Multiple Operator-valued Kernel Learning

Hachem Kadri, Alain Rakotomamonjy, Francis Bach, Philippe Preux

INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Multiple Operator-valued Kernel Learning

Hachem Kadri — Alain Rakotomamonjy — Francis Bach — Philippe Preux

N° 7900

February 2012

Thème COG

ISSN 0249-6399 ISRN INRIA/RR--7900--FR+ENG

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE
INRIA
centre de recherche LILLE - NORD EUROPE

Multiple Operator-valued Kernel Learning

Hachem Kadri*, Alain Rakotomamonjy†, Francis Bach‡, Philippe Preux§

Thème COG — Systèmes cognitifs
Équipes-Projets SequeL

Rapport de recherche n° 7900 — February 2012 — 17 pages

Abstract: This paper addresses the problem of learning a finite linear combination of operator-valued kernels. We study this problem in the case of kernel ridge regression for functional responses with a ℓ_r-norm constraint on the combination coefficients ($r \geq 1$). We propose a multiple operator-valued kernel learning algorithm based on solving a system of linear operator equations by using a block coordinate descent procedure. We experimentally validate our approach on a functional regression task in the context of finger movement prediction in Brain-Computer Interface (BCI).

Key-words: Operator-valued kernels, multiple kernel learning, nonparametric functional data analysis, function-valued reproducing kernel Hilbert spaces

* Sequel Team, INRIA Lille. E-mail: hachem.kadri@inria.fr
† LITIS, Université de Rouen. E-mail: alain.rakotomamonjy@insa-rouen.fr
‡ Sierra Team/INRIA, Ecole Normale Supérieure. E-mail: francis.bach@inria.fr
§ Sequel/INRIA-Lille, LIFL/CNRS. E-mail: philippe.preux@inria.fr

Centre de recherche INRIA Lille – Nord Europe
Parc Scientifique de la Haute Borne
40, avenue Halley, 59650 Villeneuve d'Ascq
Téléphone : +33 3 59 57 78 00 — Télécopie : +33 3 59 57 78 50

Apprentissage de Noyaux à Valeurs Opérateurs Multiples

Résumé : Dans cet article, nous proposons une méthode d'apprentissage de noyaux multiples à valeurs opérateurs dans le cas d'une régression ridge à réponse fonctionnelle. Notre méthode est basée sur la résolution d'un système d'équations linéaires d'opérateurs en utilisant une procédure de type Iterative Coordinate Descent. Nous validons expérimentalement notre approche sur un problème de prédiction de mouvement de doigt dans un contexte d'Interface Cerveau-Machine.

Mots-clés : noyaux à valeurs opérateurs, apprentissage de noyaux multiples, analyse des données fonctionnelles, espace de Hilbert à noyau reproduisant

1 Introduction

During the past decades, a large number of algorithms have been proposed to deal with learning problems in the case of single-valued functions (*e.g.*, binary-output function for classification or a real number output for regression). Recently, there has been considerable interest in estimating vector-valued functions (Micchelli and Pontil, 2005b). Much of this interest has arisen from the need to learn tasks where the target is a complex entity, not a scalar variable as usual. Typical learning situations include multitask learning (Evgeniou et al., 2005), structured output learning (Brouard et al., 2011; Tsochantaridis et al., 2005), manifold-valued regression (Hein, 2009) and object localization for image understanding (Blaschko and Lampert, 2008).

In this paper, we are interested in the problem of structured output classification and regression in the context of Brain-Computer Interface (BCI) design. More precisely, we are interested in finger movement prediction from electrocorticographic signals (Schalk et al., 2007). Indeed, from a set of signals measuring brain surface electrical activity on d channels during a given period of time, we want to predict, for any instant of that period whether a finger is moving or not and the amplitude of the finger flexion. Formally, the problem consists in learning a functional dependency between a set of d signals and a vector of labels and between the same set of signals and vector of real values (the amplitude). While, it is clear that this problem can be formalized as functional regression problem, from our point of view, such problem can benefit from multiple operator-valued kernel learning framework. Indeed, for these problems, one of the difficulties arises from the unknown latency between the signal set-on related to the finger movement and the actual movement (Pistohl et al., 2008). Hence, instead of fixing in advances some value of this latency in the regression model, our framework allows to learn it from the data by means of several operator-valued kernels.

Data streams are more and more commonly encountered in data mining and machine learning; for instance, data streams are very common in Brain Computer Interface research. A data stream may be either of discrete nature, or of continuous nature; in any case, it comes as a discrete series of objects (scalars, or more complex entities). In the case of continuous streams, the stream is really a function rather than a vector. If we wish to address such data in the sound framework of reproducing kernel Hilbert spaces (RKHS), we have to consider RKHS which elements are operators that map a function to an other function space, possibly source and target function spaces being different. Working in such RKHS, we are able to draw on the important core of work that have been performed on real RKHS, and multi-valued real RKHS. Such a functional RKHS framework and associated operator-valued kernels has been introduced very recently (Kadri et al., 2010, 2011); the present paper aims at building on these early works, and addresses in particular the problem of learning simultaneously a function-valued function and the operator-valued kernel.

Reproducing kernels play an important role in statistical learning theory and functional estimation. Scalar-valued kernels are widely used to design nonlinear learning methods which have been successfully applied in several machine learning applications (Schölkopf and Smola, 2002). Moreover, their extension to matrix-valued kernels has helped to bring about additional improvements in learning vector-valued functions (Micchelli and Pontil, 2005b;

Reisert and Burkhardt, 2007). The most common and most successful applications of matrix-valued kernel methods are in multi-task learning (Evgeniou et al., 2005), even though some successful applications in other areas, for example image processing (Ha Quang et al., 2010; Reisert and Burkhardt, 2007), also exist. A basic question always present with reproducing kernels is how to build these kernels and what is the optimal kernel choice.

In order to overcome the need of choosing a kernel before the learning process, several works have tried to address the problem of learning the kernel jointly with the decision function (Lanckriet et al., 2004; Bach et al., 2004). Since these seminal works, many efforts have been carried out in order to theoretically analyze the kernel learning framework (Cortes et al., 2010) or in order to provide efficient algorithms (Kloft et al., 2011; Aflalo et al., 2011). While a lot of works have been devoted to multiple scalar kernel learning, this problem of kernel learning have been barely investigated for operator-valued kernels. Thus, in this paper, we bridge the gap between multiple kernel learning and operator-valued kernels by proposing a framework and an algorithm for learning a finite linear combination of operator-valued kernels. The MKL framework can be extended without major difficulties to operator-valued kernels, however the resulting optimization problem is very challenging as it involves linear systems with sum of operators which are hardly invertible. For coping with this issue, we propose an elegant algorithm based on a variational reformulation of the problem and block-coordinate descent.

It should be pointed out that in a recent work (Dinuzzo et al., 2011), the authors formulated the problem of learning the output kernel as an optimization problem over the cone of positive semidefinite matrices, and proposed a block-coordinate descent method to solve it. However, they did not focus on learning the input kernel. In contrast, our Multiple operator-valued Kernel Learning (MovKL) formulation can be seen as a way of learning simultaneously input and output kernels, although we consider a linear combination of kernels fixed in advance.

The paper is organized as follows. In Section 2, we begin by a brief review of reproducing kernel Hilbert spaces with operator-valued kernels, and derive the Multiple operator-valued Kernel Learning (MovKL) problem. Section 3 shows how to solve this problem using a block coordinate descent algorithm. The proposed algorithm is experimentally evaluated in Section 4 on a functional regression task in the context of Brain-Computer Interface (BCI) design. Finally, Section 5 presents some conclusions and future work directions.

2 Problem Setting

First, we briefly review notions and properties of reproducing kernel Hilbert spaces with operator-valued kernels and show their connection to learning from multiple response data (multiple outputs ; see Micchelli and Pontil (2005b) for discrete data and Kadri et al. (2010) for continuous data). We first consider the problem of estimating a function f such that $f(x_i) = y_i$ when observed data $(x_i, y_i)_{i=1,\ldots,n}$ are assumed to be elements of infinite dimensional Hilbert spaces. In the following we denote by $\mathcal{G}_x$ and $\mathcal{G}_y$ the domains of x_i and y_i respectively. $X = \{x_1, \ldots, x_n\}$ denotes the training set with corresponding targets $Y = \{y_1, \ldots, y_n\}$. Since $\mathcal{G}_x$ and $\mathcal{G}_y$ are spaces of functions, the problem

can be thought of as an operator estimation problem, where the desired operator maps a Hilbert space of factors to a Hilbert space of targets. We can define the regularized operator estimate of $f \in \mathcal{F}$

$$f_\lambda \triangleq \arg\min_{f\in\mathcal{F}} \sum_{i=1}^{n} \|y_i - f(x_i)\|^2_{\mathcal{G}_y} + \lambda\|f\|^2_{\mathcal{F}} \tag{1}$$

In this work, we are looking for a solution to this minimization problem in a reproducing kernel Hilbert space $\mathcal{F}$ of function-valued functions on some infinite-dimensional input space $\mathcal{G}_x$. We start by introducing function-valued reproducing kernel Hilbert spaces and showing the correspondence between such spaces and positive operator-valued kernels. Bijection between scalar-valued kernel and RKHS was first established by Aronszajn (1950). Then Schwartz (1964) shows that this was a particular case of a more general situation. More recently, interest has grown in exploring Hilbert spaces of vector random functions for learning vector-valued functions (Micchelli and Pontil, 2005b; Carmeli et al., 2006). The function-valued RKHS approach extends the vector-valued case to infinite-dimensional output data (Wahba, 1992; Kadri et al., 2010).

Let $\mathcal{G}_x$ and $\mathcal{G}_y$ be infinite-dimensional Hilbert spaces and $\mathcal{F}$ a linear space of operators on $\mathcal{G}_x$ with values in $\mathcal{G}_y$. We assume that $\mathcal{F}$ is a Hilbert space with inner product $\langle\cdot,\cdot\rangle_{\mathcal{F}}$. Let $\mathcal{L}(\mathcal{G}_y)$ be the set of bounded linear operators from $\mathcal{G}_y$ to $\mathcal{G}_y$.

Definition 1 *(function-valued RKHS)*
A Hilbert space $\mathcal{F}$ of functions from $\mathcal{G}_x$ to $\mathcal{G}_y$ is called a reproducing kernel Hilbert space if there is a nonnegative $\mathcal{L}(\mathcal{G}_y)$-valued kernel $K_{\mathcal{F}}(w,z)$ on $\mathcal{G}_x \times \mathcal{G}_x$ such that:

i. the function $z \longmapsto K_{\mathcal{F}}(w,z)g$ belongs to $\mathcal{F}$, $\forall z \in \mathcal{G}_x$, $w \in \mathcal{G}_x$, $g \in \mathcal{G}_y$,

ii. $\forall f \in \mathcal{F}, \langle f, K_{\mathcal{F}}(w,\cdot)g\rangle_{\mathcal{F}} = \langle f(w), g\rangle_{\mathcal{G}_y}$ (reproducing property).

Definition 2 *(operator-valued kernel)*
An $\mathcal{L}(\mathcal{G}_y)$-valued kernel $K_{\mathcal{F}}(w,z)$ on $\mathcal{G}_x$ is a function $K_{\mathcal{F}}(\cdot,\cdot) : \mathcal{G}_x \times \mathcal{G}_x \longrightarrow \mathcal{L}(\mathcal{G}_y)$; furthermore:

i. $K_{\mathcal{F}}$ is Hermitian if $K_{\mathcal{F}}(w,z) = K_{\mathcal{F}}(z,w)^*$, where $K_{\mathcal{F}}(z,w)^*$ is the adjoint operator of $K_{\mathcal{F}}(z,w)$

ii. $K_{\mathcal{F}}$ is nonnegative on $\mathcal{G}_x$ if it is Hermitian and for every natural number r and all $\{(w_i,u_i)_{i=1,\ldots,r}\} \in \mathcal{G}_x \times \mathcal{G}_y$, the block matrix with ij-th entry $\langle K_{\mathcal{F}}(w_i,w_j)u_i, u_j\rangle_{\mathcal{G}_y}$ is nonnegative.

Theorem 1 *(bijection between function valued RKHS and operator-valued kernel)*
A $\mathcal{L}(\mathcal{G}_y)$-valued kernel $K_{\mathcal{F}}(w,z)$ on $\mathcal{G}_x$ is the reproducing kernel of some Hilbert space $\mathcal{F}$, if and only if it is positive definite.

2.1 Functional Response Ridge Regression in Dual Variables

We can write the ridge regression with functional response optimization problem (1) in the following form:

$$\begin{aligned} &\min_{f\in\mathcal{F}} \|f\|^2_{\mathcal{F}} + \tfrac{1}{\lambda}\sum_i \|\xi_i\|^2_{\mathcal{G}_y} \\ &\text{with } \xi_i = y_i - f(x_i) \end{aligned} \tag{2}$$

Now, we introduce the Lagrangian multipliers $\alpha_i, i = 1, \ldots, n$ which are functional variables since the output space is the space of functions $\mathcal{G}_y$. Let $\alpha = (\alpha_i)_{i=1,\ldots,n} \in \mathcal{G}_y^n$ the vector of functions containing the Lagrangian multipliers, the Lagrangian function is defined as

$$L(f, \alpha, \xi) = \|f\|_{\mathcal{F}}^2 + \frac{1}{\lambda}\|\xi\|_{\mathcal{G}_y^n}^2 + \langle \alpha, y - f(x) - \xi \rangle_{\mathcal{G}_y^n} \tag{3}$$

where $\alpha = (\alpha_1, \ldots, \alpha_n) \in \mathcal{G}_y^n$, $y = (y_1, \ldots, y_n) \in \mathcal{G}_y^n$, $f(x) = (f(x_1), \ldots, f(x_n)) \in \mathcal{G}_y^n$, $\xi = (\xi_1, \ldots, \xi_n) \in \mathcal{G}_y^n$ and $\forall a, b \in \mathcal{G}_y^n$, $\langle a, b \rangle_{\mathcal{G}_y^n} = \sum_{i=1}^{n} \langle a_i, b_i \rangle_{\mathcal{G}_y}$

Differentiating (3) in f and setting to zero, we obtain

$$f(.) = \frac{1}{2}\sum_{i=1}^{n} K(x_i, .)\alpha_i \tag{4}$$

where $K : \mathcal{G}_x \times \mathcal{G}_x \longrightarrow \mathcal{L}(\mathcal{G}_y)$ is the operator-valued kernel of $\mathcal{F}$.

Substituting this into (3), the problem (2) becomes

$$\min_{\xi} \max_{\alpha} -\frac{1}{4}\langle \mathbf{K}\alpha, \alpha \rangle_{\mathcal{G}_y^n} + \frac{1}{\lambda}\|\xi\|_{\mathcal{G}_y^n}^2 + \langle \alpha, y - \xi \rangle_{\mathcal{G}_y^n} \tag{5}$$

where $\mathbf{K} = [K(x_i, x_j)]_{i,j=1}^n$ is the block operator kernel matrix.

Differentiating (5) in ξ, we obtain $\xi = \frac{\lambda}{2}\alpha$ and then the dual of the functional response ridge regression problem is given by

$$\max_{\alpha} -\lambda\|\alpha\|_{\mathcal{G}_y^n}^2 - \langle \mathbf{K}\alpha, \alpha \rangle_{\mathcal{G}_y^n} + 4\langle \alpha, y \rangle_{\mathcal{G}_y^n} \tag{6}$$

2.2 MovKL in Dual Variables

Let us now consider that the function $f(\cdot)$ is sum of M functions $\{f_k(\cdot)\}_{k=1}^M$ where each f_k belongs to an operator-valued RKHS of kernel $K_k(\cdot, \cdot)$. Similarly to scalar-valued multiple kernel learning, we can cast the problem of learning these functions f_k as

$$\begin{array}{cc} \min\limits_{\mathbf{d} \in \mathcal{D}} & \min\limits_{f_k \in \mathcal{F}_k} \sum_{k=1}^{M} \frac{\|f_k\|_{\mathcal{F}_k}^2}{d_k} + \frac{1}{\lambda}\sum\limits_i \|\xi_i\|_{\mathcal{G}_y}^2 \\ & \text{with } \xi_i = y_i - \sum_{k=1}^{M} f_k(x_i) \end{array} \tag{7}$$

with $\mathbf{d} = [d_1, \cdots, d_M]$, $\mathcal{D} = \{\mathbf{d} : \forall k, d_k \geq 0 \text{ and } \sum_k d_k^r \leq 1\}$ and $1 \leq r \leq \infty$. Following the lines of Rakotomamonjy et al. (2008), a partial dualization of this problem leads to the following equivalent one

$$\min_{\mathbf{d} \in \mathcal{D}} \max_{\alpha \in \mathcal{G}_y^n} -\lambda\|\alpha\|_{\mathcal{G}_y^n}^2 - \langle \mathbf{K}\alpha, \alpha \rangle_{\mathcal{G}_y^n} + 4\langle \alpha, y \rangle_{\mathcal{G}_y^n} \tag{8}$$

where $\mathbf{K} = \sum_{k=1}^{M} d_k \mathbf{K}_k$ and $\mathbf{K}_k$ is the block operator kernel matrix associated to the operator-valued kernel K_k. The KKT conditions also state that at optimality we have $f_k(\cdot) = \frac{d_k}{2}\sum_i K_k(x_i, \cdot)\alpha_i$.

3 Solving the MovKL Problem

After having presented the framework, we now devise an algorithm for solving this multiple operator-valued kernel learning.

3.1 Block-coordinate descent algorithm

Since the optimization problem has the same structure as a multiple scalar kernel learning problem, we can build our algorithm upon the MKL literature. Hence, we propose to borrow from Kloft et al. (2011), and consider a block-coordinate descent method. The convergence of a block coordinate descent algorithm which is related closely to the Gauss-Seidel method was studied in works of Tseng (2001), Tseng and Yun (2009) and others. The difference here is that we have operators and block operator matrices rather than matrices and block matrices, but this doesn't increase the complexity if the inverse of the operators are computable (analytically or by spectral decomposition). Our algorithm iteratively solves the problem with respects to α with $\mathbf{d}$ being fixed and then with respects to $\mathbf{d}$ with α being fixed. This boils down to the following steps :

1. with $\{d_k\}$ fixed , the resulting optimization problem with respects to α has a simple form which solution is given by:

$$(\mathbf{K} + \lambda I)\alpha = 2y \tag{9}$$

   where $\mathbf{K} = \sum_{k=1}^{M} d_k \mathbf{K}_k$. While the form of solution is rather simple, solving this linear system is very challenging and we propose an algorithm for its resolution in the sequel.

2. with $\{f_k\}$ fixed, according to problem (7), we can rewrite the problem as

$$\min_{\mathbf{d}\in\mathcal{D}} \sum_{k=1}^{M} \frac{\|f_k\|^2_{\mathcal{F}_k}}{d_k} \tag{10}$$

   which has a closed-form solution and for which optimality occurs at: (see Micchelli and Pontil (2005a))

$$d_k = \frac{\|f_k\|^{\frac{2}{r+1}}}{(\sum_k \|f_k\|^{\frac{2r}{r+1}})^{1/r}} \tag{11}$$

This algorithm is similar to that of Cortes et al. (2009) and Kloft et al. (2011) both being based on alternate optimization. The difference here is the fact that we have to invert a block operator kernel matrix which is the combination of basic kernel matrices associated to M operator-valued kernels. This makes the resolution of the system very challenging, and we present an algorithm for solving it in the next paragraph.

3.2 Solving a linear system with multiple block operator-valued kernels

One common way to construct operator valued kernels is to build scalar-valued ones which are carried over to the vector-valued (resp. function-valued) setting

Algorithm 1 ℓ_r norm MovKL

Input $\mathbf{K}_k$ for $k = 1, \ldots, M$

$d_k^1 \longleftarrow \frac{1}{M}$ for $k = 1, \ldots, M$

$\alpha \longleftarrow 0$

for $t = 1, 2, \ldots$ **do**

  $\alpha' \longleftarrow \alpha$

  $\mathbf{K} \longleftarrow \sum_k d_k^t \mathbf{K}_k$

  $\alpha \longleftarrow$ solution of $(\mathbf{K} + \lambda I)\alpha = 2y$

  if $\|\alpha - \alpha'\| < \epsilon$ **then**

    break

  end if

  $d_k^{t+1} \longleftarrow \frac{\|f_k\|^{\frac{2}{r+1}}}{(\sum_k \|f_k\|^{\frac{2r}{r+1}})^{1/r}}$ for $k = 1, \ldots, M$

end for

by a positive definite matrix (resp. operator). In this setting an operator-valued kernel has the following form:

$$K(w, z) = G(w, z)T$$

where G is a scalar-valued kernel and T is an operator in $\mathcal{L}(\mathcal{G}_y)$. In multi-task learning, T is a finite dimensional matrix that is expected to share information between tasks (Evgeniou et al., 2005; Caponnetto et al., 2008). More recently and for supervised functional output learning problems, T is chosen to be a multiplication or an integral operator (Kadri et al., 2010, 2011). This choice is motivated by the fact that functional linear models for functional responses (Ramsay and Silverman, 2005) are based on these operators and then such kernels provide an interesting alternative to extend these models to nonlinear contexts. In addition, some works on functional regression and structured output learning consider operator-valued kernels constructed from the identity operator as in Lian (2007) and Brouard et al. (2011). In this work we adopt a functional data analysis point of view and then we are interested in a finite combination of operator-valued kernels constructed from the identity, multiplication and integral operators. A problem encountered when working with operator-valued kernels in infinite dimensional spaces is that of solving the system of linear operator equations (9). In the following we show how to solve this problem for two cases of operator-valued kernel combinations.

Case 1: multiple scalar-valued kernels and one operator. This is the simpler case where the combination of operator-valued kernels has the following form

$$K(w, z) = \sum_{k=1}^{M} d_k G_k(w, z)T \tag{12}$$

In this setting, the block operator kernel matrix $\mathbf{K}$ can be expressed by a Kronecker product between the multiple scalar-valued kernel matrix $\mathbf{G} = \sum_{k=1}^{M} d_k \mathbf{G}_k$,

Algorithm 2 Gauss-Seidel Method

choose an initial vector of functions $\alpha^{(0)}$
for $t = 1, 2, \ldots$
  for $i = 1, 2, \ldots, n$
    $\alpha_i^{(t)} \longleftarrow$ sol. of (14): $[K(x_i, x_i) + \lambda I]\alpha_i^{(t)} = s_i$
  end for
  check convergence; continue if necessary
end for

where $\mathbf{G}_k = [G_k(x_i, x_j)]_{i,j=1}^n$, and the operator T. Thus we can compute an analytic solution of the system of equations (9) by inverting $\mathbf{K} + \lambda I$ using the eigendecompositions of $\mathbf{G}$ and T as in Kadri et al. (2011).

Case 2: multiple scalar-valued kernels and multiple operators. This is the general case where multiple operator-valued kernels are combined as follows

$$K(w, z) = \sum_{k=1}^{M} d_k G_k(w, z) T_k \tag{13}$$

Inverting the associated block operator kernel matrix $\mathbf{K}$ is not feasible in this case, that is why we propose a Gauss-Seidel iterative procedure (see Algorithm 2) to solve the system of linear operator equations (9). Starting from an initial vector of functions $\alpha^{(0)}$, the idea is to iteratively compute, until a convergence condition is satisfied, the functions α_i according to the following expression

$$\begin{aligned} [K(x_i, x_i) + \lambda I]\alpha_i^{(t)} &= 2y_i - \sum_{j=1}^{i-1} K(x_i, x_j)\alpha_j^{(t)} \\ &\quad - \sum_{j=i+1}^{n} K(x_i, x_j)\alpha_j^{(t-1)} \end{aligned} \tag{14}$$

where t is the iteration index. This problem is still challenging because the kernel $K(\cdot, \cdot)$ still involves a positive combination of operator-valued kernels. Our algorithm is based on the idea that instead of inverting the finite combination of operator-valued kernels $[K(x_i, x_i) + \lambda I]$, we can consider the variational formulation of this system

$$\min_{\alpha_i^{(t)}} \frac{1}{2} \langle \sum_{k=1}^{M+1} K_k(x_i, x_i)\alpha_i^{(t)}, \alpha_i^{(t)} \rangle_{\mathcal{G}_y} - \langle s_i, \alpha_i^{(t)} \rangle_{\mathcal{G}_y}$$

where

$$s_i = 2y_i - \sum_{j=1}^{i-1} K(x_i, x_j)\alpha_j^{(t)} - \sum_{j=i+1}^{n} K(x_i, x_j)\alpha_j^{(t-1)},$$

$K_k = d_k G_k T_k$, $\forall k \in \{1, \ldots, M\}$, and $K_{M+1} = \lambda I$. Now, by means of a variable splitting approach, we are able to decouple the role of the different kernels.

Indeed, the above problem is equivalent to the following one :

$$\min_{\boldsymbol{\alpha}_i^{(t)}} \frac{1}{2} \langle \hat{\mathbf{K}}(x_i, x_i)\boldsymbol{\alpha}_i^{(t)}, \boldsymbol{\alpha}_i^{(t)} \rangle_{\mathcal{G}_y^M} - \langle \mathbf{s_i}, \boldsymbol{\alpha}_i^{(t)} \rangle_{\mathcal{G}_y^M}$$
$$\text{with} \quad \alpha_{i,1}^{(t)} = \alpha_{i,k}^{(t)} \quad \text{for} \quad k = 2, \ldots, M+1$$

where $\hat{\mathbf{K}}(x_i, x_i)$ is the $M+1 \times M+1$ diagonal matrix $[K_k(x_i, x_i)]_{k=1}^{M+1}$. $\boldsymbol{\alpha}_i^{(t)}$ is the vector $(\alpha_{i,1}^{(t)}, \ldots, \alpha_{i,M+1}^{(t)})$ and the $M+1$-dimensional vector $\mathbf{s_i} = (s_i, 0, \ldots, 0)$. We now have to deal with a quadratic optimization problem with equality constraints. Writing down the Lagrangian of this optimization problem and then deriving its first-order optimality conditions leads us to the following set of linear equations

$$\begin{cases} K_1(x_i, x_i)\alpha_{i,1} - s + \sum_{k=1}^{M} \gamma_k &= 0 \\ K_k(x_i, x_i)\alpha_{i,k} - \gamma_k &= 0 \\ \alpha_{i,1} - \alpha_{i,k} &= 0 \end{cases} \tag{15}$$

where $k = 2, \cdots, M+1$ and $\{\gamma_k\}$ are the Lagrangian multipliers related to the M equality constraints. Finally, in these set of equations, the operator-valued kernels have been decoupled and thus, if their inversion can be easily computed (which is the case in our experiments), one can solve the problem (15) by means of another Gauss-Seidel algorithm.

4 Experiments

In order to highlight the benefit of our multiple operator-valued kernel learning approach, we have considered a series of experiments on a real dataset, involving structured output prediction in a Brain-Computer Interface framework. The problem we addressed is a sub-problem related to finger movement decoding from electrocorticographic signals (ECoG). We focus on the problem of estimating a finger is moving or not and also on the direct estimation of the finger movement amplitude from the ECoG signals. The development of the full BCI application is beyond the scope of this paper and our objective here is to prove that this problem of predicting finger movement can benefit from multiple kernel learning. To this aim, the fourth dataset from the BCI Competition IV (Miller and Schalk, 2008) was used. The subjects were 3 epileptic patients who had platinium electrode grids placed on the surface of their brain. The number of electrodes varies between 48 to 64 depending on the subject and their position on the cortex was unknown. Electrocorticographic (ECoG) signals of the subject were recorded at a 1KHz sampling using BCI2000 (Schalk et al., 2004). A band-pass filter from 0.15 to 200Hz was applied to the ECoG signals. The finger flexion of the subject was recorded at 25Hz and up-sampled to 1KHz by means of a data glove which measures the finger movement amplitude. Due to the acquisition process, a delay appears between the finger movement and the measured ECoG signal (Miller and Schalk, 2008). One of our hopes is that this time-lag can be properly learnt by means of multiple integral-operator kernels. Features from the ECoG signals are built by computing some band-specific amplitude modulation feature, which is defined as the sum of the square of the band-specific filtered ECoG signals during a time bin of δt.

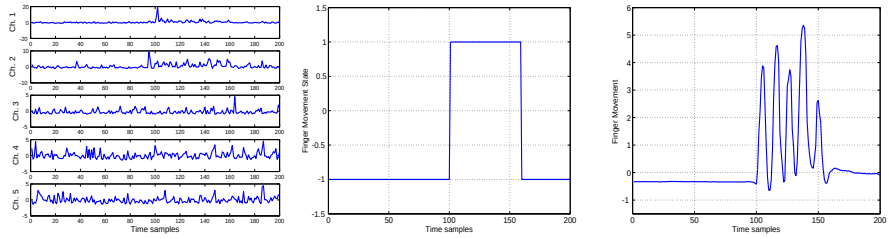

Figure 1: Example of a couple of input-output signals in our BCI task. (left) Amplitude modulation features extracted from ECoG signals over 5 pre-defined channels. (middle) Signal of labels denoting whether the considered finger is moving or not. (right) Real amplitude movement of the finger.

For our finger movement prediction task, we have kept 5 channels that have been manually selected and split the AM feature signals in portions of 200 samples. For each of these time segments, we have the label of whether at each time sample, the finger is moving or not as well as the real movement amplitudes. The dataset is composed of 487 couples of input-output signals, the output signals being either the binary movement labels or the real amplitude movement. An example of input-output signals are depicted in Figure 1. In a nutshell, the problem boils down to be a supervised signal segmentation task.

To evaluate the performance of the multiple operator-valued kernel learning approach, we use both: (1) the percentage number of labels correctly recognized (LCR) defined by $(W_r/T_n) \times 100\%$, where W_r is the number of well recognized labels and T_n the total number of labels to be recognized; (2) the residual sum of squares error (RSSE) as evaluation criterion for curve prediction

$$RSSE = \int \sum_i \{y_i(t) - \widehat{y}_i(t)\}^2 dt \tag{16}$$

where $\widehat{y}_i(t)$ is the prediction of the function $y_i(t)$ corresponding to real finger movement or the finger movement state.

For the multiple operator-valued kernels having the form (13), we have used a Gaussian kernel with 5 different bandwidths and a polynomial kernel of degree 1 to 3 combined with three operators T: identity $Ty(t) = y(t)$, multiplication operator associated with the function e^{-t^2} defined by $Ty(t) = e^{-t^2}y(t)$, and the integral Hilbert-Schmidt operator with the kernel $e^{-|t-s|}$ proposed in Kadri et al. (2011) $Ty(t) = \int e^{-|t-s|}y(s)ds$. The inverses of these operators are computed analytically. While the inverses of the identity and the multiplication operators are easily and directly computable from the analytic expressions of the operators, the inverse of the integral operator is computed from its spectral decomposition as in Kadri et al. (2011). The number of eigenfunctions as well as the regularization parameter λ are fixed using "one-curve-leave-out cross-validation" (Rice and Silverman, 1991) with the aim of minimizing the residual sum of squares error.

Empirical results on the BCI dataset are summarized in Table 1 and Table 2. The dataset was randomly partitioned into 65% training and 35% test sets. We compare our approach in the case of ℓ_1 and ℓ_2-norm constraint on the combination coefficients with: (1) the baseline scalar-valued kernel ridge regression

Table 1: Results for the movement state prediction. Residual Sum of Squares Error (RSSE) and the percentage number of Labels Correctly Recognized (LCR) of : (1) baseline KRR with the Gaussian kernel, (2) functional response KRR with the integral operator-valued kernel, (3) MovKL with ℓ_∞, ℓ_1 and ℓ_2-norm constraint.

Algorithm	**RSSE**	**LCR(%)**
KRR - scalar-valued -	68.32	72.91
KRR - functional response -	49.40	80.20
MovKL - ℓ_∞ norm -	45.44	81.34
MovKL - ℓ_1 norm -	48.12	80.66
MovKL - ℓ_2 norm -	**39.36**	**84.72**

Table 2: Residual Sum of Squares Error (RSSE) results for finger movement prediction.

Algorithm	**RSSE**
KRR - scalar-valued -	88.21
KRR - functional response -	79.86
MovKL - ℓ_∞ norm -	76.52
MovKL - ℓ_1 norm -	78.24
MovKL - ℓ_2 norm -	**75.15**

algorithm by considering each output independently of the others, (2) functional response ridge regression using an operator-valued kernel constructed from the integral operator (Kadri et al., 2011), (3) kernel ridge regression with evenly-weighted sum of operator-valued kernels, which we denote by ℓ_∞-norm MovKL.

As in the scalar case, using multiple operator-valued kernels leads to better results. By directly combining kernels constructed from identity, multiplication and integral operators we could reduce the residual sum of squares error and enhance the label classification accuracy. Best results are obtained using the MovKL algorithm with ℓ_2-norm constraint on the combination coefficients. RSSE and LCR of the baseline kernel ridge regression are significantly outperformed by the operator-valued kernel based functional response regression. These results confirm that by taking into account the relationship between outputs we can improve performance. This is due to the fact that an operator-valued kernel induces a similarity measure between two pairs of input/output.

5 Conclusion

In this paper we have presented a new method for learning simultaneously an operator and a finite linear combination of operator-valued kernels. We have extended the MKL framework to deal with functional response kernel ridge re-

gression and we have proposed a block coordinate descent algorithm to solve the resulting optimization problem. The method is applied on a BCI dataset to predict finger movement in a functional regression setting. Experimental results show that our algorithm achieves good performance outperforming existing methods. It would be interesting for future work to thoroughly compare the proposed MKL method for operator estimation with previous related methods for multi-class and multi-label MKL (Zien and Ong, 2007; Tang et al., 2009), in the contexts of structured output learning (Tsochantaridis et al., 2005) and collaborative filtering (Abernethy et al., 2009).

Appendix

Convergence of Algorithm 1

In this appendix, we present a proof of convergence of Algorithm 1. The proof is an extension to infinite dimensional Hilbert spaces with operator-valued reproducing kernels of results obtained by Argyriou et al. (2008) and more recently by Rakotomamonjy et al. (2011). Let $R(f,d)$ be the objective function of the MovKL problem defined by (7):

$$R(f,d) = L + \sum_{k=1}^{M} \frac{\|f_k\|^2_{\mathcal{F}_k}}{d_k}$$

where $L = \frac{1}{\lambda}\sum_i \|y_i - \sum_{k=1}^M f_k(x_i)\|^2_{\mathcal{G}_y}$. Substituting Equation (11) in R we obtain the objective function:

$$S(f) := R(f, d(f)) = L + \left(\sum_{k=1}^{M} \|f_k\|_{\mathcal{F}_k}^{\frac{2r}{r+1}}\right)^{\frac{r+1}{r}}$$

The function S is **strictly convex** since L is convex and $f \longmapsto \left(\sum_{k=1}^M \|f_k\|^{\frac{2r}{r+1}}\right)^{\frac{r+1}{r}}$ is strictly convex (this follows directly from strict convexity of the function $x \longmapsto x^p$ when $x \geq 0$ and $p > 1$). Thus, $S(f)$ admits a **unique minimizer**.

Now let us define the function g by:

$$g(f) = \min_u \{R(u, d(f))\}.$$

The function g is **continuous**. This comes from the fact that the function:

$$G(d) = \min_u \{R(u,d)\}$$

is continuous. Indeed, G is the minimal of value of a functional response kernel ridge regression problem in a function-valued RKHS associated to an operator-valued kernel K. So, $G(d) = R(d, u^*)$ with $u^* = \big(\mathbf{K}(d) + \lambda I\big)^{-1} y$ (see Equation (9)). u^* is continuous, and hence $G(d)$ is also continuous.

By definition we have $S(f) = R(f, d(f))$, and since $d(f)$ minimizes $R(f, \cdot)$, we obtain that:

$$S(f^{(n+1)}) \leq g(f^{(n)}) \leq S(f^{(n)})$$

where n is the number of iteration. So, the sequence $\{S(f^{(n)}),\ n \in \mathbb{N}\}$ is nonincreasing and then it is bounded since L is bounded from below. Thus, as $n \longrightarrow \infty$, $S(f^{(n)})$ converges to a number which we denote by S^*. $\{S(f^{(n)})\}$ is convergent and S is a coercive function, then the sequence $\{\|f^{(n)}\|,\ n \in \mathbb{N}\}$ is bounded. Consequently, the sequence $\{f^{(n)},\ n \in \mathbb{N}\}$ is **bounded**.

Next we show the following subsequence convergence property which underlies the convergence of Algorithm 1.

Proposition 2 *The sequence $\{f^{(n)},\ n \in \mathbb{N}\}$, since it is bounded, has a convergent subsequence*

Proof. The analogue of the Bolzano-Weierstrass theorem[1] in Hilbert spaces states that there exists a **weakly convergent** subsequence $\{f^{(n_l)},\ l \in \mathbb{N}\}$ of the bounded sequence $\{f^{(n)}\}$. By definition of weakly convergence, we have $\forall g \in \mathcal{F}$ ($\mathcal{F}$ is a RKHS with the operator-valued kernel K):

$$\lim_{n_l \to \infty} \langle f^{(n_l)}(\cdot), g\rangle = \langle f(\cdot), g\rangle$$

Let $g = K(x, \cdot)\beta$. Using the reproducing property

$$\begin{aligned}&\lim_{n_l \to \infty} \langle f^{(n_l)}(\cdot), g\rangle = \lim_{n_l \to \infty} \langle f^{(n_l)}(x), \beta\rangle\\ &\Rightarrow \langle f(\cdot), \beta\rangle = \langle f(x), \beta\rangle\end{aligned}$$

Thus the subsequence $\{f^{(n_l)}\}$ is **pointwise convergent**.

Now we show that:

$$\lim_{n_l \to \infty} \|f^{(n_l)}(\cdot)\| = \|f(\cdot)\| \qquad (*)$$

Since $f^{(n_l)} \in \mathcal{F}$, it can be written as $\sum_i K(x_i, \cdot)\alpha_i^{(n_l)}$, and then we have:

$$\begin{aligned}&\lim_{n_l \to \infty} \|f^{(n_l)}(\cdot)\|^2 = \lim_{n_l \to \infty} \langle f^{(n_l)}(\cdot), \sum_i K(x_i, \cdot)\alpha_i^{(n_l)}\rangle\\ &= \lim_{n_l \to \infty} \sum_i \langle f^{(n_l)}(x_i), \alpha_i^{(n_l)}\rangle = \sum_i \langle f(x_i), \alpha\rangle\\ &= \langle f(\cdot), \sum_i K(x_i, \cdot)\alpha\rangle = \|f\|^2\end{aligned}$$

Using $(*)$ and weak convergence, we obtain the **strong convergence** of the subsequence $\{f^{(n_l)}\}$.

$$\begin{aligned}&\lim_{n_l \to \infty} \|f^{(n_l)} - f\|^2 = \lim_{n_l \to \infty} \langle f^{(n_l)} - f, f^{(n_l)}\rangle - \langle f^{(n_l)} - f, f\rangle\\ &= \lim_{n_l \to \infty} \|f^{(n_l)}\|^2 - \|f\|^2 \quad \text{(using weak convergence)}\\ &= 0 \quad \text{(using}(*)\text{)} \qquad \square\end{aligned}$$

By Proposition 2, there exists a convergent subsequence $\{f^{(n_l)},\ l \in \mathbb{N}\}$ of the bounded sequence $\{f^{(n)},\ n \in \mathbb{N}\}$, whose limit we denote by f^*. Since

[1] The Bolzano-Weierstrass theorem states that each bounded sequence in $\mathbb{R}^n$ has a convergent subsequence. For infinite-dimensional spaces, strong convergence of the subsequence is not reached and only weak convergence is obtained. Proposition 2 shows that strong convergence can be reached in Hilbert spaces with reproducing kernels.

$S(f^{(n+1)}) \leq g(f^{(n)}) \leq S(f^{(n)})$, $g(f^{(n)})$ converges to S^*. Thus, by the continuity of g and S, $g(f^*) = S(f^*)$. This implies that f^* is a minimizer of $R(\cdot, d(f^*))$, because $R(f^*, d(f^*)) = S(f^*)$. Moreover, $d(f^*)$ is the minimizer of $R(\cdot, f^*)$ subject to the constraints on d. Thus, since the objective function R is smooth, the pair $(f^*, d(f^*))$ is the minimizer of R.

At this stage, we have shown that any convergent subsequence of $\{f^{(n)},\ n \in \mathbb{N}\}$ converges to the minimizer of R. Since the sequence $\{f^{(n)},\ n \in \mathbb{N}\}$ is bounded, it follows that the whole sequence **converges** to minimizer of R.

References

J. Abernethy, F. Bach, T. Evgeniou, and J. P. Vert. A new approach to collaborative filtering: Operator estimation with spectral regularization. *Journal of Machine Learning Research*, 10:803–826, 2009.

J. Aflalo, A. Ben-Tal, C. Bhattacharyya, J. Saketha Nath, and S. Raman. Variable sparsity kernel learning. *Journal of Machine Learning Research*, 12: 565–592, 2011.

A. Argyriou, T. Evgeniou, and M. Pontil. Convex multi-task feature learning. *Machine Learning*, 73(3):243–272, 2008.

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

F. Bach, G. Lanckriet, and M. Jordan. Multiple kernel learning, conic duality, and the SMO algorithm. In *Proceedings of the 21st International Conference on Machine Learning*, pages 41–48, 2004.

M. B. Blaschko and C. H. Lampert. Learning to localize objects with structured output regression. In *ECCV*, 2008.

C. Brouard, F. d'Alché-Buc, and M. Szafranski. Semi-supervised penalized output kernel regression for link prediction. In Lise Getoor and Tobias Scheffer, editors, *ICML 2011*, Seattle, USA, 2011. ACM press.

A. Caponnetto, C. A. Micchelli, M. Pontil, and Y. Ying. Universal multi-task kernels. *Journal of Machine Learning Research*, 68:1615–1646, 2008.

C. Carmeli, E. De Vito, and A. Toigo. Vector valued reproducing kernel Hilbert spaces of integrable functions and mercer theorem. *Analysis and Applications*, 4:377–408, 2006.

C. Cortes, M. Mohri, and A. Rostamizadeh. L_2 regularization for learning kernels. In *Proceedings of the 25th Conference in Uncertainty in Artificial Intelligence*, 2009.

C. Cortes, M. Mohri, and A. Rostamizadeh. Generalization bounds for learning kernels. In *Proceedings of the 27th Annual International Conference on Machine Learning*, 2010.

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

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

M. Ha Quang, S. H. Kang, and T. M. Le. Image and video colorization using vector-valued reproducing kernel Hilbert spaces. *Journal of Mathematical Imaging and Vision*, 37(1):49–65, 2010.

M. Hein. Robust nonparametric regression with metric-space valued output. In Y. Bengio, D. Schuurmans, J. Lafferty, C. K. I. Williams, and A. Culotta, editors, *Advances in Neural Information Processing Systems 22*, pages 718–726. 2009.

H. Kadri, E. Duflos, P. Preux, S. Canu, and M. Davy. Nonlinear functional regression: a functional rkhs approach. In *AISTATS 2010, JMLR: W&CP 9*, pages 111–125, Chia Laguna, Sardinia, Italy, 2010.

H. Kadri, A. Rabaoui, P. Preux, E. Duflos, and A. Rakotomamonjy. Functional regularized least squares classification with operator-valued kernels. In *ICML 2011*, Seattle, USA, 2011.

M. Kloft, U. Brefeld, S. Sonnenburg, and A. Zien. ℓ_p-norm multiple kernel learning. *Journal of Machine Learning Research*, 12:953–997, 2011.

G. Lanckriet, N. Cristianini, L. El Ghaoui, P. Bartlett, and M. Jordan. Learning the kernel matrix with semi-definite programming. *Journal of Machine Learning Research*, 5:27–72, 2004.

H. Lian. Nonlinear functional models for functional responses in reproducing kernel Hilbert spaces. *The Canadian Journal of Statistics*, 35:597–606, 2007.

C. Micchelli and M. Pontil. Learning the kernel function via regularization. *Journal of Machine Learning Research*, 6:1099–1125, 2005a.

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

K. J. Miller and G. Schalk. Prediction of finger flexion: 4th brain-computer interface data competition. BCI Competition IV, 2008.

T. Pistohl, T. Ball, A. Schulze-Bonhage, A. Aertsen, and C. Mehring. Prediction of arm movement trajectories from ecog-recordings in humans. *Journal of Neuroscience Methods*, 167(1):105–114, 2008.

A. Rakotomamonjy, F. Bach, Y. Grandvalet, and S. Canu. SimpleMKL. *Journal of Machine Learning Research*, 9:2491–2521, 2008.

A. Rakotomamonjy, R. Flamary, G. Gasso, and S. Canu. l(p)-l(q) penalty for sparse linear and sparse multiple kernel multitask learning. *IEEE Trans. Neural Netw.*, 22(8):1307–20, 2011.

J. O. Ramsay and B. W. Silverman. *Functional Data Analysis, 2nd ed.* Springer Verlag, New York, 2005.

M. Reisert and H. Burkhardt. Learning equivariant functions with matrix valued kernels. *Journal of Machine Learning Research*, 8:385–408, 2007.

John A. Rice and B. W. Silverman. Estimating the Mean and Covariance Structure Nonparametrically When the Data are Curves. *Journal of the Royal Statistical Society. Series B (Methodological)*, 53(1):233–243, 1991.

G. Schalk, D. J. McFarland, T. Hinterberger, N. Birbaumer, and J. R. Wolpaw. Bci2000: a general-purpose brain-computer interface (bci) system. *Biomedical Engineering, IEEE Transactions on*, 51(6):1034–1043, June 2004.

G. Schalk, J. Kubanek, K. J. Miller, N. R. Anderson, E. C. Leuthardt, J. G. Ojemann, D. Limbrick, D. Moran, L. A. Gerhardt, and J. R. Wolpaw. Decoding two-dimensional movement trajectories using electrocorticographic signals in humans. *Journal of Neural Engineering*, 4(3):264–275, 2007.

B. Schölkopf and A. J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, Cambridge, MA, USA, 2002.

L. Schwartz. Sous-espaces hilbertiens d'espaces vectoriels topologiques et noyaux associés (noyaux reproduisants). *Journal d'Analyse Mathématique*, 13:115–256, 1964.

L. Tang, J. Chen, and J. Ye. On multiple kernel learning with multiple labels. In *International Joint Conferences on Artificial Intelligence (IJCAI'09)*, pages 1255–1260, 2009.

P. Tseng. Convergence of block coordinate descent method for nondifferentiable minimization. *J. Optim. Theory Appl.*, 109:475–494, 2001.

P. Tseng and S. Yun. A coordinate gradient descent method for nonsmooth separable minimization. *Mathematical Programming*, 117(1):387–423, 2009.

I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun. Large margin methods for structured and interdependent output variables. *Journal of Machine Learning Research*, 6:1453–1484, 2005.

G. Wahba. Multivariate function and operator estimation, based on smoothing splines and reproducing kernels. In M. Casdagli and S. Eubank, editors, *Nonlinear Modeling and Forecasting*, volume 12 of *Proc. of the Santa Fe Institute*, pages 95–112. Addison Wesley, 1992.

A. Zien and C. S. Ong. Multiclass multiple kernel learning. In *ICML 2007*, pages 1191–1198, 2007.

Centre de recherche INRIA Lille – Nord Europe
Parc Scientifique de la Haute Borne - 40, avenue Halley - 59650 Villeneuve d'Ascq (France)

Centre de recherche INRIA Bordeaux – Sud Ouest : Domaine Universitaire - 351, cours de la Libération - 33405 Talence Cedex
Centre de recherche INRIA Grenoble – Rhône-Alpes : 655, avenue de l'Europe - 38334 Montbonnot Saint-Ismier
Centre de recherche INRIA Nancy – Grand Est : LORIA, Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex
Centre de recherche INRIA Paris – Rocquencourt : Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex
Centre de recherche INRIA Rennes – Bretagne Atlantique : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex
Centre de recherche INRIA Saclay – Île-de-France : Parc Orsay Université - ZAC des Vignes : 4, rue Jacques Monod - 91893 Orsay Cedex
Centre de recherche INRIA Sophia Antipolis – Méditerranée : 2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)
http://www.inria.fr
ISSN 0249-6399