Journal of Machine Learning Research 8 (2007) 385-408 Submitted 6/06; Revised 12/06; Published 3/07

Learning Equivariant Functions with Matrix Valued Kernels

Marco Reisert REISERT@INFORMATIK.UNI-FREIBURG.DE
Hans Burkhardt BURKHARDT@INFORMATIK.UNI-FREIBURG.DE
LMB, Georges-Koehler-Allee 52
Albert-Ludwig University
79110 Freiburg, Germany

Editor: Leslie Pack Kaelbing

Abstract

This paper presents a new class of matrix valued kernels that are ideally suited to learn vector valued equivariant functions. Matrix valued kernels are a natural generalization of the common notion of a kernel. We set the theoretical foundations of so called equivariant matrix valued kernels. We work out several properties of equivariant kernels, we give an interpretation of their behavior and show relations to scalar kernels. The notion of (ir)reducibility of group representations is transferred into the framework of matrix valued kernels. At the end to two exemplary applications are demonstrated. We design a non-linear rotation and translation equivariant filter for 2D-images and propose an invariant object detector based on the generalized Hough transform.

Keywords: kernel methods, matrix kernels, equivariance, group integration, representation theory, Hough transform, signal processing, Volterra series

1. Introduction

In the last decade kernel techniques have gained much attention in machine learning theory. There is a large variety of problems which can be naturally formulated in a kernelized manner, ranging from regression and classification problems, over feature transformation algorithms to coding schemes. There is a large literature on this subject. We recommend Schoelkopf and Smola (2002) and Shawe-Taylor and Cristianini (2004) and references therein.

The notion of a kernel as a kind of similarity between two given patterns can be naturally generalized to matrix valued kernels. A matrix valued kernel can carry more information than only similarity, like information about the relative pose or configuration of the two patterns. It is one way to overcome the 'information bottleneck' of the kernel matrix. First Burbea and Masani (1984) studied the theory of vector-valued reproducing kernel Hilbert spaces leading to the notion of a matrix (or operator) valued kernel. Amodei (1996) applied it for the solution of Partial Differential Equations and more recently Micchelli and Pontil (2005) used it in the context of machine learning.

In this paper we introduce a new class of matrix valued kernels with a specific transformation behavior. The new type of kernel is motivated by a vector valued regression problem. The function to be learned is desired to be equivariant, meaning that group actions on the input space of the function translate to corresponding group actions on the output space. To get an intuition: from signal processing theory one is familiar with the notion of a 'time-invariant linear filter'. In our words the term 'time-invariant' means that the filter is equivariant to the group of time-shifts. More exactly, the time-shift equivariance is expressed by the fact that the group-representations of time-

©2007 Marco Reisert and Hans Burkhardt.

shifts and the action of the filter (in this case a linear convolution) commute. Equivariance is of high interest in signal-/image processing problems and geometrically related learning problems.

We give a constructive way to obtain kernels, whose linear combinations lead to equivariant functions. These kernels are based on the concept of group integration. Group integration is widely used for invariant feature extraction. For a introduction see Burkhardt and Siggelkow (2001). Recently Haasdonk et al. (2005) used group integration to get invariance in kernel methods.

The paper is organized as follows: In Section 2 we give a first motivation for our kernels by solving an equivariant regression problem. We give an illustrative interpretation of the kernels and present an equivariant Representer Theorem, which justifies our lax motivation from the beginning. Further we uncover a relation of our framework to Volterra theory. In Section 3 we give rules how to construct matrix kernels out of matrix kernels and give constraints how to preserve equivariance. Section 4 introduces the notion of irreducibility for kernels and show how the traces of matrix valued kernels are related to scalar valued kernels. Two possible application of matrix valued kernels are given in Section 5: a rotation equivariant non-linear filter and a rotation invariant object detector. Finally, Section 6 gives a conclusion and an outlook for future direction of research and applications.

2. Group Integration Matrix Kernels

In this section we propose the basic approach. After some preliminaries we give a first motivation for our idea. Then we give an interpretation of the proposed kernels and present an equivariant Representer Theorem.

2.1 Preliminaries and Notation

We consider compact (including finite), linear, unimodular groups $\mathcal{G}$. A group representation ρ_g is a group homomorphism $\mathcal{G} \mapsto L(\mathcal{V})$, where $\mathcal{V}$ is a finite dimensional Hilbert space usually called the representation space. Sometimes we write $g\mathbf{x}$ meaning $\rho_g\mathbf{x}$, where the patterns $\mathbf{x} \in \mathcal{V}$ are always in bold face. In the continuous case ($\mathcal{G}$ is a Lie group), the representation should also be continuous. In this case one can show that any representation of the group is equivalent to an unitary representation. For finite groups this statement also holds (for a proof see, for example, Lenz, 1990). So we can restrict ourselves with no loss of generality to unitary group representation, that is, always $\rho_{g^{-1}} = \rho_g^\dagger$ holds. We denote the Haar Integral (or Group Integral) by $\int_\mathcal{G} f(g)\, dg$ regardless whether we deal with finite groups or continuous groups. Due to the unimodularity all reparametrizations are possible. For further reading on these topics we recommend Lenz (1990), Gaal (1973), Nachbin (1965), and Miller (1991); Schur (1968).

We want to learn functions $\mathbf{f} : \mathcal{X} \mapsto \mathcal{Y}$, where $\mathcal{X}, \mathcal{Y}$ are finite dimensional Hilbert spaces. Tensor- or Kronecker products are denoted by $\otimes$. An equivariant function is a function fulfilling $\mathbf{f}(g\mathbf{x}) = g\mathbf{f}(\mathbf{x})$. The group actions on $\mathcal{X}$ and $\mathcal{Y}$ can be chosen such that they apply to the current application or problem. Inner products are denoted by $\langle\cdot|\cdot\rangle_\mathcal{X}$, where the subscript indicates in which space the arguments are living. Due to the unitary group representation the inner product is invariant in the sense $\langle g\mathbf{x}_1|g\mathbf{x}_2\rangle_\mathcal{X} = \langle\mathbf{x}_1|\mathbf{x}_2\rangle_\mathcal{X}$. A scalar valued kernel is a symmetric, positive definite function $k : \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}$ fulfilling the Mercer property. More precisely, we use the term positive definite in the sense of semi-positive definiteness. If strict positive definiteness is forced, we make this explicit. We also assume invariance to group actions of $\mathcal{G}$, that is, $k(g\mathbf{x}_1, g\mathbf{x}_2) = k(\mathbf{x}_1, \mathbf{x}_2)$ for all $g \in \mathcal{G}$. The function space spanned by all vector-valued linear combination of kernel evaluations of k is called

$\mathcal{Z}_k$ (isomorphic to the vector-valued RKHS of k), where the vector-valued coefficients live in $\mathcal{Y}$, that is, $\mathcal{Z}_k = \{\mathbf{f}(\mathbf{x}) = \sum_i k(\mathbf{x},\mathbf{x}_i)\mathbf{a}_i \mid \mathbf{x}_i \in \mathcal{X},\ \mathbf{a}_i \in \mathcal{Y}\}$.

We assume the reader is familiar with basics in kernel methods, tensor algebra and representation theory.

2.2 Motivation

Our goal is to learn functions $\mathbf{f} : \mathcal{X} \mapsto \mathcal{Y}$ from learning samples $\{(\mathbf{x}_i,\mathbf{y}_i)|i = 1..n\}$ in an equivariant manner, that is, the function has to satisfy $\mathbf{f}(g\mathbf{x}) = g\mathbf{f}(\mathbf{x})$ for all $g \in \mathcal{G}$, while $\mathbf{f}(\mathbf{x}_i) = \mathbf{y}_i$ should be fulfilled as accurate as possible.

Due to the Representer Theorem by Kimeldorf and Wahba (1971) minimizer in $\mathcal{Z}_k$ of arbitrary risk functionals are of the form

$$\mathbf{f}(\mathbf{x}) = \sum_{i=1}^{n} k(\mathbf{x},\mathbf{x}_i)\mathbf{a}_i, \tag{1}$$

where the $\mathbf{a}_i \in \mathcal{Y}$ are some vector valued coefficients, which have to be learned. To obtain an equivariant behavior we pretend to present the training samples in all possible poses $\{(g\mathbf{x}_i, g\mathbf{y}_i)|i = 1..n,\ g \in \mathcal{G}\}$. Since no pose g should be favored, the coefficients $\mathbf{a}_i$ are not allowed to depend on g explicitly, but they have to turn its pose according to the actions in the input space. And hence the solution has to look like

$$\mathbf{f}(\mathbf{x}) = \int_{\mathcal{G}} \sum_{i=1}^{n} k(\mathbf{x}, g\mathbf{x}_i)\ g\mathbf{a}_i\ dg, \tag{2}$$

where the integral ranges over the whole group $\mathcal{G}$. As desired the following lemma holds

Lemma 2.1 *Every function of form (2) is an equivariant function with respect to $\mathcal{G}$.*

Proof Due to the invariance of the kernel we have

$$\mathbf{f}(g\mathbf{x}) = \int_{\mathcal{G}} \sum_{i} k(\mathbf{x}, g^{-1}g'\mathbf{x}_i)\ g'\mathbf{a}_i\ dg',$$

and using the unimodularity of $\mathcal{G}$ we reparametrize the integral by $h = g^{-1}g'$ and obtain the desired result

$$\mathbf{f}(g\mathbf{x}) = \int_{\mathcal{G}} \sum_{i} k(\mathbf{x}, h\mathbf{x}_i)\ gh\mathbf{a}_i\ dh = g\mathbf{f}(\mathbf{x}),$$

using the linearity of the group representation. ■

So we have a constructive way to obtain equivariant functions. Since we can exchange summation with integration and the group action is linear we can rewrite (2) by

$$\mathbf{f}(\mathbf{x}) = \sum_{i=1}^{n} \left(\int_{\mathcal{G}} k(\mathbf{x}, g\mathbf{x}_i)\ \rho_g\ dg \right) \mathbf{a}_i,$$

where we can interpret the expression inside the brackets as a matrix valued kernel. First we want to characterize its basic properties in the following

Proposition 2.2 *Let $K : \mathcal{X} \times \mathcal{X} \mapsto L(\mathcal{Y})$ for every $\mathbf{x}_1, \mathbf{x}_2 \in \mathcal{X}$ be defined by*

$$K(\mathbf{x}_1, \mathbf{x}_2) = \int_{\mathcal{G}} k(\mathbf{x}_1, g\mathbf{x}_2)\ \rho_g\ dg,$$

where $\rho_g \in L(\mathcal{Y})$ is a group representation and k is any symmetric, $\mathcal{G}$-invariant function. The following properties hold for a function above,

a) For every $\mathbf{x}_1, \mathbf{x}_2 \in \mathcal{X}$ and $g, h \in \mathcal{G}$, we have that

$$K(g\mathbf{x}_1, h\mathbf{x}_2) = \rho_g\ K(\mathbf{x}_1, \mathbf{x}_2)\ \rho_h^\dagger,$$

that is, K is equivariant in the first argument and anti-equivariant in the second, we say K is equivariant.

b) It holds $K(\mathbf{x}_1, \mathbf{x}_2) = K(\mathbf{x}_2, \mathbf{x}_1)^\dagger$.

Proof To prove a) we just have to follow the reasoning in the proof of Lemma 2.1. For b) we use the invariance and symmetry of k and the unimodularity of $\mathcal{G}$ and get

$$K(\mathbf{x}_1, \mathbf{x}_2) \quad = \quad \int_{\mathcal{G}} k(g^{-1}\mathbf{x}_1, \mathbf{x}_2)\ \rho_g\ dg = \int_{\mathcal{G}} k(\mathbf{x}_2, g\mathbf{x}_1)\ \rho_{g^{-1}}\ dg = K(\mathbf{x}_2, \mathbf{x}_1)^\dagger,$$

as asserted. ■

There are basically two ways to introduce matrix valued kernels in almost the same manner as for scalar-valued kernels. Following Aronszajn (1950) one can introduce a vector valued reproducing kernel Hilbert space (RKHS) and then implicitly assume that there exists a kind of evaluation functional $K_{\mathbf{x}}$ in a certain Hilbert space of vector valued functions (for a more recent introduction in the context of matrix valued kernels see Pontil and Micchelli (2004)). Basically the approach is as follows: by the Riesz Lemma there is a $K_{\mathbf{x}}$ such that $\langle \mathbf{y} | \mathbf{f}(\mathbf{x}) \rangle_{\mathcal{Y}} = \langle K_{\mathbf{x}}\mathbf{y} | \mathbf{f} \rangle$, where the inner product on the right hand side works in the Hilbert space of vector valued functions. As the upper expression is linear in $\mathbf{y}$ one can define a linear operator in $\mathcal{Y}$, the so called matrix valued kernel $K(\mathbf{x}_1, \mathbf{x}_2) \in L(\mathcal{Y})$, by the following $K(\mathbf{x}_1, \mathbf{x}_2)\mathbf{y} := (K_{\mathbf{x}_2}\mathbf{y})(\mathbf{x}_1)$ equation.

A second possibility to introduce the notion of a matrix valued kernel is to demand the existence of a feature mapping into a certain feature space. For example for scalar valued kernels this is done by Shawe-Taylor and Cristianini (2004). For matrix valued kernels the feature space can be identified as the tensor product space of linear mappings on $\mathcal{Y}$ times an arbitrary high dimensional feature space.

Both, the RKHS approach and the feature space approach are equivalent due to a generalized Mercer Theorem. We decided for the latter alternative. It is more appropriate for our purposes, because we can explicitly access the structure of the induced feature space.

Definition 2.3 (Matrix Valued Kernel) *A function $K : \mathcal{X} \times \mathcal{X} \mapsto L(\mathcal{Y})$ is called a matrix valued kernel if there is a sesqui-linear form such that for all $\mathbf{x}_1, \mathbf{x}_2 \in \mathcal{X}$*

$$K(\mathbf{x}_1, \mathbf{x}_2) = \langle \mathbf{\Psi}(\mathbf{x}_1) | \mathbf{\Psi}(\mathbf{x}_2) \rangle_{\mathcal{H}}$$

holds, where $\mathbf{\Psi}$ is a mapping from $\mathcal{X}$ into a feature-space $\mathcal{H} = \mathcal{F} \otimes L(\mathcal{Y})$ with the property that for all $0 \neq \mathbf{\Psi} \in \mathcal{H}$ the matrix $\langle \mathbf{\Psi} | \mathbf{\Psi} \rangle_{\mathcal{H}} \in L(\mathcal{Y})$ is positive definite.

The elements of the feature-space $\mathcal{H}$ can be imagined as $L(\mathcal{Y})$-valued or matrix-valued vectors, just elements of $\mathcal{F} \otimes L(\mathcal{Y})$. Actually it carries the structure of a $L(\mathcal{Y})$-bimodule. A module is a generalization of a vector space where the field elements are replaced by noncommutative ring elements. For a bimodule right and left multiplication are defined (see, for example, R. G. Douglas, 1989). Additionally we have to define the adjoint for the ring-elements $L(\mathcal{Y})$, which simply turns out to be the adjoint of the linear mapping. The connection of the feature space representation to the RKHS of Aronszjan is simple. Similarly to the case of the scalar kernels, the evaluation functionals $K_{\mathbf{x}}$ have to be associated with the elements $\Psi(\mathbf{x})$ living in the feature space $\mathcal{F} \otimes L(\mathcal{Y})$. In the RKHS the functions are represented by linear combinations $\sum_i K_{\mathbf{x}_i}\mathbf{a}_i$; in the feature space the linear combination takes the form $\sum_i \Psi(\mathbf{x}_i)\mathbf{a}_i$ which is an element of $\mathcal{F} \otimes \mathcal{Y}$. The space $\mathcal{F} \otimes \mathcal{Y}$ carries the structure of a $L(\mathcal{Y})$-left module, because multiplication from the left by elements in $L(\mathcal{Y})$ is allowed. Note that this a fundamental difference to scalar kernels. In the case of scalar kernels the representation of functions and the evaluation functionals live in the same space. For matrix kernels it is different. The evaluation functional are elements of a $L(\mathcal{Y})$-bimodule or explicitly $\mathcal{F} \otimes L(\mathcal{Y})$; the functions are elements of the $L(\mathcal{Y})$-left module $\mathcal{F} \otimes \mathcal{Y}$.

Remark 2.4 *In Definition 2.3 the matrix-valued inner product in $\mathcal{H}$ also induces a scalar inner product by the trace $tr(\langle\Psi|\Psi'\rangle_{\mathcal{H}})$ and it indeed holds that $tr(\langle\Psi|\Psi\rangle_{\mathcal{H}}) \geq 0$ for all $\Psi \neq 0$. Hence the space $\mathcal{H}$ is naturally attached with a (semi-)norm $||\Psi|| = \sqrt{tr(\langle\Psi|\Psi\rangle_{\mathcal{H}})}$.*

Theorem 2.5 (GIM-kernels) *If the function k is a scalar kernel, then the function K given in Proposition 2.2 is a matrix valued kernel. We call this kernel a Group Integration Matrix kernel or short GIM-kernel.*

Proof To show this, we give the feature mapping Ψ by the use of the feature mapping Φ corresponding to the scalar kernel k and show that the GIM-kernel can be written as a positive matrix valued inner product in the new feature space $\mathcal{H} = \mathcal{F} \otimes L(\mathcal{Y})$. Let Ψ be given by

$$\Psi(\mathbf{x}) = \frac{1}{\sqrt{\mu(\mathcal{G})}} \int_{\mathcal{G}} \Phi(g\mathbf{x}) \otimes \rho_g \, dg,$$

where $\mu(\mathcal{G}) = \int_{\mathcal{G}} dg$ is the volume of the group. The matrix valued inner product in $\mathcal{H}$ is given by the rule

$$\langle\Phi_1 \otimes \rho_1|\Phi_2 \otimes \rho_2\rangle_{\mathcal{H}} := \rho_1^\dagger \rho_2 \langle\Phi_1|\Phi_2\rangle_{\mathcal{F}}$$

and its linear extension, that is, it is a sesqui-linear mapping of type $\mathcal{H} \times \mathcal{H} \mapsto L(\mathcal{Y})$. And the so defined product is indeed positive. Any $\Psi \in \mathcal{H}$ can be written as a sum (integral) of Kronecker products $\Psi = \sum_i \Phi_i \otimes \rho_i$. So for any $\Psi \in \mathcal{H}$ and any $\mathbf{y} \in \mathcal{Y}$ we have

$$\langle\mathbf{y}|\langle\Psi|\Psi\rangle_{\mathcal{H}}\mathbf{y}\rangle_{\mathcal{Y}} = \sum_{i,j}\langle\Phi_i|\Phi_j\rangle_{\mathcal{F}}\langle\rho_i\mathbf{y}|\rho_j\mathbf{y}\rangle_{\mathcal{Y}} = \sum_{i,j} k_{ij}\langle\rho_i\mathbf{y}|\rho_j\mathbf{y}\rangle_{\mathcal{Y}} = \sum_{i,j} k_{ij}\langle\mathbf{y}_i|\mathbf{y}_j\rangle_{\mathcal{Y}} \geq 0$$

is positive, because the matrix k_{ij} is positive definite by the Mercer property of k.

Using the above rule for the inner product we can compute

$$\begin{aligned}\langle\Psi(\mathbf{x}_1)|\Psi(\mathbf{x}_2)\rangle_{\mathcal{H}} &= \frac{1}{\mu(\mathcal{G})} \int_{\mathcal{G}^2} \langle\Phi(g\mathbf{x}_1) \otimes \rho_g|\Phi(h\mathbf{x}_2) \otimes \rho_h\rangle_{\mathcal{H}} \, dg \, dh \\ &= \frac{1}{\mu(\mathcal{G})} \int_{\mathcal{G}^2} \langle\Phi(g\mathbf{x}_1)|\Phi(h\mathbf{x}_2)\rangle_{\mathcal{F}} \, \rho_{g^{-1}h} \, dg \, dh.\end{aligned}$$

Inserting the scalar kernel k and reparametrizing by $g' = g^{-1}h$ gives

$$\langle \Psi(\mathbf{x}_1)|\Psi(\mathbf{x}_2)\rangle_{\mathcal{H}} = \frac{1}{\mu(\mathcal{G})}\int_{\mathcal{G}^2} k(\mathbf{x}_1, g'\mathbf{x}_2)\rho_{g'}\, dg'\, dh = \int_{\mathcal{G}} k(\mathbf{x}_1, g\mathbf{x}_2)\rho_g\, dg = K(\mathbf{x}_1,\mathbf{x}_2)$$

which is the desired result. ■

2.3 Examples and Interpretation of Equivariant Kernels

To get more intuition how equivariant kernels, not necessarily GIM-kernels, behave, we want to sketch how such kernels can be interpreted as estimates of the relative pose of two objects. First we consider the probably most simple equivariant kernel, the complex hermitian inner product itself. The group $U(1)$ consists of all complex numbers of unit length. If we define the representation of $U(1)$ acting on the $\mathbb{C}^n$ by a simple scalar multiplication with a complex unit number, then the ordinary inner product is obviously equivariant, that is

$$\langle g_1\mathbf{x}_1|g_2\mathbf{x}_2\rangle = \langle e^{\mathbf{i}\phi_1}\mathbf{x}_1|e^{\mathbf{i}\phi_2}\mathbf{x}_2\rangle = e^{\mathbf{i}\phi_1}\langle \mathbf{x}_1|\mathbf{x}_2\rangle e^{-\mathbf{i}\phi_2} = g_1\langle \mathbf{x}_1|\mathbf{x}_2\rangle g_2^\dagger.$$

The absolute value of $\langle \mathbf{x}_1|\mathbf{x}_2\rangle$ can be interpreted as some kind of similarity. But how one should interpret the complex phase of $\langle \mathbf{x}_1|\mathbf{x}_2\rangle$? Therefore consider the distance $J(\phi) = ||\mathbf{x}_1 - e^{\mathbf{i}\phi}\mathbf{x}_2||$ for different values of ϕ. It is easy to show that J is minimal if ϕ is chosen according to $e^{\mathbf{i}\phi} = \frac{\langle \mathbf{x}_1|\mathbf{x}_2\rangle}{|\langle \mathbf{x}_1|\mathbf{x}_2\rangle|}$. Thus, the phase of the inner product carries information about the relative pose of the two objects. Actually this result can be generalized for equivariant kernels. Assume that we want to align the featuremaps $\Psi(\mathbf{x}_1)$ and $\Psi(\mathbf{x}_2)$ of an equivariant kernel optimally with respect to the group $\mathcal{G}$, that is, we try to minimize

$$J(g) = ||\Psi(\mathbf{x}_1) - \Psi(g\mathbf{x}_2)||^2,$$

Reformulating the objective leads to maximizing the trace

$$J'(g) = \mathrm{tr}(K(\mathbf{x}_1,\mathbf{x}_2)\rho_g^\dagger + \rho_g K(\mathbf{x}_2,\mathbf{x}_1)),$$

Let us assume that the group representation is surjective, that is, for any unitary matrix there is a $g \in \mathcal{G}$ such that ρ_g is identical to this matrix. Then we can use the Singular Value Decomposition (SVD) of the kernel to get the optimal solution. Let $K(\mathbf{x}_1,\mathbf{x}_2) = V\Sigma W$ the SVD, then one can show that $\rho_{g^*} = VW$ gives the optimal alignment, where J' takes $J'(g^*) = 2\mathrm{tr}(\Sigma)$ in the optimum. Interpreting this result, we can say that the sum of the singular values of a equivariant kernel expresses the similarity of the two given objects, while the unitary parts U and V contain information about the relative pose of the objects. And in fact, if $\mathbf{x}_1 = g'\mathbf{x}_2$, then $g^* = g'$ holds.

We want to demonstrate the proposed behavior with another example. Consider the finite group of cyclic translations C_n acting on the n dimensional vector space $\mathcal{X} = \mathbb{C}^n$. The group elements g_k with $k = 0, \ldots, n-1$ act on this space by $[\rho_{g_k}\mathbf{x}]_l := [\mathbf{x}]_{(l-k) \bmod n}$. This is just one particular representation of C_n. Another representation for example acts by scalar multiplications $\rho'_{g_k}\mathbf{y} := e^{\mathbf{i}\frac{2\pi}{n}k}\mathbf{y}$. Let us consider a linear GIM-kernel whose input space is $\mathcal{X} = \mathbb{C}^n$ with the representation ρ_g

and whose output space is $\mathcal{Y} = \mathbb{C}$ with the representation ρ'_g. One can choose $\mathcal{Y}$ one dimensional because ρ'_g just acts by scalar multiplications. The GIM-kernel $K : \mathcal{X} \times \mathcal{X} \mapsto \mathbb{C}$ is given by

$$K(\mathbf{x}_1, \mathbf{x}_2) = \sum_{k=0}^{n-1} \langle \mathbf{x}_1 | \rho_{g_k} \mathbf{x}_2 \rangle \; e^{\mathbf{i}\frac{2\pi}{n}k}.$$

To interpret the behavior of K and understand with respect to which properties the objects are aligned we consider the feature $\Psi(\mathbf{x})$. Having a closer look the feature is nothing else than the first coefficient of a discrete Fourier expansion of $\mathbf{x}$. Thus, the kernel is just the product of the first coefficients of object $\mathbf{x}_1$ with the complex conjugate coefficient of $\mathbf{x}_2$. Consequently, the unitary part of the kernel, the complex phase, represents the relative phase of the first Fourier modes of the objects. The alignment with respect to those is a very common procedure in image processing applications to obtain complete invariant feature sets for the cyclic translation group C_n or other abelian groups like the two dimensional rotations $SO(2)$ (see, for example, Canterakis, 1986). The considerations get more intricate when non-abelian groups are considered. For 3D rotations this subject is discussed by Reisert and Burkhardt (2005) in more detail.

2.4 Equivariant Representer Theorem

In the beginning of this section we motivated our approach by pretending to present training patterns in all possible poses, where the pose of a target is implicitly turned by letting the group act on the regression coefficients. We did not clarify whether this is the optimal solution to obtain an equivariant behavior. Are there other possible solutions which are more general but also equivariant?

It is easy to check that the subspace $\mathcal{E} \subset \mathcal{Z}_k$ of equivariant functions is a linear subspace. The projection $\Pi_\mathcal{E}$ onto this subspace is given by

$$(\Pi_\mathcal{E}\mathbf{f})(\mathbf{x}) = \frac{1}{\mu(\mathcal{G})} \int_\mathcal{G} g^{-1}\mathbf{f}(g\mathbf{x}) \; dg. \tag{3}$$

This projection operator is the core of the construction principle presented in this work. In classical invariant theory this operator is also known as the Reynolds operator (Mumford et al., 1994).

Lemma 2.6 *The projection $\Pi_\mathcal{E}$ of (1) admits the proposed representation given in (2).*

Proof Applying the projection on (1) yields

$$\begin{aligned}(\Pi_\mathcal{E}\mathbf{f})(\mathbf{x}) &= \frac{1}{\mu(\mathcal{G})} \int_\mathcal{G} \sum_{i=1}^{n} k(g\mathbf{x}, \mathbf{x}_i) \; g^{-1}\mathbf{a}_i \; dg = \frac{1}{\mu(\mathcal{G})} \int_\mathcal{G} \sum_{i=1}^{n} k(\mathbf{x}, g^{-1}\mathbf{x}_i) \; g^{-1}\mathbf{a}_i \; dg \\ &= \frac{1}{\mu(\mathcal{G})} \int_\mathcal{G} \sum_{i=1}^{n} k(\mathbf{x}, h\mathbf{x}_i) h\mathbf{a}_i \; dh,\end{aligned}$$

where we used the substitution $h = g^{-1}$. ■

In fact, this projection is an unitary projection with respect to the naturally induced scalar product in the feature space. Before stating this more precisely we show how vector valued functions which are based on scalar kernels (as introduced in Eq. (1)) look in feature space. As already mentioned the evaluation functionals and the representations of the functions live in different spaces. The function of form (1) has the following representation $\mathbf{f}(\mathbf{x}) = \langle \Phi(\mathbf{x}) \otimes I_{\mathcal{Y}} | \Psi_{\mathbf{f}} \rangle_{\mathcal{H}}$. The matrix $I_{\mathcal{Y}} \in L(\mathcal{Y})$ denotes the identity matrix on $\mathcal{Y}$. The corresponding feature-space representation $\Psi_{\mathbf{f}}$ is an element of the contracted feature-space $\mathcal{F} \otimes \mathcal{Y}$, a $L(\mathcal{Y})$-left-module. This is easy to see when you think of $\mathcal{F} \otimes \mathcal{Y}$ as vectors whose components itself are column vectors. These column vectors only admit multiplication by matrices from the left as we know from ordinary matrix calculus. The feature space representation of the function is obviously of the form $\Psi_{\mathbf{f}} = \sum_i \Phi(\mathbf{x}_i) \otimes \mathbf{a}_i$, where the $\mathbf{a}_i$ are vector-valued expansion coefficients. The naturally induced scalar-valued inner product in this space is given by

$$\langle \Phi \otimes \mathbf{a} | \Phi' \otimes \mathbf{a}' \rangle_{\mathcal{F}\otimes\mathcal{Y}} = \langle \Phi | \Phi' \rangle_{\mathcal{F}} \langle \mathbf{a} | \mathbf{a}' \rangle_{\mathcal{Y}}. \tag{4}$$

The projection operator $\Pi_{\mathcal{E}}$ as defined in Eq. (3) acts directly on the functions $\mathbf{f}$. To show that $\Pi_{\mathcal{E}}$ is unitary we must examine how it acts on the feature space representation. Therefore, we define the group action on $\mathcal{F} \otimes \mathcal{Y}$

$$g(\Phi(\mathbf{x}) \otimes \mathbf{a}) := \Phi(g^{-1}\mathbf{x}) \otimes (g^{-1}\mathbf{a}). \tag{5}$$

By the use of this we define a new operator $\tilde{\Pi}_{\mathcal{E}}$ acting on $\mathcal{F} \otimes \mathcal{Y}$ and show that it corresponds to $\Pi_{\mathcal{E}}$ and that it is unitary with respect to the inner product given in Eq. (4).

Proposition 2.7 *If $\tilde{\Pi}_{\mathcal{E}}$ is given by*

$$\tilde{\Pi}_{\mathcal{E}} \Psi_{\mathbf{f}} := \frac{1}{\mu(\mathcal{G})} \int_{\mathcal{G}} g \Psi_{\mathbf{f}} \, dg$$

then $\tilde{\Pi}_{\mathcal{E}} \Psi_{\mathbf{f}} = \Psi_{\Pi_{\mathcal{E}}\mathbf{f}}$ holds and $\tilde{\Pi}_{\mathcal{E}}$ is a unitary projection.

Proof Let $\Psi_{\mathbf{f}} = \sum_i \Phi(\mathbf{x}_i) \otimes \mathbf{a}_i$ and correspondingly $f(\mathbf{x}) = \sum_{i=1}^n k(\mathbf{x}, \mathbf{x}_i)\mathbf{a}_i$. Then

$$\begin{aligned} (\Pi_{\mathcal{E}}\mathbf{f})(\mathbf{x}) &= \frac{1}{\mu(\mathcal{G})} \int\limits_{\mathcal{G}} \sum_{i=1}^n k(g\mathbf{x}, \mathbf{x}_i) \, g^{-1}\mathbf{a}_i \, dg = \frac{1}{\mu(\mathcal{G})} \int\limits_{\mathcal{G}} \langle \Phi(g\mathbf{x}) \otimes \rho_g | \Psi_{\mathbf{f}} \rangle_{\mathcal{H}} \\ &= \langle \Phi(\mathbf{x}) \otimes I_{\mathcal{Y}} | \frac{1}{\mu(\mathcal{G})} \int_{\mathcal{G}} g \Psi_{\mathbf{f}} \rangle_{\mathcal{H}} = \langle \Phi(\mathbf{x}) \otimes I_{\mathcal{Y}} | \tilde{\Pi}_{\mathcal{E}} \Psi_{\mathbf{f}} \rangle_{\mathcal{H}} \end{aligned}$$

this yields $\tilde{\Pi}_{\mathcal{E}} \Psi_{\mathbf{f}} = \Psi_{\Pi_{\mathcal{E}}\mathbf{f}}$. The operator $\tilde{\Pi}_{\mathcal{E}}$ is obviously a projection, because $\tilde{\Pi}_{\mathcal{E}}^2 = \tilde{\Pi}_{\mathcal{E}}$. The unitarity is due to unimodularity of the group. It is easily proved by showing that $\tilde{\Pi}_{\mathcal{E}}$ is hermitian with respect to the inner product in Eq. (4). ■

In conclusion, due to the unitartity the projection $\tilde{\Pi}_{\mathcal{E}}$ gives the best equivariant approximation of an arbitrary function in $\mathcal{Z}_k$. The space of equivariant function $\mathcal{E}$ is nothing else than the space of fix points with respect to the group action defined in Eq. (5), that is $\mathcal{E} = \{\Psi \in \mathcal{F} \otimes \mathcal{Y} | \forall g \in \mathcal{G} : g\Psi = \Psi\}$.

The proof of the following theorem is similar to the proof given in Schoelkopf and Smola (2002) for the general Representer Theorem. Note that we allow coupling between the training samples.

Theorem 2.8 (Equivariant Representer Theorem) *Let $\Omega : \mathbb{R}^+ \mapsto \mathbb{R}$ be a strictly monotonically increasing function and $c : (\mathcal{X} \times \mathcal{Y} \times \mathcal{Y})^n \mapsto \mathbb{R}$ a loss function. Then each equivariant minimizer $\mathbf{f} \in \mathcal{Z}_k$ of*

$$R(\mathbf{f}) = c(\mathbf{x}_1, \mathbf{y}_1, \mathbf{f}(\mathbf{x}_1), ..., \mathbf{x}_n, \mathbf{y}_n, \mathbf{f}(\mathbf{x}_n)) + \Omega(||\mathbf{f}||^2)$$

is of the form

$$\mathbf{f}(\mathbf{x}) = \int_{\mathcal{G}} \sum_{i=1}^{n} k(\mathbf{x}, g\mathbf{x}_i)\; g\mathbf{a}_i \; dg,$$

Proof We can decompose a function $\mathbf{f} \in \mathcal{Z}_k$ orthogonally (in the sense of the inner product given in (4)) into a part in the span of

$$\{\Phi(\mathbf{x}_i) \otimes \mathbf{y} \in \mathcal{F} \otimes \mathcal{Y} | i = 1..n, \mathbf{y} \in \mathcal{Y}\}$$

and its orthogonal complement;

$$\mathbf{f}(\mathbf{x}) \quad = \quad \mathbf{f}_{||}(\mathbf{x}) + \mathbf{f}_{\perp}(\mathbf{x}).$$

We know by construction that $\mathbf{f}_{\perp}(\mathbf{x}_i) = 0$ for all $i = 1..n$. Every equivariant function in $\mathcal{Z}_k$ is a projection of the form $\Pi_{\mathcal{E}}\mathbf{f}$. Considering the projection of the decomposition

$$(\Pi_{\mathcal{E}}\mathbf{f})(\mathbf{x}) \quad = \quad (\Pi_{\mathcal{E}}\mathbf{f}_{||})(\mathbf{x}) + (\Pi_{\mathcal{E}}\mathbf{f}_{\perp})(\mathbf{x})$$

we also know that the second term $(\Pi_{\mathcal{E}}\mathbf{f}_{\perp})(\mathbf{x}_i) = 0$ vanishes for all training samples. Thus the loss function in $R(\Pi_{\mathcal{E}}\mathbf{f})$ stays unchanged if one neglects $\Pi_{\mathcal{E}}\mathbf{f}_{\perp}$. By Lemma 2.6 we know that $\Pi_{\mathcal{E}}\mathbf{f}_{||}$ is of the proposed form

$$\Pi_{\mathcal{E}}\mathbf{f}_{||} = \int_{\mathcal{G}} \sum_{i=1}^{n} k(\mathbf{x}, g\mathbf{x}_i)\; g\mathbf{a}_i \; dg.$$

Due to the orthogonality we also know

$$||\Pi_{\mathcal{E}}\mathbf{f}||^2 \quad = \quad ||\Pi_{\mathcal{E}}\mathbf{f}_{||}||^2 + ||\Pi_{\mathcal{E}}\mathbf{f}_{\perp}||^2$$

and hence $\Omega(||\Pi_{\mathcal{E}}\mathbf{f}||^2) \geq \Omega(||\Pi_{\mathcal{E}}\mathbf{f}_{||}||^2)$. And thus for fixed values of $\mathbf{a}_i$ we get $R(\Pi_{\mathcal{E}}\mathbf{f}) \geq R(\Pi_{\mathcal{E}}\mathbf{f}_{||})$, so we have to choose $\Pi_{\mathcal{E}}\mathbf{f}_{\perp} = 0$ to minimize the objective. As this also has to hold for the solution, the theorem holds. ■

Note that the above theorem makes only statements for vector valued function spaces induced by a scalar kernel. There are also matrix valued kernels that are not based on a scalar kernels, that is, GIM-kernels are not the only way to obtain equivariant kernels. In Section 4 we will work out that one important class of equivariant kernels are GIM-kernels, namely those kernels whose corresponding group representation is irreducible.

2.5 Non-Compact Groups and Relation to Volterra Filters

In this work we consider compact unimodular groups. This is indeed very restrictive. Demanding that the functions $f(g) = k(\mathbf{x}_1, g\mathbf{x}_2)$ are all functions of compact support, it seems possible to generalize our theory for non-compact groups.

In signal-processing the group of time-shifts is elementary, which is actually a non-compact unimodular group. Convolutions are known to be linear time-invariant mappings. In our terms convolutions are equivariant to time-shifts. The so called Volterra series (Boyd et al., 1984) are generalized non-linear convolutions. The Volterra theory states that a nonlinear system, that maps a function $\mathbf{x}$ defined on the time line on a function $\mathbf{f}$ in an equivariant manner, can be modeled as infinite sums of multidimensional convolutions of increasing order

$$[\mathbf{f}(\mathbf{x})]_t = h_0 + \int_{g\in\mathcal{G}} h_1(g)[g\mathbf{x}]_t \, dg + \int_{g\in\mathcal{G}} \int_{g'\in\mathcal{G}} h_2(g,g')[g\mathbf{x}]_t[g'\mathbf{x}]_t \, dg dg' + ...,$$

where $[\mathbf{x}]_t$ denotes the value of $\mathbf{x}$ at time t and $\mathcal{G}$ is the group of time-shifts. In fact, Volterra series of degree n can be modeled with polynomial matrix kernels of degree n, that is, the scalar basis kernel is $k(\mathbf{x}_1,\mathbf{x}_2) = (1+\langle\mathbf{x}_1|\mathbf{x}_2\rangle)^n$. In other words, the RKHS of the induced matrix kernel is the space of n-th order Volterra series. Using the exponential kernel $k(\mathbf{x}_1,\mathbf{x}_2) = e^{\lambda\langle\mathbf{x}_1|\mathbf{x}_2\rangle}$ the RKHS is actually the space of infinite degree Volterra series. For higher order polynomial kernels or even exponential kernels a kernelization is quite useful, because the run-time is linear in the number of training samples instead of proportional to the size of the induced feature space. There is already work by Dodd and Harrison (2003) which proposes a similar kernelized version of Volterra theory, but our theory is much more general. Our work tries to give a kernelized generalization of Volterra theory for arbitrary unimodular groups from a machine learning point of view.

3. Constructing Kernels

Similar to scalar valued kernels there are also several building rules to obtain new matrix and scalar valued kernels from existing ones. Most of them are similar to the scalar case, but there are also 'new' ones. In particular, the rule for building a kernel out of two kernels by multiplying them splits into two rules, either using the tensor product or the matrix product.

Proposition 3.1 (Closure Properties) *Let $K_1 : \mathcal{X}\times\mathcal{X} \mapsto L(\mathcal{Y})$ and $K_2 : \mathcal{X}\times\mathcal{X} \mapsto L(\mathcal{Y}')$ be matrix valued kernels and $A \in L(\mathcal{V},\mathcal{Y})$ with full row-rank, then the following functions are kernels.*

a) Let $\mathcal{Y} = \mathcal{Y}'$. $K(\mathbf{x}_1,\mathbf{x}_2) = K_1(\mathbf{x}_1,\mathbf{x}_2) + K_2(\mathbf{x}_1,\mathbf{x}_2)$

b) $K(\mathbf{x}_1,\mathbf{x}_2) = A^\dagger K_1(\mathbf{x}_1,\mathbf{x}_2)A$

c) $K(\mathbf{x}_1,\mathbf{x}_2) = K_1(\mathbf{x}_1,\mathbf{x}_2) \otimes K_2(\mathbf{x}_1,\mathbf{x}_2)$

d) Let $\mathcal{Y} = \mathcal{Y}'$. If for all $\mathbf{x} \in \mathcal{X}$, $K_1(\mathbf{x},\mathbf{x})$ and $K_2(\mathbf{x},\mathbf{x})$ commute, then the matrix product $K(\mathbf{x}_1,\mathbf{x}_2) = K_1(\mathbf{x}_1,\mathbf{x}_2)\, K_2(\mathbf{x}_1,\mathbf{x}_2)$ is a kernel.

e) $K(\mathbf{x}_1,\mathbf{x}_2) = K_1(\mathbf{x}_1,\mathbf{x}_2)^\dagger$

f) $k(\mathbf{x}_1,\mathbf{x}_2) = tr(K_1(\mathbf{x}_1,\mathbf{x}_2))$

Proof We omit proofs for a) and b) since the reasoning directly translates from the scalar case. To show c) for matrix kernels we give the corresponding feature-maps in terms of the already known feature-maps Ψ_1,Ψ_2 for kernel K_1 and K_2. From ordinary tensor calculus we know that

$$\langle\Psi_1\otimes\Psi_2|\Psi_1'\otimes\Psi_2'\rangle_{K_1\otimes K_2} = \langle\Psi_1|\Psi_1'\rangle_{K_1} \otimes \langle\Psi_2|\Psi_2'\rangle_{K_2},$$

so the new feature-map for the tensor-product kernel c) is obviously $\Psi = \Psi_1 \otimes \Psi_2$. The positivity is also given, since the tensor product of two positive definite matrices is again positive definite. For d) the feature-map is actually the same as for c), but we define a different inner product by

$$\langle \Psi_1 \otimes \Psi_2 | \Psi_1' \otimes \Psi_2' \rangle_{K_1 K_2} := \langle \Psi_1 | \Psi_1' \rangle_{K_1} \langle \Psi_2 | \Psi_2' \rangle_{K_2},$$

which extends uniquely to the whole space and is well defined. Since we assume that the diagonal kernels $K_1(\mathbf{x},\mathbf{x})$ and $K_2(\mathbf{x},\mathbf{x})$ commute, the positive definiteness is also given, because the eigenvalues of the matrix product are just the products of the positive eigenvalues of its factors. The proof of e) is trivial. For the proof of statement f), we only have to recall Remark 2.4, where we mentioned that the trace of a positive matrix valued product is an ordinary positive scalar inner product.

■

Let us have a look how such rules can be applied to equivariant kernels and under what circumstances equivariance is preserved. Rule a) can be applied if K_1 and K_2 have the same transformation behavior, meaning that the underlying group representations are identical. Rule b) preserves equivariance if the matrix A is unitary. One can see this by redefining the group representation on $\mathcal{Y}$ by $\rho_g' = A^\dagger \rho_g A$. Since ρ_g' is again an unitary representation, K is an equivariant kernel. Rule c) can also be used to construct equivariant kernels. Supposing an equivariant kernel K_1 and an invariant scalar kernel k in sense that $k(\mathbf{x}_1, g\mathbf{x}_2) = k(\mathbf{x}_1,\mathbf{x}_2)$ for all $g \in \mathcal{G}$, then rule c) implies that $K = K_1 \otimes k = K_1 k$ is also a kernel and in fact equivariant. But how can we construct an invariant kernel k. The simplest approach is to use invariant features, but our framework also offers two possibilities. A GIM-kernel based on the trivial representation (just $\rho_g = 1$) is obviously invariant (this special case covers the approach proposed by Haasdonk et al., 2005). Another possibility is to use rule d) and f) to obtain an invariant kernel. But therefore we have to force the kernel to be normal, that is, the kernel K must always commute with its transpose $K^\dagger$.

Lemma 3.2 *If K is a normal equivariant kernel, then $k = tr(K^\dagger K)$ is an invariant kernel in the following sense $k(\mathbf{x}_1, g\mathbf{x}_2) = k(g'\mathbf{x}_1,\mathbf{x}_2) = k(\mathbf{x}_1,\mathbf{x}_2)$ for all $g, g' \in \mathcal{G}$.*

Proof Since K is normal $K(\mathbf{x},\mathbf{x})$ commutes with its transpose and hence $K^\dagger K$ is a kernel by rule d). By rule f) we know that $\mathrm{tr}(K^\dagger K)$ is a kernel. The invariance of the trace

$$k(\mathbf{x}_1, g\mathbf{x}_2) = \mathrm{tr}((K\rho_g^\dagger)^\dagger K \rho_g^\dagger) = \mathrm{tr}(\rho_g K^\dagger K \rho_g^\dagger) = \mathrm{tr}(K^\dagger K) = k(\mathbf{x}_1,\mathbf{x}_2)$$

proves invariance of the kernel. ■

4. Irreducible Kernels

The question arises which representations should be used to construct GIM-kernels. There are many possibilities, but we want to choose, of course, the most simple and computationally most efficient ones. A matrix kernel which is diagonal seems to be the most appropriate; this means that the kernel has to be diagonal for every pair of input patterns. If the kernel is diagonal the number of kernel entries that has to be computed is just the dimension of the output space $\mathcal{Y}$. But not every matrix kernel can be written in diagonal form, maybe there is only a block diagonal form or even only a fully occupied matrix. To formalize this more precisely we introduce the notion of *irreducibility* for matrix kernels, which should cover the intuition that a kernel cannot be decomposed in a more simple, more diagonal kernel.

Definition 4.1 *A kernel $K(\mathbf{x}_1,\mathbf{x}_2) \in L(\mathcal{Y})$ is reducible if there is a nonempty subspace $\mathcal{W} \subset \mathcal{Y}$, such that for every $\mathbf{y} \in \mathcal{W}$ and for every $\mathbf{x}_1,\mathbf{x}_2 \in \mathcal{X}$, we have $K(\mathbf{x}_1,\mathbf{x}_2)\mathbf{y} \in \mathcal{W}$, otherwise the kernel is called irreducible.*

This definition is very close to the notion of *irreducibility* for group representations. This notion plays the central role in the harmonic analysis of groups (see Miller, 1991). A group representation $\rho_g \in L(\mathcal{V})$ is called reducible if there exists a nonempty subspace $\mathcal{W} \subset \mathcal{V}$ such that for all $\mathbf{w} \in \mathcal{W}$ also $\rho_g\mathbf{w} \in \mathcal{W}$. Then $\mathcal{W}$ is called a invariant subspace. If a representation is not reducible it is called irreducible. For all abelian groups the irreducible representations are very simple and all one-dimensional. They are just the complex numbers of unit length. Most of the readers may know them as Fourier transformations. For example, for the group of two dimensional rotation $SO(2)$, they are just $\rho_g^{(l)} = e^{\mathbf{i}l\phi}$. The invariant subspace on which $\rho_g^{(l)}$ is acting is just $\mathbb{C}$, a one dimensional vector space. Suppose a given function on the circle, then the lth component of the Fourier representation of this function is exactly the representation space on which the representation $\rho_g^{(l)}$ is acting on. For abelian groups the basis function on which the irreducible representation are working have the same formal appearance as the irreducible representations itself which might be confusing. For non-abelian groups it is different. Consider the group of three dimensional rotations $SO(3)$. The corresponding irreducible representations are the so called D-Wigner matrices which are not one dimensional anymore. They are matrices of increasing size $\rho_g^{(l)} \in \mathbb{C}^{(2l+1)\times(2l+1)}$. They act obviously on $2l+1$ dimensional vector spaces. For a given function defined on a three-dimensional sphere the expansion in the invariant subspaces is the so called Spherical Harmonic expansion which is also sometimes called the Fourier representation of the 2-Sphere. One may call harmonic analysis the generalization of Fourier theory to non-abelian groups.

If an equivariant kernel transforms by an irreducible representation, then it can be deduced that also the kernel itself is irreducible. But the opposite direction is not true in general, otherwise any equivariant kernel would be a GIM-kernel. We will later discuss this in more detail. But first two basic properties.

Lemma 4.2 *If the representation ρ of a strictly positive definite, equivariant kernel K is irreducible then K is also irreducible.*

Proof Assume that K is reducible, then there is a subspace $\mathcal{W} \subset \mathcal{Y}$, such that $\mathbf{y} \in \mathcal{W} \Rightarrow K(\mathbf{x}_1,\mathbf{x}_2)\mathbf{y} \in \mathcal{W}$. Since ρ is irreducible, we can choose $\mathbf{y} \in \mathcal{W}$ such that there exists a g with $\rho_g\mathbf{y} = \mathbf{w}+\mathbf{w}^\perp \in \mathcal{W} \oplus \mathcal{W}^\perp$. But we also know that $K(\mathbf{x}_1, g^{-1}\mathbf{x}_2)\mathbf{y} = K(\mathbf{x}_1,\mathbf{x}_2)\rho_g\mathbf{y} \in \mathcal{W}$. In particular, we can choose $\mathbf{x}_1 = \mathbf{x}_2 = \mathbf{x}$. Due to the reducibility of K and the regularity (because of strictly pd) of $K(\mathbf{x},\mathbf{x})$, we know that $0 \neq K(\mathbf{x},\mathbf{x})\mathbf{w}^\perp \in \mathcal{W}^\perp$ and hence $K(\mathbf{x},\mathbf{x})\rho_g\mathbf{y} \notin \mathcal{W}$. Contradiction! ■

Since we know from representation theory (Gaal, 1973) that any unitary representation can be decomposed in a direct sum of irreducible representations we can make a similar statement for kernels.

Corollary 4.3 *Any GIM-kernel K can be decomposed in a direct sum of irreducible GIM-kernels associated with its irreducible representations.*

Proof Let ρ be the representation of the GIM-kernel. Since any ρ can be decomposed in a direct sum $\rho = \rho^{(1)} \oplus ... \oplus \rho^{(n)}$, we directly see that

$$K = K_1 \oplus .. \oplus K_n,$$

where $K_l(\mathbf{x}_1, \mathbf{x}_2) = \int_{\mathcal{G}} k(\mathbf{x}_1, g\mathbf{x}_2)\rho_g^{(l)} dg$. And the K_l are irreducible due to Lemma 4.2. ■

By the Peter-Weyl-Theorem (Gaal, 1973) we know that the entries of the irreducible representations of a compact group $\mathcal{G}$ form a basis for the space of square-integrable function on $\mathcal{G}$. We also know that the entries of the representations are orthogonal with respect to the canonical dot-product. The following Lemma is based on that.

Lemma 4.4 *If the representation ρ of an equivariant kernel K is irreducible then K is a GIM-kernel.*

Proof We define the corresponding scalar kernel by

$$k = \frac{n}{\mu(\mathcal{G})} \mathrm{tr}(K),$$

where n is the dimensionality of the associated group representation. Then

$$\int_{\mathcal{G}} k(\mathbf{x}_1, g\mathbf{x}_2)\rho_g dg = \frac{n}{\mu(\mathcal{G})} \int_{\mathcal{G}} \mathrm{tr}(K(\mathbf{x}_1, \mathbf{x}_2)\rho_g^\dagger)\rho_g dg.$$

By the orthogonality relations for irreducible group representations we know that

$$\frac{n}{\mu(\mathcal{G})} \int_{\mathcal{G}} \mathrm{tr}(K(\mathbf{x}_1, \mathbf{x}_2)\rho_g^\dagger)\rho_g dg = K(\mathbf{x}_1, \mathbf{x}_2).$$

■

It seems that we should concentrate on the irreducible representations of the considered group, and this is indeed the canonical way. Any scalar kernel can be written in terms of its irreducible GIM-kernel expansion.

Proposition 4.5 *Let k be a scalar kernel, then*

$$k(\mathbf{x}_1, \mathbf{x}_2) = tr(K(\mathbf{x}_1, \mathbf{x}_2)),$$

where $K = K_1 \oplus K_2 \oplus ...$ is the direct sum of its irreducible GIM-kernels as given in Corollary 4.3

Proof Due to the Peter-Weyl-Theorem we can expand the scalar kernel in terms of the irreducible representation of $\mathcal{G}$, where the expansion coefficients are by definition the corresponding GIM-kernels.

$$k(\mathbf{x}_1, g\mathbf{x}_2) = \sum_{l=0}^{\infty} \mathrm{tr}(K_l(\mathbf{x}_1, \mathbf{x}_2)(\rho_g^{(l)})^\dagger)$$

where $K_l(\mathbf{x}_1,\mathbf{x}_2) = \int_{\mathcal{G}} k(\mathbf{x}_1, g\mathbf{x}_2)\rho_g^{(l)} dg$. And hence we have

$$k(\mathbf{x}_1, g\mathbf{x}_2) = \sum_{l=0}^{\infty} \mathrm{tr}(K_l(\mathbf{x}_1, g\mathbf{x}_2)) = \mathrm{tr}(K_1(\mathbf{x}_1, g\mathbf{x}_2) \oplus K_2(\mathbf{x}_1, g\mathbf{x}_2) \oplus \ldots) = \mathrm{tr}(K(\mathbf{x}_1, g\mathbf{x}_2)),$$

which proves the statement. ■

Hence we have a one-to-one correspondence between the scalar basis kernel k and the GIM-kernels K_l formed by the irreducible group representations $\rho_g^{(l)}$. So we have for GIM-kernels always the duality between the matrix- and scalar-valued kernels. For Non-GIM-kernels this one-to-one correspondence does not hold.

4.1 Non GIM-kernels

There is another very simple way to construct equivariant kernels. For example assume that $\mathcal{X} = \mathcal{Y}$ then $K(\mathbf{x}_1,\mathbf{x}_2) = |\mathbf{x}_1\rangle\langle\mathbf{x}_2|$ is obviously an equivariant kernel. This kernel may be seen as the matrix-valued analogon to the linear scalar-valued kernel $k(\mathbf{x}_1,\mathbf{x}_2) = \langle\mathbf{x}_2|\mathbf{x}_1\rangle_{\mathcal{X}}$ and indeed $k = \mathrm{tr}(K)$ holds. Note that it is in general not possible to reconstruct K from $\mathrm{tr}(K)$, this is only possible for GIM-kernels by Proposition 4.5. Such a non GIM-kernel can be easily obtained by the construction principle from above. But how to check whether a kernel is a GIM-kernel or not?

Corollary 4.6 *Let K be an irreducible equivariant kernel and let its corresponding representation be reducible, then K is not a GIM-kernel.*

Proof Assume K is of GIM-type then by Corollary 4.3 it is reducible. Contradiction! ■

This statement is the counterpart to Lemma 4.4, which says that if the group-action is irreducible we know that we have a GIM-kernel.

The simple kernel $K(\mathbf{x}_1,\mathbf{x}_2) = |\mathbf{x}_1\rangle\langle\mathbf{x}_2|$ is not of practical value, because regression functions which are built out of it are always proportional their input. So one needs clever feature-maps to get useful kernels.

4.2 Kernel Response for Symmetric Patterns

GIM-kernels provide an intrinsic mechanism to cope with symmetric patterns. A pattern $\mathbf{x} \in \mathcal{X}$ is called $\mathcal{U}$-symmetric if there is a subgroup $\mathcal{U}$ of $\mathcal{G}$ such that for all $u \in \mathcal{U}$ we have $u\mathbf{x} = \mathbf{x}$. To characterize the kernel response for such patterns we define a linear unitary projection on the space of $\mathcal{U}$-symmetric patterns as follows

$$\pi_{\mathcal{U}}\mathbf{x} := \frac{1}{\mu(\mathcal{U})} \int_{\mathcal{U}} g\mathbf{x}\, dg. \tag{6}$$

The operator $\pi_{\mathcal{U}}$ is a unitary projection due to the same arguments as used in the proof of Proposition 2.7. Using this operator we can state the following

Lemma 4.7 *Let K be a GIM-kernel. If* $\mathbf{x}_1$ *is* $\mathcal{U}_1$*-symmetric and* $\mathbf{x}_2$ *is* $\mathcal{U}_2$*-symmetric then*

$$K(\mathbf{x}_1,\mathbf{x}_2) = \pi_{\mathcal{U}_1} K(\mathbf{x}_1,\mathbf{x}_2) \pi_{\mathcal{U}_2}$$

holds.

Proof Let $\mathcal{G}/\mathcal{U}$ the quotient group. If $\mathbf{x}_2$ is $\mathcal{U}_2$-symmetric then

$$\begin{aligned} K(\mathbf{x}_1,\mathbf{x}_2) &= \frac{1}{\mu(\mathcal{G})}\int\limits_{\mathcal{G}} k(\mathbf{x}_1,g\mathbf{x}_2)\rho_g \, dg = \frac{1}{\mu(\mathcal{G})}\int\limits_{\mathcal{G}/\mathcal{U}_2}\int\limits_{\mathcal{U}_2} k(\mathbf{x}_1,gu\mathbf{x}_2)\rho_{gu}\, du\, dg \\ &= \frac{1}{\mu(\mathcal{G}/\mathcal{U}_2)}\int\limits_{\mathcal{G}/\mathcal{U}_2} k(\mathbf{x}_1,g\mathbf{x}_2)\, dg \underbrace{\left(\frac{1}{\mu(\mathcal{U}_2)}\int_{\mathcal{U}_2}\rho_u \, du\right)}_{\pi_{\mathcal{U}_2}}. \end{aligned}$$

As $\pi_{\mathcal{U}}$ is a projection and thus $\pi_{\mathcal{U}}^2 = \pi_{\mathcal{U}}$ we can insert a $\pi_{\mathcal{U}_2}$ in the last line from above for free and reverse the whole computation an obtain $K(\mathbf{x}_1,\mathbf{x}_2) = K(\mathbf{x}_1,\mathbf{x}_2)\pi_{\mathcal{U}_2}$. If $\mathbf{x}_1$ is $\mathcal{U}_1$-symmetric the reasoning is very similar. Due to the unimodularity of the group and its subgroups it is no problem to exchange g and u in the last expression from the first line. Hence, we can use the same arguments as for the $\pi_{\mathcal{U}_2}$-symmetry. ∎

Assume now we have a trained function $\mathbf{f}(\mathbf{x})$ composed as a linear combination of GIM-kernels. What is the response of $\mathbf{f}(\mathbf{x})$ if $\mathbf{x}$ is $\mathcal{U}$-symmetric. Due to Lemma 4.7 it is obvious that $\mathbf{f}(\mathbf{x})$ is also $\mathcal{U}$-symmetric and the symmetrization is obtained via the operator defined in Eq. (6). This behavior is reasonable, because there is no reason why the output should violate the symmetry if the input does not spend information beyond. For example, if the output space $\mathcal{Y}$ is a space of probability distributions the idea of symmetrization via $\pi_{\mathcal{U}}$ is a good idea. Because, if we are uncertain about the original pose of an input object due to symmetry, we should vote for all possible outputs allowed by the symmetry in a additive manner. And the operator $\pi_{\mathcal{U}}$ actually works this way.

Now let us consider the opposite view. Suppose we want to build a function that maps a $\mathcal{U}$-symmetric input $\mathbf{x}_0$ to a not symmetric output $\mathbf{y}_0$, which is obviously an ill-posed problem. How does the function behave? We are not able to make a general statement, because it depends on the learning algorithm applied. But to get an intuition consider the function $\mathbf{f}(\mathbf{x}) = K(\mathbf{x},\mathbf{x}_0)\mathbf{y}_0$, which essentially behaves as desired, if $K(\mathbf{x},\mathbf{x}) \propto I_{\mathcal{Y}}$. By Lemma 4.7 we know that $\mathbf{f}(\mathbf{x}) = K(\mathbf{x},\mathbf{x}_0)(\pi_{\mathcal{U}}\mathbf{y}_0)$ holds, thus the function behaves as if the desired output was symmetrized beforehand.

5. Applications

We show two application examples that basically can be described as image processing applications. We present a rotation-equivariant filter, which can be used for example, for image restoration or image enhancement. Thereby each pixel neighborhood is handled as one training instance, which leads to an enormous number of training samples. Hence, we let the filter work in the featurespace of a second-order matrix kernel. Secondly, we build a rotation equivariant object detector. Specific interest points in the image are detected. Relative to local features around these interest points the center of the object is learned. Of course ambiguities can occur. For example, relative to a corner of a rectangle there are two possible centers of the rectangle (only for a square there is a unique center). To cope with such ambiguities we learn the probability distribution that the center of the object lies in a specific direction relative to the local features around the interest points.

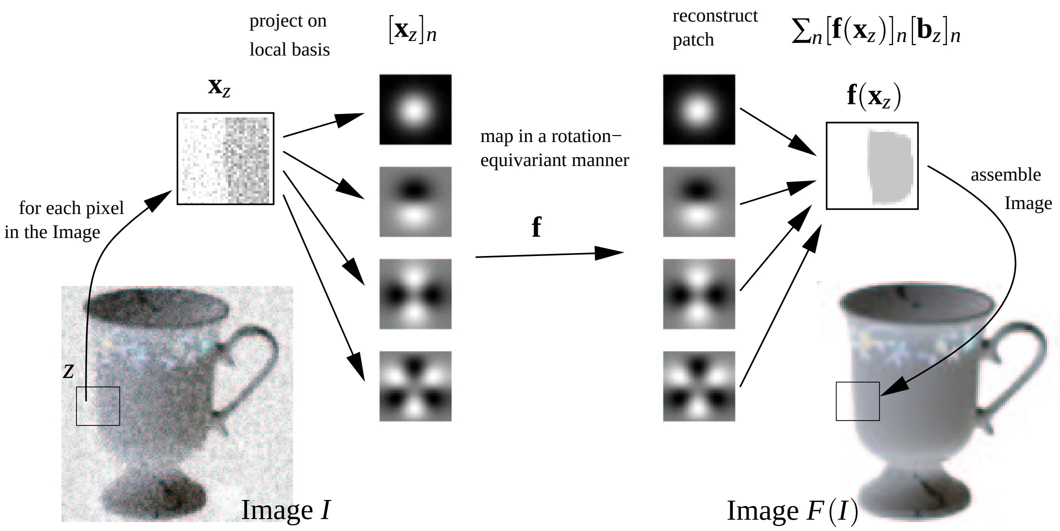


Figure 1: The workflow of the rotation equivariant filter for the example of image denoising.

5.1 Equivariant Kernels for 2D Rotations

In both experiments we want to learn functions $\mathbf{f} : \mathcal{X} \mapsto \mathcal{X}$, where $\mathcal{X}$ is the space of functions defined on the unit circle. The irreducible representations of 2D-rotations are $e^{\mathbf{i}n\phi}$ and the irreducible subspaces correspond to the Fourier representation of the function. Hence in Fourier representation the GIM-kernel is a diagonal kernel with entries

$$K_n(\mathbf{x}_1, \mathbf{x}_2) = \int_0^{2\pi} k(\mathbf{x}_1, g_\phi \mathbf{x}_2) e^{\mathbf{i}n\phi} \, d\phi \tag{7}$$

on the diagonal. Discrete approximations of this integral can be computed quickly by a Fast Fourier Transform (FFT). Assuming k is a dot-product kernel $k(\langle \mathbf{x}_1 | \mathbf{x}_2 \rangle_{\mathcal{X}})$, then a kernel evaluation looks as follows: Compute the cross-correlation $c(\phi) = \langle \mathbf{x}_1 | g_\phi \mathbf{x}_2 \rangle_{\mathcal{X}}$ using the FFT, apply the nonlinearity $k(c(\phi))$ and transform it back into Fourier domain $K_n = \int_0^{2\pi} k(c(\phi)) e^{\mathbf{i}n\phi} \, d\phi$. If the patterns are already in Fourier domain, we need two FFT per kernel evaluation. This approach is used in the second experiment.

The probably most simple non-linear scalar kernel is $k(\mathbf{x}_1, \mathbf{x}_2) = \langle \mathbf{x}_1 | \mathbf{x}_2 \rangle_{\mathcal{X}}^2$. Using this kernel as a scalar base kernel the matrix-valued feature space looks very simple. The feature map is given by

$$[\Psi(\mathbf{x})]_{nl} \propto [\mathbf{x}]_{-l} \, [\mathbf{x}]_{l+n}, \tag{8}$$

where $[\mathbf{x}]_l$ denotes lth component of the Fourier representation of $\mathbf{x}$. The induced matrix valued bilinear product in this space is the ordinary

$$K_n(\mathbf{x}_1, \mathbf{x}_2) = \sum_l [\Psi(\mathbf{x}_1)]_{nl} \, [\Psi(\mathbf{x}_2)]^*_{nl}.$$

hermitian inner product. We use a similar feature space representation in the first experiment.

5.2 A Rotation Equivariant Filter for Images

For convenience we represent a 2D gray-value image I in the complex plane, where we denote the gray value at point $z \in \mathbb{C}$ by I_z. The idea is to learn a rotation-equivariant function $\mathbf{f} : \mathcal{X} \mapsto \mathcal{X}$, which maps each neighborhood $\mathbf{x}_z \in \mathcal{X}$ of a pixel z onto a new neighborhood $\mathbf{f}(\mathbf{x}_z) \in \mathcal{X}$ with properties given by learning samples. The neighborhood is represented by projections on local basis functions $[\mathbf{b}_z]_n = z^n e^{-\lambda|z|^2}$ for $n = 0, \ldots, N-1$. This basis is actually a orthogonal basis with respect to the standard inner product. In Figure 1 the appearance of the basis is depicted. In particular, for each pixel z' the inner product

$$[\mathbf{x}_{z'}]_n = \langle I_{z-z'} | z^n e^{-\lambda|z|^2} \rangle = \sum_{z \in [1,256]^2} I_{z-z'} z^n e^{-\lambda|z|^2}$$

of the image with the basis images is computed. As the basis images are only considerable different from zero in a small neighborhood around the origin, the N expansion coefficients $[\mathbf{x}_{z'}]_n$ represent the appearance of neighborhood of a pixel z. Actually the computation of the local basis expansion is nothing else than a cross correlation of the image with the basis images, which can be expressed as a convolution:

$$[\mathbf{x}_z]_n = I_z * [\mathbf{b}_{-z}]_n^\dagger.$$

Here $*$ denotes the convolution operator. In Figure 1 the approach is depicted for an image denoising application. The basis functions from above have the advantage that their transformation behavior is the same as for functions defined on a circle, that is, $[\mathbf{x}_z]_n \mapsto [\mathbf{x}_z]_n e^{\mathbf{i}n\phi}$ for rotations around z. The vector-components $[\mathbf{f}(\mathbf{x}_z)]_n$ are also interpreted as expansion coefficients of the neighborhood in this basis. Since the neighborhoods overlap we formulate one global function which maps the whole image I on a new image

$$F(I_z) \quad = \quad \sum_n [\mathbf{f}(\mathbf{x}_z)]_n * [\mathbf{b}_z]_n = \mathbf{f}(\mathbf{x}_z) \underline{*}\, \mathbf{b}_z.$$

This function is translation- and rotation-equivariant, since $\mathbf{f}$ is rotation-equivariant and the convolution is naturally translation-equivariant. The scalar basis kernel k for the GIM-kernel expansion of $\mathbf{f}$ is chosen by

$$k(\mathbf{x}_z, \mathbf{x}'_z) = (1 + \langle \mathbf{x}_z | \mathbf{x}'_z \rangle_{\mathcal{X}})^2.$$

The corresponding feature-map is nearly the same as in (8). Given two images I^{src} and I^{dest} our goal is to minimize the cost

$$\sum_{z \in [1,256]^2} |I_z^{\text{dest}} - F_{\mathbf{w}}(I_z^{\text{src}})|^2 + \gamma ||\mathbf{w}||^2,$$

where $\mathbf{w}$ is the parameter vector yielding $\mathbf{f}(\mathbf{x}) = \langle \Psi(\mathbf{x}) | \mathbf{w} \rangle_{\mathcal{H}}$, where $\mathcal{H}$ is the feature space of the above kernel. Hence $F_{\mathbf{w}}$ is linear in $\mathbf{w}$ since the convolution is a linear operation. The parameter γ is a regularization parameter. The actual training procedure is a simple ridge regression scheme.

In our experiments we use images of size 256^2. Note that each pixel-neighborhood is regarded as one training sample. Due to the overlap of the neighborhoods the cost function couples training samples. The grayvalues of the images are scaled to $[0,1]$. The neighborhood of a pixel is represented by $N = 8$ coefficients, where the width of the Gaussian is around 10 pixels. The corresponding feature-space is of dimension $N(N+1)/2 + N = 44$. As already mentioned the feature-space dimension is much less than the number of training-samples (256^2) and hence working in feature-space is recommended.

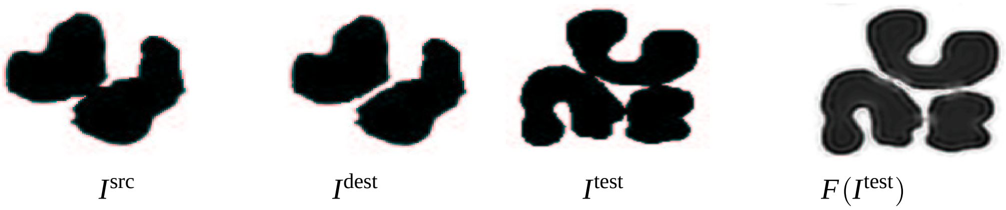


Figure 2: Results for the image transformation example. Images I^{src} and I^{dest} where used for training. Image I^{test} show the image for testing, the three connected should be separated. $F(I^{\text{src}})$ shows the image after application of the trained image transformation.

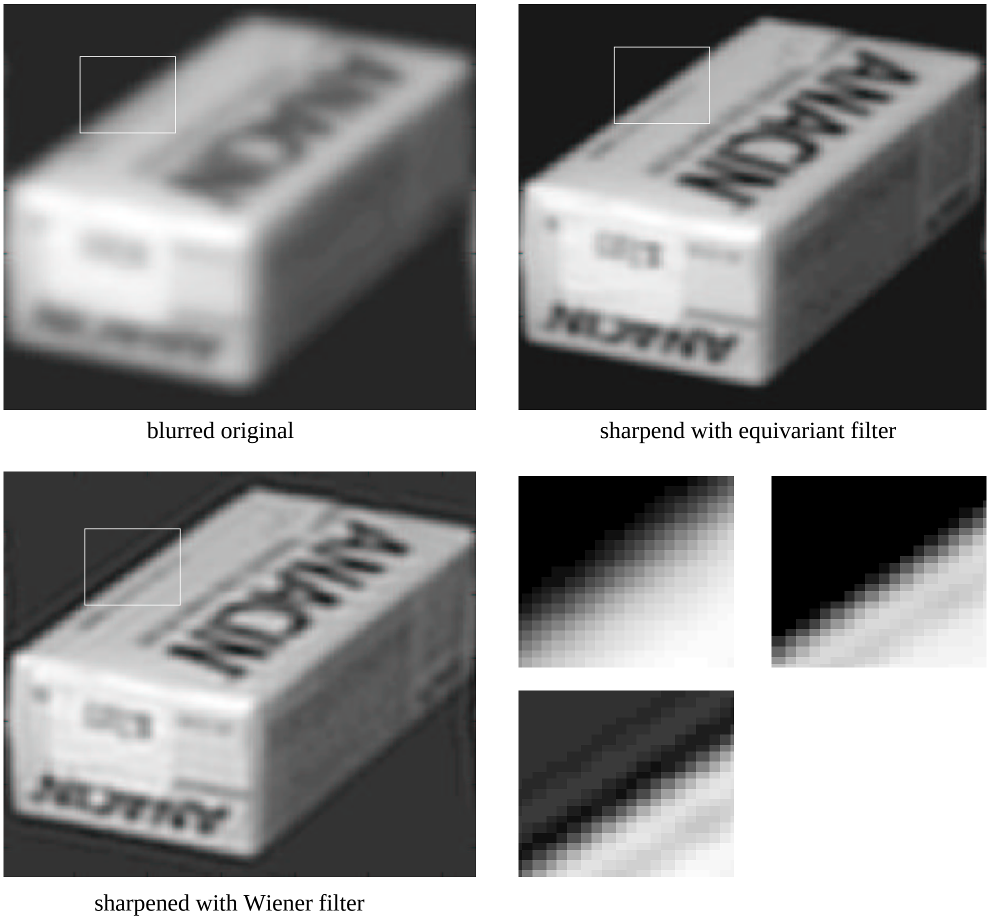
We conduct two experiments. First we train the filter to separate connected clusters of pixels. The left two images in Figure 2 show the images I^{src}, I^{dest} used for training. As the equation system is highly overdetermined the regularization parameter γ can be chosen very small. If the training images are distorted by a small amount of independent Gaussian noise we can actually choose $\gamma = 0$. In general we can conclude that the higher the regularization γ is the less effective is our filter, but the more robust the filter is against additive noise. This is a well known behavior.

The right two images show a test image in its original appearance and after the application of the trained filter. One can see that the filter actually works properly. Of course, some artifacts are created. One can nicely see the equivariance of the filter. The behavior depends neither on the orientation of the touching areas nor the absolute position. In fact, the filter may be interpreted as a locally applied quadratic (because of the second-order polynomial kernel) Volterra filter. The test image I^{test} was chosen such that common methods would not work properly. For example, if one use a distance transform, the agglomerate would be divided in five instead three clusters, because the distance transform gives five local maxima.

As a second experiment we trained the filter to perform a deconvolution. For training we used a black disk on white background as target I^{dest} and a blurred version as I^{src} (blurred with a Gaussian of width 2 percent of the image size). In Figure 3 a) we show a blurred image which has to be sharpened (blurred with the same width as used for training). We compare our filter with an ordinary Wiener filter for deconvolution (see, for example, Jain, 1989). Assuming decorrelated Gaussian noise models, the Wiener filter only depends on the signal-to-noise ratio, which controls the intensity of the sharpening of the image. In our approach the regularization parameter γ plays a similar role as the signal-to-noise ratio for the Wiener filter. We tuned the parameters such that the obtained sharpness for both approaches are visually comparable. In Figure 3 b) we give the result obtained by our method, in c) the result for the Wiener filter. Our method produces a more natural looking image and more importantly it produces no over- and undershoots at the sharp edges like the Wiener filter.

5.3 A Rotation Equivariant Object Detector

The goal of the object detector is to find a specific object in an image independently of its rotational or translational pose. The proposed object detector may be seen as some kind of generalized hough transform (see Ballard, 1981), where the lookup-table is is replaced by an equivariant function,



blurred original

sharpend with equivariant filter

sharpened with Wiener filter

Figure 3: Results for the image restoration example. The Wiener filter produces overshoot- and undershoot-artifacts; this is a well known problem for inverse linear filters. Our non-linear filter creates a fine and accurate edge while having the same sharpness as the Wiener filter. (have a look at these results in an electronic form, a printer may destroy the fine differences)

which is learned from a shape template. It has also some similarities to the object recognition system using the so called SIFT features (Lowe, 2004).

At first, stable interest points in the image have to be detected, for example, points of high variation or something similar. Relative to these interest points the algorithm votes for a predefined center of the object. Assume we want to detect rectangles in an image. Corners are an important and stable feature of a rectangle. Relative to a corner of a rectangle we have obviously two hypotheses for the putatively rectangle center. To cope with such multiple hypotheses in general we do not make discrete votes but vote for all possible points with its corresponding probability that there may be the center of the object. Our goal is to learn a mapping that maps features from the local neighborhood of the interest points to a probability distribution for the object's center. To detect the center of the object independently of its rotational pose this mapping has to be rotation-equivariant.

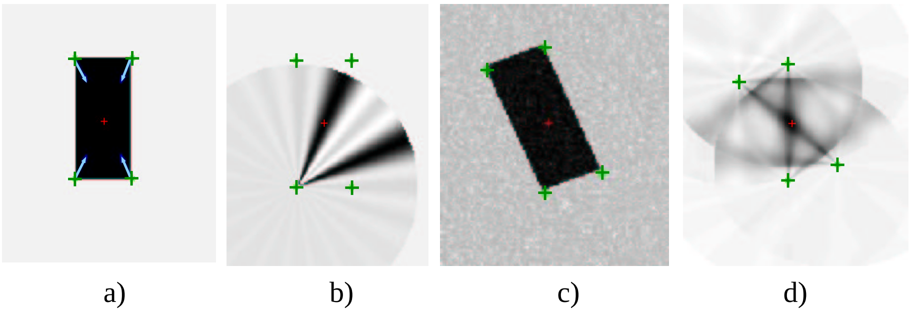


Figure 4: Example for the rectangle. a) The rectangle with its four interest points and its center. b) The response of the trained voting function $\mathbf{p}(\mathbf{x})$ for the lower left interest point (as mentioned the distance to the center is not learned, only the direction). The function votes for the center of the shown rectangle and for the center of a putatively rotated rectangle by 90 degrees. c) an rotated version of the training object with additive Gaussian noise. d) the superimposed responses of all four interest points. The maxima is obviously in the center of the rectangle.

For simplicity we want to restrict ourselves to vote for points in the same specific direction with the same probability or weight, that is, the output of the equivariant-mapping is a probability distribution defined on the circle.

For the features we use the same projections on local basis-functions as in the previous experiment. To find the interest points we search for local maxima of the absolute value $|[\mathbf{x}_z]_2|$, which basically gives responses for corner-like structures. To eliminate responses at edges, which are rather unstable, we only take those local maxima whose values for the quotient $|[\mathbf{x}_z]_2|/|[\mathbf{x}_z]_1|$ are above a given threshold. Let $\mathbf{x}$ be a feature vector around some interest point and ϕ the angle to the center of the object. We are interested to learn the conditional distribution $p(\phi|\mathbf{x})$ from training samples $(\phi_i, \mathbf{x}_i)$ belonging to the training object's interest points. As we want to detect the objects in a rotation invariant manner, the distribution has to fulfill rotation equivariance $p(\phi|g_\varphi \mathbf{x}) = p(\phi + \varphi|\mathbf{x})$. In Section 4 we worked out that it is advantageous to work with irreducible kernels. As already mentioned the irreducible kernels of the 2D-rotation group act on the Fourier representation of functions, so we denote $p(\phi|\mathbf{x})$ by the vector $\mathbf{p}(\mathbf{x})$, where its components are the Fourier coefficients of $p(\phi|\mathbf{x})$ in respect to ϕ. We denote the cutoff-frequency of the Fourier representation by N.

We do not rigorously model $\mathbf{p}$ as a probability distribution. Let $\mathbf{e}_\phi$ be the unit-pulse in Fourier representation, then we minimize

$$\sum_i ||\mathbf{e}_{\phi_i} - \mathbf{p}(\mathbf{x}_i)||^2,$$

which is nothing else then the ordinary least-square regression scheme. But the solution basically behaves as we want it to. To get an idea, consider again the rectangle example with its four corners as interest points. If we neglect noise the local features $\mathbf{x}_i$ of the corners are all the same $\mathbf{x}_i = g_{\varphi_i}\mathbf{x}_0$ up to rotations g_{φ_i} of $0, 90, 180, 270$ degree. Then the equivariant least-square minimizer $\mathbf{p}(\mathbf{x}_0)$ is proportional to the Fourier transformed histogram of the relative angles $\phi_i + \varphi_i$. The reader should have an antipicatory look at Figure 4 a) and b), where this is illustrated. One can see that $\mathbf{p}(\mathbf{x}_0)$ give contributions for both possible corner directions.

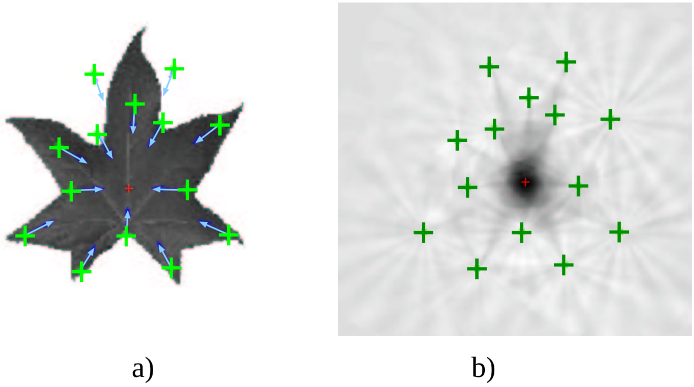


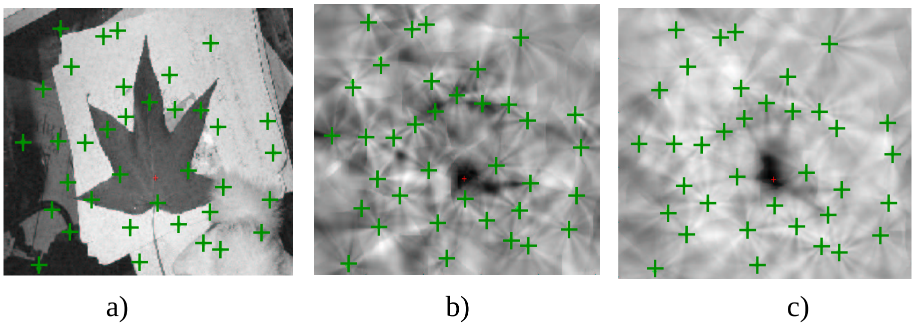
Figure 5: Training image a) The only training leaf. The interest points for training are marked and its corresponding direction to the center. b) The voting map for the training object. Black means high probability for the center, white low probability.

The scalar basis kernel is chosen to be the Gaussian kernel $k(\mathbf{x}_1,\mathbf{x}_2) = e^{-\lambda||\mathbf{x}_1-\mathbf{x}_2||^2}$. The diagonal matrix kernel is approximated by using the FFT as depicted in Section 5.1. To get good approximations we have to perform frequency padding. In the experiments we used a FFT of size $6N$ (this is enough to compute a third-order polynomial kernel accurately). For accurate computation of the exponential kernel an infinite size FT would be needed, but the experiments have shown that an approximation is enough. The final learning procedure is very simple. We just have to solve N linear equations of the form

$$\mathbf{K}_n\mathbf{a}_n = \mathbf{b}_n \qquad n = 0,\ldots,N-1,$$

where the matrix entries of $(\mathbf{K}_n)_{i,j} = K_n(\mathbf{x}_i,\mathbf{x}_j)$ are the approximations of Equation (7), and the entries of the target are $[\mathbf{b}_n]_i = e^{\mathbf{i}n\phi_i}$.

In Figure 4 we illustrate results for the rectangle example. In a) the training example with its four interest points is shown and b) shows the contribution of the voting function of the lower left corner. The sinusoidal artifacts are stemming from the finite Fourier representation. In c) and d) we applied our object detector to a rotated and noisy rectangle. The results are satisfying, the voting function $\mathbf{p}(\mathbf{x})$ obviously behaves in an equivariant manner and is robust against small distortions. To make a more realistic example, we train our object detector to find leaves on natural background. In Figure 5 a) a segmented leaf for training is shown. We found by hand that 14 interest points are relatively stable. Of course, this number, depends on the size of the Gaussian used for feature computation (in our case the width is four percent of the image size, that is, rather large). After training we applied the object detector on the train image itself, which is shown in Figure 5 b). For each interest point we superimpose its contributions within a distance of about forty percent of the image size. To get equal contribution from each point we normalize the output of our voting function $\mathbf{p}(\mathbf{x})/||\mathbf{p}(\mathbf{x})||$. To show the stability and generalization ability of our object detector, we applied it on a more complex image shown in Figure 6 a). The leaf is from the same species as the training leaf but obviously different. The resulting voting map is shown in Figure 6 b), in fact, the overall maximum is in the center of the leaf, but it seems that the maximum is relatively unstable depending on the background. To improve the result, we modified the kernel. As explained in Section 3 we can use Lemma 3.2 to compute an invariant kernel and modify our matrix kernel by



a)

b)

c)

Figure 6: A test image of a leaf with background and small clutter. a) The testimage with the found interest points. b) The voting map for test image with the simple matrix kernel. c) The voting map achieved with the enhanced kernel for better selectivity.

$K' = K\ \mathrm{tr}(K^{\dagger}K)$ to give it more selectivity to the features itself. In Figure 6 c) the voting map with this enhanced kernel is shown, the maximum seems to be much more robust.

6. Conclusion and Outlook

We presented a new type of matrix valued kernel for learning equivariant functions. Several properties were shown and connections to representation theory were established. We showed with two illustrative examples that the theory is applicable.

The usage in nonlinear signal and image processing are apparent. Problems like image denoising, enhancement and image morphing can be tackled. Another simple task for the equivariant filter would be to gaps in road networks.

The proposed object detector shows in spite of its simplicity a nice behavior and generalization ability and should be worth to improve. Generalizations of our proposed experiments to 3D-rotations are straightforward.

From a theoretical point an extension of our framework to non-compact groups would be satisfying. Another important challenge is to find sparse learning algorithms with an unitary invariant loss functional. Unfortunately, the unitary extension of the ε-insensitive loss is not solvable via quadratic programming anymore. A last important issue is the local nature of transformations. In hand-written digit recognition invariance under rotations might turn a ”6” into a ”9”, while rotations by 5 degrees are ok. A generalization of our theory using notions like approximate or partial equivariance are necessary.

Acknowledgments

The authors would like to acknowledge the anonymous reviewers and the action editor for their useful suggestions and comments that have greatly improved the presentation of the paper. Further we would like to thank Nikos Canterakis and Bernard Haasdonk for carefully reading the manuscript in an early stage. And finally, thanks to Olaf Ronneberger for the fruitful discussions and inspiration.

References

L. Amodei. Reproducing kernels of vector-valued function spaces. In *Surface Fitting and Multiresolution Methods, A. Le Maut C. Rabut and L. L. Schumaker (eds.)*, pages 17–26, 1996.

N. Aronszajn. Theory of reproducing kernels. *Trans. AMS*, 686:337404, 1950.

D.H. Ballard. Generalizing the hough transform to detect arbitrary shapes. *Pattern Recognition*, 13-2, 1981.

S. Boyd, L. O. Chua, and C. A. Desoer. Analytical foundations of Volterra series. *IMA Journal of Mathematical Control and Information*, 1:243–282, 1984.

J. Burbea and P. Masani. Banach and Hilbert spaces of vector-valued functions. *Pitman Research Notes in Mathematics*, 90, 1984.

H. Burkhardt and S. Siggelkow. Invariant features in pattern recognition - fundamentals and applications. In *In Nonlinear Model-Based Image/Video Processing and Analysis*, pages 269–307. John Wiley and Sons, 2001.

N. Canterakis. Vollstaendige minimale systeme von polynominvarianten fuer die zyklischen gruppen g(n) und fuer g(n) x g(n). In *In Tagungsband des 9. Kolloquiums - DFG Schwerpunkt: Digitale Signalverarbeitung*, pages 13–17, 1986.

T.J. Dodd and R.F. Harrison. Estimating Volterra filters in Hilbert space. In *Proceedings of IFAC Conference on Intelligent Control Systems and Signal Processing*, 2003.

A. Gaal. *Linear Analysis and Represenatation Theory*. Springer Verlag, New York, 1973.

B. Haasdonk, A. Vossen, and H. Burkhardt. Invariance in kernel methods by haar-integration kernels. In *Proceedings of the 14th Scandinavian Conference on Image Analysis*, pages 841–851, 2005.

A.K. Jain. *Fundamentals of Digital Image Processing*. Prentice-Hall, Inc., Englewood Cliffs, N.J., USA, 1989.

G.S. Kimeldorf and G. Wahba. Some results on Tchebycheffian spline functions. *Journal of Mathematical Analysis and Applications*, 33:82–95, 1971.

R. Lenz. *Group Theoretical Methods in Image Processing*. Springer Verlag, Lecture Notes, 1990.

D.G. Lowe. Distinct image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60:91–110, 2004.

C.A. Micchelli and M. Pontil. On learning vector-valued functions. *Neural Computation*, 17:177–204, 2005.

W. Miller. Topics in harmonic analysis with applications to radar and sonar. *IMA Volumes in Mathematics and its Applications*, 1991.

D. Mumford, J. Fogarty, and F. Kirwan. *Geometric Invariant Theory*. Springer, 1994.

L. Nachbin. *The Haar Integral*. D. van Nostrand Company, Inc., Princenton, New Jersey, Toronto, New York, London, 1965.

M. Pontil and C.A. Micchelli. Kernels for multi-task learning. In *Proceeding of the NIPS*, 2004.

V. I. Paulsen R. G. Douglas. *Hilbert Modules over Function Algebras*. Wiley New York, 1989.

M. Reisert and H. Burkhardt. Averaging similarity weighted group representations for pose estimation. In *Proceedings of IVCNZ'05*, 2005.

B. Schoelkopf and A. J. Smola. *Learning with Kernels*. The MIT Press, 2002.

I. Schur. *Vorlesungen ueber Invariantentheorie*. Springer Verlag, Berlin, Heidelberg, New York, 1968.

J. Shawe-Taylor and N. Cristianini. *Kernel Methods for Pattern Analysis*. Cambridge University Press, 2004.