Learning low-rank output kernels

Francesco Dinuzzo FDINUZZO@TUEBINGEN.MPG.DE
Max Planck Institute for Intelligent Systems
Spemannstrasse 38,
72076 Tübingen, Germany

Kenji Fukumizu FUKUMIZU@ISM.AC.JP
The Institute of Statistical Mathematics
10-3 Midori-cho,
Tachikawa, Tokyo 190-8562 Japan

Editor: Chun-Nan Hsu and Wee Sun Lee

Abstract

Output kernel learning techniques allow to simultaneously learn a vector-valued function and a positive semidefinite matrix which describes the relationships between the outputs. In this paper, we introduce a new formulation that imposes a low-rank constraint on the output kernel and operates directly on a factor of the kernel matrix. First, we investigate the connection between output kernel learning and a regularization problem for an architecture with two layers. Then, we show that a variety of methods such as nuclear norm regularized regression, reduced-rank regression, principal component analysis, and low rank matrix approximation can be seen as special cases of the output kernel learning framework. Finally, we introduce a block coordinate descent strategy for learning low-rank output kernels.

Keywords: Output kernel learning, learning the kernel, RKHS, coordinate descent

1. Introduction

Methods for learning vector-valued functions are becoming popular subjects of study in machine learning, motivated by applications to multi-task learning, multi-label and multi-class classification. In these problems, selecting a model that correctly exploits the relationships between the different output components is crucial to ensure good learning performances.

Within the framework of regularization in reproducing kernel Hilbert spaces (RKHS) of vector-valued functions (Aronszajn, 1950; Micchelli and Pontil, 2005), one can directly encode relationships between the outputs by choosing a suitable operator-valued kernel. A simple and well studied model assumes that the kernel can be decomposed as the product of a scalar positive semidefinite kernel on the input space (*input kernel*), and a linear operator on the output space (*output kernel*), see e.g. (Evgeniou et al., 2005; Bonilla et al., 2008; Caponnetto et al., 2008; Baldassarre et al., 2010; Dinuzzo et al., 2011). Covariance functions (kernels) of this form have been also studied in geostatistics, in the context of the so-called intrinsic coregionalization model, see e.g. (Goovaerts, 1997; Alvarez and Lawrence, 2011).

The choice of the output kernel may significantly influence learning performance. When prior knowledge is not sufficient to fix the output kernel in advance, it is necessary to adopt automatic techniques to learn it from the data. A multiple kernel learning approach has been proposed by Zien and Ong (2007), while Bonilla et al. (2008) propose to choose the output kernel by minimizing the marginal likelihood within a Bayesian framework. Recently, a methodology to learn simultaneously a vector-valued function in a RKHS and a kernel on the output space has been proposed (Dinuzzo et al., 2011). Such a technique is based on the optimization of a non-convex functional that, nevertheless, can be globally optimized in view of invexity (Mishra and Giorgi, 2008). The method of Dinuzzo et al. (2011) directly operates on the full output kernel matrix which, in general, is full-rank. However, when the dimensionality of the output space is very high, storing and manipulating the full matrix may not be efficient or feasible.

In this paper, we introduce a new output kernel learning method that enforces a rank constraint on the output kernel and directly operates on a factor of the kernel matrix. In section 2, we recall some preliminary results about RKHS of vector valued functions and decomposable kernels, and introduce some matrix notations. In section 3, we introduce the low-rank output kernel learning model and the associated optimization problem. In section 4, we show that the proposed output kernel learning problem can be seen as the kernelized version of nuclear norm regularization. In view of such connection, a variety of methods such as reduced-rank regression (Anderson, 1951; Izenman, 1975; Reinsel and Velu, 1998), principal component analysis (Jolliffe, 1986), and low rank matrix approximation (Eckart and Young, 1936) can be seen as particular cases. In section 5 we develop an optimization algorithm for output kernel learning based on a block coordinate descent strategy. Finally, in section 6, performances of the algorithm are investigated using synthetic multiple time series reconstruction datasets, and compared with previously proposed methods. In Appendix A, we discuss an alternative formulation of the low rank output kernel learning problem and derive a suitable optimality condition. All the proofs are given in Appendix B.

2. Preliminaries

In sub-section 2.1, we review basic definitions regarding kernels and reproducing kernel Hilbert spaces of vector-valued functions (Micchelli and Pontil, 2005), and then introduce a class of decomposable kernels that will be the focus of the paper. In sub-section 2.2, we introduce some matrix notations needed in the paper.

2.1. Reproducing Kernel Hilbert Spaces

Let $\mathcal{X}$ denote a non-empty set and $\mathcal{Y}$ a Hilbert space with inner product $\langle \cdot, \cdot \rangle_{\mathcal{Y}}$. Throughout the paper, all Hilbert spaces are assumed to be real. Let $\mathcal{L}(\mathcal{Y})$ denote the space of bounded linear operators from $\mathcal{Y}$ into itself.

Definition 1 (Positive semidefinite $\mathcal{Y}$-kernel) *A symmetric function* $H : \mathcal{X} \times \mathcal{X} \to \mathcal{L}(\mathcal{Y})$ *is called* positive semidefinite $\mathcal{Y}$-kernel on $\mathcal{X}$ *if, for any natural number* ℓ*, the following holds*

$$\sum_{i=1}^{\ell} \sum_{j=1}^{\ell} \langle y_i, H(x_i, x_j) y_j \rangle_{\mathcal{Y}} \geq 0, \qquad \forall (x_i, y_i) \in (\mathcal{X}, \mathcal{Y}) .$$

It is often convenient to consider the function obtained by fixing one of the two arguments of the kernel to a particular point.

Definition 2 (Kernel section) *Let H denote a $\mathcal{Y}$-kernel on $\mathcal{X}$. A* kernel section *centered on $\bar{x} \in \mathcal{X}$ is a map $H_{\bar{x}} : \mathcal{X} \to \mathcal{L}(\mathcal{Y})$ defined as*

$$H_{\bar{x}}(x) = H(\bar{x}, x).$$

The class of positive semidefinite $\mathcal{Y}$-kernels can be associated with a suitable family of Hilbert spaces of vector-valued functions.

Definition 3 (RKHS of $\mathcal{Y}$-valued functions) *A Reproducing Kernel Hilbert Space of $\mathcal{Y}$-valued functions $g : \mathcal{X} \to \mathcal{Y}$ is a Hilbert space $\mathcal{H}$ such that, for all $x \in \mathcal{X}$, there exists $C_x \in \mathbb{R}$ such that*

$$\|g(x)\|_{\mathcal{Y}} \leq C_x \|g\|_{\mathcal{H}}, \qquad \forall g \in \mathcal{H}.$$

It turns out that every RKHS of $\mathcal{Y}$-valued functions $\mathcal{H}$ can be associated with a unique positive semidefinite $\mathcal{Y}$-kernel H, called the *reproducing kernel*, such that the following *reproducing property* holds:

$$\langle g(x), y \rangle_{\mathcal{Y}} = \langle g, H_x y \rangle_{\mathcal{H}}, \qquad \forall (x, y, g) \in (\mathcal{X}, \mathcal{Y}, \mathcal{H}).$$

Conversely, given a positive semidefinite $\mathcal{Y}$-kernel H on $\mathcal{X}$, there exists a unique RKHS of $\mathcal{Y}$-valued functions defined over $\mathcal{X}$ whose reproducing kernel is H. The standard definition of positive semidefinite *scalar kernel* and RKHS (of real-valued functions) can be recovered by letting $\mathcal{Y} = \mathbb{R}$. The following definition introduces a specific class of $\mathcal{Y}$-kernels that will be the focus of this paper.

Definition 4 (Decomposable Kernel) *A positive semidefinite $\mathcal{Y}$-kernel on $\mathcal{X}$ is called* decomposable *if it can be written as*

$$H_{\mathbf{L}} = K \cdot \mathbf{L},$$

where K is a positive semidefinite scalar kernel on $\mathcal{X}$ and $\mathbf{L} : \mathcal{Y} \to \mathcal{Y}$ is a self-adjoint positive semidefinite operator, i.e.

$$\langle y, \mathbf{L}y \rangle_{\mathcal{Y}} \geq 0, \quad \forall y \in \mathcal{Y}.$$

In the previous definition, K takes into account the similarity between the inputs (*input kernel*), and $\mathbf{L}$ measures the similarity between the output's components (*output kernel*).

2.2. Matrix notation

The identity matrix is denoted as $\mathbf{I}$, the vector of all ones is denoted by e. For any matrix $\mathbf{A}$, $\mathbf{A}^T$ denote the transpose, $\mathrm{tr}(\mathbf{A})$ the trace, $\mathrm{rank}(\mathbf{A})$ the rank, $\mathrm{rg}(\mathbf{A})$ the range, and $\mathbf{A}^\dagger$ the Moore-Penrose pseudo-inverse. For any pair of matrices of the same size $\mathbf{A}, \mathbf{B}$, let $\langle \mathbf{A}, \mathbf{B} \rangle_F := \mathrm{tr}(\mathbf{A}^T \mathbf{B})$ denote the Frobenius inner product, and $\|\mathbf{A}\|_F := \sqrt{\langle \mathbf{A}, \mathbf{A} \rangle_F}$ the induced norm. In addition, $\|\mathbf{A}\|_* := \mathrm{tr}\left(\left(\mathbf{A}^T \mathbf{A}\right)^{1/2}\right)$ denote the nuclear norm, which coincides with the trace when $\mathbf{A}$ is square symmetric and positive semidefinite. The symbols $\otimes$, $\odot$, and $\oslash$ denote the Kronecker product, the Hadamard (element-wise) product, and the element-wise division, respectively. Finally, let $\mathbb{S}_+^m$ denote the closed cone of positive semidefinite matrices or order m, and

$$\mathbb{S}_+^{m,p} = \left\{ \mathbf{A} \in \mathbb{S}_+^m : \mathrm{rank}(\mathbf{A}) \leq p \right\} \subseteq \mathbb{S}_+^m$$

the set of positive semidefinite matrices whose rank is less than or equal to p.

3. Low-rank output kernel learning

Let $\mathcal{Y} = \mathbb{R}^m$, and let $\mathcal{H}$ denote an RKHS of $\mathcal{Y}$-valued functions whose reproducing kernel is decomposable (see Definition 4). Given a set of ℓ input-output data pairs $(x_i, y_i) \in \mathcal{X} \times \mathcal{Y}$, and a positive real number $\lambda > 0$, consider the following problem (low-rank *output kernel learning*):

$$\min_{\mathbf{L} \in \mathbb{S}_+^{m,p}} \left[\min_{g \in \mathcal{H}} \left(\sum_{i=1}^{\ell} \frac{\|y_i - g(x_i)\|_2^2}{2\lambda} + \frac{\|g\|_{\mathcal{H}}^2}{2} + \frac{\mathrm{tr}(\mathbf{L})}{2} \right) \right]. \tag{1}$$

First of all, application of the representer theorem (Kimeldorf and Wahba, 1971; Schölkopf et al., 2001; Micchelli and Pontil, 2005) to the inner minimization problem of (1) yields

$$g = \mathbf{L} \left(\sum_{i=1}^{\ell} c_i K_{x_i} \right).$$

Then, by introducing the input kernel matrix $\mathbf{K} \in \mathbb{S}_+^{\ell}$ such that $\mathbf{K}_{ij} = K(x_i, x_j)$, and matrices $\mathbf{Y}, \mathbf{C} \in \mathbb{R}^{\ell \times m}$ such that

$$\mathbf{Y} = (y_1, \dots, y_\ell)^T, \qquad \mathbf{C} = (c_1, \dots, c_\ell)^T,$$

problem (1) can be rewritten as

$$\min_{\mathbf{L} \in \mathbb{S}_+^{m,p}} \min_{\mathbf{C} \in \mathbb{R}^{\ell \times m}} \left(\frac{\|\mathbf{Y} - \mathbf{KCL}\|_F^2}{2\lambda} + \frac{\langle \mathbf{C}^T \mathbf{KC}, \mathbf{L} \rangle_F}{2} + \frac{\mathrm{tr}(\mathbf{L})}{2} \right). \tag{2}$$

Although the objective functional of (1) is separately convex with respect to $\mathbf{C}$ and $\mathbf{L}$, it is not jointly (quasi)-convex with respect to the pair $(\mathbf{C}, \mathbf{L})$. Nevertheless, by using techniques similar to Dinuzzo et al. (2011), is it possible to prove invexity, so that stationary points are global minimizers. Unfortunately, in presence of the rank constraint, stationary points are not guaranteed to be feasible points. In Appendix A, we derive a sufficient condition for global optimality based on a reformulation of problem (2).

3.1. Output kernel learning as a kernel machine with two layers

In this subsection, we present an alternative interpretation of problem (1). Such formulation allows us to apply the model to data compression and visualization problems, and will be also useful for optimization purposes. Consider a map $g : \mathcal{X} \to \mathcal{Y}$ of the form:

$$g(x) = (g_2 \circ g_1)(x),$$

where $g_1 : \mathcal{X} \to \mathbb{R}^p$ is a non-linear vector-valued function, and $g_2 : \mathbb{R}^p \to \mathcal{Y}$ is a linear function. One can interpret g_1 as a map that performs a non-linear feature extraction or dimensionality reduction, while the operator g_2 linearly combines the extracted features to produce the output vector. In particular, assume that g_1 belongs to an RKHS $\mathcal{H}_1$ of vector valued functions whose kernel is decomposable as $H = K \cdot \mathbf{I}$, and g_2 belongs to the space $\mathcal{H}_2$

of linear operators of the type $g_2(z) = \mathbf{B}z$, endowed with the Hilbert-Schmidt (Frobenius) norm. Consider the following regularization problem

$$\min_{g_2 \in \mathcal{H}_2} \left[\min_{g_1 \in \mathcal{H}_1} \left(\sum_{i=1}^{\ell} \frac{\|y_i - (g_2 \circ g_1)(x_i)\|_2^2}{2\lambda} + \frac{\|g_1\|_{\mathcal{H}_1}^2}{2} + \frac{\|g_2\|_{\mathcal{H}_2}^2}{2} \right) \right] \tag{3}$$

According to the representer theorem for vector-valued functions, the inner minimization problem admits a solution of the form

$$g_1 = \sum_{i=1}^{\ell} a_i K_{x_i}$$

and thus we have

$$g = \mathbf{B} \left(\sum_{i=1}^{\ell} a_i K_{x_i} \right).$$

By introducing matrices $\mathbf{K}, \mathbf{Y}$ as in the previous section, and letting $\mathbf{A} \in \mathbb{R}^{\ell \times p}$ such that

$$\mathbf{A} = (a_1, \dots, a_\ell)^T,$$

the regularization problem can be rewritten as

$$\min_{\mathbf{B} \in \mathbb{R}^{m \times p}} \min_{\mathbf{A} \in \mathbb{R}^{\ell \times p}} Q(\mathbf{A}, \mathbf{B}), \tag{4}$$

where

$$Q(\mathbf{A}, \mathbf{B}) := \frac{\|\mathbf{Y} - \mathbf{K}\mathbf{A}\mathbf{B}^T\|_F^2}{2\lambda} + \frac{\langle \mathbf{A}, \mathbf{K}\mathbf{A} \rangle_F}{2} + \frac{\|\mathbf{B}\|_F^2}{2}$$

The following result shows that problems (1) and (3) are equivalent.

Theorem 5 *The optimal solutions g for problems (1) and (3) coincide.*

3.2. Kernelized auto-encoder

In general, if the output data of problem (3) coincide with the inputs, i.e. $y_i = x_i$, the model can be seen as a kernelized auto-encoder with two layers, where the first layer g_1 performs a non-linear data compression, and the second-layer g_2 linearly decompresses the data. The compression ratio can be controlled by simply choosing the value of p. Alternatively, if the goal is data visualization in, say, two or three dimensions, one can simply set $p = 2$ or $p = 3$.

Kernel PCA (Schölkopf et al., 1998) is another related technique which uses positive semidefinite kernels to perform non-linear feature extraction, and can be also interpreted as an auto-encoder in the feature space. A popular application of kernel PCA is pattern denoising, which requires the non-linear extraction of features followed by the solution of a pre-image problem (Schölkopf et al., 1999). Observe that the kernelized auto-encoder obtained by solving (3) differs from kernel PCA, since it minimizes a reconstruction error in the original input space, and also introduces a suitable regularization on the non-linear feature extractor. Differently from kernel PCA, the kernel machine with two layers discussed in this section performs non-linear denoising without requiring the solution of a pre-image problem.

4. A kernelized nuclear norm regularization problem

The following result shows the connection between problem (2) and a kernelized nuclear norm regularization problem with rank constraint.

Lemma 6 *If* $\mathbf{\Theta}$ *solves the following problem:*

$$\min_{\mathbf{\Theta}\in\mathbb{R}^{\ell\times m}} \left[\frac{\|\mathbf{Y}-\mathbf{K}\mathbf{\Theta}\|_F^2}{2\lambda} + \operatorname{tr}\left(\left(\mathbf{\Theta}^T\mathbf{K}\mathbf{\Theta}\right)^{1/2}\right)\right], \quad \text{subject to} \quad \operatorname{rank}(\mathbf{\Theta}) \leq p, \tag{5}$$

then the pair

$$\mathbf{L} = \left(\mathbf{\Theta}^T\mathbf{K}\mathbf{\Theta}\right)^{1/2}, \quad \mathbf{C} = \mathbf{L}^{\dagger}\mathbf{\Theta},$$

is an optimal solution of problem (2).

4.1. Special cases

In this subsection, we show that a variety of techniques such as nuclear norm regularization, reduced-rank regression, principal component analysis, and low-rank matrix approximation can be all seen as particular instances of the low-rank output kernel learning framework.

4.1.1. Linear input kernel

Let $\mathbf{X} \in \mathbb{R}^{\ell\times n}$ denote a data matrix, and assume that the input kernel is linear:

$$\mathbf{K} = \mathbf{X}\mathbf{X}^T.$$

Then, letting $\mathbf{\Phi} = \mathbf{X}^T\mathbf{\Theta}$, problem (5) reduces to *nuclear norm regularization* with a rank constraint:

$$\min_{\mathbf{\Phi}\in\mathbb{R}^{\ell\times m}} \left[\frac{\|\mathbf{Y}-\mathbf{X}\mathbf{\Phi}\|_F^2}{2\lambda} + \|\mathbf{\Phi}\|_*\right], \quad \text{subject to} \quad \operatorname{rank}(\mathbf{\Phi}) \leq p.$$

Indeed, the optimal $\mathbf{\Phi}$ for this last problem must automatically be in the range of $\mathbf{X}^T$, see e.g. (Argyriou et al., 2009, Lemma 21). Observe that the nuclear norm regularization already enforces a low rank solution (Fazel et al., 2001). Therefore, for sufficiently large values of λ, the rank constraint is not active. On the other hand, when $\lambda \to 0^+$ the solution of the previous problem converges to the *reduced-rank regression* solution:

$$\min_{\mathbf{\Phi}\in\mathbb{R}^{\ell\times m}} \|\mathbf{Y}-\mathbf{X}\mathbf{\Phi}\|_F^2, \quad \text{subject to} \quad \operatorname{rank}(\mathbf{\Phi}) \leq p.$$

If $\mathbf{Y} = \mathbf{X}$, and $\mathbf{X}$ is centered, then we obtain *principal component analysis*:

$$\min_{\mathbf{\Phi}\in\mathbb{R}^{\ell\times m}} \|\mathbf{X}\left(\mathbf{I}-\mathbf{\Phi}\right)\|_F^2, \quad \text{subject to} \quad \operatorname{rank}(\mathbf{\Phi}) \leq p.$$

Indeed, the optimal solution $\mathbf{\Phi}$ of this last problem coincides with the projection operator over the subspace spanned by the first p principal components of $\mathbf{X}$. In all these linear problems, the output kernel $\mathbf{L}$ is simply given by

$$\mathbf{L} = \left(\mathbf{\Phi}^T\mathbf{\Phi}\right)^{1/2}.$$

4.1.2. Low-rank matrix approximation

For any non-singular input kernel matrix $\mathbf{K}$, letting $\boldsymbol{\Theta} = \mathbf{K}^{-1}\mathbf{Z}$, the solution of (5) for $\lambda \to 0^+$ tends to the *low-rank matrix approximation* solution:

$$\min_{\mathbf{Z} \in \mathbb{R}^{\ell \times m}} \|\mathbf{Y} - \mathbf{Z}\|_F^2, \quad \text{subject to} \quad \text{rank}(\mathbf{Z}) \leq p.$$

The optimal $\mathbf{Z}$ coincides with a reduced singular value decomposition of $\mathbf{Y}$ (Eckart and Young, 1936).

5. Block coordinate descent for low-rank output kernel learning

In this section, we introduce an optimization algorithm (Algorithm 1) for solving problem (2) based on a block coordinate descent strategy. In particular, we consider the equivalent formulation (4), and alternate between optimization with respect to the two factors $\mathbf{A}$ and $\mathbf{B}$. First of all, we assume that an eigendecomposition of the input kernel matrix $\mathbf{K}$ is available:

$$\mathbf{K} = \mathbf{U}_X \text{diag}\left\{\lambda_X\right\} \mathbf{U}_X^T.$$

Observe that the eigendecomposition can be computed once for all at the beginning of the optimization procedure.

5.1. Sub-problem w.r.t matrix A

When $\mathbf{B}$ is fixed in equation (4), the optimization with respect to $\mathbf{A}$ is a convex quadratic problem. A necessary and sufficient condition for optimality is

$$\mathbf{0} = \frac{\partial Q}{\partial \mathbf{A}} = -\frac{\mathbf{K}\left(\mathbf{Y} - \mathbf{K}\mathbf{A}\mathbf{B}^T\right)\mathbf{B}}{\lambda} + \mathbf{K}\mathbf{A}$$

A sufficient condition is obtained by choosing $\mathbf{A}$ as the unique solution of the linear matrix equation

$$\mathbf{K}\mathbf{A}(\mathbf{B}^T\mathbf{B}) + \lambda\mathbf{A} = \mathbf{Y}\mathbf{B}.$$

Now, given the eigendecomposition

$$\mathbf{B}^T\mathbf{B} = \mathbf{U}_Y \text{diag}\left\{\lambda_Y\right\} \mathbf{U}_Y^T,$$

we have

$$\mathbf{A} = \mathbf{U}_X \mathbf{V} \mathbf{U}_Y^T,$$

where

$$\mathbf{V} = \mathbf{Q} \oslash \left(\lambda_X \lambda_Y^T + \lambda e e^T\right), \qquad \mathbf{Q} = \mathbf{U}_X^T \mathbf{Y} \mathbf{B} \mathbf{U}_Y.$$

As shown in the following, during the coordinate descent procedure it is not necessary to explicitly compute $\mathbf{A}$. In fact, it is sufficient to compute $\mathbf{V}$ as in lines 5-7 of Algorithm 1.

5.2. Sub-problem w.r.t matrix B

For any fixed $\mathbf{A}$, the sub-problem with respect to $\mathbf{B}$ is quadratic and strongly convex. The unique solution is obtained by setting

$$\mathbf{0} = \frac{\partial Q}{\partial \mathbf{B}} = -\frac{\left(\mathbf{Y} - \mathbf{K}\mathbf{A}\mathbf{B}^T\right)^T \mathbf{K}\mathbf{A}}{\lambda} + \mathbf{B},$$

which can be rewritten as

$$\mathbf{B} = \mathbf{Y}^T\mathbf{K}\mathbf{A}\left(\mathbf{A}^T\mathbf{K}^2\mathbf{A} + \lambda\mathbf{I}\right)^{-1}.$$

Now, assume that $\mathbf{A}$ is optimized as in the previous subsection. Then, after some manipulations, taking into account the eigendecomposition of $\mathbf{K}$ and $\mathbf{B}^T\mathbf{B}$, the update for $\mathbf{B}$ can be reduced to lines 9-10 of Algorithm 1.

5.3. Practical aspects

The solution in correspondence with different values of the regularization parameter λ can be computed efficiently by using a warm start procedure: the output kernel is initialized by using the result of the previous optimization, while moving from the highest value of the regularization parameter to the lowest. For each value of λ, we stop the coordinate descent procedure by checking whether the Frobenius norm of the variation of $\mathbf{B}$ from an iteration to the next is lower than a specified tolerance.

Observe that the eigenvectors of the input kernel matrix are used only outside the main loop of Algorithm 1, to properly rotate the outputs, and to reconstruct the factor $\mathbf{A}$. The eigendecomposition of $\mathbf{K}$ can be computed once for all the values of the regularization parameter: standard algorithms require $O(\ell^3)$ operations. The two key steps in each iteration of Algorithm 1 are the computation of the eigendecomposition in line 5 and the solution of the linear system in line 10. They can be both performed in $O\left(\max\{p^2 m, p^3\}\right)$ operations. The memory required to store all the matrices scales as $O(\max\{mp, \ell^2\})$.

If the number of outputs m is very large, one can choose low values of p to control both computational complexity and memory requirements. On the other hand, if there are no limitations in memory and computation time, one could set $p = m$ and use only λ to control the complexity of the model. By doing this, one is also guaranteed to obtain the global minimizer of the optimization problem. Notice that the parameters p and λ both control the rank of the resulting model.

6. Experiments: reconstruction of multiple signals

We apply low rank output kernel learning to reconstruct and denoise multiple signals. We present two experiments. In the first experiment, we compare the learning performances of Algorithm 1 with previously proposed techniques. Also, we investigate the dependence of learning performances and training time on the rank parameter. In the second experiment, we demonstrate that low rank output kernel learning scales well to datasets with a very large number of outputs, whereas full-rank techniques cannot be applied anymore. All the experiments have been run in a MATLAB environment with an Intel i5 CPU 2.4 GHz, 4 GB RAM.

Algorithm 1 Low-rank output kernel learning with block-wise coordinate descent

```
Compute eigendecomposition: $\mathbf{K} = \mathbf{U}_X \mathrm{diag}\left\{\lambda_X\right\} \mathbf{U}_X^T$
$\mathbf{B} \leftarrow \mathbf{I}_{m \times p}$
$\widetilde{\mathbf{Y}} \leftarrow \mathbf{U}_X^T \mathbf{Y}$
repeat
    Compute eigendecomposition: $\mathbf{B}^T\mathbf{B} = \mathbf{U}_Y \mathrm{diag}\left\{\lambda_Y\right\} \mathbf{U}_Y^T$
    $\mathbf{Q} \leftarrow \widetilde{\mathbf{Y}}\mathbf{B}\mathbf{U}_Y$
    $\mathbf{V} \leftarrow \mathbf{Q} \oslash \left(\lambda_X \lambda_Y^T + \lambda e e^T\right)$
    $\mathbf{B}_p \leftarrow \mathbf{B}$
    $\mathbf{E} \leftarrow \mathrm{diag}\left\{\lambda_X\right\} \mathbf{V}$
    $\mathbf{B} \leftarrow \widetilde{\mathbf{Y}}^T \mathbf{E} \left(\mathbf{E}^T\mathbf{E} + \lambda \mathbf{I}\right)^{-1} \mathbf{U}_Y^T$
until $\|\mathbf{B} - \mathbf{B}_p\|_F \geq \delta$
$\mathbf{A} \leftarrow \mathbf{U}_X \mathbf{V} \mathbf{U}_Y^T$
```

First of all, we generated 50 independent realizations Z_k, $(k = 1, \ldots, 50)$ of a Gaussian Process on the interval $[-1, 1]$ of the real line with zero-mean and covariance function

$$K(x_1, x_2) = \exp(-10|x_1 - x_2|).$$

We then generated m new processes U_j as

$$U_j = \sum_{k=1}^{50} B_{jk} Z_k,$$

where the mixing coefficients B_{jk} are independently drawn from a uniform distribution on the interval $[0, 1]$. Output data have been generated by sampling the processes U_j in correspondence with 200 uniformly spaced input points in the interval $[-1, 1]$, and corrupting them by adding a zero-mean Gaussian noise with a signal to noise ratio of 1:1.

6.1. Experiment 1

In the first experiments, we have set $m = 200$, and randomly extracted $\ell = 100$ samples to be used for training, using the remaining 100 for tuning the regularization parameter λ. The results of method (1) are compared with the baseline obtained by fixing the output kernel to the identity, and the method of Dinuzzo et al. (2011) which uses a squared Frobenius-norm regularization on the output kernel. For each value of the rank parameter $p = 1, \ldots, m$, the solution is computed for 25 logarithmically spaced values of the regularization parameter λ in the range

$$\lambda \in \left[10^{-5}\alpha, \alpha\right], \qquad \alpha = \sqrt{\|\mathbf{Y}^T\mathbf{K}\mathbf{Y}\|_2}.$$

In view of Lemma 8 in Appendix A, the optimal output kernel $\mathbf{L}$ is null for $\lambda > \alpha$. For each value of p, we used the warm start procedure (see subsection 5.3) for computing the solution in correspondence with different values of the regularization parameter. Figure 1 reports the MSE (mean squared error) on the left panel and the training time to compute the solution for all the values of λ on the right panel, as a function of the rank parameter p. In terms of

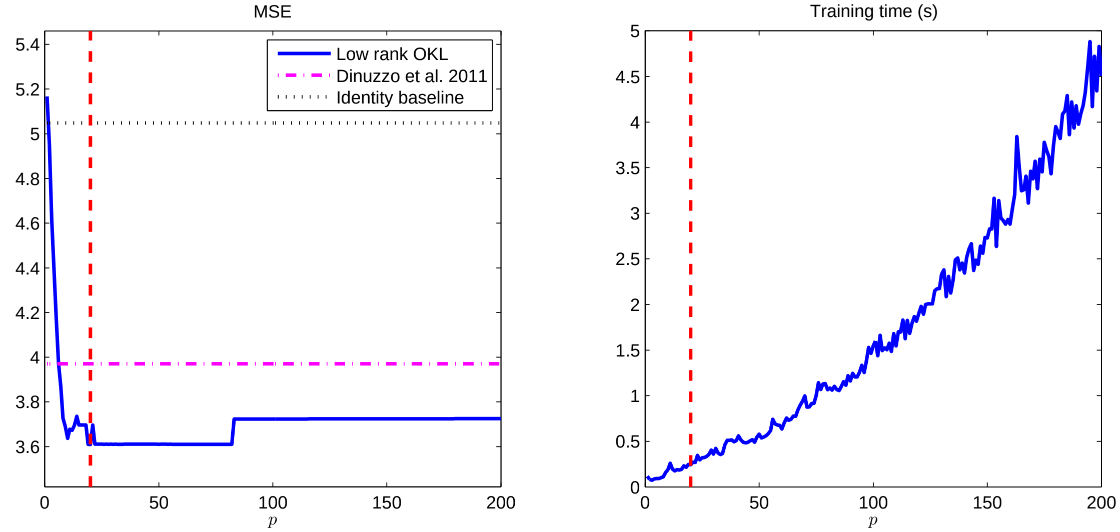

Figure 1: Experiment 1: The continuous lines represent the MSE (left panel) and training time (right panel) of Algorithm 1, as a function of the rank p. In the left panel, the horizontal baseline obtained by fixing the output kernel to the identity is dotted, while the performance of the method of (Dinuzzo et al., 2011) is dash-dotted. The vertical dashed line corresponds to the value of p that minimizes MSE.

reconstruction error, the output kernel learning method (1) outperforms both the baseline model obtained by fixing $\mathbf{L} = \mathbf{I}$, and the Frobenius-norm regularization method, within a broad range of parameter p. The best performances are observed in correspondence with low values of the rank parameter (the vertical line corresponds to $p = 20$). Finally, from the right panel of Figure 1, one can observe the dependence of the training time on p. In particular, observe that the training time in correspondence with the best low rank model is reduced by more than an order of magnitude with respect to the full rank model.

6.2. Experiment 2

Within the same setting of the previous subsection, this time we generated $m = 10^5$ processes U_j. Again, we randomly choose 100 samples from the uniform grid to be used for training and measure learning performances using the MSE (mean squared error). Observe that in this case it is not possible to learn a full-rank output kernel matrix since it would not fit into the memory. For the same reason, the method of Dinuzzo et al. (2011) (that store the full matrix $\mathbf{L}$) cannot be applied. Therefore, this time we fixed the rank parameter $p = 50$, and report performances of the Algorithm 1 for 25 different values of the regularization parameter chosen on a logarithmic scale. Performances are compared with the baseline obtained by fixing $\mathbf{L} = \mathbf{I}$. From the left panel of Figure 2, we see that the low rank output kernel performs better than the identity. The overall training time to compute the solution of (1) for the 25 values of the regularization parameter is about 45 seconds. From the right

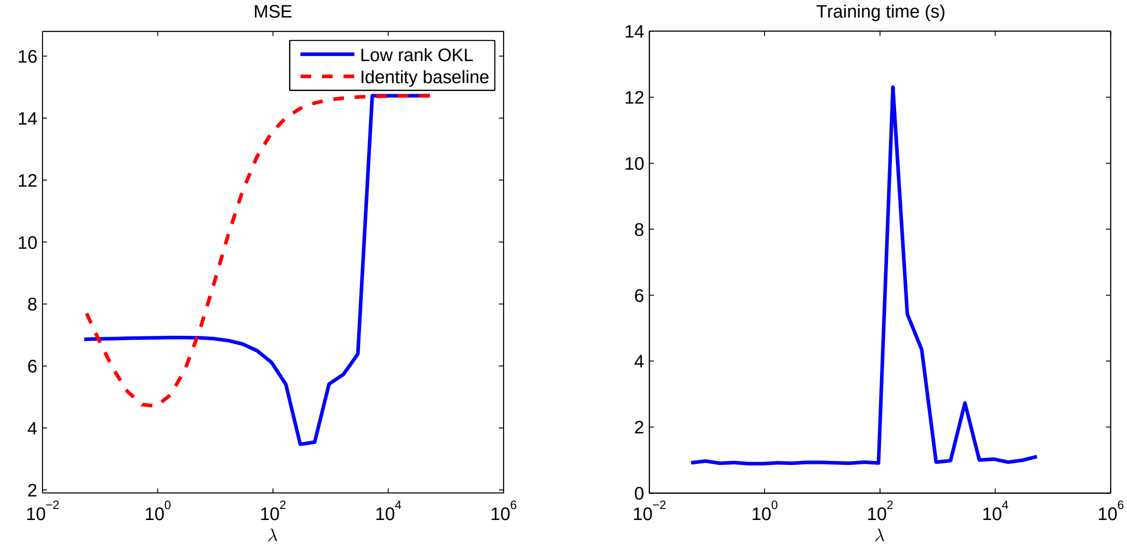

Figure 2: Experiment 2: The continuous lines represent the Mean Squared Error (left panel) and training time (right panel) of Algorithm 1, as a function of the regularization parameter λ. In the left plot, the baseline obtained by fixing the output kernel to the identity is dashed. In the right plot, the training time refers to an iteration of the warm-start procedure (see subsection 5.3).

panel of Figure 2, we can see that most of the computation time (using the warm-start procedure described in subsection 5.3) is spent in correspondence with intermediate values of λ. These values of λ also correspond to the best reconstruction performances. The right panel of Figure 2 also shows that the warm-start procedure does a good job at exploiting the continuity of the solution along a regularization path.

7. Conclusions

We have presented a new regularization-based methodology to learn low rank output kernels. The new model encodes the assumption that the output components of the vector-valued function to be learned lie on a low-dimensional subspace. We have shown that the proposed framework can be interpreted as a kernel machine with two layers as well as the kernelized counterpart of nuclear norm regularization. Finally, we have developed an effective block coordinate descent technique to solve the proposed optimization problem. By manipulating only low-rank matrices, the new approach allows to limit memory requirements and improve performances with respect to previous techniques.

Appendix A. Optimality condition

Observe that the inner optimization problem in (2) is an unconstrained quadratic minimization with respect to $\mathbf{C}$. By eliminating $\mathbf{C}$ from the objective functional and letting

$y = \mathrm{vec}(\mathbf{Y})$, where $\mathrm{vec}(\cdot)$ denotes the vectorization operator, we obtain the following problem in $\mathbf{L}$ only

$$\min_{\mathbf{L}\in\mathbb{S}_+^{m,p}} \left(\frac{y^T \left(\mathbf{L}\otimes\mathbf{K}+\lambda\mathbf{I}\right)^{-1} y}{2} + \frac{\mathrm{tr}(\mathbf{L})}{2} \right). \tag{6}$$

Problem (6) is of the form

$$\min_{\mathbf{L}\in\mathbb{S}_+^{m,p}} f(\mathbf{L}), \tag{7}$$

where f is a differentiable convex functional. Observe that, although the objective functional is convex, the feasible set is non-convex for $p < m$. Equivalently, problem (7) can be rewritten as an unconstrained minimization of a non-convex functional:

$$\min_{\mathbf{B}\in\mathbb{R}^{m\times p}} f(\mathbf{B}\mathbf{B}^T). \tag{8}$$

Now, letting

$$\mathbf{G}(\mathbf{B}) = \frac{\partial f\left(\mathbf{B}\mathbf{B}^T\right) + \partial f\left(\mathbf{B}\mathbf{B}^T\right)^T}{2},$$

we have the following result.

Lemma 7 *If $\mathbf{B}_*$ is a global minimizer for (8), then*

$$\mathbf{G}(\mathbf{B}_*)\mathbf{B}_* = \mathbf{0}. \tag{9}$$

Conversely, if (9) holds and, in addition, we have

$$\mathbf{G}(\mathbf{B}_*) \in \mathbb{S}_+^m, \tag{10}$$

then $\mathbf{B}_*$ is a global minimizer for (8).

The following result specializes the more general result of Lemma 7 to problem (6).

Lemma 8 *If $\mathbf{L} = \mathbf{B}\mathbf{B}^T$ is an optimal solution of problem (6), then there exists $\mathbf{C}\in\mathbb{R}^{\ell\times m}$ such that*

$$\mathbf{K}\mathbf{C}\mathbf{L} + \lambda\mathbf{C} = \mathbf{Y}, \tag{11}$$

$$\left(\mathbf{C}^T\mathbf{K}\mathbf{C}\right)\mathbf{B} = \mathbf{B}, \tag{12}$$

Conversely, if (11)-(12) hold and, in addition, we have

$$\|\mathbf{C}^T\mathbf{K}\mathbf{C}\|_2 \le 1, \tag{13}$$

then $\mathbf{L}$ is an optimal solution of problem (6).

A corollary of Lemma 8 is the following: if $\|\mathbf{Y}^T\mathbf{K}\mathbf{Y}\|_2 \le \lambda^2$, then $\mathbf{L} = \mathbf{0}$ is an optimal solution of problem (6). To see this, it suffices to set $\mathbf{C} = \mathbf{Y}/\lambda$. Another simple observation is that, in view of (12), the rank of any optimal $\mathbf{L}$ is always less or equal than the rank of $\mathbf{K}$. These two observations imply that the range of parameters λ and p, which define the low-rank output kernel learning algorithm, can be restricted as follows:

$$0 < \lambda \le \sqrt{\|\mathbf{Y}^T\mathbf{K}\mathbf{Y}\|_2}, \qquad 0 < p \le \min\left\{\mathrm{rank}(\mathbf{K}), m\right\}$$

without any loss of generality.

Appendix B. Proofs

Proof [of Theorem 5] First of all, we show that problems (2) and (4) are equivalent. Consider problem (2) and observe that, in view of the rank constraint, we have the decomposition $\mathbf{L} = \mathbf{B}\mathbf{B}^T$, where $\mathbf{B} \in \mathbb{R}^{m \times p}$. By letting

$$\mathbf{A} = \mathbf{C}\mathbf{B}, \tag{14}$$

problem (2) boils down to (4). Conversely, take an optimal $\mathbf{A} \in \mathbb{R}^{\ell \times p}$ for problem (4), and observe that it admits a unique decomposition of the form

$$\mathbf{A} = \mathbf{C}\mathbf{B} + \mathbf{U}, \qquad \mathbf{U}\mathbf{B}^T = \mathbf{0}.$$

We have

$$\mathbf{K}\mathbf{A}\mathbf{B}^T = \mathbf{K}\mathbf{C}\mathbf{B}\mathbf{B}^T = \mathbf{K}\mathbf{C}\mathbf{L},$$

and

$$\frac{\langle \mathbf{A}, \mathbf{K}\mathbf{A}\rangle_F}{2} = \frac{\langle \mathbf{U}, \mathbf{K}\mathbf{U}\rangle_F}{2} + \frac{\langle \mathbf{C}^T\mathbf{K}\mathbf{C}, \mathbf{L}\rangle_F}{2} \geq \frac{\langle \mathbf{C}^T\mathbf{K}\mathbf{C}, \mathbf{L}\rangle_F}{2}.$$

It follows that we can set $\mathbf{U} = \mathbf{0}$, so that $\mathbf{A}$ can be written as in (14) without any loss of generality. Finally, in view of (14), the optimal g for problems (1) and (3) coincide. ■

Proof [of Lemma 6] Letting $\mathbf{\Theta} = \mathbf{C}\mathbf{L}$, problem (2) can be rewritten as

$$\min_{\mathbf{\Theta} \in \mathbb{R}^{\ell \times m}} \left[\frac{\|\mathbf{Y} - \mathbf{K}\mathbf{\Theta}\|_F^2}{2\lambda} + \min_{\mathbf{L} \in \mathbb{S}_+^{m,p}} \left(\frac{\langle \mathbf{\Theta}^T\mathbf{K}\mathbf{\Theta}, \mathbf{L}^\dagger \rangle_F}{2} + \frac{\operatorname{tr}(\mathbf{L})}{2} \right) \right], \quad \text{subject to} \quad \operatorname{rg}(\mathbf{\Theta}) \subseteq \operatorname{rg}(\mathbf{L}).$$

Now, it turns out that a closed-form solution of the inner minimization problem is given by

$$\mathbf{L} = \left(\mathbf{\Theta}^T\mathbf{K}\mathbf{\Theta}\right)^{1/2}.$$

Indeed, such a solution is the only stationary point of the unconstrained functional, and also satisfies the constraints, provided that

$$\operatorname{rank}(\mathbf{\Theta}) = \operatorname{rank}(\mathbf{L}) \leq p.$$

By plugging the expression of $\mathbf{L}$ into the objective functional, we obtain problem (5). ■

Proof [of Lemma 7] Let

$$g(\mathbf{B}) = f\left(\mathbf{B}\mathbf{B}^T\right),$$

and observe that

$$\operatorname{vec}(\nabla g(\mathbf{B})) = 2\operatorname{vec}(\mathbf{G}(\mathbf{B}))\left(\mathbf{B} \otimes \mathbf{I}\right) = \operatorname{vec}(2\mathbf{G}(\mathbf{B})\mathbf{B}),$$

so that

$$\nabla g(\mathbf{B}) = 2\mathbf{G}(\mathbf{B})\mathbf{B}.$$

Since g is differentiable, every global minimizer must be a stationary point, therefore equation (9) follows. Now, assume that (9) and (10) holds. Then, by convexity of f we have

$$\begin{aligned} g(\mathbf{B}_*) - g(\mathbf{B}) &= f\left(\mathbf{B}_*\mathbf{B}_*^T\right) - f\left(\mathbf{B}\mathbf{B}^T\right) \leq \langle \nabla f\left(\mathbf{B}_*\mathbf{B}_*^T\right), \mathbf{B}_*\mathbf{B}_*^T - \mathbf{B}\mathbf{B}^T\rangle_F \\ &= \langle \mathbf{G}(\mathbf{B}_*), \mathbf{B}_*\mathbf{B}_*^T - \mathbf{B}\mathbf{B}^T\rangle_F \\ &= -\langle \mathbf{G}(\mathbf{B}_*), \mathbf{B}\mathbf{B}^T\rangle_F \\ &= -\text{tr}\left(\mathbf{B}^T\mathbf{G}(\mathbf{B}_*)\mathbf{B}\right) \leq 0, \end{aligned}$$

so that

$$g(\mathbf{B}_*) \leq g(\mathbf{B}), \quad \forall \mathbf{B} \in \mathbb{R}^{m\times p},$$

and $\mathbf{B}_*$ is a global minimizer. ■

Proof [of Lemma 8] Apply Lemma 7 to the objective functional of problem (6). It suffices to observe that $\mathbf{G}(\mathbf{B})$ can be expressed as

$$\mathbf{G}(\mathbf{B}) = \frac{1}{2}\left(\mathbf{I} - \mathbf{C}^T\mathbf{K}\mathbf{C}\right),$$

where $\mathbf{C}$ satisfies the linear matrix equation (11). Then, conditions (9)-(10) boil down to (12)-(13), respectively. ■

References

M. A. Alvarez and N. D. Lawrence. Computationally efficient convolved multiple output Gaussian processes. *Journal of Machine Learning Research*, 12:1459–1500, 2011.

T. W. Anderson. Estimating linear restrictions on regression coefficients for multivariate normal distributions. *Annals of Mathematical Statistics*, 22:327–351, 1951.

A. Argyriou, C. A. Micchelli, and M. Pontil. When is there a representer theorem? Vector versus matrix regularizers. *Journal of Machine Learning Research*, 10:2507–2529, 2009.

N. Aronszajn. Theory of reproducing kernels. *Transactions of the American Mathematical Society*, 68:337–404, 1950.

L. Baldassarre, L. Rosasco, A. Barla, and A. Verri. Vector field learning via spectral filtering. In *Machine Learning and Knowledge Discovery in Databases*, volume 6321, pages 56–71. Springer, 2010.

E. Bonilla, K. Ming Chai, and C. Williams. Multi-task Gaussian process prediction. In J.C. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems*, pages 153–160. MIT Press, Cambridge, MA, 2008.

A. Caponnetto, C. A. Micchelli, M. Pontil, and Y. Ying. Universal multi-task kernels. *Journal of Machine Learning Research*, 9:1615–1646, 2008.

F. Dinuzzo, C. S. Ong, P. Gehler, and G. Pillonetto. Learning output kernels with block coordinate descent. In *Proceedings of the 28th Annual International Conference on Machine Learning*, Bellevue, WA, USA, 2011.

C. Eckart and G. Young. The approximation of one matrix by another of lower rank. *Psychometrika*, (1):211–218, 1936.

T. Evgeniou, C. A. Micchelli, and M. Pontil. Learning multiple tasks with kernel methods. *Journal of Machine Learning Research*, 6:615–637, 2005.

M. Fazel, H. Hindi, and S. Boyd. A rank minimization heuristic with application to minimum order system approximation. In *Proceedings of the American Control Conference*, volume 6, pages 4734–4739, Arlington, Virginia, June 2001.

P. Goovaerts. *Geostatistics For Natural Resources Evaluation.* Oxford University Press, USA, 1997.

A. J. Izenman. Reduced-rank regression for the multivariate linear model. *Journal of Multivariate Analysis*, 5(2):248–264, 1975.

I. T. Jolliffe. *Principal Component Analysis.* Springer, New York, NY, USA, 1986.

G. Kimeldorf and G. Wahba. Some results on Tchebycheffian spline functions. *Journal of Mathematical Analysis and Applications*, 33(1):82–95, 1971.

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

S. K. Mishra and G. Giorgi. *Invexity and optimization.* Nonconvex Optimization and Its Applications. Springer, Dordrecht, 2008.

G. H. Reinsel and R. P. Velu. *Multivariate reduced rank regression: theory and applications.* Springer, New York, 1998.

B. Schölkopf, A. J. Smola, and K-R Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10(5):1299–1319, 1998.

B. Schölkopf, S. Mika, C.J.C. Burges, P. Knirsch, K.-R. Müller, G. Rätsch, and A.J. Smola. Input space versus feature space in kernel-based methods. *IEEE Transactions on Neural Networks*, 10(5):1000 –1017, sep 1999.

B. Schölkopf, R. Herbrich, and A. J. Smola. A generalized representer theorem. *Neural Networks and Computational Learning Theory*, 81:416–426, 2001.

A. Zien and C. S. Ong. Multiclass multiple kernel learning. In *Proceedings of the 24th international conference on Machine learning*, pages 1191–1198, 2007.