Eötvös Loránd University

Faculty of Science

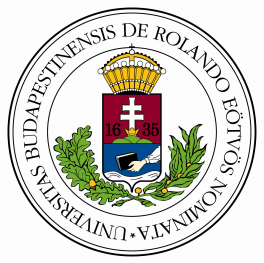
Gergely Márton Szathmári

Kernel Ridge Regression in image reconstruction

BSc Thesis

Mathematics

Supervisor:
Balázs Csanád Csáji
Department of Probability Theory and Statistics



Budapest, 2025

Acknowledgements

I would like to thank to my thesis supervisor, Balázs Csanád Csáji, for introducing me the topic and pointing out new perspectives for this thesis. I would also like to express my gratitude towards my family for their patience and constant support.

Contents

Chapter 1

Introduction

In the last few decades machine learning has become one of the most rapidly developing fields in science. Its main focus is to extract patterns from the observations to make accurate and robust predictions on previously unseen data. The learning process relies heavily on statistics and mathematical optimization.

The aim of this thesis is to review the theoretical foundations of kernel ridge regression. As we will see we will have to solve a regression problem. In most cases the simple linear regression is not sufficient to make proper predictions since the relation of the data points is nonlinear. When this is the situation we have to take a more sophisticated approach. Kernel methods enable us to transform the data into high, even infinite dimensional, spaces, where we can learn complicated relationships, and still, the computations the process requires remain manageable.

Beyond the theoretical foundations, I will demonstrate the practical applications of kernel regression in image denoising, missing pixel estimation and super-resolution.

Chapter 2

Regression

Let $\mathcal{X}$ be a non-empty set and $n \in \mathbb{N}$. In a regression problem, we are given a $(x_1, y_1), \ldots, (x_n, y_n) \in \mathcal{X} \times \mathcal{Y}$ training set ($\mathcal{Y}$ is typically assumed to be $\mathbb{R}$). Furthermore we assume that there exists a function $f^* : \mathcal{X} \to \mathcal{Y}$ such that $\forall i : y_i = f^*(x_i) + \xi_i$, where ξ_i is a 0 mean random variable for all i. By including such random variables we may represent noise in the input or variability of the target values, which are not due to the input. Our goal is to find a function $\hat{f} : \mathcal{X} \to \mathcal{Y}$ that not only performs well on the training data but also makes reasonable predictions on unseen points of $\mathcal{X}$. To assess the performance of a function at a given training point we introduce the notion of the loss function.

Definition 2.0.1 *(1, Definition 3.1)*
Denote by $(x, y, f(x)) \in \mathcal{X} \times \mathcal{Y} \times \mathcal{Y}$ the triplet consisting of a pattern x, an observation y and a prediction $f(x)$. Then the map $c : \mathcal{X} \times \mathcal{Y} \times \mathcal{Y} \to [0, \infty)$ with the property $c(x, y, y) = 0$ for all $x \in \mathcal{X}$ and $y \in \mathcal{Y}$ is called a ***loss function****.*

Remark 2.0.2 *The loss function may depend on the input x (for instance, in image reconstruction we may require higher accuracy in the center of the image than at the edges). However for the sake of simplicity it is often assumed that the loss function is location-independent and it is generally a function of $y - f(x)$.*

Remark 2.0.3 *For practical learning applications the loss function must satisfy additional properties to its definition: It should be cheap to compute, continuously differentiable and robust in a sense that it is resistant to outliers. Convexity is also desirable to ensure uniqueness of the optimal solution.*

In regression the most frequently used loss function is the squared error:

$$c(x, y, f(x)) = (y - f(x))^2$$

Now we want to find a way to combine the computed errors, so that we can assess how good a given estimate f is. In the following, we assume that there is a probability distribution $P(x, y)$ on $\mathcal{X} \times \mathcal{Y}$, that captures the randomness of the data generation and how the y values depend on the x values. Moreover we assume that $P(x, y)$ has a density function $p(x, y)$ and that the data $(x_i, y_i), i \in [m]$ are drawn independently and identically distributed from $P(x, y)$.

Definition 2.0.4 *(1, Definition 3.3)*
*The **expected risk** with respect to $P(x, y)$ and the loss function c is:*

$$R(f) = E(c(X, Y, f(X))) = \int_{\mathcal{X}\times\mathcal{Y}} c(x, y, f(x))p(x, y)dxdy$$

Naturally this is not a practical definition, as if we knew $p(x, y)$, we would be able to make the best possible predictions without learning. However we can approximate $p(x, y)$ with the empirical density $p_{\text{emp}}(x, y) = \frac{1}{m}\sum_{i=1}^{m} \delta_{x_i}(x)\delta_{y_i}(y)$ and thus arrive at the concept of empirical risk.

Here informally $\delta_{x_i}(x) = \begin{cases} \infty & \text{if } x_i = x \\ 0 & \text{if } x_i \neq x \end{cases}$ and formally δ_{x_i} satisfies $\int \delta_{x_i}(x)f(x)dx = f(x_i)$.

Definition 2.0.5 *(1, Definition 3.4)*
*The **empirical risk** of the function f is defined as:*

$$R_{emp}(f) = \int_{\mathcal{X}\times\mathcal{Y}} c(x, y, f(x))p_{emp}(x, y)dxdy = \frac{1}{m}\sum_{i=1}^{m} c(x_i, y_i, f(x_i))$$

Remark 2.0.6 *If we let c be the squared error, then $R_{emp}(f) = \frac{1}{m}\sum_{i=1}^{m}(y_i - f(x_i))^2$ and we call it the mean squared error (MSE).*

During the learning process we wish to minimize the empirical risk. However, it is easy to construct functions that are optimal for this minimization, but perform very poorly on the unseen data. For example for all i let $\hat{f}(x_i) = y_i$ and let $\hat{f}(x) = 0$ otherwise. Although this function has 0 empirical risk, $\hat{f}$ would not be able to generalize to yet unseen data. For this reason the following 2 methods play an important role in regression.

- Restricting the space of functions, where we do the minimization (the restriction should be done prior to seeing the data).
- Regularization: introducing a punishing term for functions that are too complex or not smooth enough.

Often the above methods are used simultaneously and referred to as inductive bias.

The above mentioned regularization will be carried out by the regularization functional $\Omega : \mathcal{F} \to \mathbb{R}_{\geq 0}$, where $\mathcal{F}$ is the set of functions from $\mathcal{X}$ to $\mathcal{Y}$.

Now we introduce a method that aims to minimize the sum of the empirical risk and the regularization functional over a specific space of functions.

2.1 Ridge regression

Let $(x_i, y_i) \in \mathcal{X} \times \mathcal{Y}$, $i \in [n]$, $\mathcal{F}$ be a subset of the set of functions from $\mathcal{X}$ to $\mathcal{Y}$, $R_{\text{emp}}(f)$ be the empirical risk of f with respect to a loss function c and $\lambda \in \mathbb{R}_{\geq 0}$. Then ridge regression is the following minimization problem:

$$\min_{f \in \mathcal{F}} R_{\text{emp}}(f) + \lambda \Omega(f)$$

Remark 2.1.1 *The coefficient λ enables us to control the effect of the regularization term.*

2.1.1 Regularized linear regression

Let $\mathcal{X} = \mathbb{R}^d$, $\mathcal{Y} = \mathbb{R}$, $(x_i, y_i) \in \mathbb{R}^{d+1}$, $i = 1, \ldots, n$ a training set and $\mathcal{F} = \{\langle \omega, \cdot \rangle | \omega \in \mathbb{R}^d\}$, so we use linear functions for prediction.
In this case it is natural to choose $\Omega(f)$ as $||\omega||^2$. One way to argue for this could be that in linear regression the predictor function is $f(x) = \sum_{i=1}^{n} \omega_i x_i$ and if for a particular j ω_j was large, then a little change in x_j could cause a big change in $f(x_j)$ and thus f would not perform well on noisy data. By penalizing large euclidean norm of ω we enforce smoother and more stable predictions.
Let us choose the loss function as $c(x, y, f(x)) = (y - f(x))^2$. This way the optimization problem becomes:

$$\underset{\omega \in \mathbb{R}^d}{\text{argmin}} \frac{1}{n} \sum_{i=1}^{n} (\langle \omega, x_i \rangle - y_i)^2 + \lambda \langle \omega, \omega \rangle$$

Note that the function we want to optimize depends only on inner products. This observation will be very important, because in the next section we will introduce the concept of a reproducing kernel Hilbert space and by using a kernel function we will be able to map the data points into this space, where the inner product of two such functions will be determined by the kernel function. The kernel function will be easy to compute and thus we will be able to use a very similar learning algorithm to regularized linear regression, but with the difference that we will be able to uncover non-linear relationships.

Chapter 3

Reproducing kernel Hilbert spaces

In this chapter we will state definitions and results that hold for both spaces $\mathbb{R}$ and $\mathbb{C}$, so $\mathbb{F}$ will be used to represent them. Let $\mathcal{X}$ be an arbitrary set and $\mathcal{F}(\mathcal{X}, \mathbb{F})$ be the set of functions from $\mathcal{X}$ to $\mathbb{F}$. The set $\mathcal{F}(\mathcal{X}, \mathbb{F})$ is a vector space over the field $\mathbb{F}$ with the following 2 operations: (f+g)(x) = f(x) + g(x) and $(\lambda f)(x) = \lambda f(x)$.

Definition 3.0.1 *Let $\mathcal{H}$ be a Hilbert-space. Then the functional $f : \mathcal{H} \to \mathbb{F}$ is bounded if there exists $M \in \mathbb{R}_{>0}$ such that $||f(x)||_{\mathbb{F}} \leq M||x||_{\mathcal{H}}$.*

Definition 3.0.2 *(2, Definition 1.1)*
*$\mathcal{H} \subset \mathcal{F}(\mathcal{X}, \mathbb{F})$ is a **reproducing kernel Hilbert space** (from now on **RKHS**) on $\mathcal{X}$, if:*

(i) $\mathcal{H}$ is a vector subspace of $\mathcal{F}(\mathcal{X}, \mathbb{F})$

(ii) $\mathcal{H}$ is endowed with an inner product, $\langle \cdot, \cdot \rangle_{\mathcal{H}}$, with respect to which $\mathcal{H}$ is a Hilbert space

*(iii) for every $x \in \mathcal{X}$ the linear **evaluation functional** $E_x : \mathcal{H} \to \mathbb{F}$, defined by $E_x(f) := f(x)$, is bounded.*

Remark 3.0.3 *E_x is indeed linear since for $\lambda_1, \lambda_2 \in \mathbb{F}$ and $f_1, f_2 \in \mathcal{H}$:*

$$E_x(\lambda_1 f_1 + \lambda_2 f_2) = (\lambda_1 f_1 + \lambda_2 f_2)(x) = \lambda_1 f_1(x) + \lambda_2 f_2(x) = \lambda_1 E_x(f_1) + \lambda_2 E_x(f_2)$$

Theorem 3.0.4 *(Riesz representation theorem)(3, Theorem 9.3.)*
Let $\mathcal{H}$ be a Hilbert space over the field $\mathbb{F}$ and $f : \mathcal{H} \to \mathbb{F}$ be a continuous linear functional. Then there exists a unique $y \in \mathcal{H}$, such that:

$$f(x) = \langle x, y \rangle_{\mathcal{H}}, \forall x \in \mathcal{H}$$

Remark 3.0.5 *If $f : \mathcal{H} \to \mathbb{F}$ is a bounded linear functional on a Hilbert space $\mathcal{H}$, then it is continuous.*

Corollary 3.0.6 *As a result of the Riesz representation theorem it follows that if $\mathcal{H}$ is an RKHS over $\mathcal{X}$, then for each $x \in \mathcal{X}$ the evaluation functional, E_x, can be represented with a unique vector $k_x \in \mathcal{H}$, which means that $f(x) = E_x(f) = \langle f, k_x \rangle_{\mathcal{H}}$, $\forall x \in \mathcal{X}$.*

Definition 3.0.7 *(2, Definition 1.2)*
The function k_x is called the ***reproducing kernel of the point x****.*

Definition 3.0.8 *(2, Definition 1.2)*
The function $K : \mathcal{X} \times \mathcal{X} \to \mathbb{F}$ defined by:

$$K(x, y) = k_y(x)$$

is called the ***reproducing kernel for $\mathcal{H}$****. The notation $k_y = K(\cdot, y)$ will be often used.*

For the previously defined K, the following simple properties hold:

- $K(x, y) = k_y(x) = E_x(k_y) = \langle k_y, k_x \rangle_{\mathcal{H}}$
- $K(x, y) = \langle k_y, k_x \rangle_{\mathcal{H}} = \overline{\langle k_x, k_y \rangle_{\mathcal{H}}} = \overline{E_y(k_x)} = \overline{k_x(y)} = \overline{K(y, x)}$
- $||E_y||^2 = ||k_y||^2_{\mathcal{H}} = \langle k_y, k_y \rangle_{\mathcal{H}} = E_y(k_y) = k_y(y) = K(y, y)$, the first equality is true because the functional E_y is represented by the vector k_y and using the Cauchy-Schwarz inequality it can be shown that they have the same norm.
- $\forall f \in \mathcal{H} : f(x) = E_x(f) = \langle f, k_x \rangle_{\mathcal{H}} = \langle f, K(\cdot, x) \rangle_{\mathcal{H}}$

In the following we give an equivalent definition for an RKHS.

Definition 3.0.9 ***(RKHS 2nd Definition)****(4, Definition 1)*
Let $\mathcal{X}$ be a nonempty set and $\mathcal{H}$ a Hilbert-space of functions $f : \mathcal{X} \to \mathbb{R}$. Then $\mathcal{H}$ is an ***RKHS****, with the inner product $\langle \cdot, \cdot \rangle_{\mathcal{H}}$ if there exists a function $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ such that:*

(i) for all $x \in \mathcal{X}$ $K(\cdot, x) \in \mathcal{H}$

(ii) K has the reproducing property: $\langle f, K(\cdot, x) \rangle_{\mathcal{H}} = f(x)$ for all $f \in \mathcal{H}$ and for all $x \in \mathcal{X}$

Remark 3.0.10 *The 2 definitions for the RKHS are indeed equivalent since the Riesz representation theorem 3.0.4 ensures that the 2nd definition follows from the first and for the other direction* $\forall x \in \mathcal{X}$, $\exists K(\cdot, x) \in \mathcal{H}$:

$$|E_x(f)| = |f(x)| = |\langle f, K(\cdot, x)\rangle_{\mathcal{H}}| \leq ||f||_{\mathcal{H}}||K(\cdot, x)||_{\mathcal{H}} = ||f||_{\mathcal{H}}\sqrt{K(x, x)}$$

where we used the Cauchy-Schwarz inequality. We can let the constant in the definition of boundedness equal $\sqrt{K(x, x)}$.

Example 3.0.11 ($\mathbb{C}^n$ as an RKHS) *(2, Example 1.2.1)*
Let $\mathcal{X} = \{1, ..., n\}$. *We can identify each* $v \in \mathbb{C}^n$ *with a function* $v' : \mathcal{X} \to \mathbb{C}$, $v'(j) := v_j$. *This way* $\mathbb{C}^n$ *is a vector space of all functions on* $\mathcal{X}$, *endowed with the usual inner product of* $\mathbb{C}^n$*:*

$$v', u' : \mathcal{X} \to \mathbb{C} : \quad \langle v', u'\rangle = \langle v, u\rangle_{\mathbb{C}^n} = \sum_{i=1}^{n} v_i\overline{u_i}$$

Let $\{e_j\}_{j=1}^n$ *be the canonical orthonormal basis for* $\mathbb{C}^n$, *the corresponding functions are* $e'_j(i) = \delta_{ij}$, *where* δ_{ij} *is the Kronecker delta. The set of functions* $\{e'_j\}_{j=1}^n$ *is precisely the set of reproducing kernels for the set* $\mathcal{X}$ *because:*

$$v'(j) = v_j = \langle v, e_j\rangle_{\mathbb{C}^n} = \langle v', e'_j\rangle$$

For $\mathbb{C}^n$ *to be an RKHS, the linear evaluation functional* $L_j(v')$ *must be bounded. This is indeed the case because* $\forall v' : \mathcal{X} \to \mathbb{C}$*:*

$$|L_j(v')| = |v'(j)| = |\langle v, e_j\rangle_{\mathbb{C}^n}| \overset{C-S.ineq.}{\leq} ||v||_{\mathbb{C}^n} \cdot ||e_j||_{\mathbb{C}^n} = ||v||_{\mathbb{C}^n} = ||v'||$$

Non-example 3.0.12 *(2, 1.2.2)*
$L^2[0, 1]$ *is not an RKHS because the evaluation functional* $E_x(f) = f(x)$ *is not bounded. Let* $x \in (0, 1)$ *and define*

$$f_n(t) = \begin{cases} \left(\frac{t}{x}\right)^n & \text{if } 0 \leq t \leq x \\ \left(\frac{1-t}{1-x}\right)^n & \text{if } x < t \leq 1 \end{cases}$$

then

$$\lim_{n\to\infty} \|f_n\|_{L^2[0,1]} = 0, \quad \text{but} \quad f_n(x) = 1 \quad \forall n.$$

thus there can not exist a $C \in \mathbb{R}$ *such that* $|E_x(f_n)| = |f_n(x)| \leq C||f_n||_{L^2[0,1]}$ *for all* n.

3.1 Properties of RKHSs

The first important property of an RKHS is that the linear span of the reproducing kernels for all $x \in \mathcal{X}$ is dense in $\mathcal{H}$. We start by defining the notion of density.

Definition 3.1.1 *Let $\mathcal{H}$ be a Hilbert-space and $S \subset \mathcal{H}$ a subset. We say that S is **dense** in $\mathcal{H}$ if $\overline{S} = \mathcal{H}$, where $\overline{S}$ is the closure of S and consists of all the limit points of Cauchy-sequences of S - with respect to the norm induced by the inner product of $\mathcal{H}$.*

Proposition 3.1.2 *(2, Proposition 2.1)*
Let $\mathcal{H}$ be an RKHS on the set $\mathcal{X}$ with kernel K. Then for $f \in \mathcal{H}$

$$f \perp S \iff f = 0$$

where S is the linear span of the functions $k_y = K(\cdot, y)$.

Proof. $\Longleftarrow$ if f(y)=0, for all $y \in \mathcal{X}$, then $0 = f(y) = \langle f, k_y \rangle$, so $f \perp S$
$\Longrightarrow$ $\forall y \in \mathcal{X} : 0 = \langle f, k_y \rangle = f(y)$, so f=0. □

Theorem 3.1.3 ***(Riesz Orthogonal Theorem)****(5, Theorem 2.16.)*
Let $\mathcal{H}$ be a Hilbert-space and $M \subset \mathcal{H}$ a closed subspace. Then for any $x \in \mathcal{H}$, there is a unique $x_1 \in M$ and $x_2 \in M^{\perp}$ such that $x = x_1 + x_2$.

With the help of Proposition 3.1.2 and the Riesz Orthogonal Theorem 3.1.3 we can prove that any element of an RKHS can be approximated arbitrarily well - in the norm induced by the inner product of $\mathcal{H}$ - with functions from S = span$\{k_y | y \in \mathcal{X}\}$, formally S is dense in $\mathcal{H}$.

Corollary 3.1.4 *(5, Proposition 2.20; 2, Proposition 2.1.)*
Let $\mathcal{H}$ be an RKHS on the set $\mathcal{X}$ with kernel K. Then S, the linear span of the functions $k_y = K(\cdot, y)$, is dense in $\mathcal{H}$.

Proof. We saw that $S^{\perp} = \{0\}$. Moreover $S^{\perp} = \overline{S}^{\perp}$, because $S^{\perp} \supseteq \overline{S}^{\perp}$ trivially and if $f \perp S$, then for any $g \in \overline{S}$, there is a sequence $\{h_n\} \subseteq S$ such that $h_n \to g$, and by the continuity of the inner product $0 = \langle f, h_n \rangle \to \langle f, g \rangle$, so $f \in \overline{S}^{\perp}$. Since $\overline{S}$ is a closed subspace of $\mathcal{H}$ and $\overline{S}^{\perp} = \{0\}$, by the Riesz Orthogonal Theorem $\mathcal{H} = \overline{S}$. □

Lemma 3.1.5 *(2, Proposition 2.2.)*
Let $\mathcal{H}$ be an RKHS on $\mathcal{X}$ and let $\{f_n\} \subseteq \mathcal{H}$. If $f_n \to f$, as $n \to \infty$, i.e., $||f_n - f||_{\mathcal{H}} \to 0$, as $n \to \infty$, then $f_n(x) \to f(x)$, as $n \to \infty$ for every $x \in \mathcal{X}$.

Proof.

$$|f_n(x) - f(x)| = |(f_n - f)(x)| = |\langle f_n - f, k_x\rangle| \overset{C-S.ineq.}{\leq} ||f_n - f|| \cdot ||k_x|| \to 0$$

□

Remark 3.1.6 *This property makes an RKHS very well-behaved for machine learning, as norm convergence of functions ensures the convergence of their evaluations. In other words, by approximating a function in the RKHS norm we are approximating its pointwise values, which makes learning more stable and reliable.*

As a consequence of the previous lemma, the following proposition states that if 2 RKHSs on $\mathcal{X}$ have the same reproducing kernel then they are, in fact, equal.

Proposition 3.1.7 *(2, Proposition 2.3.)*
Let $\mathcal{H}_i$, i=1, 2 be RKHSs on $\mathcal{X}$ with kernels K_i, i=1, 2. Let $||\cdot||_i$ denote the norm on the space $\mathcal{H}_i$. If $K_1(x, y) = K_2(x, y)$ for all x, y$\in \mathcal{X}$, then $\mathcal{H}_1 = \mathcal{H}_2$.

Proof. Let $K(x, y) = K_1(x, y) = K_2(x, y)$ and W_i =span$\{k_x \in \mathcal{H}_i : x \in \mathcal{X}\}$, i=1,2. Then for any $f \in W_i$, we have that $f(x) = \sum_j \alpha_j k_{x_j}(x) = \sum_j \alpha_j K(x, x_j)$, thus the values of f are independent of wether we regard f as an element of W_1 or W_2.
Also for any $f \in W_1 = W_2$: $||f||_1^2 = \sum_{i,j} \alpha_i \overline{\alpha}_j \langle k_{x_i}, k_{x_j}\rangle = \sum_{i,j} \alpha_i \overline{\alpha}_j K(x_j, x_i) = ||f||_2^2$.
Now consider the case when $f \in \mathcal{H}_1$, then since W_1 is dense in $\mathcal{H}_1$ (Corollary 3.1.4) we have a sequence $\{f_n\} \subseteq W_1$ such that $f_n \to f$, as $n \to \infty$. Since $\{f_n\}$ is Cauchy in W_1 it is also Cauchy in W_2, because the norms are equal. Consequently there exists $g \in \mathcal{H}_2$ with $f_n \to g$, as $n \to \infty$. By the above Lemma $f(x) = \lim_n f_n(x) = g(x)$, so $f \in \mathcal{H}_2$. The same can be done the other way around resulting in $\mathcal{H}_1 = \mathcal{H}_2$. □

Since we already know that the reproducing kernel of an RKHS is a kernel function, we prove that it is unique.

Proposition 3.1.8 ***(Uniqueness of the reproducing kernel)***
Let $\mathcal{H}$ be an RKHS on $\mathcal{X}$. Assume that the kernel functions K_1 and K_2 are reproducing kernels of $\mathcal{H}$. Then $K_1 = K_2$.

Proof. For any $x \in \mathcal{X}$:

$$||K_1(\cdot, x) - K_2(\cdot, x)||_{\mathcal{H}}^2 = \langle K_1(\cdot, x) - K_2(\cdot, x), K_1(\cdot, x) - K_2(\cdot, x)\rangle_{\mathcal{H}}$$

$$= \langle K_1(\cdot, x) - K_2(\cdot, x), K_1(\cdot, x)\rangle - \langle K_1(\cdot, x) - K_2(\cdot, x), K_2(\cdot, x)\rangle_{\mathcal{H}}$$

$$= K_1(x, x) - K_2(x, x) - K_1(x, x) + K_2(x, x) = 0$$

The equalities followed from the linearity of the inner product and the reproducing properties of K_1 and K_2.
Since $||\cdot||_{\mathcal{H}}$ is a norm $K_1(\cdot, x) = K_2(\cdot, x)$ are equal as functions, so $K_1(y, x) = K_2(y, x)$, $\forall y \in \mathcal{X}$, proving the proposition. □

3.2 Kernel functions

Up to this point we always started with an RKHS and we examined the properties of the corresponding reproducing kernel. In the next section our main goal is to characterize when a function $K : \mathcal{X} \times \mathcal{X} \to \mathbb{C}$ is the reproducing kernel for some RKHS. Since the concept of positive semi-definiteness of a matrix will be of high importance we start by defining it.

Definition 3.2.1 *Let $A = (a_{i,j})$ be a complex Hermitian matrix. A is **positive semi-definite** if and only if for every $\alpha_1, ..., \alpha_n \in \mathbb{C}$ we have that $\sum_{i,j=1}^{n} \overline{\alpha}_i \alpha_j a_{i,j} = \langle A\alpha, \alpha\rangle \geq 0$.*

Definition 3.2.2 *(2, Definition 2.12.)*
*Let $\mathcal{X}$ be a set and $K : \mathcal{X} \times \mathcal{X} \to \mathbb{C}$. Then K is called a **kernel function** if for every n and for every choice of n distinct points, $\{x_1, ..., x_n\} \subseteq \mathcal{X}$, the kernel matrix $\boldsymbol{K}$, $\boldsymbol{K}_{i,j} = K(x_i, x_j)$ is positive semi-definite.*

For later purposes we define a kernel function also in the case when the target space of K is $\mathbb{R}$.

Definition 3.2.3 *If $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, then K is a **kernel function** if $\boldsymbol{K}$ is a symmetric positive semi-definite matrix.*

The following proposition states that the reproducing kernel of an RKHS is always a kernel function.

Proposition 3.2.4 *(2, Proposition 2.13.)*
Let $\mathcal{X}$ be a set and let $\mathcal{H}$ be an RKHS on $\mathcal{X}$ with reproducing kernel K. Then K is a kernel function.

Proof. Fix $\{x_1, ..., x_n\} \subseteq \mathcal{X}$ and $\alpha_1, ..., \alpha_n \in \mathbb{C}$. Then

$$\sum_{i,j=1}^{n} \overline{\alpha}_i \alpha_j K(x_i, x_j) = \sum_{i=1}^{n} \overline{\alpha_i} \langle k_{x_i}, \sum_{j=1}^{n} \alpha_j k_{x_j} \rangle = \langle \sum_{i=1}^{n} \alpha_i k_{x_i}, \sum_{j=1}^{n} \alpha_j k_{x_j} \rangle$$

$$= || \sum_{j=1}^{n} \alpha_j k_{x_j} ||^2 \geq 0$$

□

In the following theorem we start with a kernel function and we build an RKHS such that its reproducing kernel is the kernel function.

Theorem 3.2.5 ***(Moore)*** *(2, Theorem 2.14.)*
Let $\mathcal{X}$ be a set and $K : \mathcal{X} \times \mathcal{X} \to \mathbb{C}$ be a function. If K is a kernel function, then there exists an RKHS $\mathcal{H}$ of functions on $\mathcal{X}$ such that K is the reproducing kernel of $\mathcal{H}$.

Proof. For $y \in \mathcal{X}$ let $k_y : \mathcal{X} \to \mathbb{C}$ be the function defined by $k_y = K(\cdot, y)$. Let W be the vector space that is the span of the set $\{k_y : y \in \mathcal{X}\}$. Let $f, g \in W$. Now we can define a function $B : \mathcal{W} \times \mathcal{W} \to \mathbb{C}$ such that $B(f, g) = B(\sum_j \alpha_j k_{y_j}, \sum_i \beta_i k_{y_i}) = \sum_{i,j} \alpha_j \overline{\beta_i} K(y_i, y_j)$, where α_j and β_i are scalars. Our first goal is to show that B is an inner product on W.
1)
Since a function in W can be expressed many different ways as a linear combination of the functions k_y, we must check whether the function B is well-defined. Firstly assume that $g = \sum_i \beta_i k_{y_i}$ and choose 2 distinct expressions of f, $f = \sum_j \alpha_j k_{y_j} = \sum_l \gamma_l k y_l$, then:

$$B(\sum_j \alpha_j k_{y_j}, g) = \sum_{i,j} \alpha_j \overline{\beta_i} K(y_i, y_j) = \sum_i \overline{\beta_i} \sum_j \alpha_j K(y_i, y_j) = \sum_i \overline{\beta_i} \sum_j \alpha_j k_{y_j}(y_i)$$

$$= \sum_i \overline{\beta_i} f(y_i) = \sum_i \overline{\beta_i} \sum_l \gamma_l k_{y_l}(y_i) = \sum_i \overline{\beta_i} \sum_l \gamma_l K(y_i, y_l) = \sum_{i,l} \gamma_l \overline{\beta_i} K(y_i, y_l)$$

$$= B(\sum_l \gamma_l k_{y_l}, g)$$

Now let $f = \sum_j \alpha_j k_{y_j}$ and choose 2 distinct expressions for $g = \sum_i \beta_i k_{y_i} = \sum_l \gamma_l k_{y_l}$.

$$B(f, \sum_i \beta_i k_{y_i}) = \sum_{i,j} \alpha_j \overline{\beta_i} K(y_i, y_j) = \sum_j \alpha_j \sum_i \overline{\beta_i} K(y_i, y_j) = \sum_j \alpha_j \sum_i \overline{\beta_i K(y_j, y_i)}$$

$$= \sum_j \alpha_j \overline{\sum_i \beta_i k_{y_i}(y_j)} = \sum_j \alpha_j \overline{g(y_j)} = \sum_j \alpha_j \overline{\sum_l \gamma_l k_{y_l}(y_j)}$$

$$= \sum_j \alpha_j \sum_l \overline{\gamma_l} K(y_l, y_j) = B(f, \sum_l \gamma_l k_{y_l})$$

During the second calculation we used the fact that $K(x,y) = \overline{K(y,x)}$, which is true since $\mathbf{K}$ is Hermitian. Overall, we showed that the value of B does not depend on the chosen expression of the functions, so it is well-defined.

2)

Furthermore we have to show that B is sesquilinear. It is linear in the first variable:

$$B(\lambda_1 f_1 + \lambda_2 f_2, g) = B(\lambda_1 \sum_j \alpha_j k_{y_j} + \lambda_2 \sum_j \gamma_j k_{y_j}, \sum_i \beta_i k_{y_i})$$

$$= B(\sum_j (\lambda_1 \alpha_j + \lambda_2 \gamma_j) k_{y_j}, \sum_i \beta_i k_{y_i})$$

$$= \sum_{i,j} (\lambda_1 \alpha_j + \lambda_2 \gamma_j) \overline{\beta_i} K(y_i, y_j)$$

$$= \lambda_1 \sum_{i,j} \alpha_j \overline{\beta_i} K(y_i, y_j) + \lambda_2 \sum_{i,j} \gamma_j \overline{\beta_i} K(y_i, y_j)$$

$$= \lambda_1 B(f_1, g) + \lambda_2 B(f_2, g)$$

To show that it is linear in the second variable we show first that $B(f,g) = \overline{B(g,f)}$:

$$B(f,g) = \sum_{i,j} \alpha_j \overline{\beta_i} K(y_i, y_j) = \overline{\sum_{i,j} \overline{\alpha_j} \beta_i K(y_j, y_i)} = \overline{B(g,f)}$$

And thus:

$$B(f, \lambda_1 g_1 + \lambda_2 g_2) = \overline{B(\lambda_1 g_1 + \lambda_2 g_2, f)} = \overline{\lambda_1 B(g_1, f) + \lambda_2 B(g_2, f)}$$

$$= \overline{\lambda_1} B(f, g_1) + \overline{\lambda_2} B(f, g_2)$$

3)

$B(f,f) \geq 0$ for all $f \in W$, because $B(f,f) = \sum_{i,j} \alpha_j \overline{\alpha_i} K(y_i, y_j) \geq 0$, since $\mathbf{K}$ positive semi-definite.

4)

$B(f,f) = 0 \iff f = 0$

if $f = 0$, then:

$$B(f,f) = \sum_{i,j} \alpha_j \overline{\alpha_i} K(y_i, y_j) = \sum_i \overline{\alpha_i} \sum_j \alpha_j k_{y_j}(y_i) = \sum_i \overline{\alpha_i} f(y_i) = 0$$

as for the other direction firstly note that for all $x \in \mathcal{X}$:

$$B(f, k_x) = B(\sum_i \alpha_i k_{y_i}, k_x) = \sum_i \alpha_i K(x, y_i) = f(x)$$

Moreover the Cauchy-Schwarz inequality already holds for B, as its proof does not depend on the last, yet unproven, requirement for an inner product. Thus for all $x \in \mathcal{X}$:

$$|f(x)| = |B(f, k_x)| \leq \sqrt{B(f, f) \cdot B(k_x, k_x)} = 0$$

which means that f=0.

Now that we have a vector space W endowed with an inner product (from now on: $B(\cdot, \cdot) = \langle \cdot, \cdot \rangle$), we can make it complete by taking equivalence classes of Cauchy sequences from W and thus we get the abstract Hilbert space $\mathcal{H}$.
Using the above $\mathcal{H}$, now our goal is to define $\hat{\mathcal{H}} \subseteq \mathcal{F}(\mathcal{X}, \mathbb{C})$, that will be the RKHS. We will introduce a linear map from $\mathcal{H}$ to $\hat{\mathcal{H}}$ and we will prove that it is a bijection between the two sets. With the help of this result we will be able to equip $\hat{\mathcal{H}}$ with an inner product so that it becomes a Hilbert space. Finally we will show that this $\hat{\mathcal{H}}$ is, in fact, an RKHS with the reproducing kernel K.
Let $\hat{h}(x) = \langle h, k_x \rangle$ for a given $h \in \mathcal{H}$ and $\hat{\mathcal{H}} = \{\hat{h} : h \in \mathcal{H}\}$, so $\hat{\mathcal{H}} \subseteq \mathcal{F}(\mathcal{X}, \mathbb{C})$. Since the map: $L : \mathcal{H} \to \hat{\mathcal{H}}$, $h \mapsto \hat{h}$ is trivially linear, $\hat{\mathcal{H}}$ is a vector space. Note that for any $f \in W$, we have that $\hat{f}(x) = f(x)$. Furthermore the linearity of L ensures that if we want to prove that it is an injective map, it is enough to show that if $\hat{h}(x) = 0, \forall x \in \mathcal{X}$, then $h = 0$.
Suppose that $\hat{h}(x) = 0$ for all x. By definition this means that $\langle h, k_x \rangle = 0$ for every x, so $h \perp W$. Since W is dense in $\mathcal{H}$, we have that h=0. L is of course a surjection by the definition of $\hat{\mathcal{H}}$. Thus L is a bijection and if we define an inner product on $\hat{\mathcal{H}}$ by $\langle \hat{f}, \hat{g} \rangle_{\hat{\mathcal{H}}} = \langle f, g \rangle_{\mathcal{H}}$, then $\hat{\mathcal{H}}$ will be a Hilbert space of functions on $\mathcal{X}$.
Finally for $\hat{\mathcal{H}}$ to be an RKHS the evaluation functional $E_y, \forall y \in \mathcal{X}$ must be bounded.

$$E_y(\hat{h}) = \hat{h}(y) = \langle h, k_y \rangle = \langle \hat{h}, \hat{k_y} \rangle_{\hat{\mathcal{H}}}$$

Thus the boundedness follows from the Cauchy-Schwarz inequality. Moreover the reproducing kernel for the point y is $\hat{k_y} = k_y$, and so the reproducing kernel for $\hat{\mathcal{H}}$ is $\hat{k_y}(x) = \langle k_y, k_x \rangle = K(x, y)$. □

Moore's theorem 3.2.5, proposition 3.2.4 and proposition 3.1.7 and proposition 3.1.8 about the uniqueness of the reproducing kernel show that there is a one-to-one correspondence between RKHSs on a set and kernel functions on the set.

3.3 Solving ride regression in an RKHS

The following theorem states that the minimizer of the regularized risk over an RKHS with the reproducing kernel K is in the linear span of the functions $k_{x_i} = K(\cdot, x_i)$, where the x_i-s are our training samples. This property ensures that an RKHS can be used for learning purposes, because instead of having to solve an optimization problem possibly in infinite dimensions, we can simply determine the coefficients in the linear combination that gives the minimizer function.

Theorem 3.3.1 ***(Representer Theorem)*** *(1,Theorem 4.2)*
Let $\mathcal{H}$ be an RKHS on $\mathcal{X}$ with the reproducing kernel K and $(x_i, y_i) \in \mathcal{X} \times \mathbb{R}$, $i \in [m]$ be a training set.
Denote by $\Omega : [0, \infty) \to \mathbb{R}$ a strictly monotonic increasing function and by $c : (\mathcal{X} \times \mathbb{R}^2)^m \to \mathbb{R} \cup \{\infty\}$ an arbitrary loss function. Then each minimizer of the regularized risk:

$$c(x_1, y_1, f(x_1), \ldots, x_m, y_m, f(x_m)) + \Omega(||f||^2_{\mathcal{H}})$$

admits a representation of the form:

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

Proof. Any $f \in \mathcal{H}$ can be decomposed into a part $f_{||}$, that is in the linear span of the functions $\mathrm{S} = \{k_{x_i} : x_i \in \mathcal{X}\}$ and into a part $f_\perp$, that is orthogonal to S. Thus

$$f = \sum_{i=1}^{m} \alpha_i K(\cdot, x_i) + f_\perp$$

Using the representing property of $\mathcal{H}$ we get the following for all $x_i \in \mathcal{X}, j \in [m]$:

$$f(x_j) = \langle f, K(\cdot, x_j)\rangle_{\mathcal{H}} = \langle \sum_{i=1}^{m} \alpha_i K(\cdot, x_i) + f_\perp, K(\cdot, x_j)\rangle_{\mathcal{H}}$$

$$= \sum_{i=1}^{m} \alpha_i K(x_j, x_i) + \langle f_\perp, K(\cdot, x_j)\rangle_{\mathcal{H}} = \sum_{i=1}^{m} \alpha_i K(x_j, x_i)$$

As a result we see that the orthogonal component has no effect on $f(x_j)$ for any $j \in [m]$, so we have to choose $f_\perp$ such that $\Omega(||f||^2_{\mathcal{H}})$ is minimal. Using the Pythagorean theorem we get:

$$\Omega(||f||^2_{\mathcal{H}}) = \Omega(||f_\perp||^2_{\mathcal{H}} + ||f_{||}||^2_{\mathcal{H}}) \geq \Omega(||f_{||}||^2_{\mathcal{H}})$$

Thus $f_\perp$ must be the function 0, resulting in $f(x) = \sum_{i=1}^{m} \alpha_i K(x, x_i)$ for the f that minimizes the regularized risk. $\square$

Remark 3.3.2 *The following simple calculation motivates why it is sensible to include the norm of the function in the regularized risk. For any $x, x' \in \mathcal{X}$ and $f \in \mathcal{H}$:*

$$|f(x) - f(x')| = |\langle f, K(\cdot, x) - K(\cdot, x')\rangle_{\mathcal{H}}| \overset{C-S.ineq.}{\leq} ||f||_{\mathcal{H}} \cdot ||K(\cdot, x) - K(\cdot, x')||_{\mathcal{H}}$$

Thus the norm of the function can be interpreted as a smoothness measure that controls how fast the value of the function changes with respect to a perturbation of x in the geometry defined by the kernel.

Based on the definition of general ridge regression now we define kernel ridge regression (KRR)

Definition 3.3.3 *Let $(x_i, y_i) \in \mathcal{X} \times \mathbb{R}$, $i \in [n]$ be a training set and K be a kernel function. By $\mathcal{H}$ denote the RKHS over $\mathcal{X}$, that is reproduced by K. $R_{emp}(f) = \frac{1}{n}\sum_{i=1}^{n} c(x_i, y_i, f(x_i))$ is the empirical risk of f with respect to the loss function $c(x, y, f(x)) = (y - f(x))^2$ and let $\lambda \in \mathbb{R}_{>0}$. Then **kernel ridge regression (KRR)** is the following minimization problem:*

$$\hat{f} = \underset{f \in \mathcal{H}}{arg\ min}\ \frac{1}{n}\sum_{i=1}^{n}(y_i - f(x_i))^2 + \lambda||f||_{\mathcal{H}}^2$$

Remark 3.3.4 *Since $\lambda > 0$ the function $\Omega(||f||_{\mathcal{H}}^2) = \lambda||f||_{\mathcal{H}}^2$ is strictly monotonic increasing, which, by the representer theorem 3.3.1, implies that $\hat{f}(x) = \sum_{i=1}^{n} \alpha_i K(x, x_i)$ for some $\alpha_i \in \mathbb{R}$.*

Using this remark we can explicitly find $\hat{f}$ by computing α_i. To proceed let $\boldsymbol{K} \in \mathbb{R}^{n \times n}$, $\boldsymbol{K}_{i,j} = (K(x_i, x_j))$ be a matrix; $\boldsymbol{\alpha} \in \mathbb{R}^n$, $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_n)$; $\boldsymbol{y} \in \mathbb{R}^n$, $\boldsymbol{y} = (y_1, \ldots, y_n)$ be column vectors. Then $(\hat{f}(x_1), \ldots, \hat{f}(x_n))^T = \boldsymbol{K}\boldsymbol{\alpha}$ and $||\hat{f}||_{\mathcal{H}}^2 = \boldsymbol{\alpha}^T\boldsymbol{K}\boldsymbol{\alpha}$.
As a result the minimization problem reduces to:

$$\underset{\boldsymbol{\alpha} \in \mathbb{R}^n}{\arg\min}\frac{1}{n}||(\boldsymbol{K}\boldsymbol{\alpha} - \boldsymbol{y})||_2^2 + \lambda\boldsymbol{\alpha}^T\boldsymbol{K}\boldsymbol{\alpha}$$

This is equivalent to:

$$\underset{\boldsymbol{\alpha} \in \mathbb{R}^n}{\arg\min}\frac{1}{n}(\boldsymbol{\alpha}^T\boldsymbol{K}^T\boldsymbol{K}\boldsymbol{\alpha} - 2\boldsymbol{y}^T\boldsymbol{K}\boldsymbol{\alpha}) + \lambda\boldsymbol{\alpha}^T\boldsymbol{K}\boldsymbol{\alpha}$$

This is a convex function of $\boldsymbol{\alpha}$ since $\boldsymbol{K}$ is symmetric, positive semi-definite and thus $\boldsymbol{K}^T\boldsymbol{K}$ is also positive semi-definite. By differentiating and setting the derivative equal to zero we can find $\boldsymbol{\alpha}^*$, the optimal point.

$$\frac{2}{n}(\boldsymbol{K}\boldsymbol{K}\boldsymbol{\alpha} - \boldsymbol{K}\boldsymbol{y}) + 2\lambda\boldsymbol{K}\boldsymbol{\alpha} = \boldsymbol{0}$$

$$\boldsymbol{K}((\boldsymbol{K} + \lambda n \boldsymbol{I})\boldsymbol{\alpha} - \boldsymbol{y}) = 0$$

Since $\lambda > 0$, the matrix $(\boldsymbol{K} + \lambda n \boldsymbol{I})$ is positive definite and thus invertible. $\boldsymbol{\alpha}^* = (\boldsymbol{K} + \lambda n \boldsymbol{I})^{-1}\boldsymbol{y}$ is a solution, so $\hat{f}(x)$ can be computed explicitly the following way:

$$\hat{f}(x) = \sum_{i=1}^{n} \alpha_i^* K(x, x_i) \tag{3.1}$$

Chapter 4

Properties of kernels

4.1 Kernels as inner products

In the previous chapter we proved that if we take a kernel function, we can construct a reproducing kernel Hilbert-space, that consists of functions that map the input space $\mathcal{X}$ to $\mathbb{R}$. Over this space we could solve the kernel ridge regression problem explicitly and found a function $\hat{f}$. Now we turn our attention to how a kernel function K can be written as an inner product in a Hilbert-space $\mathcal{H}$. To achieve this we use the map $\Phi : \mathcal{X} \to \mathcal{H}$ and write $K(x, y) = \langle \Phi(x), \Phi(y) \rangle_{\mathcal{H}}$. We call $\mathcal{H}$ the feature space and Φ the feature map. The following proposition states that such an expression of a function K exists if and only if it is a kernel function.

Proposition 4.1.1 *Let $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ be a function, $n \in \mathbb{N}$ and $\{x_1, x_2, \dots x_n\} \subseteq \mathcal{X}$. Then K is a kernel function if and only if there exists an inner product space $\mathcal{H}$ and a map $\Phi : \mathcal{X} \to \mathcal{H}$ such that $K(x, y) = \langle \Phi(x), \Phi(y) \rangle_{\mathcal{H}}$.*

Proof. If there exists such a map then we need that $\sum_{i,j=1}^{n} \alpha_i \alpha_j K(x_i, x_j) \geq 0$ for every $n \in \mathbb{N}$ and $\alpha_1, ..., \alpha_n \in \mathbb{R}$. Since $K(x_i, x_j) = \langle \Phi(x_i), \Phi(x_j) \rangle_{\mathcal{H}}$ we have that:

$$\sum_{i,j=1}^{n} \alpha_i \alpha_j K(x_i, x_j) = \sum_{i,j=1}^{n} \alpha_i \alpha_j \langle \Phi(x_i), \Phi(x_j) \rangle_{\mathcal{H}} = || \sum_{i=1}^{n} \alpha_i \Phi(x_i) ||^2_{\mathcal{H}} \geq 0.$$

For the other direction we can use the reproducing kernel map that appeared in the proof of Moore's theorem: $\Phi(x) = K(\cdot, x)$ and thus the theorem guarantees that $K(x, y) = \langle \Phi(x), \Phi(y) \rangle_{\mathcal{H}}$, where $\mathcal{H}$ is the RKHS associated with the kernel function K. □

Remark 4.1.2 ***(Kernel trick)***
Given an algorithm that depends only on a kernel function K, we can construct an alternative algorithm by replacing K with an other kernel function K'.

Due to the previous proposition the kernel trick is a sensible method since we can think of the original algorithm as an inner product based algorithm operating on the data $\Phi(x_1), \ldots, \Phi(x_n)$ and by replacing K with K' we have the same inner product based algorithm, with the modification that it now acts on $\Phi'(x_1), \ldots, \Phi'(x_n)$.

Remark 4.1.3 *Observe that kernel ridge regression is just the kernelized version of the simple regularized linear regression.*

Although the RKHS belonging to a kernel function is unique, there are more feature spaces, where the kernel function computes the inner product. In the following we will construct another Hilbert-space that accomplishes this. To motivate the infinite dimensional case, first we assume that the input space $\mathcal{X}$ is finite, so $\mathcal{X} = \{x_1, x_2, \ldots, x_N\}$. In this case the kernel function $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is completely defined by the $N \times N$ symmetric positive semi-definite matrix $\boldsymbol{K}$. We know that such a matrix can be diagonalized on an orthonormal basis with non-negative eigenvalues. Let the eigenvalues be $0 \leq \lambda_1 \leq \cdots \leq \lambda_N$ and let the eigenvectors be $u_1, \ldots, u_N$. Then

$$K(x_i, x_j) = \sum_{k=1}^{N} \lambda_k [u_k]_i [u_k]_j$$

$$= \langle (\sqrt{\lambda_1}[u_1]_i, \ldots, \sqrt{\lambda_N}[u_N]_i), (\sqrt{\lambda_1}[u_1]_j, \ldots, \sqrt{\lambda_N}[u_N]_j) \rangle_{\mathbb{R}^{\mathbb{N}}}$$

thus with the feature map $\Phi(x_i) = (\sqrt{\lambda_1}[u_1]_i, \ldots, \sqrt{\lambda_N}[u_N]_i)$ we have expressed K as the inner product in $\mathbb{R}^N$.
Moving on to the case where $\mathcal{X}$ is not finite first we define eigenfunctions and eigenvalues and then state Mercer's theorem.

Definition 4.1.4 *Let T be a linear operator on a vector space V. $0 \neq \psi \in V$ is an* ***eigenfunction****, corresponding to the* ***eigenvalue*** *λ, of T if*

$$T\psi = \lambda\psi$$

Theorem 4.1.5 ***(Mercer)*** *(13, page 338.)*
Let $\mathcal{X}$ be a compact metric space and μ be a nondegenerate Borel measure on $\mathcal{X}$, i.e for

any nonempty open set $S \subset \mathcal{X} : \mu(S) > 0$. Suppose that $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is a continuous kernel function. Let us define the integral operator:

$$T_K : L_2(\mathcal{X}) \to L_2(\mathcal{X})$$

$$(T_K f)(x) = \int_{\mathcal{X}} K(x, x') f(x') d\mu(x')$$

Let $\psi_j \in L_2(\mathcal{X})$, $j = 1, \ldots, N_{\mathcal{H}}$ be the normalized orhtogonal eigenfunctions of T_K (it can be proven that they exist and can be chosen this way), with the eigenvalues $\lambda_j > 0$, sorted in non-increasing order.
Then

1. $$\sum_{j=1}^{N_{\mathcal{H}}} \lambda_j < \infty$$

2. $$K(x, x') = \sum_{j=1}^{N_{\mathcal{H}}} \lambda_j \psi_j(x) \psi_j(x') \text{ holds for all } x, x' \in \mathcal{X}$$

$N_{\mathcal{H}}$, the number of eigenfunctions is either finite or countably infinite. In the second case the series converges absolutely for each $x, x' \in \mathcal{X}$ and uniformly on $\mathcal{X} \times \mathcal{X}$.

Remark 4.1.6 *From the second statement of Mercer's Theorem 4.1.5 it follows that $K(x, x') = \langle \Phi(x), \Phi(x') \rangle_{\ell_2^{N_{\mathcal{H}}}}$ for all $x, x' \in \mathcal{X}$ with*

$$\Phi : \mathcal{X} \to \ell_2^{N_{\mathcal{H}}}$$

$$x \mapsto (\sqrt{\lambda_j} \psi_j(x))_{j=1,\ldots,N_{\mathcal{H}}}$$

where $\ell_2^{N_{\mathcal{H}}}$ denotes the Hilbert-space of finite vectors or sequences with finite 2-norm.

A kernel that satisfies the conditions of Mercer's theorem is called a Mercer kernel. In the following section we look at a method that can be used to approximate the kernel matrix of a Mercer kernel. This way we can reduce the computational cost of kernel ridge regression, which is normally $\mathcal{O}(n^3)$, due to the computation of the inverse of $\boldsymbol{K} + \lambda n \boldsymbol{I}$.

4.2 Reduce computational costs

This section is based on [6] Christopher Williams and Matthias Seeger (2000).

4.2.1 Approximating the kernel matrix

Let $x_1, \ldots, x_n$ be an input set from $\mathcal{X}$, K a Mercer kernel and $\boldsymbol{K}$ be the corresponding kernel matrix. The main idea of the approximation is to drop all but the first p terms of the expansion of $K(x, x')$ in Mercer's Theorem 4.1.5. This is sensible, because we saw that $\lambda_n \to 0$, as $n \to 0$. Let us assume that we know the eigenfunctions ψ_i of the integral operator T_K (defined at Mercer's Theorem 4.1.5) and they are sorted so that the corresponding eigenvalues are in non-increasing order. Let $U \in \mathbb{R}^{n \times p}$ be a matrix, whose ith column is the vector $(\psi_i(x_1), \ldots, \psi_i(x_n))^\top \in \mathbb{R}^n$ and let $\Lambda = \operatorname{diag}(\lambda_1, \ldots, \lambda_p)$. This way we have $\boldsymbol{K} \approx U \Lambda U^\top$, based on $K(x, x') \approx \sum_{j=1}^{p} \lambda_j \psi_j(x) \psi_j(x')$.

Note that the above described method assumes that we have the eigendecomposition available. If we wanted to compute it directly, the computational cost would be $\mathcal{O}(n^3)$, so we would not be able to achieve any acceleration of the KRR algorithm. Therefore we approximate the eigenvalues and the eigenfunctions evaluated at the training points using the Nyström method.

4.2.2 Nyström method

ψ_i is an eigenfunction of the integral operator T_K, with an eigenvalue λ_i, if we have for all $y \in \mathcal{X}$:

$$\int_{\mathcal{X}} K(y, x) \psi_i(x) d\mu(x) = \lambda_i \psi_i(y)$$

Let us assume that there is a probability density function p(x), such that:

$$\int K(y, x) \psi_i(x) p(x) dx = \lambda_i \psi_i(y)$$

We want to approximate this eigenfunction equation with an i.i.d. sample $\{x_1, \ldots, x_q\}$ drawn from p(x), so we replace p with the empirical density and approximate the integral with the following sum:

$$\frac{1}{q} \sum_{k=1}^{q} K(y, x_k) \psi_i(x_k) \approx \lambda_i \psi_i(y) \tag{4.1}$$

If we plug x_j for $j = 1, \ldots, q$ for the y into this approximation, we get the following:

$$\frac{1}{q} \boldsymbol{K}^{(q)} \psi_i \approx \lambda_i \psi_i$$

where $\boldsymbol{K}^{(q)}$ is the kernel matrix of the q input points and $\psi_i = (\psi_i(x_1), \ldots, \psi_i(x_q))^\top \in \mathbb{R}^q$. This motivates us to calculate the eigendecomposition of $\boldsymbol{K}^{(q)}$ (it can be done, because it is symmetric):

$$\boldsymbol{K}^{(q)} = U^{(q)} \Lambda (U^{(q)})^\top$$

where U is orthonormal and Λ is diagonal with entries $\lambda_1^{(q)} \geq \lambda_2^{(q)} \geq \cdots \geq \lambda_q^{(q)} \geq 0$.
As a result we get the following approximations:

$$\psi_i(x_j) \approx \sqrt{q} U^q_{j,i}, \quad \lambda_i \approx \frac{\lambda_i^{(q)}}{q}$$

This is a valid approximation, because:

$$\frac{1}{q}\boldsymbol{K}^{(q)}\psi_i \approx \frac{1}{q}\boldsymbol{K}^{(q)}\sqrt{q}U^q_{.,i} = \frac{1}{\sqrt{q}}\lambda_i^{(q)}U^q_{.,i} = \frac{1}{q}\lambda_i^{(q)}\psi_i \approx \lambda_i\psi_i$$

Plugging this back to equation 4.1, we can approximate $\psi_i(y)$ for any $y \in \mathcal{X}$ by:

$$\psi_i(y) \approx \frac{\sqrt{q}}{\lambda_i^{(q)}} \sum_{k=1}^{q} K(y, x_k) U^{(q)}_{k,i} = \frac{\sqrt{q}}{\lambda_i^{(q)}} K(y, .) U^{(q)}_{.,i} \tag{4.2}$$

where $K(y, .) = [K(y, x_1), \ldots, K(y, x_k)]^\top$.
Going back to the previous subsection, where we approximated the kernel matrix for an input set of size n, now we have the approximation of the eigenfunctions evaluated at these points, so we can give an explicit formula for U and Λ. For the approximation of the eigenfunctions we use a sample set of size q with $p \leq q < n$. Let the ith column of U be $U_{.,i}$ and the corresponding eigenvalue be λ_i Then by using equation 4.2 we obtain the following approximations:

$$U_{.,i} \approx \frac{1}{\sqrt{n}} \frac{\sqrt{q}}{\lambda_i^{(q)}} K_{n,q} U^{(q)}_{.,i}, \quad \lambda_i \approx \frac{n}{q}\lambda_i^{(q)}$$

where $U^{(q)}_{.i}$ is the ith eigenvector and $\lambda_i^{(q)}$ is the ith eigenvector of the eigenproblem described above, $K_{n,q}$ is the appropriate $n \times q$ submatrix of $\boldsymbol{K}$.

Now that we can calculate a low-rank approximation for $\boldsymbol{K}$, we look at how it can be used to reduce the computational cost of KRR.

4.2.3 Speeding up KRR

During KRR the most computationally expensive task is to invert the matrix $\boldsymbol{K} + \lambda n \boldsymbol{I}$. Using the approximation described in section 4.2.1, we have to invert the matrix $U\Lambda U^\top +$

$\lambda n \boldsymbol{I}$, which is still of size $n \times n$, but the Woodbury matrix identity will allow us to compute the inverse faster.

Proposition 4.2.1 *(Woodbury matrix identity) Let A be an $n \times n$, U be an $n \times p$, C be a $p \times p$ and V be a $p \times n$ matrix. Assume that the matrices $A + UCV$, A, C, $C^{-1} + VA^{-1}U$ are invertible. Then*

$$(A + UCV)^{-1} = A^{-1} - A^{-1}U\left(C^{-1} + VA^{-1}U\right)^{-1}VA^{-1}$$

In the Woodbury identity we can let $A = \lambda n \boldsymbol{I}$, $U = U$, $C = \Lambda$, $V = U^\top$. This way instead of a $\mathcal{O}(n^3)$ inverse calculation we have to do $\mathcal{O}(np^2)$ calculations, where $p \ll n$.

4.3 The RKHS of a Mercer kernel

Proposition 4.3.1 *Any Mercer kernel is also a kernel function.*

Proof. Let $\boldsymbol{\alpha} \in \mathbb{R}^n$. Then we have

$$\sum_{i,j} \alpha_i \alpha_j K(x_i, x_j) = \sum_{i,j} \alpha_i \alpha_j \langle \Phi(x_i), \Phi(x_j) \rangle = || \sum_i \alpha_i \Phi(x_i) ||^2 \geq 0$$

□

Since every Mercer kernel is a kernel function, by Moore's theorem 3.2.5 we know that there is a unique RKHS belonging to each. The following theorem gives a construction for this RKHS using the eigenfunctions and eigenvalues of the integral operator defined by the Mercer kernel.

Theorem 4.3.2 *Let K be a Mercer kernel and let ψ_j, $j = 1, 2, \ldots$ be the eigenfunctions and λ_j, $j = 1, 2, \ldots$ be the positive eigenvalues of the integral operator defined in Mercer's theorem. Then*

$$\mathcal{H} = \left\{ f = \sum_{j=1}^{\infty} a_j \psi_j, \text{ with } \sum_{j=1}^{\infty} \frac{a_j^2}{\lambda_j} < \infty \right\}$$

is an RKHS with the inner product

$$\langle f, g \rangle_{\mathcal{H}} = \sum_{k=1}^{\infty} \frac{a_k b_k}{\lambda_k}$$

where $f = \sum_k a_k \psi_k$, $g = \sum_k b_k \psi_k$.

Proof. It is technical to prove that it is a Hilbert space of functions from $\mathcal{X}$ to $\mathbb{R}$, so we will only prove the 2 points of definition 3.0.9.
(i) $K(\cdot, x) \in \mathcal{H}$ for all $x \in \mathcal{X}$ part: Let $x \in \mathcal{X}$ and $a_j = \lambda_j \psi_j(x)$ for all j. Then

$$\sum_{j=1}^{\infty} \frac{a_j^2}{\lambda_j} = \sum_{j=1}^{\infty} \lambda_j \psi_j(x)\psi_j(x) = K(x,x) < \infty$$

where the last equality followed from Mercer's theorem. Therefore we know that there is a $\phi_x \in \mathcal{H}$ such that $\phi_x = \sum_{j=1}^{\infty} a_j \psi_j$. By lemma 3.1.5 we know that convergence in the RKHS norm implies point-wise convergence, so for any $y \in \mathcal{X}$ we have

$$\phi_x(y) = \sum_{j=1}^{\infty} a_j \psi_j(y) = \sum_{j=1}^{\infty} \lambda_j \psi_j(x)\psi_j(y) = K(y,x)$$

thus $\phi_x = K(\cdot, x) \in \mathcal{H}$.
(ii) The reproducing property part: Let $f \in \mathcal{H}$, so we can write it as $f = \sum_{j=1}^{\infty} a_j \psi_j$. Let $x \in \mathcal{X}$, we have seen that $K(\cdot, x) = \sum_{j=1}^{\infty} \lambda_j \psi_j(x)\psi_j$. Thus

$$\langle f, K(\cdot, x)\rangle_{\mathcal{H}} = \sum_{j=1}^{\infty} \frac{a_j \lambda_j \psi_j(x)}{\lambda_j} = \sum_{j=1}^{\infty} a_j \psi_j(x) = f(x)$$

□

Remark 4.3.3 *If some eigenvalues are equal to 0 then $\mathcal{H}$ is the subspace spanned by the eigenfunctions with positive eigenvalues.*

Remark 4.3.4 *Note that the eigenfunctions depend on the choice of the measure μ over $\mathcal{X}$, but since we know that the RKHS of a kernel function is unique, $\mathcal{H}$ does not depend on μ.*

The concrete choice of the kernel has a significant effect on our algorithm, so it is important to choose one that has a lot of "expressive power", in a sense that we can learn a wide range of functions with it.

4.4 Choosing a kernel

Let K be a kernel function and x_i, $i = 1, \dots N$ be a training set in the input space. We saw that due to the representer theorem, the functions we learn have the following form $f = \sum_{i=1}^{N} c_j K(\cdot, x_i)$. The following definition encapsulates the idea that we want to approximate any target function arbitrarily well as the number N increases without bound. The approximation will be done in the uniform norm.

Definition 4.4.1 ***(Universal kernel)*** *(7)*
Let $\mathcal{X}$ be a metric space, K be a continuous kernel function on $\mathcal{X} \times \mathcal{X}$, $\mathcal{Z}$ be a compact subset of $\mathcal{X}$ and $\mathcal{C}(\mathcal{Z})$ be the set of continuous functions from $\mathcal{Z}$ to $\mathbb{R}$ equipped with the maximum norm $||\cdot||_{\mathcal{Z}}$. For any $y \in \mathcal{Z}$ define $K_y : \mathcal{X} \to \mathbb{R}$, $K_y(x) = K(x, y)$ for all $x \in \mathcal{X}$. $K(\mathcal{Z}) := \overline{span}\{K_y : y \in \mathcal{Z}\}$, i.e., the set of all functions which are uniform limits of linear combinations of K_y functions. K is ***universal*** *if for any compact subset $\mathcal{Z}$ of $\mathcal{X}$, for any $\epsilon > 0$ and for any $f \in \mathcal{C}(\mathcal{Z})$, there is a function $g \in K(\mathcal{Z})$ such that $||f - g||_{\mathcal{Z}} \leq \epsilon$.*

A significant example of a universal kernel is the Gaussian kernel.

Definition 4.4.2 ***(Gaussian kernel)***

$$K(x, x') = \exp(-\frac{||x - x'||^2}{2\sigma^2})$$

where $\sigma \in \mathbb{R}_{>0}$.

In the following chapter we introduce Gaussian process regression, which is deeply connected to kernel ridge regression, but instead of the deterministic, optimization-based approach, we saw at KRR, we will look at regression from a probabilistic, Bayesian point of view.

Chapter 5

Another perspective, Gaussian process regression

This chapter is based on the 2nd chapter of [8] Gaussian processes for machine learning by Christopher Williams and Carl Edward Rasmussen (2006) and our goal is to establish a connection between Gaussian process regression and the KRR method.

The main idea of Gaussian process (GP) regression is that we describe a prior distribution over functions, where we give higher probabilities to functions that we consider more likely. In most cases smooth functions will be preferred. Then with the Bayesian mindset, we update the prior distribution so that it fits better to the observed data and thus we get the posterior distribution. Although the proposed method sounds promising, the fact that there are uncountably infinite many functions could cause computational difficulties. This is where it comes in handy that we use a Gaussian process, that is the generalization of a Gaussian probability distribution, to describe random functions, because if we ask for the properties of the function only for a finite number of points - as we would do in a regression problem - then inference in the GP gives the same answer, as if we would have taken all the infinitely many points into account. However, the drawback of this method is that we assume Gaussian data, which sometimes turns out to be unrealistic.

Definition 5.0.1 *(Gaussian process)[Dudley 2002 p 443] Let $\mathcal{X}$ be a non-empty set, $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ be a kernel function and $m : \mathcal{X} \to \mathbb{R}$ be any function. Then a random function $f : \mathcal{X} \to \mathbb{R}$ is said to be a Gaussian process (GP) with mean function m and covariance*

function K, denoted by $f \sim \mathcal{GP}(m, K)$, if for any finite set $X = \{x_1, \ldots, x_n\} \subset \mathcal{X}$ of any size $n \in \mathbb{N}$, the random vector

$$(f(x_1), \ldots, f(x_n)) \in \mathbb{R}^n$$

follows the multivariate normal distribution with covariance matrix $K(X, X) = (K(x_i, x_j))_{i,j=1}^n \in \mathbb{R}^{n \times n}$ and mean vector $(m(x_1), \ldots, m(x_n)) \in \mathbb{R}^n$.

Remark 5.0.2 *From now on the mean function m will be chosen to be 0. This implies that $K(x, x') = cov(f(x), f(x')) = \mathbb{E}(f(x)f(x'))$*

Example 5.0.3 *Let $\mathcal{X} \subset \mathbb{R}$. In this case the covariance function is often chosen to be the Radial Basis Function, so*

$$cov(f(x_1), f(x_2)) = K(x_1, x_2) = exp(-\frac{1}{2}|x_1 - x_2|^2)$$

Note that the covariance is almost 1 if the input points are close to each other and it gets closer and closer to 0 as the distance of the input points increases.

The choice of a covariance function K implies a prior distribution over functions, which we can sample for a set $X = \{x_1, \ldots, x_n\}$, by calculating the covariance matrix $\boldsymbol{K}$ and sampling the normal distribution $\mathcal{N}(0, K(X, X))$. This way we get the sample pairs $(x_i, f(x_i))$.

In the following we describe how the prior distribution can be updated, based on the observed data.
Let $f \sim \mathcal{GP}(0, K)$. We will assume that we have observations with additive Gaussian noise, i.e., we have (x_i, y_i) pairs for $i = 1, \ldots, n$, such that $y_i = f'(x_i) + \epsilon_i$, for a function f' and independent ϵ_i-s, with $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$, for $i = 1, \ldots, n$. Let $\boldsymbol{y} = (y_1, \ldots, y_n)$ and $X = \{x_1, \ldots, x_n\}$. By $\mathbb{E}(\epsilon_i) = 0$ and the independence of the ϵ_i-s, we have

$$\text{cov}(y_i, y_j) = \mathbb{E}(f(x_i)f(x_j)) + \sigma^2\delta_{ij} = K(x_i, x_j) + \sigma^2\delta_{ij}$$

where δ_{ij} is the Kronecker delta or in matrix form

$$\text{cov}(\boldsymbol{y}) = K(X, X) + \sigma^2 I$$

Let the test inputs (the inputs, for which we want to make predictions) be $X^* = \{x_1^*, \ldots, x_{n^*}^*\}$

and the corresponding test outputs $\boldsymbol{f^*} = (f(x_1^*), \ldots, f(x_n^*)) \in \mathbb{R}^{n^*}$. Then the prior distribution of the joint vector $[\boldsymbol{y}, \boldsymbol{f^*}] \in \mathbb{R}^{n+n^*}$ becomes

$$\begin{bmatrix} \boldsymbol{y} \\ \boldsymbol{f^*} \end{bmatrix} \sim \mathcal{N}\left(0, \begin{bmatrix} K(X,X) + \sigma^2 I & K(X,X^*) \\ K(X^*,X) & K(X^*X^*) \end{bmatrix}\right)$$

where $K(X, X^*) \in R^{n \times n^*}$ is the matrix containing the covariances evaluated at all pairs of observed and test points.
Since $\boldsymbol{y}$ is already observed we only need those instantiations that agree with the observations. Mathematically this can be carried out by conditioning $\boldsymbol{f^*}$ on $\boldsymbol{y}$. To find the resulting distribution we need a proposition.

Proposition 5.0.4 *(8, Appendix A.2)*
Let $\mathbf{x}$ *and* $\mathbf{y}$ *be jointly Gaussian random vectors*

$$\begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mu_{\mathbf{x}} \\ \mu_{\mathbf{y}} \end{bmatrix}, \begin{bmatrix} A & C \\ C^\top & B \end{bmatrix}\right)$$

then the conditional distribution of $\mathbf{y}$ *given* $\mathbf{x}$ *is*

$$\mathbf{y}|\mathbf{x} \sim \mathcal{N}\left(\mu_{\mathbf{y}} + C^\top A^{-1}(\mathbf{x} - \mu_{\mathbf{x}}),\, B - C^\top A^{-1} C\right).$$

Using this proposition we have

$$\boldsymbol{f^*}|\boldsymbol{y} \sim \mathcal{N}(\mu, \Sigma) \tag{5.1}$$

where

$$\mu = K(X^*, X)[K(X,X) + \sigma^2 I]^{-1}\boldsymbol{y}$$
$$\Sigma = K(X^*, X^*) - K(X^*, X)[K(X,X) + \sigma^2 I]^{-1} K(X, X^*)$$

Thus function values $\boldsymbol{f^*}$, corresponding to the test inputs X^*, can be generated by evaluating μ and Σ and generating samples from the distribution $\mathcal{N}(\mu, \Sigma)$. The best prediction for the output values can be obtained by evaluating the posterior mean μ at the input values $x_1^*, \ldots, x_n^*$.

Remark 5.0.5 *In the case when we want to make a prediction for a single input* $x^* \in \mathcal{X}$*, the posterior mean* μ *simplifies to* $\mu = K(x^*, X)[K(X,X) + \sigma^2 I]^{-1}\boldsymbol{y}$*, where* $K(x^*, X)$

is the vector containing the covariances between value of the test point and the values of the n observed input points. This way $f(x^*) = \mu = \sum_{i=1}^{n} \alpha_i K(x^*, x_i)$*, where* $\boldsymbol{\alpha} = [K(X,X) + \sigma^2]^{-1}\boldsymbol{y}$*. If we compare this expression with the solution of the KRR, which we derived using the representer theorem, we find that if* $n\lambda = \sigma^2$ *we obtain the same predictive function, where* λ *was the regularization constant.*

Note that 5.1 is true for any set of test inputs X^* of any size $n^* \in \mathbb{N}$. This way we can summarize the above calculations in a theorem.

Theorem 5.0.6 *(9, Theorem 3.1)*
Assume additive i.i.d., 0 mean Gaussian additive noise and let $f \sim \mathcal{GP}(0, K)$*. Let* $X = (x_1, \ldots, x_n) \in \mathbb{R}^n$ *and* $\boldsymbol{y} = (y_1, \ldots, y_n)^\top \in \mathbb{R}^n$*. Then we have*

$$f|\boldsymbol{y} \sim \mathcal{GP}(m_{posterior}, K_{posterior})$$

where

$$m_{posterior}(x) = K(x, X)[K(X,X) + \sigma^2 I]^{-1}\boldsymbol{y}$$

$$K_{posterior}(x, x') = K(x, x') - K(x, X)[K(X,X) + \sigma^2 I]^{-1}K(X, x')$$

$K(x, X)$ *and* $K(X, x')$ *are the row and column vectors containing the covariances between f(x) and f(x') and the value of the n observed input points respectively.*

Chapter 6

Image processing

In the following section we introduce the basics of image processing based on the first chapter of the book 10 Image processing: the fundamentals.

6.1 Basics of image processing

Definition 6.1.1 ***(Panchromatic image)***
A ***panchromatic image*** *is a bivariate real valued function $f(x, y)$, where x and y are spatial coordinates and the value of f at the point (x, y) is proportional to the brightness of the depicted scene at that point.*

Definition 6.1.2 ***(Multispectral image)***
A ***multispectral image*** *is a vector valued function $f(x, y) \to \mathbb{R}^d$, each component of which indicates the brightness of the scene at (x, y) at the corresponding range of wavelengths of the electromagnetic spectrum. The components are called channels.*

The difference between a panchromatic image and a multispectral image is that, in the first case the sensor, which observes the scene, captures the intensity of light in a single spectral band, while in the second there are multiple sensors calibrated to capture different ranges of wavelengths.

Remark 6.1.3 *In the case of multispectral images, d is often equal to 3, because we use 3 sensors to capture the intensity of light at the wavelength ranges: red, green, blue.*

Remark 6.1.4 *The processes we apply to images will be presented for panchromatic images, because they can be easily extended to multispectral images by doing the same process in all 3 channels.*

Based on Definition 6.1.1 we think of a digital image as an image, which has been discretised both in spatial coordinates (pixels) and in brightness. It is represented by a matrix $I \in \mathbb{R}^{H \times W}$, where H and W are the height and width of the image respectively, and $I_{i,j} = f(i,j)$, with $0 \leq f(x,y) \leq G$, for some integer $G = 2^n - 1$. G is usually equal to 255 and denotes the maximum pixel value.

In the next section we will be concerned with 3 types of image processing tasks.

6.2 Tasks

The first two tasks can arise when we have to work with images that have been corrupted during the data acquisition process. This can be due to a noisy sensor or transmission errors. The 3rd task is super-resolution where our goal is to give more detail to an image.

1. **Missing pixels/Image inpainting:** We want to estimate the brightness values of some missing pixels, by using the observed brightness values of pixels in their vicinity.

2. **Image denoising:** In this case we received an image, where each pixel was corrupted with some additive noise (modeled as i.i.d. 0 mean Gaussian noise) and our task is to remove this noise as much as possible.

3. **Super-resolution:** Given a digital image, we want to refine the grid, it is defined on. To achieve this we need to estimate $f(x,y)$ at the pixel (x,y), that is not part of our input data.

Remark 6.2.1 *Super-resolution is not only useful when we want to make images aesthetically nicer, but also when we want to compress images, so that their storage requires less memory. This is done by downsampling the data and when we want to use it we apply the super-resolution method to obtain an image "close" to the original.*

In the next section we define some ways to measure the "closeness" of images to each other.

6.3 Image quality metrics

After performing the above mentioned image processing techniques we will want to measure how similar the obtained and the original images are.

Definition 6.3.1 *Based on the Remark 2.0.6 we can define the* ***mean squared error (MSE)*** *between 2 images represented by the matrices* $A, B \in \mathbb{R}^{H \times W}$. *MSE(*$A, B$*)* $= \frac{1}{HW} \sum_{i=1}^{H} \sum_{j=1}^{W} (A_{i,j} - B_{i,j})^2$.

The first metric we define is an MSE based similarity measure.

Definition 6.3.2 *(11)*
The ***peak-signal-to-noise ratio (PSNR)*** *between 2 images represented by the matrices* A, B *is defined as PSNR(*A, B*)=*$10 \log_{10}(\frac{G^2}{MSE(A,B)})$, *where* G *is the maximum pixel value.*

Remark 6.3.3 *Note that as the MSE decreases to 0 the PSNR approaches infinity, so a higher PSNR value indicates a better processing method. On the other hand a small PSNR is caused by big numerical differences between the brightness values of the 2 pictures.*

The second metric we will rely on, was designed in a way that it correlates with the quality perception of our eyes. To demonstrate this we introduce all the 3 image distortion factors that it is composed of.

Definition 6.3.4 *(12)*
The ***structural similarity index measure (SSIM)*** *between 2 images represented by the matrices* A, B *is the product of the following 3 quantities:*

1. $l(A, B) = \frac{2\mu_A\mu_B + C_1}{\mu_A^2 + \mu_B^2 + C_1}$, *where* μ_A *is the mean value of the matrix* A *and can be viewed as the estimate for the luminance of the image.*

2. $c(A, B) = \frac{2\sigma_A\sigma_B + C_2}{\sigma_A^2 + \sigma_B^2 + C_2}$, *where* σ_A^2 *denotes the variance of the matrix* A *and can be viewed as an estimate for the contrast of the image.*

3. $s(A, B) = \frac{\sigma_{A,B} + C_3}{\sigma_A\sigma_B + C_3}$, *where* $\sigma_{A,B}$ *is the covariance of* A *and* B, *and it measures the tendency of* A *and* B *to vary together, thus it indicates structural similarity.*

The constants C_1, C_2, C_3 *are used to avoid a null denominator a are chosen as* $C_1 = (K_1 G)^2, C_2 = (K_2 G)^2, C_3 = \frac{C_2}{2}$, *where* $K_1, K_2 \ll 1$ *constants and* G *is the maximum pixel value.*
SSIM$(A, B) = l(A, B) \times c(A, B) \times s(A, B) = \frac{(2\mu_A\mu_B + C_1)(2\sigma_{A,B} + C_2)}{(\mu_A^2 + \mu_B^2 + C_1)(\sigma_A^2 + \sigma_B^2 + C_2)}$

Chapter 7

Experiments

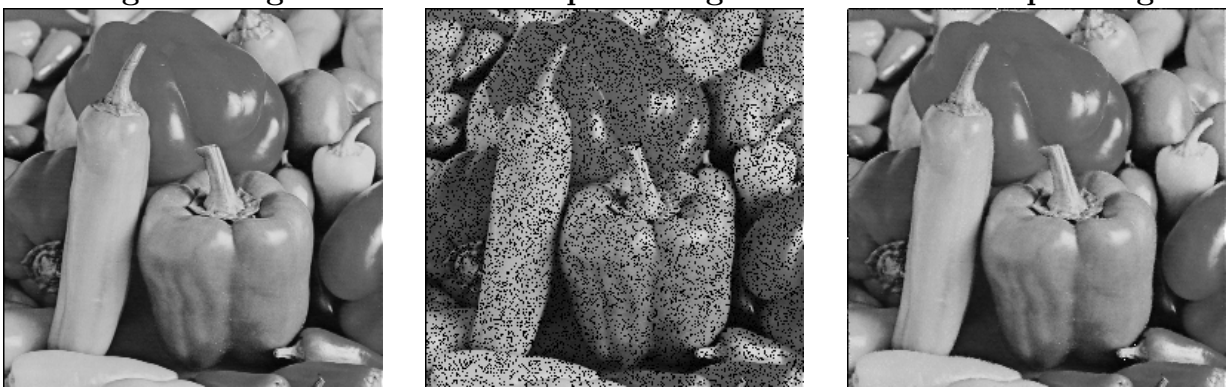
The images used in this chapter are from the Set12 dataset which contains 12 grayscale images with resolution 256×256.

Remark 7.0.1 *A grayscale image is similar to a panchromatic image in the sense that it has a single channel, but it is obtained by combining the channels of the multispectral image with appropriately chosen weights.*

7.1 Image inpainting

In this section we assume that we lost some percentage of the original image. To model this, the "missing pixels" are chosen randomly from the image and we corrupt them by turning them black. We slide a 4×4 patch over every ruined pixel and use the non-corrupted pixels in the patch and their corresponding brightness values as a training set. We calculate the α^* in equation 3.1 and do the prediction for the brightness value of the corrupted pixel. There was no need to do hyperparameter-optimization here, due to the good initial results.

Original image | **Corrupted image** | **After inpainting**



20% of pixels corrupted

$\lambda = 0.01$; $\sigma = 2$

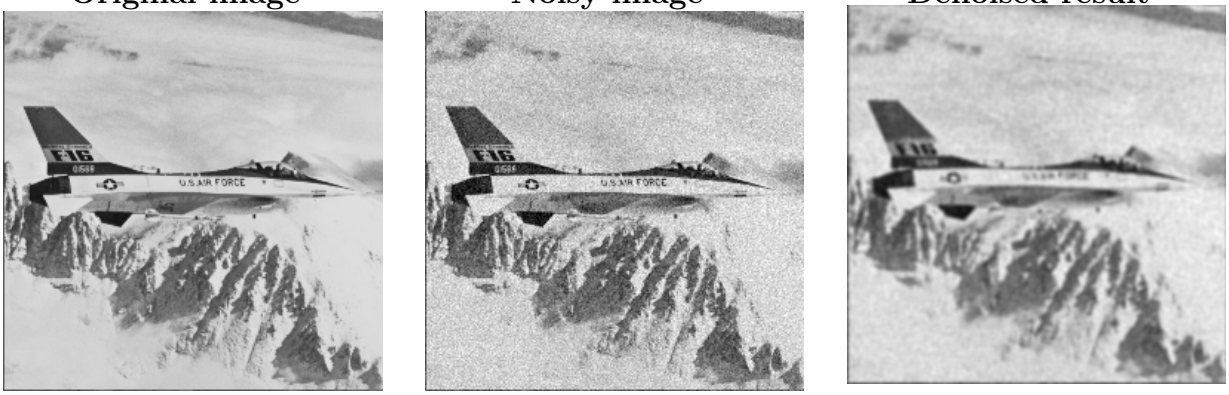
SSIM=0.97; PSNR=34.56

Figure 7.1: Comparison of original, corrupted, and corrected images.

7.2 Image denoising

In this case we added Gaussian noise with 0 mean and 225 variance to each pixel's intensity. We implemented Gaussian kernel ridge regression by moving a 5×5 patch through the image and using these pixels as a training set to estimate the intensity of the central pixel. Neither the use of a smaller nor a bigger patch resulted in significant improvement of the image quality metrics. The regularization constant λ and the σ parameter of the Gaussian kernel was obtained with grid search with respect to the SSIM metric.

Original image | **Noisy image** | **Denoised result**



$\mathcal{N}(0, 15^2)$ noise

$\lambda = 0.05$; $\sigma = 2$

SSIM=0.76; PSNR=23

Figure 7.2: Comparison of original, noisy, and denoised airplane images.

7.3 Super-resolution

In this section we start off with a low-resolution image, containing a lower amount of pixels, and our goal is to multiply the number of pixels in the image. This can be done by "naive" image upsampling methods like nearest neighbors or bicubic interpolation. If we want to create an image that is not only high resolution, but also high-fidelity and aesthetically pleasing we apply super-resolution. The first step is to downsample the 256×256 image, by a factor of 2 to obtain the 128×128 low-resolution image. Then we sweep through the 256×256 grid with a 9×9 patch and the central pixel's intensity value is estimated based on the intensity values that correspond to those low-resolution pixels that fall into the 9×9 square. By performing grid search with respect to the SSIM metric we found:

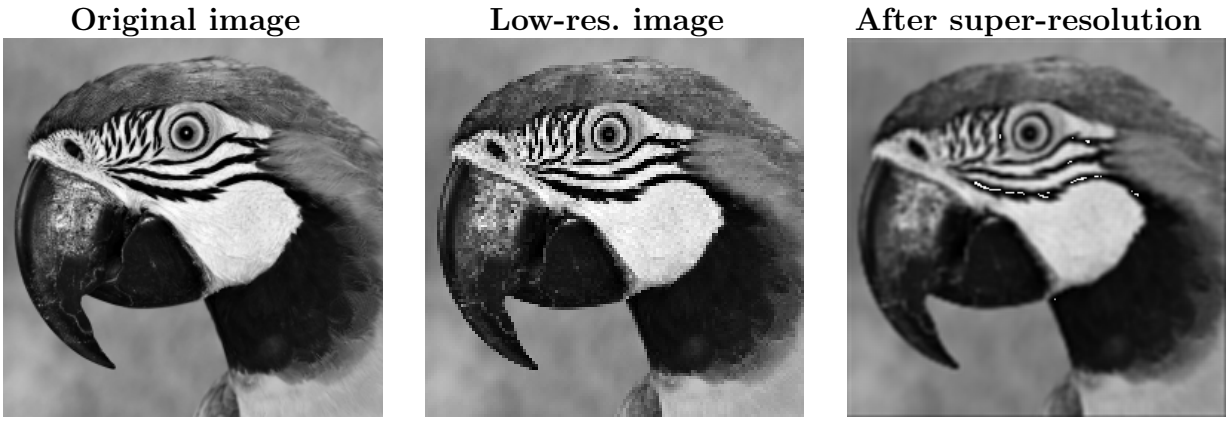
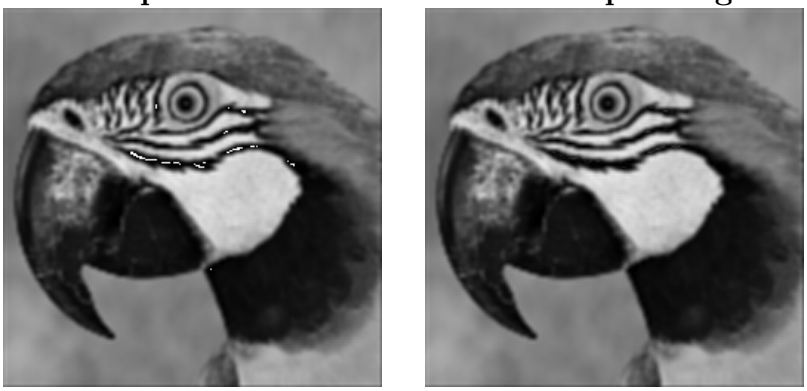


Figure 7.3: Comparison of original, low-res., and reconstructed high-res. images.

We can observe some corrupted pixels in the picture at those areas where there are thin black lines on a white background. This should come as no surprise because in these pixels our method uses mostly white pixels for prediction. To fix this we can use the super-resolved picture as an input for our image inpainting technique, although in that section we assumed that we know which pixels were corrupted. To find these pixels we can slide a 3×3 patch through the image and if the difference of the average intensity in the patch and the intensity of the central pixel is greater than a pre-specified threshold then we mark the pixel as corrupted. We perform the image inpainting several times in a row and we obtain:

After super-resolution **After inpainting**



SSIM=0.83; PSNR=21.67 SSIM=0.83; PSNR=22.48

Figure 7.4: Comparison of high-res. and inpainted high-res. images

Conclusion

After the introduction, in the second chapter of the thesis we introduced the basic concepts of regression and we defined the empirical risk, which gave us a way to quantify how well a regression function performs on a training set. Then we introduced ridge regression, where we added a term that penalized the complexity of the functions. In the third chapter, motivated by regularized linear regression, we introduced RKHSs, which had the important property that function evaluations could be written in the form of an inner product. Since RKHSs provide the basis for kernel methods first we looked at some properties of these function spaces, then defined kernel functions and ultimately arrived at the conclusion that there is a one-to-one correspondence between kernel functions and RKHSs. Then we proved the representer theorem and it enabled us to give an analytical solution to kernel ridge regression, which is an optimization exercise in an infinite dimensional space. In the fourth chapter we saw that kernel functions compute inner products in feature spaces, which enables us to modify inner product based algorithms with the kernel trick. Later we stated Mercer's theorem, which not only granted another feature space, where the kernel function computes the inner product, but also could be used to reduce the computational costs of kernel ridge regression via the Nyström method. Finally we gave another representation of the RKHS of a Mercer kernel and defined the Gaussian kernel, which is a universal kernel. The next chapter contained the overview of Gaussian process regression and it became clear that it is deeply connected to kernel ridge regression. In the following chapter we introduced the basics of image processing, formulated the tasks that we want to experiment with and defined some image quality metrics. In the last section we performed some experiments, the image inpainting and super-resolution were relatively successful. The denoising task proved to be too difficult for our model, the resulting image was blurred, low-fidelity. Some future improvements could be made by experimenting with different kernels like the Paley-Wiener kernel. Using bigger patches in the above mentioned tasks is also worth considering, however due to the increased

computational costs, this would require the implementation of an approximation method, like the one described in Section 4.2.

The codes and images used in Chapter 7 are available in this Github repository.

Bibliography

[1] Bernhard Schölkopf and Alexander J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond.* MIT Press, (2002).

[2] Vern I. Paulsen and Mrinal Raghupathi. *An Introduction to the Theory of Reproducing Kernel Hilbert Spaces.* Cambridge University Press, (2016).

[3] Ferenc Izsák, Zsigmond Tarcsay, Dániel Tüzes. *Analízis jegyzetek I-III.* (2018).

[4] Berlinet, Alain, and Christine Thomas-Agnan. *Reproducing kernel Hilbert spaces in probability and statistics.* Springer Science + Business Media, (2001).

[5] János Karátson. *Numerikus Funkcionálanalízis.*, ELTE Institute of Mathematics, (2014).

[6] Christopher K. I. Williams and Matthias Seeger. *Using the Nyström method to speed up kernel machines.* Advances in neural information processing systems 13, (2000).

[7] Charles A. Micchelli, Yuesheng Xu and Haizhang Zhang. *Universal Kernels.* Journal of Machine Learning Research 7.12, (2006).

[8] Christopher K. I. Williams and Carl Edward Rasmussen. *Gaussian processes for machine learning.* (Vol. 2, No. 3, p. 4). Cambridge, MA: MIT press, (2006).

[9] Motonobu Kanagawa, Philipp Hennig, Dino Sejdinovic, Bharath K. Sriperumbudur. *Gaussian processes and kernel methods: A review on connections and equivalences.* arXiv preprint arXiv:1807.02582, (2018).

[10] Maria Petrou and Costas Petrou. *Image processing: the fundamentals.* John Wiley and Sons, (2010).

[11] Alain Horé and Djemel Ziou. *Image quality metrics: PSNR vs. SSIM.* 2010 20th international conference on pattern recognition. IEEE, (2010).

[12] Zhou Wang, Eero P. Simoncelli and Alan C. Bovik. *Multiscale structural similarity for image quality assessment.* The Thrity-Seventh Asilomar Conference on Signals, Systems and Computers, 2003. Vol. 2. Ieee, (2003).

[13] Hongwei Sun. *Mercer theorem for RKHS on noncompact sets.* Journal of Complexity 21.3 337-349., (2005).

Alulírott Szathmári Gergely Márton nyilatkozom, hogy szakdolgozatom elkészítése során az alább felsorolt feladatok elvégzésére a megadott MI alapú eszközöket alkalmaztam:

Feladat	**Felhasznált eszköz**	**Felhasználás helye**	**Megjegyzés**
LaTeX kód generálása	GPT-4-turbo	3., 4. és 5. Fejezet	
Nyelvhelyesség ellenőrzése	GPT-4-turbo	Teljes dolgozat	

A felsoroltakon túl más MI alapú eszközt nem használtam.