Vector Field Learning via Spectral Filtering

Luca Baldassarre[1], Lorenzo Rosasco[2,3], Annalisa Barla[1], and Alessandro Verri[1]

[1] Università degli Studi di Genova - DISI
Via Dodecaneso 35, Genova, Italy
{baldassarre,barla,verri}@disi.unige.it
[2] Istituto Italiano di Tecnologia,
Via Morego, 30 16163 Genova, Italy
[3] CBCL, Massachusetts Institute of Technology
Cambridge, MA 02139 - USA
lrosasco@mit.edu

Abstract. In this paper we present and study a new class of regularized kernel methods for learning vector fields, which are based on filtering the spectrum of the kernel matrix. These methods include Tikhonov regularization as a special case, as well as interesting alternatives such as vector valued extensions of L2-Boosting. Our theoretical and experimental analysis shows that spectral filters that yield iterative algorithms, such as L2-Boosting, are much faster than Tikhonov regularization and attain the same prediction performances. Finite sample bounds for the different filters can be derived in a common framework and highlight different theoretical properties of the methods. The theory of vector valued reproducing kernel Hilbert space is a key tool in our study.

Keywords: Vector-valued Functions; Multi-task; Regularization; Spectral Filtering; Kernels.

1 Introduction

In this paper we study theoretical and computational properties of a class of kernel methods for learning a vector valued function. These methods are based on filtering the spectrum of the kernel matrix rather than empirical risk minimization. The idea of using kernel methods for vector field learning has been considered in [1] where the framework of vector valued reproducing kernel Hilbert spaces was adopted and the representer theorem for Tikhonov regularization was generalized to the vector valued setting. Our work can be seen as an extension of the work in [1] aimed in particular at: 1) investigating the application of spectral filtering schemes [2] to learning vector fields; 2) establishing consistency and finite sample bounds for Tikhonov regularization as well as for all spectral filters in the setting of vector valued learning. One of the main outcomes of our study is that iterative algorithms based on spectral filtering outperform Tikhonov regularization from the computational perspective, while preserving the good prediction performances.

J.L. Balcázar et al. (Eds.): ECML PKDD 2010, Part I, LNAI 6321, pp. 56–71, 2010.

Classical supervised learning focuses on the problem of estimating functions with scalar outputs: a real number in regression and one between two possible labels in binary classification. The starting point of our investigation is the observation that in many practical problems it is convenient to model the object of interest as a function with multiple outputs. In machine learning, this problem typically goes under the name of multi-task or multi-output learning and has recently attracted a certain attention. It is interesting to recognize at least two classes of problems with multiple output functions. The first class, that we might call multi-task learning, corresponds to the situation in which we have to solve several standard scalar learning problems (each with its own training set) that we assume to be related, so that we can expect to obtain a better solution if we attempt to solve them simultaneously. Application in user recommendation systems can be taken as an example. The second class of problems corresponds to learning vector valued functions. This situation is better described as a supervised learning problem where the outputs are vector valued and we have a single training set. For example, a practical problem is that of estimating the velocity field of an incompressible fluid from scattered spatial measurements.

The two problems are clearly related. Indeed, we can view tasks as components of a vector valued function or equivalently learning each component of a vector valued function as one of many scalar tasks. Nonetheless, there are also some differences that make the two problems different both from a practical and a theoretical point of view. For example, in multi-task learning the input points for each task can be represented by different features and the sample size might vary from one task to the other. In particular, each task can be sampled in a different way so that, in some situations, by assuming that the tasks are highly correlated, we can essentially augment the number of effective points available for each individual task. This effect does not occur while learning vector fields where each component is sampled at the same input points. Since the sampling procedures are different, the error analyses for multi-task and vector valued learning are also different. The latter case is closer to the scalar setting, whereas in the multi-task case the situation is more complex: one might have different cardinalities for the various tasks or be interested to evaluate the individual performance of each task.

In this paper, we focus primarily on vector field learning as a natural extension of the classical scalar setting. In particular, some of the theoretical results are specific to vector valued functions, but many of the computational ideas we discuss apply to general multi-task problems. We propose a new class of algorithms to learn multi-output functions, called *spectral filters*, that cannot be described in terms of penalized empirical risk minimization. Each algorithm performs a different filtering of the spectrum of the kernel matrix, designed to suppress contributions corresponding to small eigenvalues. These algorithms are motivated by the results connecting learning theory and regularization of ill-posed problems [3] and the relations between stability and generalization [4,5]. We provide a detailed computational analysis that takes into account the specific form of the kernel as well the regularization parameter choice step, from which it is clear

that the various methods have different computational properties. The bound on the excess risk further illustrates these differences. Our experiments confirm that iterative spectral filters, that can be seen as extensions of L2 boosting [6] are often preferable to Tikhonov regularization.

The plan of the paper follows. After discussing previous work in the next subsection, in Sect.2 we recall some basic concepts. in Sect.3 we present the class of algorithms under study, while in Sect.4 we review examples of kernels. The finite sample bound on the excess risk and computational issues are discussed in Sect.5. The experimental analysis is conducted in Sect.6 and we conclude in Sect.7 proposing some future work.

1.1 Previous Work

Several recent works considered multi-output learning, especially multi-task, and proposed a variety of approaches. Starting from the work of [7], related ideas have been developed in the context of regularization methods [8], Gaussian processes [9,10]. The specific problem of learning a vector valued function has received less attention in machine learning. In statistics we mention the Curds & Whey method [11], Reduced Rank Regression [12], Filtered Canonical y-variate Regression [13] and Partial Least Squares [14]. Estimating vector fields is common in the context of geophysics and goes under the name of co-kriging [15]. Some attempts to extend machine learning algorithms from the scalar to the vector setting have been made [16,17]. A study of vector valued learning with kernel methods is started in [1], where regularized least squares are analyzed from the computational point of view. The error analysis of vector valued Tikhonov regularization is given in [18]. To the best of our knowledge the application of spectral filtering techniques to vector field learning has not yet been studied.

2 Basic Concepts

We start by presenting the setup of the problem, as well as the basic notions behind the theory of vector valued reproducing kernels.

Supervised Learning. The problem of supervised learning amounts to inferring an unknown functional relation given a finite *training set* of input-output pairs $\mathbf{z} = \{(x_i, y_i)\}_{i=1}^n$ that are assumed to be identically and independently distributed according to a fixed, but unknown probability measure $\rho(x, y) = \rho_X(x)\rho(y|x)$ on $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$. Here we are interested in vector valued learning where $\mathcal{Y} \subseteq \mathbb{R}^T$. A learning algorithm is a map from a training set $\mathbf{z}$ to an estimator $f_{\mathbf{z}} : \mathcal{X} \to \mathcal{Y}$. A good estimator should generalize to future examples and, if we choose the square loss, this translates into the requirement of having small *expected risk*

$$\mathcal{E}(f) = \int_{\mathcal{X}\times\mathcal{Y}} \|y - f(x)\|_T^2 d\rho(x, y) \ ,$$

where $\|\cdot\|_T$ denotes the euclidean norm in $\mathbb{R}^T$. The ideal estimator is the minimizer of the expected risk, that is the regression function $f_\rho(x) = \int_{\mathcal{Y}} y\rho(y|x)$,

but cannot be directly calculated since ρ is unknown. Further, the search for a solution is often restricted to some space of hypotheses $\mathcal{H}$. In this case the best attainable error is $\mathcal{E}(f_{\mathcal{H}}) = \inf_{f\in\mathcal{H}} \mathcal{E}(f)$. The quality of an estimator can then be assessed considering the distribution of the *excess risk*, $\mathcal{E}(f_{\mathbf{z}}) - \mathcal{E}(f_{\mathcal{H}})$, and in particular we say that an estimator is consistent if

$$\lim_{n\to\infty} \mathrm{P}\left[\mathcal{E}(f_{\mathbf{z}}) - \mathcal{E}(f_{\mathcal{H}}) \geq \varepsilon\right] = 0$$

for all positive ε. A more quantitative result is given by finite sample bounds,

$$\mathrm{P}\left[\mathcal{E}(f_{\mathbf{z}}) - \mathcal{E}(f_{\mathcal{H}}) \geq \varepsilon(\eta, n)\right] \leq 1 - \eta\ , \quad 0 < \eta \leq 1\ .$$

In the following we'll be interested into hypotheses space defined by a kernel.

Vector Valued RKHS. The development of the theory of RKHS in the vector case is essentially the same as in the scalar case and we refer to [1,19] for further details and references. We consider functions having values in some euclidean space $\mathcal{Y} \subseteq \mathbb{R}^T$ with scalar product (norm) $\langle\cdot,\cdot\rangle_T$, ($\|\cdot\|_T$). A RKH space is a Hilbert space of functions $f : \mathcal{X} \to \mathbb{R}^T$, with scalar product (norm) denoted by $\langle\cdot,\cdot\rangle_\Gamma$ ($\|\cdot\|_\Gamma$), defined by a matrix valued kernel $\Gamma : \mathcal{X}\times\mathcal{X} \to \mathcal{B}(\mathbb{R}^T)$, where $\mathcal{B}(\mathbb{R}^T)$ is the space of $T \times T$ positive semi-definite matrices.
The kernel Γ has the following reproducing property: for all $c \in \mathbb{R}^T$ and $x \in \mathcal{X}$

$$\langle f(x), c\rangle_T = \langle f, \Gamma(\cdot, x)c\rangle_\Gamma\ . \tag{1}$$

We assume throughout that $\sup_{x\in\mathcal{X}} ||\Gamma(x,x)|| = \kappa < \infty$, where $\|\cdot\|$ is the operator norm, which implies $\|f(x)\|_T \leq \kappa\|f\|_\Gamma$. Similarly to the scalar case, it can be shown that for any reproducing kernel Γ, a unique RKHS can be defined.

3 Learning Vector Fields with Spectral Filtering

In this section we present the new class of algorithms that we study in this paper. Towards this end it is instructive to preliminarily recall the main features of Tikhonov regularization for scalar and vector problems.

3.1 Tikhonov Regularization

In the scalar case, Tikhonov regularization [20,21] in a RKHS $\mathcal{H}$, with kernel K, corresponds to the minimization problem

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

Its solution is given by

$$f^\lambda_{\mathbf{z}}(\cdot) = \sum_{i=1}^{n} K(x_i,\cdot)c_i \qquad \text{with} \qquad (\mathbf{K} + \lambda n I)\mathbf{c} = \mathbf{y}\ , \tag{2}$$

with $\mathbf{K}_{ij} = K(x_i, x_j)$, $\mathbf{y} = (y_1, \ldots, y_n)$, $c_i \in \mathbb{R}$ and $\mathbf{c} = (c_1, \ldots, c_n)$. The final estimator $f_{\mathbf{z}}$ is determined by a parameter choice $\lambda_n = \lambda(n, \mathbf{z})$, so that $f_{\mathbf{z}} = f_{\mathbf{z}}^{\lambda}$.

In the case of vector valued output, i.e. $\mathcal{Y} \subseteq \mathbb{R}^T$, the simplest idea is to consider a naïve extension of Tikhonov regularization, reducing the problem to learning each component independently. Namely, the solution is assumed to belong to $\mathcal{H} = \mathcal{H}^1 \times \mathcal{H}^2 \cdots \times \mathcal{H}^T$, where the spaces $\mathcal{H}^1, \mathcal{H}^2, \ldots, \mathcal{H}^T$ are RKHS with norms $\|\cdot\|_{\mathcal{H}^1}, \ldots, \|\cdot\|_{\mathcal{H}^T}$. Then $f = (f^1, \ldots, f^T)$ and $\|f\|_I^2 = \sum_{j=1}^T \|f^j\|_{\mathcal{H}^j}^2$ and Tikhonov regularization amounts to solving the following problem

$$\min_{f^1 \in \mathcal{H}^1, \ldots, f^T \in \mathcal{H}^T} \{ \frac{1}{n} \sum_{i=1}^{n} \sum_{j=1}^{T} (y_i^j - f^j(x_i))^2 + \lambda \sum_{j=1}^{T} \|f^j\|_{\mathcal{H}^j}^2 \} \,, \tag{3}$$

which is equivalent to solve T *independent* scalar problems. Within the framework of vector valued RKHSs, the choice above corresponds to a diagonal matrix valued kernel of the form $\Gamma(x, x') = diag(K_1(x, x'), \ldots, K_T(x, x'))$.

Recently, a regularization scheme of the form (3) has been studied in [1] for general matrix valued kernels. In this case there is *no straightforward* decomposition of the problem and one of the main results in [1] shows that the regularized solution can be written as

$$f_{\mathbf{z}}^{\lambda}(\cdot) = \sum_{i=1}^{n} \Gamma(\cdot, x_i) c_i \qquad \text{with} \qquad (\mathbf{\Gamma} + \lambda n I)\mathbf{C} = \mathbf{Y} \,, \tag{4}$$

where $c_i \in \mathbb{R}^T$, $\mathbf{C} = (c_1, \ldots, c_n)$, $\mathbf{Y} = (y_1, \ldots, y_n)$ and the kernel matrix $\mathbf{\Gamma}$ is a $n \times n$ block matrix, where each block is a $T \times T$ scalar matrix, so that $\mathbf{\Gamma}$ is a $nT \times nT$ scalar matrix.

The above discussion highlights a first interesting observation. Assuming the components of the vector field to be independent results in a block diagonal kernel matrix. Conversely, working with a non-diagonal matrix, hence with a general kernel, it is possible to exploit the functional relations that exists among the components of the vector field.

3.2 Regularization via Spectral Filtering

We present the class of regularized kernel methods under study, referring to [2,22] for the scalar case. We call these methods *spectral filters* because they achieve a stable, hence generalizing, solution by *filtering out* the unstable components of the kernel matrix, that is the directions corresponding to small eigenvalues. The interpretation of regularization as a way to restore stability is classical in ill-posed inverse problems, where many algorithms besides Tikhonov regularization are used [23]. The connection between learning and regularization theory of ill-posed problems [3] motivates considering spectral filtering techniques.

Adding a penalty to the empirical risk has a stabilizing effect from a numerical point of view, since it transforms the problem from $\mathbf{\Gamma}\mathbf{C} = \mathbf{Y}$ to $(\mathbf{\Gamma} + \lambda n \mathbf{I})\mathbf{C} = \mathbf{Y}$. Hence, the penalty reduces the instability due to the eigenvectors corresponding to the small eigenvalues of the kernel matrix.

The idea of spectral filtering algorithms is that other regularized matrices $g_\lambda(\mathbf{\Gamma})$ besides $(\mathbf{\Gamma} + \lambda n \mathbf{I})^{-1}$ can be defined. Each algorithm corresponds to a specific filter function and in general there is no natural interpretation in terms of penalized empirical risk minimization. The matrix valued function $g_\lambda(\mathbf{\Gamma})$ is described by a scalar function g_λ using spectral calculus. More precisely, if $\mathbf{\Gamma} = \mathbf{USU}^*$ is the eigendecomposition of $\mathbf{\Gamma}$ with $\mathbf{S} = diag(\sigma_1, \ldots, \sigma_n)$, then $g_\lambda(\mathbf{S}) = diag(g_\lambda(\sigma_1), \ldots, g_\lambda(\sigma_n))$ and $g_\lambda(\mathbf{\Gamma}) = \mathbf{U} g_\lambda(\mathbf{S}) \mathbf{U}^*$. For example, in the case of Tikhonov regularization $g_\lambda(\sigma) = \frac{1}{\sigma + n\lambda}$.

Suitable choices of filter functions g_λ define estimators of the form (4) with coefficients given by

$$\mathbf{C} = g_\lambda(\mathbf{\Gamma})\mathbf{Y} \ . \tag{5}$$

From the computational perspective, a key point that we show in the following is that many filter functions allow to compute the coefficients $\mathbf{C}$ without explicitly computing the eigen-decomposition of $\mathbf{\Gamma}$.

Clearly not all filter functions are admissible. Roughly speaking, as λ decreases an admissible filter function $g_\lambda(\mathbf{\Gamma})$ should approximate $\mathbf{\Gamma}^{-1}$ and its condition number should increase.

Remark 1. Note that in the scalar case, manipulations of the kernel matrix have been extensively used to define (and learn) new kernels to be used in Tikhonov regularization [24]. In the approach we present here, rather than defining a new kernel, each spectral filter g_λ defines an *algorithm* which is not based on empirical risk minimization.

3.3 Examples of Spectral Regularization Algorithms

We proceed giving several examples of spectral filtering algorithms.

L2 Boosting. In the scalar setting this method has been interpreted as a way to combine weak classifiers corresponding to splines functions at the training set points [6] and is called Landweber iteration in inverse problems literature [23]. The method can also be seen as the gradient descent minimization of the empirical risk on the whole RKHS, with no further constraint. Regularization is achieved by early stopping of the iterative procedure, hence the regularization parameter is the number of iterations.
The coefficients (5) can be found by setting $\mathbf{C}^0 = 0$ and considering for $i = 1, \ldots, t$ the following iteration

$$\mathbf{C}^i = \mathbf{C}^{i-1} + \eta(\mathbf{Y} - \mathbf{\Gamma}\mathbf{C}^{i-1}) \ ,$$

where the step size η can be chosen to make the iterations converge to the minimizer of the empirical risk, see below. If we use (4) to write the empirical risk as $\|\mathbf{\Gamma}\mathbf{C} - \mathbf{Y}\|^2$, it is easy to see that this is simply gradient descent. Further, it is can be shown by induction that the solution at the $t-$th iteration is given by

$$\mathbf{C}^t = \eta \sum_{i=0}^{t-1} (I - \eta\mathbf{\Gamma})^i \mathbf{Y} \ ,$$

and it follows that the filter function is $G_\lambda(\sigma) = \eta \sum_{i=0}^{t-1}(I - \eta\sigma)^i$. Interestingly, this filter function has another interpretation that can be seen recalling that, given a matrix A, $\|A\| \in (0,1)$ ($||\cdot||$ is the operator norm), $\sum_{i=0}^{\infty} A^i = \frac{1}{\mathbf{I}-A}$. If we replace A with $I-\eta\mathbf{\Gamma}$ and $\|I-\eta\mathbf{\Gamma}\| < 1$, we get $\mathbf{\Gamma}^{-1} = \eta \sum_{i=0}^{\infty}(I-\eta\mathbf{\Gamma})^i$. Therefore, the filter function of L2 Boosting corresponds to the truncated power series expansion of $\mathbf{\Gamma}^{-1}$. The last reasoning also shows a possible way to choose the step-size. In fact, choosing $\eta = 1/\sigma_{max}$, where σ_{max} is the maximum eigenvalue of the kernel matrix $\mathbf{\Gamma}$, we are guaranteed that $\|I - \eta\mathbf{\Gamma}\| < 1$.

Accelerated L2 Boosting. This method is also called the ν-method and it is particularly interesting since it is significantly faster than L2 boosting. Usually, it can find the same solution in only $\sqrt{t}$ steps. The coefficients are found by setting $\mathbf{C}^0 = 0$, $\omega_1 = (4\nu + 2)/(4\nu + 1)$, $\mathbf{C}^1 = \mathbf{C}^0 + \frac{\omega_1}{n}(\mathbf{Y} - \mathbf{\Gamma}\mathbf{C}^0)$ and considering for $i = 2, \ldots, t$ the iteration given by

$$\mathbf{C}^i = \mathbf{C}^{i-1} + u_i(\mathbf{C}^{i-1} - \mathbf{C}^{i-2}) + \frac{\omega_i}{n}(\mathbf{Y} - \mathbf{\Gamma}\mathbf{C}^{i-1}) \, .$$

The derivation of the filter function is considerably more complicated and is given in [23], where the parameters ν, ω_i and u_i are also defined. The filter function is shown to be $G_t(\sigma) = p_t(\sigma)$ with p_t a polynomial of degree $t-1$. This method can be proved to be faster than L2 boosting since the ν-method can find in $\sqrt{t}$ steps the same solution found by L2 boosting after t iterations. regularization parameter is the *square root* of the iteration number rather than the iteration number itself.

Iterated Tikhonov. This method is a combination of Tikhonov regularization and L2 boosting where we set $\mathbf{C}^0 = 0$ and consider for $i = 1, \ldots, t$ the iteration $(\mathbf{\Gamma} + n\lambda I)\mathbf{C}^i = \mathbf{Y} + n\lambda\mathbf{C}^{i-1}$. The filter function is:

$$G_\lambda(\sigma) = \frac{(\sigma + n\lambda)^t - (n\lambda)^t}{\sigma(\sigma + n\lambda)^t} \, .$$

This methods is motivated by the desire to circumvent some of the limitations of Tikhonov regularization, namely a saturation effect that prevents exploiting the smoothness of the target function beyond a given critical value– see [23,2] for further details.

Truncated Singular Values Decomposition. This method is akin to a projection onto the first principal components in a vector valued setting. The number of components depends on the regularization parameter. The filter function is defined as $G_\lambda(\sigma) = 1/\sigma$ if $\sigma \geq \lambda/n$ and 0 otherwise.

4 Matrix Valued Kernels

In this section, we briefly review some matrix valued kernels for vector fields learning - see [25,26,27]. In particular we discuss a class of kernels that leads to a faster implementation of spectral filtering algorithms. Before doing this we give an example of a general kernel that we will consider in our experimental section.

Divergence free and curl free fields. These kernels have been used in [28] for the problem of reconstructing divergence-free or curl-free vector fields and they can be used only for vector fields whose input and the output spaces have the same dimensions. The divergence-free kernel is

$$\Gamma_{df}(x, x') = \frac{1}{\sigma^2} e^{-\frac{||x-y||^2}{2\sigma^2}} A_{x,x'} \tag{6}$$

where

$$A_{x,x'} = \frac{(x - x')(x - x')^T}{\sigma^2} + \left((T-1) - \frac{||x - x'||^2}{\sigma^2} \right) \mathbf{I}$$

and the curl-free is

$$\Gamma_{cf}(x, x') = \frac{1}{\sigma^2} e^{-\frac{||x-x'||^2}{2\sigma^2}} \left(\mathbf{I} - \frac{(x - x')(x - x')^T}{\sigma^2} \right) . \tag{7}$$

It is possible to consider a convex linear combination of these two kernels for learning any vector field and for reconstructing its divergence-free and curl-free parts separately (see the experiments in Sect.6).

4.1 Design of Decomposable Kernels.

A general class of kernels consists of kernels of the form

$$\Gamma(x, x') = K(x, x')A \tag{8}$$

where K is a scalar kernel and A a positive semidefinite $T \times T$ matrix that encodes how the outputs are related. This class of kernels allows to decouple the role played by the input and output spaces. As we show in Sect.5.1, it is possible to derive more efficient learning schemes using these kernels. The role of the matrix A can be understood by linking it to a regularizer on the components of the vector field.

Proposition 1. *Let Γ be a product kernel of the form in (8). Then the norm of any function in the corresponding RKHS can be written as*

$$||f||_\Gamma^2 = \sum_{\ell,q=1}^{T} A^\dagger_{\ell q} < f^\ell, f^q >_K , \tag{9}$$

where $A^\dagger$ is the pseudoinverse of A.

Proof. A function in the RKHS defined by the matrix valued kernel $\Gamma = KA$ can be written as $f(x) = \sum_i \Gamma(x, x_i)c_i = \sum_i K(x, x_i)Ac_i$ with $c_i \in \mathbb{R}^T$, so that the ℓ-th component is $f^\ell(x) = \sum_i \sum_{t=1}^T K(x, x_i)A_{\ell t}c_i^t$, where $c_i^t \in \mathbb{R}$ is the t-th component of c_i. Therefore, each f^ℓ belongs to $\mathcal{H}_K$ and it is a linear combination of the coefficients $\{c_i\}_{i=1}^n$ that depends on the ℓ-th row of the matrix A.

The norm of f is $||f||_\Gamma^2 = \sum_{i,j} \sum_{\ell,q=1}^T K(x_i, x_j)c_i^\ell A_{\ell q}c_j^q$ and the scalar products between its components are given by $\langle f^\ell, f^q \rangle_K = \sum_{i,j} \sum_{t,s=1}^T K(x_i, x_j)A_{\ell t}c_i^t A_{qs}c_j^s$. Combining these expressions, it is straightforward to obtain (9). □

The above result shows how to design a kernel by defining a penalty on the components of the vector field.

Common similarity. This kernel forces the components of the vector field to be similar to their average, $\Gamma_\omega(x, x') = K(x, x')(\omega \mathbf{1} + (1 - \omega)\mathbf{I})$. In fact, it is straightforward to show that the corresponding regularizer is

$$A_\omega \sum_{\ell=1}^{T} ||f^\ell||_K^2 + B_\omega \sum_{\ell=1}^{T} ||f^\ell - \frac{1}{T} \sum_{q=1}^{T} f^q||_K^2 \,, \tag{10}$$

where A_ω and B_ω are coefficients that depend on ω.

Graph regularization. A regularizer that forces stronger or weaker similarity between the components [29] is defined as

$$\frac{1}{2} \sum_{\ell,q=1}^{T} ||f^\ell - f^q||_K^2 M_{\ell q} + \sum_{\ell=1}^{T} ||f^\ell||_K^2 M_{\ell\ell} \,, \tag{11}$$

where M is a $T \times T$ positive weight matrix. The corresponding kernel is $\Gamma = KL^\dagger$, where $L = D - M$ and $D_{\ell q} = \delta_{\ell q} \left(\sum_{h=1}^{T} M_{\ell h} + M_{\ell q} \right)$.

Output components clustering. This regularizer is based on the idea of grouping the components into r clusters and enforcing the components in each cluster to be similar to their average [25]

$$J(f) = \epsilon_1 \sum_{c=1}^{r} \sum_{l \in I(c)} ||f^l - \overline{f}_c||_K^2 + \epsilon_2 \sum_{c=1}^{r} m_c ||\overline{f}_c||_K^2 \,, \tag{12}$$

where $\overline{f}_c$ is the mean of the components in cluster c and $I(c)$ is the index set of the components that belong to cluster c. Simple calculations show that the corresponding kernel is $\Gamma = KG^\dagger$, where $G_{lq} = \epsilon_1 \delta_{lq} + (\epsilon_2 - \epsilon_1) M_{lq}$ and $M_{lq} = \frac{1}{m_c}$ if components l and q belong to the same cluster c of cardinality m_c, $M_{lq} = 0$ otherwise.

5 Computational and Sample Complexity

We present a computational complexity analysis of the spectral filters taking into account the choice of kernel. We show that for a specific class of matrix valued kernels it is possible to greatly reduce the computational complexity of the algorithms. Finally, we give the bound on the excess risk that leads to consistency.

5.1 Faster Implementation for Decomposable Kernels

The main point we make in this section is that, for kernels of the form $\Gamma = KA$, we can use the eigen-sytem of the matrix A to define a new coordinate system

where the problem can be solved in a computational faster way. The outcome of this analysis is that the vector field learning problem can be reduced to solving T scalar regression problems.

If we denote with $u_1, \ldots, u_T$ the eigenvectors of A, we can write the vector $\mathbf{C} = (c_1, \ldots, c_n)$, with $c_i \in \mathbb{R}^T$, as $\mathbf{C} = \sum_{j=1}^{T} \tilde{c}^j \otimes u_j$, where $\otimes$ is the tensor product and $\tilde{c}^j = (\langle c_1, u_j \rangle_T, \ldots, \langle c_n, u_j \rangle_T)$. Similarly $\mathbf{Y} = \sum_{j=1}^{T} \tilde{y}^j \otimes u_j$, with $\tilde{y}^j = (\langle y_1, u_j \rangle_T, \ldots, \langle y_n, u_j \rangle_T)$. The above transformations are simply rotations in the output space. Moreover the kernel matrix $\mathbf{\Gamma}$ is given by the tensor product of the $n \times n$ scalar kernel matrix $\mathbf{K}$ and A, that is $\mathbf{\Gamma} = \mathbf{K} \otimes A$.
If we denote with λ_i, v_i $(i = 1, \ldots, n)$, the eigenvalues and eigenvectors of $\mathbf{K}$ and with σ_j $(j = 1, \ldots, T)$ the eigenvalues of A, we have the following result.

Proposition 2. *The solution of the vector valued problem* $\mathbf{C} = g_\lambda(\mathbf{\Gamma})\mathbf{Y}$ *can be obtained by solving* T *scalar problems*

$$\tilde{c}^j = g_\lambda(\sigma_j \mathbf{K})\tilde{y}^j \ , \qquad j = 1, \ldots, T \ . \tag{13}$$

Proof. Substituting the expressions for $\mathbf{C}$ and $\mathbf{Y}$ into $\mathbf{C} = g_\lambda(\mathbf{\Gamma})\mathbf{Y}$, we obtain

$$\sum_{j=1}^{T} \tilde{c}^j \otimes u_j = \sum_{j=1}^{T} g_\lambda(\mathbf{K} \otimes A)\tilde{y}^j \otimes u_j \ .$$

Working in the eigen-system $v_i \otimes u_j$ $(i = 1, \ldots, n$ and $j = 1, \ldots, T)$ of the matrix $\mathbf{K} \otimes A$ and recalling that the spectral filters operate on the eigenvalues of the kernel matrix, we have

$$\sum_{j=1}^{T} \tilde{c}^j \otimes u_j = \sum_{j=1}^{T} \sum_{i=1}^{n} g_\lambda(\lambda_i \sigma_j)\langle \tilde{y}^j, v_i \rangle v_i \otimes u_j = \sum_{j=1}^{T} g_\lambda(\sigma_j \mathbf{K})\tilde{y}^j \otimes u_j \ .$$

Since the eigenvectors u_j are orthonormal, the two sides of the equation must be equal term by term. It follows that $\tilde{c}^j = g_\lambda(\sigma_j \mathbf{K})\tilde{y}^j$ for $j = 1, \ldots, T$. □

The above equation shows that in the new coordinate system $\{u_1, \ldots, u_T\}$, we have to solve T essentially independent problems. Indeed, after rotating the outputs (and the coefficients) the only coupling is the rescaling of each kernel matrix by σ_j. For example, in the case of Tikhonov regularization, the j-th component is found solving $\tilde{c}^j = (\sigma_j \mathbf{K} + \lambda n \mathbf{I})^{-1}\tilde{y}^j = (\mathbf{K} + \frac{\lambda}{\sigma_j} n \mathbf{I})^{-1} \frac{\tilde{y}^j}{\sigma_j}$ and we see that the scaling term is changing the scale of the regularization parameter and of the outputs. The above calculation shows that all kernels of this form allow for a simple implementation at the price of the eigen-decomposition of the matrix A. Also, it shows that the coupling among the different tasks can be seen as a rotation and rescaling of the output points.

5.2 Regularization Path and Computational Complexity

Here we discuss the complexity of the whole *regularization path* for Tikhonov regularization and accelerated L2 boosting, since this algorithm turns out to be

among the fastest. The regularization path is the set of solutions corresponding to many parameter values.

On one hand, when using Tikhonov regularization, for each value of the regularization parameter the solution is found by inverting a $nT \times nT$ matrix. On the other, most iterative methods require only matrix vector multiplications, and each step corresponds to a solution for a value of the regularization parameter, so that at step N we have computed the entire regularization path up to N. Therefore, in general, if we consider N parameter values we will have $O(N(nT)^3)$ time complexity for Tikhonov regularization and $O(N(nT)^2)$ for iterative methods.

In the special case of kernels of the form $\Gamma = KA$, we can diagonalize the matrix A and then work in a new coordinate system where the kernel matrix is block diagonal and all the blocks are the same, up to a rescaling. In this case the complexity of the vector field algorithm is essentially the same of T scalar problems – $O(TNn^3)$ for Tikhonov and $O(TNn^2)$ for iterative methods – plus the cost of computing the eigen-decomposition of A, which is $O(T^3)$.

5.3 Sample Complexity

Our main theoretical result is a finite sample bound on the excess risk for all algorithms based on spectral filtering. This result immediately leads to consistency and can be proven in a unified framework. Each algorithm is characterized by specific constants that might change from one algorithm to the other and we refer to [22] for their computation. In particular, here we underline to role played by the one of such constants, namely the qualification number $\overline{r}$, which controls the best achievable learning rate of the corresponding algorithm, as is illustrated in the following theorem. We need to assume that the input space is a separable metric space (not necessarily compact) and that the output space is a bounded set in $\mathbb{R}^T$, that is $sup_{y \in \mathcal{Y}} ||y||_T = M < \infty$. For the sake of simplicity we also assume that a minimizer of the expected risk on $\mathcal{H}$ exists and denote it with $f_{\mathcal{H}}$. Let us also define the integral operator $T_\Gamma f(x) = \int_{\mathcal{X}} \Gamma(x, x') f(x') \rho(x')$.

Theorem 1. *Assume* $||(T_\Gamma)^{-\nu} f_{\mathcal{H}}||_\Gamma \leq R$, *where* $\nu = r - \frac{1}{2}$. *If*

$$\frac{1}{2} \leq r \leq \overline{r} \qquad \textit{and} \qquad \lambda_n = C n^{-\frac{1}{2r+1}} \log \frac{4}{\eta} \,,$$

then, for $f_{\mathbf{z}} = f_{\mathbf{z}}^{\lambda_n}$, *we have, with probability* $1 - \eta$,

$$\mathcal{E}(f_{\mathbf{z}}) - \mathcal{E}(f_{\mathcal{H}}) \leq C' n^{-\frac{2r}{2r+1}} \log^2 \frac{4}{\eta} \,, \tag{14}$$

where C, C' *are constants that depend on* R *and* r, *but not on* n.

The index r describes the simplicity of the field estimation problem and $r > 1/2$ implies that $f_{\mathcal{H}}$ exists. The simpler the problem (r large), the faster the learning rate. The qualification number $\overline{r}$ represents how much each specific filter can exploit the simplicity of the learning problem. Tikhonov regularization, for instance, has a qualification number $\overline{r} = 1$, that yields a rate proportional to

$n^{-\frac{2}{3}}$, while for some iterative algorithms, such as the ν-method, the qualification number $\overline{r}$ can be arbitrarily large, yielding a bound arbitrarily close to n^{-1}. Note that the result above directly leads to consistency, since the limit for $n \to \infty$ is zero and, even when the expected risk does not achieve a minimum in $\mathcal{H}$, one can still show that there is a parameter choice ensuring convergence to $\inf_{f \in \mathcal{H}} \mathcal{E}(f)$. If the kernel is universal [30,27,31], then universal consistency [32] is ensured. In particular, note that the results in [31] allow to work on a non compact domain, at least if the kernel is well-behaved. Due to space reasons, we omit the proof which will be included in a longer version of this paper.

6 Empirical Analysis

We present a synthetic 2-dimensional vector field estimation problem in order to illustrate the benefits of using matrix valued kernels and the computational advantages of iterative spectral filters with respect to Tikhonov regularization. We consider the setting of [28] and first compare our vector valued regression approach with estimating each component of the field independently, in both cases using the ν-method, which is the fastest algorithm when the matrix valued kernel is not of the form $\Gamma = KA$. Finally, we compare the computation time of Tikhonov regularization and the ν-method to show that the latter is significantly faster and scales better with the number of training points.

The vector field is generated from a scalar field defined by the sum of 5 gaussians centered at $(0,0)$, $(1,0)$, $(0,1)$, $(-1,0)$ and $(0,-1)$ respectively. The covariances are all set to be $0.45\mathbf{I}$, where $\mathbf{I}$ is the 2×2 identity matrix. We compute the gradient of the scalar field and its perpendicula field. We then consider a convex combination of these two vector fields, controlled by a parameter γ. Examples of the resulting fields for $\gamma = 0$ and $\gamma = 1$ are shown in Fig.1.

Firstly, we consider the noiseless case. The vector field is constructed specifying a value of the parameter γ. The field is then computed on a 70×70 grid over the square $[-2,2] \times [-2,2]$. The models are trained on a uniform random

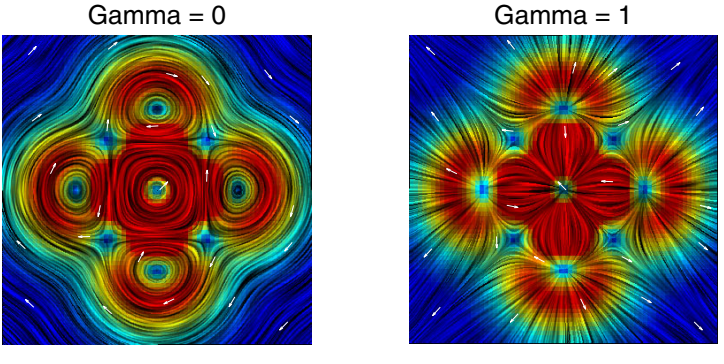

Fig. 1. Visualization of the 2-dimensional vector field for $\gamma = 0$, resulting in a divergence-free field, and for $\gamma = 1$ that yields a curl-free field

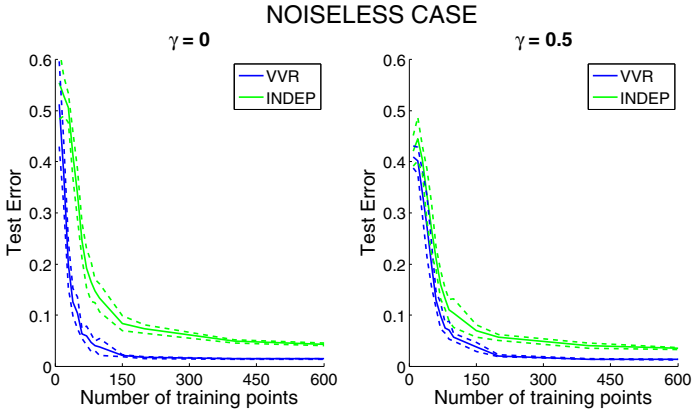

Fig. 2. Noiseless case. Test errors for the proposed vector valued approach (VVR) and for learning each component of the field independently (INDEP) as a function of the number of training points used for learning. Solid lines represent average test error, while dotted lines show the average test error plus/minus one standard deviation.

sample of points from this grid and their predictions on the whole grid (except the training points) compared to the correct field. The number of training examples is varied from 10 to 600. For each cardinality of the training set, the training and prediction process is repeated 10 times with a different randomization of the training points. We use a convex combination of the divergence-free (6) and curl-free (7) kernels, controlled by the parameter $\tilde{\gamma}$. We adopt a 5-fold cross validation to select the optimal number of iterations for the ν-method and the parameter $\tilde{\gamma}$. The width, σ, of these kernels was set to be 0.8.

We use an angular measure of error to compare two fields [33]. If $v_o = (v_o^1, v_o^2)$ and $v_e = (v_e^1, v_e^2)$ are the original and estimated fields, we consider the transformation $v \rightarrow \tilde{v} = \frac{1}{||(v^1, v^2, 1)||}(v^1, v^2, 1)$. The error measure is then $err = arccos(\tilde{v}_e \cdot \tilde{v}_o)$. This error measure was derived by interpreting the vector field as a velocity field and it is convenient because it handles large and small signals without the amplification inherent in a relative measure of vector differences.

The results for the noiseless case are reported in Fig.2, which clearly shows the advantage of using a vector valued approach with the combination of curl-free and divergence-free kernels. We present only the results for the field generated with $\gamma = 0$ and $\gamma = 0.5$ since for the remaining fields the errors are set within these two examples. The prediction errors of the proposed approach via the ν-method are always lower than the errors obtained by regressing on each component independently, even when the training set is very large. The average value of the estimated parameter $\tilde{\gamma}$, converges to the true value of γ as the number of training points increases (result not shown for brevity), indicating that it is possible for the model to learn the field decomposition in an automatic way.

We then consider the case with normal noise whose standard deviation is independent from the signal and is chosen to be 0.3. We follow the same experimental

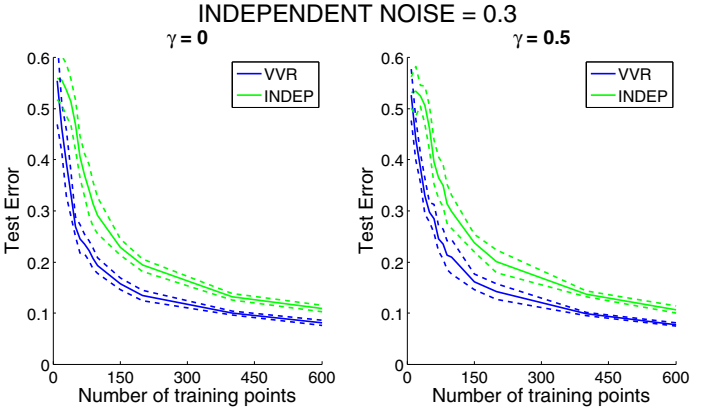

Fig. 3. Independent noise of standard deviation 0.3. Test errors for the proposed vector valued approach (VVR) and for learning each component of the field independently (INDEP) as a function of the number of training points used for learning. Solid lines represent average test error, while dotted lines show the average test error plus/minus one standard deviation.

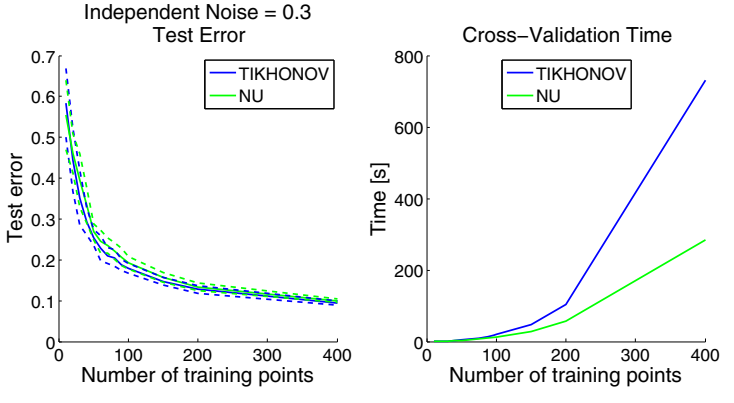

Fig. 4. (Left) Independent noise of standard deviation 0.3. Test errors for the proposed vector valued approach solved with Tikhonov regularization or the ν-method. (Right) Computation time for the whole regularization path for the ν-method and for Tikhonov regularization. The experiment was performed on Matlab on a desktop PC with AMD Athlon X2 64 3.2GHz and 2GB RAM.

protocol adopted for the noiseless case. The results are reported in Fig.3 and indicate that also in the presence of noise the proposed approach consistently outperforms regressing on each component independently. The advantage is stronger when fewer training points are available, but it is still present even at higher training set cardinalities. Again, the estimated value for the parameter $\hat{\gamma}$ well

approximates the true value used for the creation of the vector field, indicating that it is possible for the model to learn the field decomposition in an automatic way also in presence of noise (result not shown for brevity).

In Fig.4 we compare Tikhonov regularization and the ν-method on the field generated with $\gamma = 0$. In the left panel are reported the test errors as a function of the number of training examples in the case with independent normal noise of standard deviation 0.3. We clearly see that the two algorithms perform equivalently. In the right plot are shown the cross validation learning times for Tikhonov regularization and the ν-method. For both methods, both the parameter $\hat{\gamma}$ and the regularization parameter have been estimated. For Tikhonov regularization 50 values of the regularization parameter, between 10^-5 and 10^-2, were assessed, while for the ν-method the iterations up to 500 were evaluated. The experimental results confirm our complexity analysis: the ν-method is significantly faster than Tikhonov regularization, while preserving its good generalization performance.

7 Conclusions

In this paper we considered the problem of learning vector valued functions and proposed a class of regularized kernel methods based on spectral filtering of the kernel matrix. Tikhonov regularization and L2 boosting are examples of methods falling in our framework. The complexity and empirical analysis showed the advantages of iterative algorithms with respect to the more standard Tikhonov regularization. A similar conclusion can be drawn comparing the learning rates of the spectral filters.

Acknowledgements. We would like to thank Ernesto De Vito for many useful discussions. This work has been partially supported by the EU Integrated Project Health-e-Child IST-2004-027749.

References

1. Micchelli, C.A., Pontil, M.: On learning vector–valued functions. Neural Computation 17 (2005)
2. Lo Gerfo, L., Rosasco, L., Odone, F., De Vito, E., Verri, A.: Spectral algorithms for supervised learning. Neural Computation 20 (2008)
3. De Vito, E., Rosasco, L., Caponnetto, A., De Giovannini, U., Odone, F.: Learning from examples as an inverse problem. JMLR 6 (2005)
4. Poggio, T., Rifkin, R., Mukherjee, S., Niyogi, P.: General conditions for predictivity in learning theory. Nature 428 (2004)
5. Bousquet, O., Elisseeff, A.: Stability and generalization. JMLR 2 (2002)
6. Bühlmann, P., Yu, B.: Boosting with the l_2-loss: Regression and classification. JASA 98 (2002)
7. Caruana, R.: Multitask learning. Mach. Learn. 28 (1997)
8. Argyriou, A., Maurer, A., Pontil, M.: An algorithm for transfer learning in a heterogeneous environment. In: Daelemans, W., Goethals, B., Morik, K. (eds.) ECML PKDD 2008, Part I. LNCS (LNAI), vol. 5211, pp. 71–85. Springer, Heidelberg (2008)

9. Boyle, P., Frean, M.: Dependent gaussian processes. In: NIPS (2005)
10. Chai, K., Williams, C., Klanke, S., Vijayakumar, S.: Multi-task gaussian process learning of robot inverse dynamics. In: NIPS (2009)
11. Breiman, L., Friedman, J.H.: Predicting multivariate responses in multiple linear regression. J. R. Statist. Soc. B 59(1) (1997)
12. Izenman, A.: Reduced-rank regression for the multivariate linear model. J. Multiv. Anal. 5 (1975)
13. van der Merwe, A., Zidek, J.V.: Multivariate regression analysis and canonical variates. Can. J. Stat. 8 (1980)
14. Wold, S., Ruhe, H., Wold, H., Dunn III, W.: The collinearity problem in linear regression. The partial least squares (pls) approach to generalizes inverses. SIAM J. Sci. Comput. 5 (1984)
15. Stein, M.L.: Interpolation of spatial data. Springer, Heidelberg (1999)
16. Brudnak, M.: Vector-valued support vector regression. In: IJCNN (2006)
17. Vazquez, E., Walter, E.: Multi output support vector regression. In: SYSID (2003)
18. Caponnetto, A., De Vito, E.: Optimal rates for regularized least-squares algorithm. Found. Comp. Math. (2006)
19. Carmeli, C., De Vito, E., Toigo, A.: Vector valued reproducing kernel Hilbert spaces of integrable functions and Mercer theorem. JAA 4 (2006)
20. Tikhonov, A.N., Arsenin, V.Y.: Solutions of Ill-posed Problems. John Wiley, Chichester (1977)
21. Evgeniou, T., Pontil, M., Poggio, T.: Regularization networks and support vector machines. Advances in Computational Mathematics 13(1), 1–50 (2000)
22. Bauer, F., Pereverzev, S., Rosasco, L.: On regularization algorithms in learning theory. J. Complex. 23(1) (2007)
23. Engl, H.W., Hanke, M., Neubauer, A.: Regularization of inverse problems. Kluwer, Dordrecht (1996)
24. Smola, A., Kondor, R.: Kernels and regularization on graphs. In: Schölkopf, B., Warmuth, M.K. (eds.) COLT/Kernel 2003. LNCS (LNAI), vol. 2777, pp. 144–158. Springer, Heidelberg (2003)
25. Evgeniou, T., Micchelli, C.A., Pontil, M.: Learning multiple tasks with kernel methods. JMLR 6 (2005)
26. Sheldon, D.: Graphical multi-task learning. Technical report, Cornell University (2008) (preprint)
27. Caponnetto, A., Micchelli, C.A., Pontil, M., Ying, Y.: Universal kernels for multi-task learning. JMLR 9 (2008)
28. Macêdo, I., Castro, R.: Learning divergence-free and curl-free vector fields with matrix-valued kernels. Technical report, Instituto Nacional de Matematica Pura e Aplicada (2008)
29. Micchelli, C.A., Pontil, M.: Kernels for multi-task learning. In: NIPS (2004)
30. Steinwart, I.: On the influence of the kernel on the consistency of support vector machines. JMLR 2 (2002)
31. Carmeli, C., De Vito, E., Toigo, A., Umanitá, V.: Vector valued reproducing kernel hilbert spaces and universality. JAA 4 (2010)
32. Devroye, L., Györfi, L., Lugosi, G.: A Probabilistic Theory of Pattern Recognition. Springer, Heidelberg (1996)
33. Barron, J., Fleet, D., Beauchemin, S.: Performance of optical flow techniques. IJCV 12(1) (1994)