $(D - W)X = DX\Lambda$.

This can be interpreted as

$$x = f(y),$$

with $\|\nabla f(y)\|$ small, so that points near y will be mapped to points near $f(y)$. Note that a map that fulfils this one average can be found via

$$\arg\min_{\|f\|_{L^2(M)}=1} \int_M \|\nabla f(y)\|^2 \, \mathrm{d}y.$$

Observing Stokes' theorem

$$\int_M \langle X, \nabla f \rangle = -\int_M \operatorname{div}(X) f,$$

we obtain

$$\int_M \|\nabla f(y)\|^2 \, \mathrm{d}y = \int_M \mathcal{L}(f) f \, \mathrm{d}y$$

with $\mathcal{L}(f)$ the Laplace-Beltrami operator,

$$\mathcal{L}(f) = -\operatorname{div}(\operatorname{grad}(f)), \quad f \in C^2(M).$$

The connection between L and $\mathcal{L}$ can be further analysed [BN08].

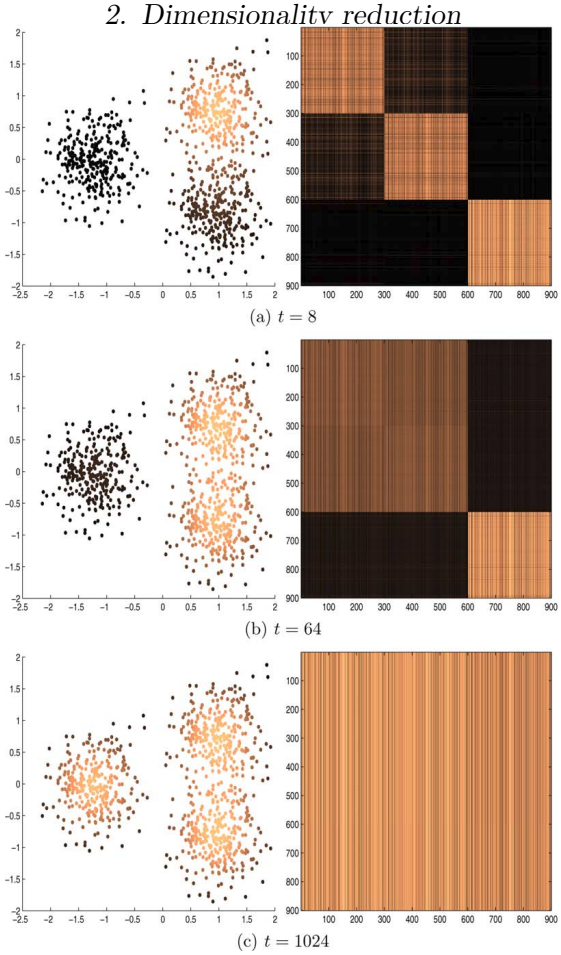
2.2.6. Diffusion Maps

We now consider the underlying Markov chain with transition matrix P and look at different times t. Then the probability of going from y_i to y_j in t (time) steps is given by the t-th power P^t of P. Looking at P^t at different steps will reveal structure of Y at different scales, as can be seen in Fig. 2.7. The weights specify the local geometry of the data and capture some geometric feature of interest. The Markov chain defines fast and slow directions of propagation, based on the weights. As one runs the walk forward, the local geometry information is propagated and accumulated.

With this view, so-called diffusion difference were defined by [Laf04; CL06]. This metric measures the similarity of two points as the probability of connecting paths between them.

$$\left(d^2_{D,t}(y_i, y_j) =\right) \quad D^2_t(y_i, y_j) = \sum_{y_k \in Y} \left| P^t_{ik} - P^t_{kj} \right|^2 .$$

Here P^t_{ik} sums the probability of all possible paths of length t between y_i and y_k. The diffusion distance is small if there are many high probability paths of length $2t$ between two points, where the path probability between y_i, y_k and y_k, y_i are roughly equal. Alternatively, we write $P^t_{ik} = p_t(y_i, y_k)$, i.e. treat p_t as a kernel function. Then y_i is close to y_j if the two kernels $p_t(y_i, \cdot)$ and $p_t(y_j, \cdot)$ are similar, i.e. the two "bumps" around y_i and y_j are similar. Since for increasing t the kernels become wider, the points y_i, y_j become more close in the diffusion distance D_t. Unlike the geodesic distance, the diffusion distance is robust to noise and topological shortcuts because it is an average over all paths connecting two points. See Fig. 2.8 for an illustration.



(a) $t = 8$

(b) $t = 64$

(c) $t = 1024$

Figure 2.7.: Heat diffusion at times $t = 8, t = 64$ and $t = 1024$ over a set containing 3 clusters. Points in Y are ordered so that the first 300 roughly correspond to the first cluster, the next 300 are in the second c luster and the last 300 in the third cluster. A graph is built with Gaussian weights and the corresponding random walk matrix P is formed. The left column represents the set, and the color encodes the intensity of diffusion from a fixed given point, that is, it corresponds to a given row of the corresponding power of P. The right column is a plot of the transition matrices P^8, P^{64} and P^{1024}. At $t = 8$, the set appears to be made of 3 distinct clusters. At $t = 64$, the two closest clusters have merged, and the data set is made of 2 clusters. Last, at $t = 1024$, all clusters have merged. Note also that P^{1024} appears to be (numerically) of rank one, as we have the approximate equality $p_{1024}(x, y) \approx \pi(y)$ for all x and y. This example illustrates, that the very notion of a cluster from a random walk point of view is a region in which the probability of escaping this region is low. This simple illustration also emphasizes the fact that, in addition to being the time parameter, t plays the role of a scale parameter. Taken form *Diffusion Maps* by [CL06].

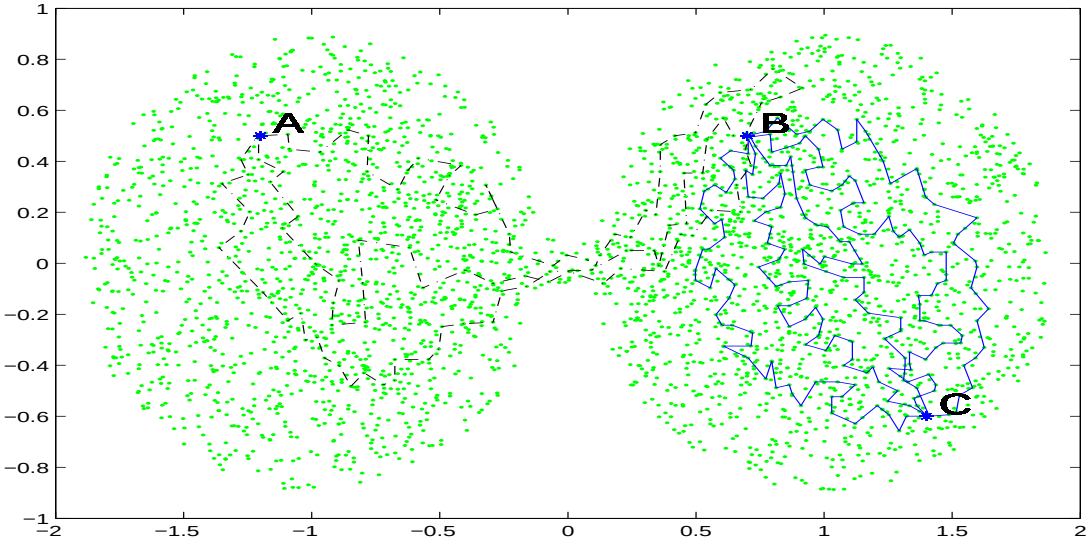


Figure 2.8.: Unlike the geodesic distance, the diffusion metric D_m is robust to short circuits. In the example above, points B and C are connected by a lot of paths and therefore are close in the sense of D_m. On the contrary, because of the presence of a bottleneck, points A and B are connected by relatively few paths, making these points very distant from each other. Image taken from [Laf04].

If P^t is symmetric, one obtains by multiplying out

$$\begin{aligned} D_t^2(y_i, y_j) &= \sum_{y_k \in Y} \Big(P_{ik}^t \underbrace{P_{ik}^t}_{=P_{ki}^t} - 2P_{ik}^t P_{kj}^t + \underbrace{P_{kj}^t}_{=P_{jk}^t} P_{kj}^t \Big) \\ &= P_{ii}^{2t} - 2P_{ij}^{2t} + P_{jj}^{2t} \\ &= p_{2t}(y_i, y_i) - 2p_{2t}(y_i, y_j) + p_{2t}(y_j, y_j). \end{aligned}$$

Again we can connect a distance measure to kernels.

Reminder: For the graph Laplacian, we had the definitions

$$\begin{aligned} L_{\text{sym}} &= I - D^{-\frac{1}{2}} W D^{-\frac{1}{2}} \\ L_{\text{rw}} &= I - D^{-1} W \end{aligned}$$

In general, to any reversible Markov process, one can associate a symmetric graph and from a symmetric graph with nonnegative weights one can construct a reversible Markov chain on the graph.

With that, we now introduce Diffusion Maps by Coifman, Lafon, 2004-2006, e.g. [Laf04; CL06]. We consider a weighted graph $[Y, E]$, where the weight function $k(x, y)$ satisfies

- $k(x, y) = k(y, x)$
- $k(x, y) \geq 0$
- k is positive semidefinite, i.e.

$$\int_Y \int_Y K(x, y) f(x) f(y) \, \mathrm{d}\mu(x) \, \mathrm{d}\mu(y) \geq 0.$$

  for all real-values bounded functions and where $\mu(y)$ is a probability measure on Y.

From this, we obtain by normalization a Markov chain as follows: For all $x \in Y$ let

$$d(x) = \int_Y k(x, y) \, \mathrm{d}\mu(y)$$

by a local measure of volume. We define

$$p(x, y) = \frac{k(x, y)}{d(x)}.$$

Surely $p(x, y) \geq 0$ holds, but p is not symmetric anymore. However, we have

$$\int_Y p(x, y) \, \mathrm{d}\mu(y) = 1.$$

So p describes a transition probability of a Markov chain. We can define a diffusion operator P, which preserves constant functions

$$Pf(x) = \int_Y p(x,y)f(y)\,\mathrm{d}\mu(y)$$

From e.g. [Chu97] one knows that there is a spectral theory for these kind of Markov chains. In particular, the integral operator $\tilde{P}$ defined on $L^2(Y)$ with the kernel

$$\tilde{p}(x,y) = p(x,y)\sqrt{\frac{d(x)}{d(y)}} = \frac{k(x,y)}{\sqrt{d(x)}\sqrt{d(y)}}$$

is symmetric. Therefore, we have a spectral decomposition of the operator $\tilde{P}$, so it holds

$$\tilde{p}(x,y) = \sum_{i\geq 0} \lambda_i \phi_i(x)\phi_i(y),$$

where under natural conditions $\tilde{p}$ has a discrete set of eigenvalues and $\lambda_0 = 1 > \lambda_1 \geq \lambda_2 \geq \ldots$. It can be seen $\phi_0 = \sqrt{\pi}$, with

$$\pi(y) = \frac{d(y)}{\int_Y d(z)\,\mathrm{d}\mu(z)}$$

the stationary distribution of the Markov chain. This implies that p satisfies

$$p(x,y) = \sum_{i\geq 0} \lambda_i \psi_i(x)\chi_i(y),$$

where

$$\psi_i(x) = \frac{\phi_i(x)}{\sqrt{\pi(x)}}, \qquad \chi_i(y) = \phi_i(y)\sqrt{\pi(y)}.$$

In particular $\psi_0(x) = 1$. The eigenvalues are as before. One can obtain analogous formula for powers P^t of P

$$p_t(x,y) = \sum_{i\geq 0} \lambda_i^t \psi_i(x)\chi_i(y) \tag{52}$$

The $\{\phi_i\}_{i\geq 0}$ form an orthonormal basis of $L^2(Y,\mathrm{d}\mu)$, consequentially the $\{\chi_i\}_{i\geq 0}$ form an orthonormal basis of of $L^2\left(Y, \frac{\mathrm{d}\mu}{\pi}\right)$.

For a fixed x, Eq. (52) can be seen as the orthogonal expansion of a function $y \mapsto P_t(x,y)$ into the orthonormal basis $\{\chi_i\}_{i\geq 0}$, where the coefficients of the expansion are the $\{\lambda_i^t\psi_i\}_{i\geq 0}$.

Definition 2.30. We define the family of *diffusion distances* $\{D_t\}_{t\in\mathbb{N}}$ by

$$\begin{aligned} D_t^2(x,y) &= \|p_t(x,\cdot) - p_t(y,\cdot)\|^2_{L^2\left(Y,\frac{\mathrm{d}\mu}{\pi}\right)} \\ &= \int_Y \left(p_t(x,u) - p_t(y,u)\right)^2 \frac{\mathrm{d}\mu(u)}{\pi(u)}. \end{aligned}$$

We define the family of *diffusion maps* $\{\Psi_t^s\}_{t\in\mathbb{N},1\le s\le N-1}$:

$$\Psi_t^s(y) \coloneqq \begin{pmatrix} \lambda_1^t\psi_1(y) \\ \lambda_2^t\psi_2(y) \\ \vdots \\ \lambda_s^t\psi_s(y) \end{pmatrix},$$

where λ_i, ψ_i come from Eq. (52), where we set $\Psi_t \coloneqq \Psi_t^{N-2}$. Each component $\lambda_i^t\psi_i$ is called *diffusion coordinate*.

We now can connect the diffusion distance with the diffusion map.

Theorem 2.31. *The diffusion distance D_t is equal to the Euclidean distance in the diffusion map space.*

$$\begin{aligned} D_t^2(x,y) &= \|\Psi_t(x) - \Psi_t(y)\|_2^2 \\ &= \sum_{i\ge 1} \lambda_i^{2t}(\psi_i(x) - \psi_i(y))^2. \end{aligned}$$

Proof. Inserting Eq. (52) into D_t gives

$$D_t^2(x,y) = \int_Y \left(\sum_{i\ge 1} \underbrace{\lambda_i^t(\psi_i(x) - \psi_i(y))}_{C(i)} \chi_i(u) \right)^2 \frac{\mathrm{d}\mu(u)}{\pi(u)}.$$

Multiplying out, exchanging integral and sum, and observing

$$\langle \chi_i, \chi_j \rangle_{L^2\left(Y,\frac{\mathrm{d}\mu}{\pi}\right)} = \delta_{ij}$$

gives

$$D_t^2(x,y) = \sum_{i\ge 1} \lambda_i^{2t} \left(\psi_i(x) - \psi_i(y)\right)^2.$$

∎

Remark. One can define this alternatively using $\tilde{p}$ and $\tilde{p}_t$ and their expansion into ϕ_i. In that formulation it is easy to see

$$\begin{aligned} \tilde{D}_t(x,y) &= \tilde{p}_{2t}(x,x) - 2\tilde{p}_{2t}(x,y) + \tilde{p}_{2t}(y,y) \\ &= \sum_{i=1} \lambda_i^{2t}(\phi_i(x) - \phi_i(y))^2. \end{aligned}$$

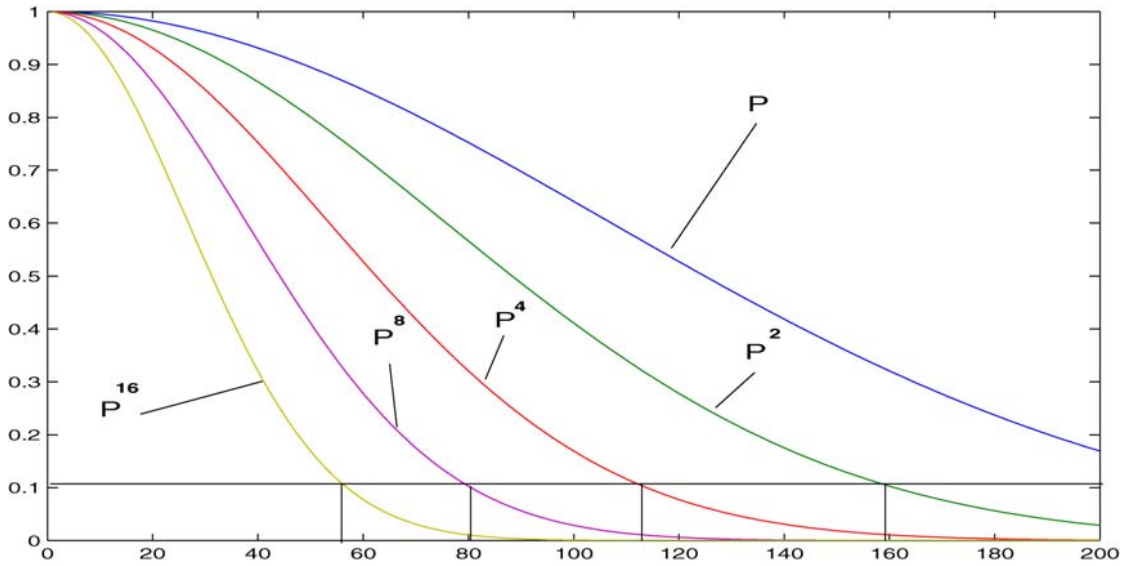


Figure 2.9.: As noted $\lambda_0 = 1 > \lambda_1 \geq \lambda_2 \geq \ldots$, so with increasing t the number of significant eigenvalues decreases for P^t, as does the numerical rank. Image from [CL06].

If we define for $\delta > 0$

$$s(\delta, t) = \max\{i \in N \mid \lambda_i^t > \delta \lambda_1^t\}$$

then we have up to relative precision δ

$$D_t(x, y) = \left(\sum_{i=1}^{s(\delta,t)} \lambda_i^{2t} (\psi_i(x) - \psi_i(y))^2 \right)^{\frac{1}{2}}.$$

Note that Coifman and Maggioni introduced a decomposition of these "probability bumps", the so-called diffusion wavelets [CM06]. To compress P^t in this view, the eigenfunctions at the beginning of the spectrum close to $\lambda_0 = 1$ have a low-frequency content, while going down the spectrum the eigenfunctions become more and more oscillatory.

Besides the manifold view, one can see the data as samples from the equilibrium distribution of stochastic dynamical systems. These two views have different implications:

- when sampled from a manifold, we aim to recover the manifold structure regardless of the data distribution
- when sampled from an equilibrium distribution, the density of the points is a quantity of interest

There is a (subtle) interplay between the statistics (in form of density) and the geometry (in form of manifold structure) of the data set.

We considered so far isotropic weights, such as

$$K(x,y) = \exp\left(-\frac{\|x-y\|^2}{t}\right)$$

in Laplacian Eigenmap (LE). We now introduce a family of anisotropic diffusion processes. We specify a parameter $\alpha \in \mathbb{R}$, which specifies the amount of influence of the density. In the end, the usual graph Laplacian normalization will be applied on a renormalized graph with non-isotropic weights.

Consider a continuous situation, i.e. $Y = M$ and we have a density $q(x)$ of the points of M. For $\alpha \in \mathbb{R}$ and a rotation invariant kernel

$$K(x,y) = h\left(\frac{d^2(x,y)}{\varepsilon}\right),$$

1. Let
$$q_K(x) = \int_Y K(x,y)q(y)\,\mathrm{d}y \tag{53}$$
and form the new kernel
$$K^{(\alpha)}(x,y) = \frac{K(x,y)}{q_K^\alpha(x)q_K^\alpha(y)}.$$

2. Let
$$d^{(\alpha)}(x) = \int_Y K^{(\alpha)}(x,y)q(y)\,\mathrm{d}y \tag{54}$$
and apply the weighted graph Laplacian normalization to $K^{(\alpha)}$ and define the anisotropic transition kernel
$$p^{(\alpha)}(x,y) = \frac{K^{(\alpha)}(x,y)}{d^{(\alpha)}(x)}.$$

Theorem 2.32. *Let $P^{(\alpha)}$ be the operator defined by*

$$P^{(\alpha)}f(x) \coloneqq \int_Y p^{(\alpha)}(x,y)f(y)q(y)\,\mathrm{d}y.$$

The eigenfunctions of $P^{(\alpha)}$ approximate the eigenfunctions of the following symmetric Schrödinger operator

$$\Delta\phi - \frac{\Delta(q^{1-\alpha})}{q^{1-\alpha}}\phi,$$

where $\phi = fq^{1-\alpha}$.

PROOF. See [CL06, Appendix B] for the precise statement and proof. ∎

Three main cases are relevant

- $\alpha = 0$ gives the normalized graph Laplacian on isotropic weights. The corresponding operator is
$$\Delta\phi - \frac{\Delta q}{q}\phi,$$
where for uniform densities the potential term vanishes. This fits with Belkin and Niyogi, who, at least implicitly, considered this case. One can note, that in practice a perfectly uniform density is difficult to achieve, or estimate. A non-constant mode in q will be amplified by $\frac{\Delta q}{q}$, therefore approximating the Laplace-Beltrami operator on the manifold is unstable. But generally, the influence of the density is maximal in this case.

- $\alpha = 1$ approximates the Laplace-Beltrami operator by taking non-uniform densities into account. Thereby, one is able to recover the Riemannian geometry of the data set, regardless of the distribution of the points.

- $\alpha = \frac{1}{2}$ relates to Fokker-Planck diffusion. The operator is
$$\Delta\phi - \frac{\Delta\sqrt{q}}{\sqrt{q}}\phi.$$
If one assumes $q = e^{-U}$ one can write this as
$$\Delta\phi - \left(\frac{\|\nabla u\|^2}{4} - \frac{\Delta u}{2}\right)\phi,$$
which leads to the forward Fokker-Planck equation
$$\frac{\partial q}{\partial t} = \nabla(\nabla q + q\nabla u),$$
where $q(y,t)$ represents the density of points at position y and time t for a dynamical system satisfies
$$\dot{y} = -\nabla u(x) + \sqrt{2}\dot{w},$$
where w is a d-dimensional Brownian motion. With this normalization one can analyze such stochastic dynamical systems. See [Nad+06] for more.

To use this for actual finite data, the integrals are approximated by finite sums, i.e. we go to the empirical mean using the data y_i which are sampled from the density $q(x)$. This gives for Eq. (53)
$$N \cdot q_K(y_i) \approx \tilde{q}_K(y_i) = \sum_{j=1}^{N} K(y_i, y_j).$$

Then we define the kernel as before:

$$\tilde{K}^{(\alpha)} = \frac{K(y_i, y_j)}{\tilde{q}_K^{\alpha}(y_i)\tilde{q}_K^{\alpha}(y_j)}.$$

Thus for Eq. (54) we get

$$N \cdot d^{(\alpha)}(y_i) \approx \tilde{d}^{(\alpha)}(y_i) = \sum_{j=1}^{N} \tilde{K}^{(\alpha)}(y_i, y_j) \left(= \sum_{j=1}^{N} \frac{K(y_i, y_j)}{\tilde{q}_K^{\alpha}(y_i)\tilde{q}_K^{\alpha}(y_j)} \right)$$

Hence, one has

$$\tilde{p}^{(\alpha)}(y_i, y_j) = \frac{\tilde{K}^{(\alpha)}(y_i, y_j)}{\tilde{d}^{(\alpha)}(y_i)}$$

Note: In the algorithm, one has

$$\hat{p}^{(\alpha)}(y_i, y_j) = \frac{\tilde{K}^{(\alpha)}(y_i, y_j)}{\sqrt{\tilde{d}^{(\alpha)}(y_i)}\sqrt{\tilde{d}^{(\alpha)}(y_j)}}$$

Due to the law of large numbers, the sums do converge to the integrals. In order to achieve a given precision with high probability, the number N of samples must grow faster than $\varepsilon^{-\frac{d}{4}-\frac{1}{2}}$ where d is the intrinsic dimension of M. Regarding noise, the approximation is valid as long as the scale parameter $\sqrt{\varepsilon}$ remains larger than the size of the perturbation.

General diffusion maps algorithm

Given: data Y, rotation invariant kernel K, $\alpha \in \mathbb{R}$, $p \in \mathbb{N}$, $t \in \mathbb{N}$

Output: embedding X

$K = [K(y_i, y_j)]_{i,j=1}^{N}$
$Q = \mathrm{diag}(K \cdot \mathbb{1})$
$K^{(\alpha)} = (Q^{\alpha})^{-1} K (Q^{\alpha})^{-1}$
$D = \mathrm{diag}(K^{(\alpha)} \cdot \mathbb{1})$
$P = D^{-\frac{1}{2}} K^{(\alpha)} D^{-\frac{1}{2}}$
$[F_p, \Lambda_p] = \mathrm{EVD}(P, p+1)$
return $X = \left[\lambda_2^t \frac{F_2}{F_1}, \lambda_3^t \frac{F_3}{F_1}, \ldots, \lambda_{p+1}^t \frac{F_{p+1}}{F_1}\right]$

Note that in

$$K(x, y) = \exp\left(\frac{-d^2(x, y)}{t}\right),$$

we can also use other distance measures, e.g. use d_G or the Dynamic Time Warping (DTW)-distance.

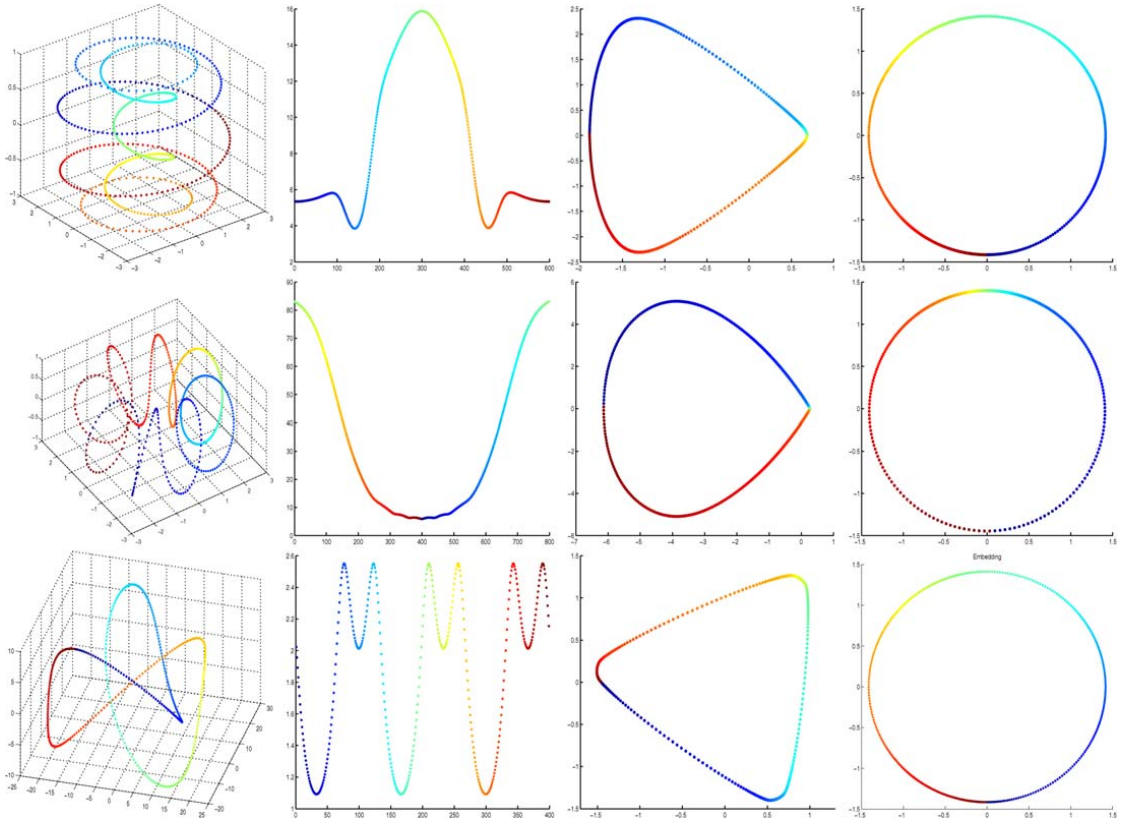


Figure 2.10.: Curves in $\mathbb{R}^3$ (two helix curves and the trefoil curve) with some nonuniform density. Although a natural ordering of these points is given by following the curve, the points were given unordered. From left to right: original curves, the densities of points on these curves, the embedding using the graph Laplacian ($\alpha = 0$), and the embedding using the Laplace-Beltrami-approximation ($\alpha = 1$). For the last case, the curve is embedded as a perfect circle and the arc length parametrization is recovered, regardless of the density of points on the original curves. The graph Laplacian tends to generate corners at regions of high density. Taken from [CL06].

2.2.7. Out of Sample Extensions

The embedding function $f : \mathbb{R}^d \to \mathbb{R}^p$ is only known point-wise, i.e. $f(y_i) = x^i$. Out of sample extensions are needed for two reasons.

1. One gets new data y and does not want to do the eigenvalue decomposition again.

2. One has too many data for an eigenvalue decomposition and wants to use a subset to compute the embedding and do 1. for the rest.

The common way to do this is the *Nyström*-extension. Denote by $v_{i,k}$ the i-th coordinate of the k-th eigenvector of $K_N = [k(y_i, y_j)]_{i,j=1}^N$ associated with the eigenvalue l_k. The Nyström formula

$$f_{k,N}(y) = \frac{\sqrt{N}}{l_k} \sum_{i=1}^{N} v_{i,k} k(y, y_i),$$

where $f_{k,N}$ is the k-th Nyström estimator with N samples.

The underlying theory of this stems from integral operators. A different view on the Nyström extension as from the standpoint of matrix completion, we can look at

$$G = \begin{pmatrix} A & B \\ B^{\mathrm{T}} & C \end{pmatrix} \begin{matrix} N \\ M \end{matrix}$$
$$\begin{matrix} N & M \end{matrix}$$

with $N \ll M$. The Nyström extension implicitly approximates C by $B^{\mathrm{T}} A^{-1} B$. The quality of approximation can be quantified by the norm of the Schur complement $\|C - B^{\mathrm{T}} A^{-1} B\|$.

To perform the Nyström extension for the studied approaches, a data-dependent kernel is used, see [Ben+04] for examples.

2.2.8. t-SNE

A motivation for t-SNE [MH08] are certain problems of distance preserving dimensionality reduction algorithms for visualization of (different) structures.

Consider a set of points that lie in high dimensions but are from

- an intrinsicially two-dimensional curved manifold

- an intrinsicially ten-dimensional curved manifold.

While one can model the small pairwise distances between data well in a two-dimensional map in the first case, it becomes problematic to model pairwise distances in two dimensions to approximate pairwise distances between points on the ten-dimensional manifold. There are two main observations, relating to the curse of dimensionality.

- in ten dimensions it is possible to have 11 points that are mutually equidistant
- volume of a sphere centered on point y_i scales as r^d

Pick any point y_i in the first situation, then all others have the same distance to that one in two dimensions, i.e. are on a circle around y_i. But while this reflects the pairwise distance to y_i, one cannot respect the other pairwise distances among the remaining points.

In the second situation, assume that some data are approximately uniformly distributed in the region around y_i on the ten-dimensional manifold. If one aims to model the distances from y_i to the other points in two-dimensions, one can observe the so-called "crowding problem": the area r^2 around x_i of the two-dimensional map is much smaller than the volume r^{d-1} around y_i. While an embedding can accommodate moderately distant points of y_i in the available area around x_i it will be overcrowded.

Therefore one can aim for relaxing the distance preservation, in particular for visualization of data in two or three dimensions.

t-SNE aims to address this by considering two distributions. One distribution p models pairwise similarities of the input data Y, that can be empirically measured, and one distribution q models pairwise similarities of the corresponding low-dimensional points in the embedding. The approach now minimizes the divergence between these two distributions to compute a low-dimensional distribution and a corresponding embedding.

Now, given a set $\{y_i\}_{i=1}^N$ and a distance function $d(y_j, y_k)$. One models the similarity of y_j to y_k by the conditional probability that y_j would pick y_k as its neighbour, where this is given by

$$p(y_k|y_j) = \frac{\exp\left(-d(y_j, y_k)/2\sigma_j^2\right)}{\sum_{l\neq j}\exp\left(-d(y_j, y_l)/2\sigma_j^2\right)}.$$

Thereby, neighbours are picked in proportion to their probability density modelled by a Gaussian centered at y_j with variance σ_j. This is at first non-symmetric, due to potentially different σ_j's. One uses instead of $p(y_k|y_j)$ directly the following symmetrisation:

$$p_{jk} \coloneqq \frac{p(y_k|y_j) + p(y_j|y_k)}{2N}$$

In view of the second observation above, points that have a moderate distance from y_i

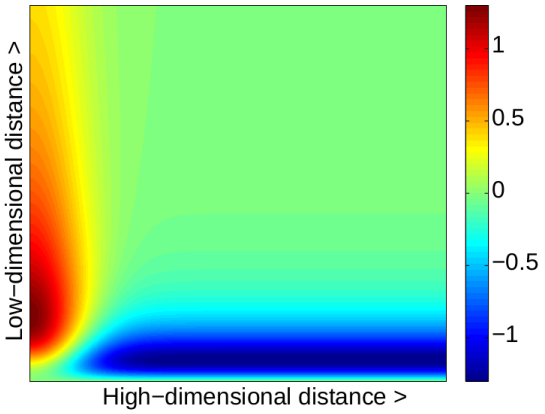


Figure 2.11.: Gradient of t-SNE as a function of the pairwise Euclidean distance between two points in the high-dimensional space and low-dimensional data representation, respectively, from [MH08]. Positive values of the gradient represent an attraction between the low-dimensional datapoints, whereas negative values represent a repulsion between the two datapoints.

will have to be placed in comparison further away in the two-dimensional map. This gives the motivation for the t in t-SNE, namely using Student's t-distribution with one degree of freedom (i.e. the Cauchy distribution) in the lower dimensional case, which is a probability distribution that has much heavier tails than a Gaussian.

$$q_{jk} = \frac{(1 + \|x_j - x_k\|^2)^{-1}}{\sum_l \sum_{m \neq l} (1 + \|x_l - x_m\|^2)^{-1}}$$

t-SNE now computes points $\{x_1, \ldots, x_N\}$ that minimize the Kullback-Leibler divergence between the distribution p for Y and q for X:

$$C(X) = KL(P|Q) = \sum_j \sum_k p_{jk} \log \frac{p_{jk}}{q_{jk}}. \tag{55}$$

Lemma 2.33. *The gradient of the Kullback-Leibler divergence between the distribution p for Y and the Student's t-based joint probability distribution q for X is:*

$$\frac{\partial C}{\partial x_j} = 4 \sum_k (p_{jk} - q_{jk})(x_j - x_k)(1 + \|x_j - x_k\|^2)^{-1}$$

Proof. Straightforward calculation, see [MH08]. ∎

The bandwidth σ_j of the Gaussian that is centered over each high-dimensional datapoint y_j is unlikely to be constant for all points, since the density of the data is likely to vary.

t-SNE has therefore a parameter, the determines σ_j such that it produces a probability distribution P_j over all of the other datapoints with a fixed perplexity that is specified by the user. The perplexity is defined as

$$Perp(P_j) = 2^{H(P_j)},$$

where $H(P_j)$ is the Shannon entropy of P_j measured in bits:

$$H_j(\sigma) = -\sum_k p(y_k|y_j) \log p(y_k|y_j).$$

Given the perplexity parameter K, the root of $H_j(\sigma) - \log K$ determines σ_j.

The embedding X is computed with a gradient descent algorithm. Since the cost function Eq. (55) is non-convex, the obtained solution depends on the initial proposal and the parameters of the descent algorithm. Therefore it may be different for different runs of t-SNE on the same data set. This is a drawback for reproducibility and interpretability of the embeddings.

The computational bottleneck is the normalization over all $N(N-1)$ pairs of points to compute p_{jk} or q_{jk}. Therefore in [Maa14] an accelerated algorithm was introduced, which computes an approximation to the gradient.

For p_{jk} one computes a sparse approximation by using the neighborhood $\mathcal{N}_j$ of the $\lfloor 3K \rfloor$ nearest neighbors of y_j for each point from Y. We redefine the pairwise similarities between the points as

$$p(y_k|y_j) = \begin{cases} \frac{\exp\left(-d(y_j,y_k)/2\sigma_j^2\right)}{\sum_{l\in\mathcal{N}_j} \exp\left(-d(y_j,y_l)/2\sigma_j^2\right)} & \text{if } k \in \mathcal{N}_j \\ 0 & \text{otherwise} \end{cases}$$

The nearest neighbor sets $\mathcal{N}_j$ can be found in $\mathcal{O}(KN \log N)$ time by using a so-called vantage-point tree on the input data [Maa14].

For the q_{jk} this is not possible, since the x_j change during the optimization. Here, a so-called Barnes-Hut algorithm can be employed [Maa14]. Observing

$$q_{jk} = \frac{(1 + \|x_j - x_k\|^2)^{-1}}{Z},$$

with

$$Z := \sum_l \sum_{m \neq l} (1 + \|x_l - x_m\|^2)^{-1},$$

we can rewrite the gradient as:

$$\begin{aligned}\frac{\partial C}{\partial x_j} &= 4\sum_k (p_{jk} - q_{jk})(x_j - x_k)(1 + \|x_j - x_k\|^2)^{-1} \\ &= 4\sum_k (p_{jk} - q_{jk})q_{jk}Z(x_j - x_k) \\ &= 4\left(\sum_k (p_{jk}q_{jk})q_{jk}Z(x_j - x_k) - \sum_k q_{jk}^2 Z(x_j - x_k)\right) \\ &= 4\left(F_{attr} + F_{rep}\right).\end{aligned}$$

Here F_{attr} denotes the sum of all attractive forces and F_{rep} denotes the sum of all repulsive forces. Computing F_{attr} can be done by summing over all non-zero elements of the sparse distribution that was constructed earlier, this can be achieved in $\mathcal{O}(KN)$.

To compute F_{rep} efficiently in $\mathcal{O}(N)$ one employs the Barnes-Hut algorithm for particle simulations in astrophysics or molecular dynamics. Consider three points x_j, x_k, and x_l with $\|x_j - x_k\| \approx \|x_j - x_l\| \gg \|x_k - x_l\|$. The contributions of x_k and x_l to F_{rep} for x_j will be very similar. This can be exploited by constructing a quad-tree (in 2D) or octtree (in 3D) for the estimate X, see Fig. 2.12, and deciding at each node, if the contribution from its cell can be used in an aggregate fashion, see Fig. 2.13. In particular, of a cell is sufficiently small and sufficiently far away from x_j, the contribution to F_{rep} will be roughly the same for all points in that cell. In such a case one uses

$$N_{cell}q_{j,cell}^2 Z(x_j - x_{cell}),$$

with x_{cell} the center of mass of that cell, N_{cell} the number of points in that cell, and

$$q_{j,cell}Z = (1 + \|x_j - x_{cell}\|^2)^{-1}.$$

Note that the tree can be built in $\mathcal{O}(N)$ time and the computation of the gradient in depth-first traversal can be performed in $\mathcal{O}(N \log N)$. See [Maa14] for details.

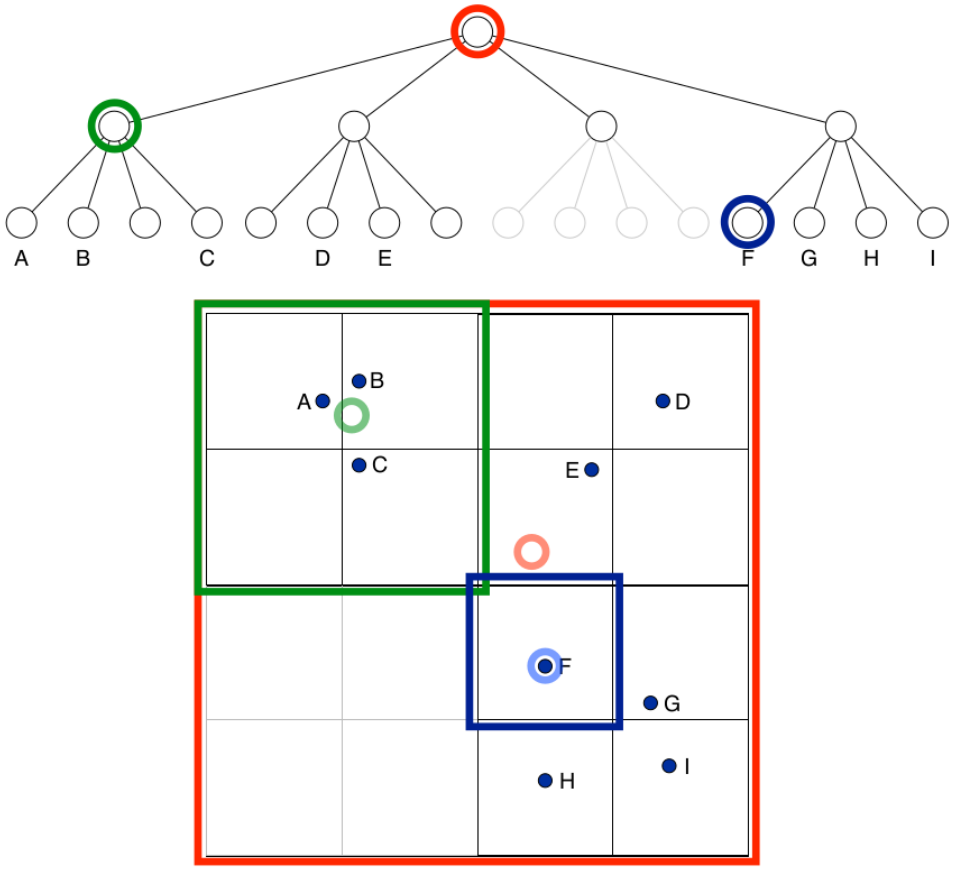


Figure 2.12.: A quad-tree for nine two-dimensional data points. Nodes in the graph correspond to square cells in the space, where for each cell we store the number of points inside and the center-of-mass of those points. From [Maa14].

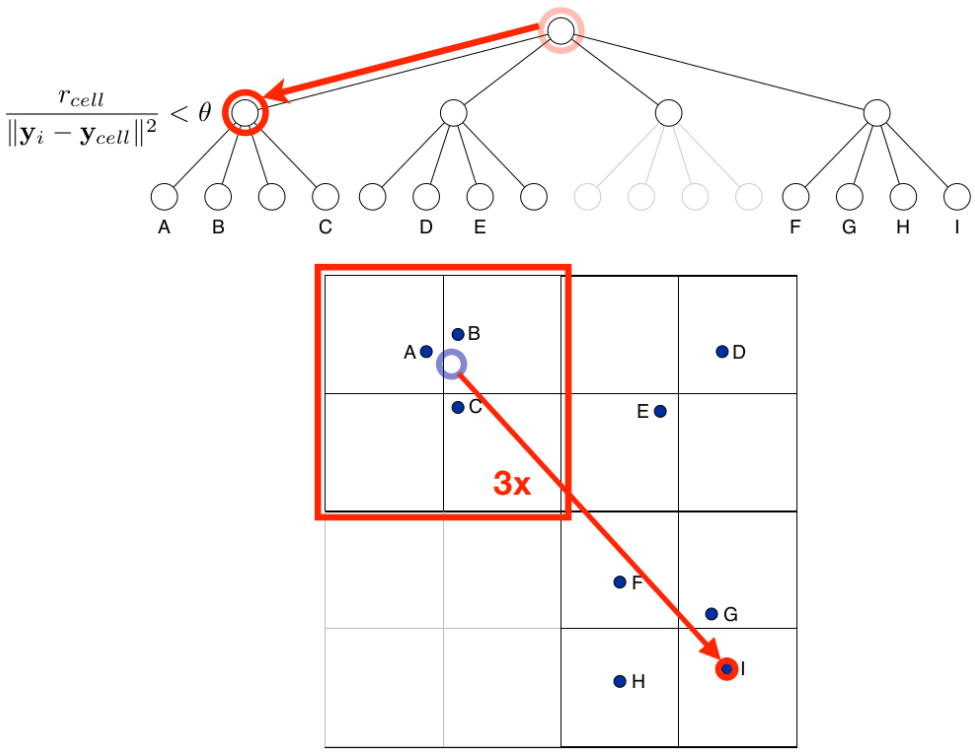


Figure 2.13.: The Barnes-Hut algorithm performs a depth-first search on the embedding quadtree, checking at every node whether or not the cell can be used as an aggregate, depending on the size of the cell and the distance to x_j. From [Maa14].

2.2.9. Autoencoder

Autoencoder (AE) are special feed-forward neural networks, which consist of so-called ***encoder*** layers, which perform the reduction step, and ***decoder*** layers, where vectors from the low-dimensional space are mapped back to the high-dimensional space. In this fashion, two parts are trained simultaneously: A dimensionality reduction architecture and a high-dimensional vector reconstruction algorithm. Most of the material for the lecture on the autoencoder is from the book [GBC16], which is available also online.

Essentially, an AE is built from two maps:

1. An encoder $E : \mathbb{R}^d \to \mathbb{R}^p$ mapping from the original data space $\mathbb{R}^d$ to the low-dimensional embedding space $\mathbb{R}^p$ with $p < d$ and

2. a decoder $D : \mathbb{R}^p \to \mathbb{R}^d$ mapping into the opposite direction.

Then, the AE f is just the concatenation of both maps

$$f := D \circ E : \mathbb{R}^d \to \mathbb{R}^d.$$

Therefore, the main question is how to build the en- and decoder.

In its most simple form, an autoencoder is just a fully-connected two-layer (feedforward) neural network, where the hidden layer contains fewer neurons than the input layer. Furthermore, the output layer has the same size as the input layer. In terms of the dimensions p and d we assigned to the encoder E and decoder D, we have d input and output neurons and $p < d$ hidden neurons. See Fig. 2.14 for $d = 6$ and $p = 2$.

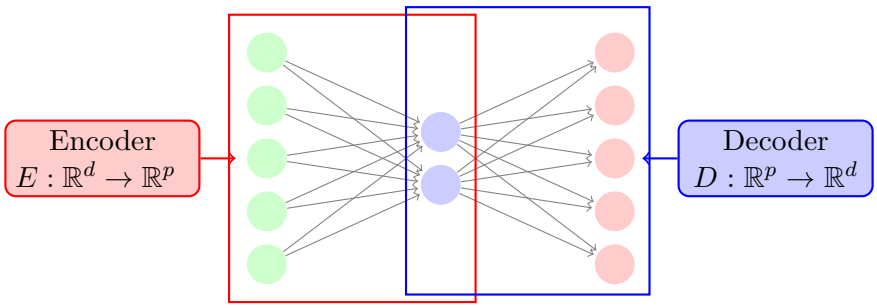


Figure 2.14.: An autoencoder with $d = 5$ and $p = 2$.

The encoder can be simply written as

$$E(y) = \phi_E(W_E y + b_E)$$

for a weight matrix $W_E \in \mathbb{R}^{p\times d}$, a bias vector $b_E \in \mathbb{R}^p$ and an activation function $\phi_E : \mathbb{R} \to \mathbb{R}$ which acts element-wise. Analogously, the decoder is

$$D(x) = \phi_D(W_D x + b_D)$$

for a weight matrix $W_D \in \mathbb{R}^{d\times p}$, a bias vector $b_D \in \mathbb{R}^d$ and an activation function $\phi_D : \mathbb{R} \to \mathbb{R}$.

Popular activation functions are

$$\phi(x) := \tanh(x) \qquad \text{or} \qquad \phi(x) = \text{sigmoid}(x) = \frac{1}{1-e^{-x}}.$$

Nowadays, a commonly used activation function is the so-called ***rectified linear unit*** (ReLU)

$$\phi(x) := \max(0, x).$$

The main idea behind autoencoders is related to the original idea behind the PCA of minimizing the (Euclidean) distance between the original vectors and their reconstructed counterpart, cf. (35). For AEs this translates to

$$f = \underset{g=D\circ E}{\arg\min} \frac{1}{N}\sum_{i=1}^{N} \|y_i - g(y_i)\|^2 = \underset{D,E}{\arg\min} \frac{1}{N}\sum_{i=1}^{N} \|y_i - D\circ E(y_i)\|^2. \tag{56}$$

In order to learn the weights and biases of E and D, we minimize the distance between data points and their reconstructions as given in (56), usually by using stochastic gradient with backpropagation. In this way, we learn the encoder and decoder simultaneously.

If $\phi_E = \phi_D = \text{id}$, there is a clear similarity to the PCA minimization problem, which searches for

$$\underset{W\in\mathbb{R}^{d\times p}, b\in\mathbb{R}^d, x_i\in\mathbb{R}^p}{\arg\min} \frac{1}{N}\sum_{i=1}^{N} \|y_i - Wx_i + b\|^2.$$

In the PCA formulation, we aim to minimize with respect to the low-dimensional data points x_i. In our autoencoder setting, they resemble the encoded inputs $E(y_i)$.

This simple architecture can be generalized by adding multiple layers to the en- and decoder networks. Note that the term *deep autoencoder* is sometimes also used to describe an architecture of several autoencoders chained together. Usually, the encoding layers shrink in size monotonically until the stage of highest compression, while the decoding layers grow in size vice versa. An example of a four-layer autoencoder with a two-layer encoder and a two-layer decoder is depicted in Fig. 2.15.

The introduction of more layers, i.e. a deeper network, greatly enhances the capabilities of autoeconders, as long as nonlinear activation functions are used. Besides fully

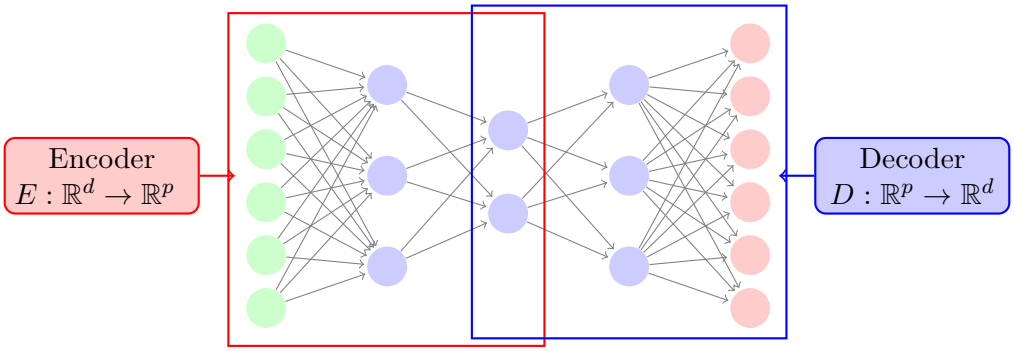


Figure 2.15.: A multilayer/deep autoencoder with $d = 6$ and $p = 2$.

connected networks, it is also possible to employ other types of networks within the AE framework. For instance, when working with pictures, it is quite natural to use so-called $2d$-convolutional layers followed by pooling layers in the encoder. The decoder is then built out of convolutional layers and upsampling layers, which enlarge the image by using bilinear interpolation for instance.

Comparing autoencoder to what we saw before for nonlinear dimensionality reduction, one key advantage would be the easy treatment of out-of-sample data, one just passed it through the network to get the latent variable. Note that if the latent variable space is high-dimensional, one case use t-SNE or other dimensionality reduction approaches to reduce the latent space to two or three dimensions for visual analysis and inspection.

On the one hand, deep autoencoder allow much flexibility in nonlinear dimensionality reduction by varying the type and number of layers in the intermediate stages. On the other hand, how to choose these is often not obvious. Further, too much capability of the decoder and encoder pair is not always useful. One can imagine an autoencoder with a one-dimensional latent space, but very expensive non-linear encoder and decoder pair, which learns 'just' a space filling curve.

A way to address this by regularization. Common are in particular three approaches, where we use a general loss function $L(y, f(y))$.

Sparse Autoencoders

With Ω a sparsity penalty on the encoding layer E one uses as the functional to minimize

$$L(y, g(y)) + \Omega(E).$$

Sparse autoencoders are typically used to learn features for another task such as clas-

sification. The sparsity penalty could be the L_1-norm or the top-K units. For specific penalties a Bayesian interpretation is possible.

Denoising Autoencoders

Here one minimizes

$$L(y, g(\tilde{y})),$$

where $\tilde{y}$ is a copy of y that has been corrupted by some form of noise. The autoencoder must now undo this corruption rather than simply copying the input.

Regularizing by Penalizing Derivatives

One can use

$$L(y, g(\tilde{y})) + \Omega(g),$$

with now

$$\Omega(g) = \lambda \sum_i \|\nabla_y (g(y))_i\|^2 \text{ or } \Omega(g) = \lambda \left\| \frac{\partial g(y)}{\partial y} \right\|_F^2.$$

This enforces some smoothness of the learned function, i.e. it does not change much when y slightly changes. This is also called contractive autoencoder.

2.2.10. Variational Autoencoder (VAE)

The lectures on the variational autoencoder are based on [KW19; Doe16].

As a different generative model successful for images are generative adversial networks (GAN).

Note that there is a recent publication questioning some of recent results and observations on VAEs [Loc+19].

2.2.11. UMAP

UMAP (Uniform Manifold Approximation and Projection) [MHM18] is constructed from a theoretical framework based in Riemannian geometry and algebraic topology. The UMAP algorithm is competitive with t-SNE for visualization quality, and arguably preserves more of the global structure with superior run time performance.

But, observe the article "Initialization is critical for preserving global data structure in both t-SNE and UMAP" [KL21] and notice that the research on and understanding of t-SNE and UMAP is still ongoing.

A. Numerical Linear Algebra

Definition A.1 (Singular Value Decomposition). If $A \in \mathbb{R}^{m\times n}$ is a real matrix, there exist two orthogonal matrices

$$U = [u_1, \dots, u_m] \in \mathbb{R}^{m\times m}, \qquad V = [v_1, \dots, v_n] \in \mathbb{R}^{n\times n}$$

such that

$$A = U\Sigma V^T, \qquad \text{with } \Sigma = \operatorname{diag}(\sigma_1, \dots, \sigma_p) \in \mathbb{R}^{m\times n} \tag{57}$$

and $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_p$, for $p = \min(m, n)$.

The σ_i are called singular values, the u_i are called left singular vectors, and the v_i are right singular vectors.

Theorem A.2 (Schmidt-Eckart-Young). *Given a matrix $A \in \mathbb{R}^{m\times n}$ of rank r, the matrix*

$$A_k := \sum_{i=1}^{k} \sigma_i u_i v_i^T, \qquad 0 \leq k \leq r, \tag{58}$$

satisfies the optimality property

$$\|A - A_k\|_F = \min_{\substack{B\in R^{m\times n} \\ \operatorname{rank}(B)\leq k}} \|A - B\|_F = \sqrt{\sum_{i=k+1}^{r} \sigma_i^2}. \tag{59}$$

A similar result holds by considering the 2-norm instead of the Frobenius norm: for any $0 < k < r$, the matrix A_k defined in Eq. (57) is also such that

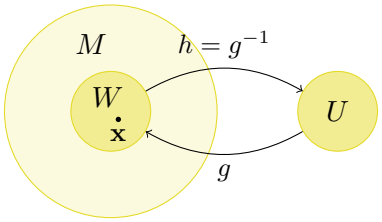
$$\|A - A_k\|_2 = \min_{B\in R^{m\times n}, \operatorname{rank}(B)\leq k} \|A - B\|_2 = \sigma_k + 1. \tag{60}$$

B. Differential Geometry

Definition B.1 (Smooth Mapping). Let $X \subset \mathbb{R}^k$ and $Y \subset \mathbb{R}^l$ be non-empty sets. A mapping $f : X \to Y$ is called *smooth mapping* if "all partial derivatives exist and are continuous".

Definition B.2 (Diffeomorphism). Let $X \subset \mathbb{R}^k$ and $Y \subset \mathbb{R}^l$ be non-empty sets. If a bijection $f : X \to Y$ and its inverse $f^{-1} : Y \to X$ both are smooth, then f is called a *diffeomorphism* (differentiable homeomorphism) and X is said to be diffeomorph to Y.

Definition B.3 (Manifold). Let $M \subset \mathbb{R}^n$ be non-empty. Assume that for all $\mathbf{x} \in M$ there exists an open set $W \subset M$, with $x \in \operatorname{int} W$, s.t. W is diffeomorph to an open set $U \subset \mathbb{R}^k$. Then M is called a k-dimensional smooth *manifold*.
A diffeomorphism $g : U \to W$ is called a parametrization of W, its inverse $h(= g^{-1}) : W \to U$ is called a coordinate mapping. The pair (W, h) is called a (local) coordinate system, or a chart, of M.



An $\mathbf{x} \in W$ has coordinates $h(\mathbf{x}) = [h_1(\mathbf{x}), h_2(\mathbf{x}), \ldots, h_m(\mathbf{x})]$. We constrain ourselves to simple manifolds, i.e. only one coordinate system.

Definition B.4. A function f on an open set $V \subset M$ is called differentiable (smooth) if $f \circ h^{-1}$ is differentiable on $h(V \cap M) = h(V)$ for the coordinate system (M, h). At $\mathbf{x} \in V$ the derivative of f is the linear mapping defined by

$$\mathrm{d}f_{\mathbf{x}} = \lim_{t \to 0} \frac{f(\mathbf{x} + t\mathbf{h}) - f(\mathbf{x})}{t}, \qquad \mathbf{h} \in \mathbb{R}^k,$$

which can be represented by the matrix

$$\mathrm{d}f_{\mathbf{x}} = \left[\frac{\partial f_i(\mathbf{x})}{\partial x_j} \right]_{i,j=1}^{l,k}.$$

Definition B.5 (Tangent space of a point on a manifold). Let $M \subset \mathbb{R}^m$ be a k-dimensional manifold, $U \subset \mathbb{R}^k$ be an open set, and $g : U \to M$ be a parametrization of the neighbourhood $g(U) \subset M$. Assume $\mathbf{p} \in M$, $\mathbf{u} \in U$ and $g(\mathbf{u}) = \mathbf{p}$. The image of the linear transformation $\mathrm{d}g_{\mathbf{u}}$ is called the *tangent space* of M at $\mathbf{p}$.

$$T_{\mathbf{p}}M := \mathrm{d}g_{\mathbf{u}}(\mathbb{R}^k).$$

A vector in $T_{\mathbf{p}}M$ is called a *tangent vector*.

Note that the set $\left\{ \frac{\partial g}{\partial u_1}, \ldots, \frac{\partial g}{\partial u_k} \right\}$ is a basis for $T_{\mathbf{p}}M$ and can be represented by the basis matrix

$$\frac{\partial g}{\partial \mathbf{u}} := \left[\frac{\partial g_i}{\partial u_j} \right]_{i,j=1}^{m,k}.$$

A tangent vector at $\mathbf{p}$ has the form

$$\mathbf{x}_{\mathbf{p}} = \sum_{i=1}^{k} \alpha_i \frac{\partial g}{\partial u_i} = \frac{\partial g}{\partial u} \vec{\alpha}.$$

Now assume f is a function on M, which has a smooth extension on an open set O of $\mathbb{R}^m$, s.t. $M \subset O$. The composite $f \circ g$ is a function on $\mathbb{R}^k$, but f can also be represented as a function of the coordinates $\mathbf{u}$, say $F(\mathbf{u})$, of course $f \circ g(\mathbf{u}) = F(\mathbf{u})$. Then the directional derivatives of F in the direction $\vec{\alpha} \in \mathbb{R}^k$ can be given by

$$\frac{\partial f \circ g}{\partial \vec{\alpha}} = \mathrm{d}f_{\mathbf{p}} \frac{\partial g}{\partial \mathbf{u}} \vec{\alpha}.$$

Riemannian metrics

Now let M be a k-dimensional manifold and $T_{\mathbf{p}}M$ the tangent space at $\mathbf{p} \in M$. g is a parametrization of M and (W, h) is the coordinate system on M. g defines a basis for

$T_{\mathbf{p}}M$ as before, represented by $\frac{\partial g}{\partial \mathbf{u}}$. The metric $\mathbf{G}_g(\mathbf{p})$ is defined by

$$\mathbf{G}_g(\mathbf{p}) = \left(\frac{\partial g}{\partial \mathbf{u}}\right)^T \left(\frac{\partial g}{\partial \mathbf{u}}\right), \qquad \mathbf{p} \in M.$$

$\mathbf{G}_g(\mathbf{p})$ is a positive semidefinite matrix, it is called *Riemann metric on* M. Take two tangent vectors $\mathbf{x}, \mathbf{y} \in T_{\mathbf{p}}M$, $\mathbf{x} = \frac{\partial g}{\partial \mathbf{u}}\vec{x}$, $\mathbf{y} = \frac{\partial g}{\partial \mathbf{u}}\vec{y}$, then their inner product is defined by

$$\langle \mathbf{x}, \mathbf{y} \rangle_{\mathbf{p}} \coloneqq \vec{x}^T \mathbf{G}(\mathbf{p}) \vec{y} \tag{61}$$

and their norm by

$$\|\mathbf{x}\|_{\mathbf{p}} \coloneqq \vec{x}^T \mathbf{G}(\mathbf{p}) \vec{x}. \tag{62}$$

Definition B.6. Let M be a k-dimensional manifold with the metric $\mathbf{G}_g(\mathbf{p})$ and defines for each pair of tangent vectors Appendix B the inner product. Then M is called a *Riemannian manifold* and is denoted by (M, g) or $(M, \mathbf{G})$.

Often one chooses isometric parametrizations so that $\mathbf{G}_g(\mathbf{p}) = \mathbf{I}_d$.

Definition B.7 (Geodesic Distance). Let M be a connected Riemannian manifold and $\gamma \subset M$ is a curve on M with the parametric equation

$$\underline{\gamma} = [\gamma_1(t), \ldots, \gamma_m(t)]^T, \qquad t \in [a, b].$$

Then the length of the curve $\underline{\gamma}$ is defined by

$$\left\|\underline{\gamma}\right\| = \int_a^b \|\dot{\gamma}(t)\| \, \mathrm{d}t.$$

The *geodesic distance* d_M is defined by

$$d_M(\mathbf{x}, \mathbf{y}) = \inf_{\substack{\gamma \subset M, \\ \gamma(a)=\mathbf{x}, \\ \gamma(b)=\mathbf{y}}} \|\gamma\|.$$

C. Neighbourhood Graph

Definition C.1 (Undirected Graph). Let V be a given finite set. An *undirected (or simple) graph* G is an ordered pair $[V, E]$ so that the elements of E are 2-element subsets of V. The elements of V are called vertices, nodes or points of the graph, and the elements of E are called edges or lines of G.
The number of edges that connect a vertex $\mathbf{x}$ is called degree. The set of all vertices that are connected to a vertex $\mathbf{x}$ is called the neighbourhood $N_G(\mathbf{x})$ of $\mathbf{x}$. We use $\mathbf{x} \notin N_G(\mathbf{x})$.

We can use an adjacency matrix to represent the edge set in a graph $G = [V, E]$

$$A_{ij} = \begin{cases} 1, & \text{if } (\mathbf{v}_i, \mathbf{v}_j) \in E \text{ or } (\mathbf{v}_j, \mathbf{v}_i) \in E \\ 0, & \text{else.} \end{cases}$$

A weight matrix is a generalization where each edge has a non-zero weight.

ε-neighbourhood

Let Y be any data set. $\mathbf{y}^i$ and $\mathbf{y}^j$ are connected if and only if $d_2(\mathbf{y}^i, \mathbf{y}^j) \leq \varepsilon$.

- Distances between all connected points are of roughly similar size.
- If ε is small, weighting the edges does not "really" absorb more information about the data into the graph.

This graph is also called r-ball graph graph, for $\epsilon = r$ in the above.

k-nearest neighbours graph

Vertex $\mathbf{y}^i$ is connected to $\mathbf{y}^j$ if $\mathbf{y}^j$ is among the k-nearest neighbours of $\mathbf{y}^i$ (w.r.t. d_2).

- Ignore directions of edges, if $(\mathbf{y}^i, \mathbf{y}^j) \in E$ also add $(\mathbf{y}^j, \mathbf{y}^i)$. (k-nearest neighbourhood relationship is not symmetric.)
- The alternative would be: Only connect $\mathbf{y}^i$ and $\mathbf{y}^j$ if k-nearest neighbours relation holds in both directions to begin with.

Here weighting the edges by the similarity of the edges can be useful.

Adaptive neighbourhood graph
For a constant $c > 1$ we define the c-neighbourhood of a point $\mathbf{y}^i \in Y$ by

$$N_c(\mathbf{y}^i) = \left\{ \mathbf{y}^j \in Y \setminus \{\mathbf{y}^i\} : d_2(\mathbf{y}^i, \mathbf{y}^j) \leq c \min_{\mathbf{y}^* \in Y \setminus \{\mathbf{y}^i\}} d_2(\mathbf{y}^i, \mathbf{y}^*) \right\}.$$

Then two vertices $\mathbf{y}^i, \mathbf{y}^j$ are connected if either $\mathbf{y}^i \in N_c(\mathbf{y}^j)$ or $\mathbf{y}^j \in N_c(\mathbf{y}^i)$. Weighting can again be useful.

Fully connected graph
All point pairs with positive similarity are connected and weighted by similarity, e.g. by Gaussian similarity function $S(\mathbf{y}^i, \mathbf{y}^j) = \exp(-0.5 d_2(\mathbf{y}^i, \mathbf{y}^j)^2/\sigma^2)$, where σ controls the width of the neighbourhood.

D. Semidefinite Programming

Let $\mathbf{P}$ and $\mathbf{Q}$ be two $N \times N$ symmetric matrices which are considered as vectors in $\mathbb{R}^{N^2}$. We define their inner product by

$$\mathbf{P} \cdot \mathbf{Q} = \sum_{i=1}^{N} \sum_{j=1}^{N} P_{i,j} Q_{i,j}.$$

Thus, $\mathbf{P}$ is a functional on the Euclidean space $\mathbb{R}^{N^2}$. Let $\mathbf{C}_i,\ 1 \leq i \leq m$, be m symmetric matrices and $\mathbf{b} = [b_1, \ldots, b_m]^T \in \mathbb{R}^m$. The following optimization is called a *semidefinite programming* (SDP) problem

$$\begin{array}{lll} \text{minimize} & \mathbf{P} \cdot \mathbf{Q} & \\ \text{s.t.} & \mathbf{C}_i \cdot \mathbf{Q} = b_i & 1 \leq i \leq m \\ & \mathbf{Q} \succeq 0. & \end{array}$$

Observe, that the collection of psd matrices is a convex set in $\mathbb{R}^{N^2}$ and each constraint is a hyperplane in $\mathbb{R}^{N^2}$. Assume that $\{\mathbf{C}_1, \ldots, \mathbf{C}_m\}$ forms a linearly independent set in $\mathbb{R}^{N^2}$. Then the intersection of the hyperplanes $\mathbf{C}_i \cdot \mathbf{Q} = b_i$, $1 \leq i \leq m$ denoted by S_m is an $(N^2 - m)$-dimensional affine space.
The SDP-problem can be seen as finding the minimal value of the linear functional $\mathbf{P} \cdot \mathbf{Q}$ on the intersection of the convex set $\mathbf{Q} \succeq 0$ and the affine space S_m. It can be that no psd matrix fulfills the constraints. If the set of matrices which fulfill the side constraints is non-empty we call the SDP feasible.

Acronyms

AE Autoencoder

CMDS classical multidimensional scaling

cpsd conditionally positive semidefinite

CV cross-validation

DTW Dynamic Time Warping

ECG electrocardiography

EDM Euclidean distance matrix

EEG electroencephalography

EVD eigenvalue decomposition

i.i.d. independent, identically distributed

k-**NN** k-Nearest Neighbour

LE Laplacian Eigenmap

MDS multidimensional scaling

MVU maximum variance unfolding

ONB orthonormal basis

PCA principal component analysis

PDE partial differential equation

psd positive semidefinite

RBF radial basis function

RKHS reproducing kernel Hilbert space

R&D research & development

SDP semidefinite programming

SVD singular value decomposition

SVM support vector machine

Index of key definitions

Bibliography

[AJP18] Ery Arias-Castro, Adel Javanmard, and Bruno Pelletier. "Perturbation Bounds for Procrustes, Classical Scaling, and Trilateration, with Applications to Manifold Learning." In: (Oct. 2018), pp. 1–24. arXiv: 1810.09569.

[AJP20] Ery Arias-Castro, Adel Javanmard, and Bruno Pelletier. "Perturbation Bounds for Procrustes, Classical Scaling, and Trilateration, with Applications to Manifold Learning." In: *Journal of Machine Learning Research* 21.15 (2020), pp. 1–37. URL: http://jmlr.org/papers/v21/18-720.html.

[AL19] Ery Arias-Castro and Thibaut Le Gouic. "Unconstrained and Curvature-Constrained Shortest-Path Distances and Their Approximation." In: *Discrete and Computational Geometry* (2019). ISSN: 14320444. DOI: 10.1007/s00454-019-00060-7.

[BCR84] Christian Berg, Jens Peter Reus Christensen, and Paul Ressel. *Harmonic Analysis on Semigroups*. Vol. 100. Graduate Texts in Mathematics. New York, NY: Springer New York, 1984. DOI: 10.1007/978-1-4612-1128-0.

[Ben+04] Yoshua Bengio, Olivier Delalleau, Nicolas Le Roux, Jean-François Paiement, Pascal Vincentand, and Marie Ouimet. "Learning Eigenfunctions Links Spectral Embedding and Kernel {PCA}." In: *Neural Comp.* 16.10 (2004), pp. 2197–2219.

[Ber+00] Mira Bernstein, Vin de Silva, John C. Langford, and Joshua B. Tenenbaum. "Graph approximations to geodesics on embedded manifolds." Dec. 2000. URL: http://isomap.stanford.edu/BdSLT.pdf.

[BG05] I. Borg and P.J.F. Groenen. *Modern Multidimensional Scaling: Theory and Applications*. Springer Series in Statistics. Springer New York, 2005.

[Bis06] Christopher M. Bishop. *Pattern recognition and machine learning*. Springer, 2006.

[BN03] Mikhail Belkin and Partha Niyogi. "Laplacian Eigenmaps for Dimensionality Reduction and Data Representation." In: *Neural Computation* 15.6 (2003), pp. 1373–1396. ISSN: 0899-7667. DOI: 10.1162/089976603321780317. arXiv: arXiv:1011.1669v3.

[BN08] Mikhail Belkin and Partha Niyogi. "Towards a theoretical foundation for Laplacian-based manifold methods." In: *Journal of Computer and System Sciences* 74.8 (2008), pp. 1289–1308. DOI: 10.1016/j.jcss.2007.08.006.

[Boh+13] Bastian Bohn, Jochen Garcke, Rodrigo Iza-Teran, Alexander Paprotny, Benjamin Peherstorfer, Ulf Schepsmeier, and Clemens-August Thole. "Analysis of Car Crash Simulation Data with Nonlinear Machine Learning Methods." In: *Procedia Computer Science, Proceedings of the ICCS 2013, Barcelona.* Vol. 18. 0. 2013, pp. 621–630. DOI: 10.1016/j.procs.2013.05.226.

[Chu97] Fan R. K. Chung. *Spectral graph theory.* Vol. 92. CBMS Regional Conference Series in Mathematics. Published for the Conference Board of the Mathematical Sciences, Washington, DC; by the American Mathematical Society, Providence, RI, 1997, pp. xii+207. ISBN: 0-8218-0315-8.

[CL06] Ronald R. Coifman and Stéphane Lafon. "Diffusion maps." In: *Applied and Computational Harmonic Analysis* 21.1 (July 2006), pp. 5–30. ISSN: 10635203. DOI: 10.1016/j.acha.2006.04.006.

[CM06] Ronald R. Coifman and M Maggioni. "Diffusion Wavelets." In: *Appl. Comput. Harmon. Anal.* 21.1 (2006), pp. 53–94.

[Doe16] Carl Doersch. "Tutorial on Variational Autoencoders." In: (June 2016), pp. 1–23. arXiv: 1606.05908. URL: http://arxiv.org/abs/1606.05908.

[Dub57] L. E. Dubins. "On Curves of Minimal Length with a Constraint on Average Curvature, and with Prescribed Initial and Terminal Positions and Tangents." In: *American Journal of Mathematics* 79.3 (July 1957), p. 497. ISSN: 00029327. DOI: 10.2307/2372560.

[Fab+17] Felix A. Faber et al. "Prediction Errors of Molecular Machine Learning Models Lower than Hybrid DFT Error." In: *Journal of Chemical Theory and Computation* 13.11 (2017), pp. 5255–5264. ISSN: 15499626. DOI: 10.1021/acs.jctc.7b00577.

[Fed59] Herbert Federer. "Curvature measures." In: *Transactions of the American Mathematical Society* 93.3 (Mar. 1959), pp. 418–418. ISSN: 0002-9947. DOI: 10.1090/S0002-9947-1959-0110078-1.

[GBC16] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning.* http://www.deeplearningbook.org. MIT Press, 2016.

[GKQ16] Charles Gawad, Winston Koh, and Stephen R. Quake. "Single-cell genome sequencing: Current state of the science." In: *Nature Reviews Genetics* 17.3 (2016), pp. 175–188. ISSN: 14710064. DOI: 10.1038/nrg.2015.16. arXiv: 1505.02710.

[Hoc89] Harry Hochstadt. *Integral equations.* Wiley Classics Library. Reprint of the 1973 original, A Wiley-Interscience Publication. John Wiley & Sons, Inc., New York, 1989, pp. x+282. ISBN: 0-471-50404-1.

[HTF09] T Hastie, R Tibshirani, and J Friedman. *The Elements of Statistical Learning, Second Edition.* Springer, 2009.

[IG19] Rodrigo Iza-Teran and Jochen Garcke. "A Geometrical Method for Low-Dimensional Representations of Simulations." In: *SIAM/ASA Journal on Uncertainty Quantification* (2019). accepted.

[Jol02] I.T. Jolliffe. *Principal Component Analysis*. Springer Series in Statistics. Springer, 2002.

[KJM20] Nils M Kriege, Fredrik D Johansson, and Christopher Morris. "A survey on graph kernels." In: *Applied Network Science* 5.1 (Dec. 2020), p. 6. ISSN: 2364-8228. DOI: 10.1007/s41109-019-0195-3. URL: https://appliednetsci.springeropen.com/articles/10.1007/s41109-019-0195-3.

[KL21] Dmitry Kobak and George C. Linderman. "Initialization is critical for preserving global data structure in both t-SNE and UMAP." In: *Nature Biotechnology* 39.2 (Feb. 2021), pp. 156–157. ISSN: 1087-0156. DOI: 10.1038/s41587-020-00809-z. URL: http://dx.doi.org/10.1038/s41587-020-00809-z%20http://www.nature.com/articles/s41587-020-00809-z.

[KW19] Diederik P. Kingma and Max Welling. "An Introduction to Variational Autoencoders." In: *Foundations and Trends® in Machine Learning* 12.4 (2019), pp. 307–392. ISSN: 1935-8237. DOI: 10.1561/2200000056. arXiv: 1906.02691.

[Laf04] Stéphane Lafon. "Diffusion Maps and Geometric Harmonics." PhD thesis. Yale University, 2004.

[Loc+19] Francesco Locatello, Stefan Bauer, Mario Lucic, Gunnar Rätsch, Sylvain Gelly, Bernhard Schölkopf, and Olivier Bachem. "Challenging Common Assumptions in the Unsupervised Learning of Disentangled Representations." In: *ICML*. 2019. arXiv: 1811.12359. URL: http://arxiv.org/abs/1811.12359.

[Lov93] László Lovász. "Random Walks on Graphs: A Survey." In: *Combinatorics, Paul Erdös is Eighty*. János Bolyai Math. Soc., 1993, pp. 353–397.

[Lux07] Ulrike von Luxburg. "A tutorial on spectral clustering." In: *Stat. Comput.* 17.4 (2007), pp. 395–416. ISSN: 0960-3174. DOI: 10.1007/s11222-007-9033-z.

[LV07] John A. Lee and Michel Verleysen. *Nonlinear dimensionality reduction*. Information Science and Statistics. Springer, New York, 2007, pp. xviii+308. ISBN: 978-0-387-39350-6. DOI: 10.1007/978-0-387-39351-3.

[Maa14] Laurens Van Der Maaten. "Accelerating t-SNE using Tree-Based Algorithms." In: *Journal of Machine Learning Research* 15 (2014), pp. 1–21.

[MH08] Laurens Van Der Maaten and Geoffrey Hinton. "Visualizing data using t-SNE." In: *Journal of Machine Learning Research* 9 (2008), pp. 2579–2605.

[MHM18] Leland McInnes, John Healy, and James Melville. "UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction." In: (Feb. 2018), pp. 1–18. arXiv: 1802.03426. URL: http://arxiv.org/abs/1802.03426.

[Mic86] Charles A. Micchelli. "Interpolation of scattered data: Distance matrices and conditionally positive definite functions." In: *Constructive Approximation* 2.1 (1986), pp. 11–22. DOI: 10.1007/BF01893414.

[Nad+06] Boaz Nadler, Stéphane Lafon, Ronald R. Coifman, and Ioannis G. Kevrekidis. "Difusion maps, spectral clustering and reaction coordinates of dynamical systems." In: *Appl. Comput. Harmon. Anal.* 21.1 (2006), pp. 113–127. ISSN: 1063-5203. DOI: 10.1016/j.acha.2005.07.004.

[RW06] Carl Edward Rasmussen and Christopher K I Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2006.

[SC08] Ingo Steinwart and Andreas Christmann. *Support Vector Machines*. Information Science and Statistics. New York, NY: Springer New York, 2008.

[Sch11] Robert Schaback. "Kernel-Based Meshless Methods." Lecture Notes. 2011. URL: http://num.math.uni-goettingen.de/schaback/teaching/AV_2.pdf.

[SS02] Bernhard Schölkopf and Alex Smola. *Learning with Kernels*. MIT Press, 2002.

[SSM98] Bernhard Schölkopf, Alex Smola, and Klaus-Robert Müller. "Nonlinear Component Analysis as a Kernel Eigenvalue Problem." In: *Neural Computation* 10.5 (1998), pp. 1299–1319. DOI: 10.1162/089976698300017467. URL: http://www.mitpressjournals.org/doi/abs/10.1162/089976698300017467.

[TSL00] J.B. Tenenbaum, V. de Silva, and J.C. Langford. "A Global Geometric Framework for Nonlinear Dimensionality Reduction." In: *Science* 290.5500 (Dec. 2000), pp. 2319–2323. ISSN: 1095-9203. DOI: 10.1126/science.290.5500.2319. URL: http://dx.doi.org/10.1126/science.290.5500.2319.

[Vis+10] S.V. N. Vishwanathan, Nicol N. Schraudolph, Risi Kondor, and Karsten M. Borgwardt. "Graph kernels." In: *Journal of Machine Learning Research* 11 (2010), pp. 1201–1242.

[Wen05] Holger Wendland. *Scattered Data Approximation*. Cambridge University Press, 2005. DOI: 10.1017/CBO9780511617539.

[WRY16] Allon Wagner, Aviv Regev, and Nir Yosef. "Revealing the vectors of cellular identity with single-cell genomics." In: *Nature Biotechnology* 34.11 (2016), pp. 1145–1160. ISSN: 15461696. DOI: 10.1038/nbt.3711. arXiv: 15334406.

[Wu92] Zongmin Wu. "Hermite-Birkhoff interpolation of scattered data by radial basis functions." In: *Approximation Theory and its Applications* 8.2 (1992), pp. 1–10. ISSN: 1573-8175. DOI: 10.1007/BF02836101.