Composite Kernel Learning

Marie Szafranski [1] MARIE.SZAFRANSKI@HDS.UTC.FR
Yves Grandvalet [1,2] YVES.GRANDVALET@IDIAP.CH
Alain Rakotomamonjy [3] ALAIN.RAKOTOMAMONJY@INSA-ROUEN.FR
[1] Heudiasyc, UMR CNRS 6599, Université de Technologie de Compiègne, 60205 Compiègne cedex, France
[2] IDIAP Research Institute, Centre du Parc, P.O. Box 592, 1920 Martigny, Switzerland
[3] LITIS EA 4051, UFR de Sciences, Université de Rouen, 76800 Saint Etienne du Rouvray, France

Abstract

The Support Vector Machine (SVM) is an acknowledged powerful tool for building classifiers, but it lacks flexibility, in the sense that the kernel is chosen prior to learning. Multiple Kernel Learning (MKL) enables to learn the kernel, from an ensemble of basis kernels, whose combination is optimized in the learning process. Here, we propose Composite Kernel Learning to address the situation where distinct components give rise to a group structure among kernels. Our formulation of the learning problem encompasses several setups, putting more or less emphasis on the group structure. We characterize the convexity of the learning problem, and provide a general wrapper algorithm for computing solutions. Finally, we illustrate the behavior of our method on multi-channel data where groups correpond to channels.

1. Motivation

Kernel methods have been extensively used in learning problems (Schölkopf & Smola, 2001). In these models, the observations are implicitly mapped in a feature space via a mapping $\Phi : \mathcal{X} \rightarrow \mathcal{H}$, where $\mathcal{H}$ is a Reproducing Kernel Hilbert Space (RKHS) with reproducing kernel $K : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$.

We address the problem of learning the kernel in Support Vector Machines (SVM) and related methods. Indeed, the kernel is crucial in many respects, and its relevance is essential to the success of kernel methods. Formally, the primary role of K is to define the evaluation functional in $\mathcal{H}$: $\forall f \in \mathcal{H},\ f(\boldsymbol{x}) = \langle f, K(\boldsymbol{x}, \cdot)\rangle_{\mathcal{H}}$, but K also defines (i) $\mathcal{H}$ itself, since $\forall f \in \mathcal{H},\ f(\boldsymbol{x}) = \sum_{i=1}^{\infty} \alpha_i K(\boldsymbol{x}_i, \boldsymbol{x})$; (ii) a metric, and hence a smoothness functional in $\mathcal{H}$: $\|f\|_{\mathcal{H}}^2 = \sum_{i=1}^{\infty}\sum_{j=1}^{\infty} \alpha_i\alpha_j K(\boldsymbol{x}_i, \boldsymbol{x}_j)$; (iii) a distance between observations: $\|\Phi(\boldsymbol{x}) - \Phi(\boldsymbol{x}')\|^2 = K(\boldsymbol{x}, \boldsymbol{x}) + K(\boldsymbol{x}', \boldsymbol{x}') - 2K(\boldsymbol{x}, \boldsymbol{x}')$.

In this paper, we devise Composite Kernel Learning (CKL), a framework where the kernel is learned in a way to favor the selection of variables or groups of variables. Section 2 motivates our approach while briefly reviewing the different means proposed to extend kernel methods beyond the predefined kernel setup. We then follow in Section 3 by considering some recent developments in variable selection that are relevant for our aims. Section 4 describes the CKL framework; the optimization algorithm is provided in Section 5, and experiments are reported in Section 6.

2. Flexible Kernel Methods

From now on, we restrict our discussion to classification, where, from a learning set $S = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$ of pairs of observations and label $(\boldsymbol{x}_i, y_i)$, one aims at building a decision rule that predicts the class label y of any observation $\boldsymbol{x}$. We furthermore focus on the binary case, where $(\boldsymbol{x}_i, y_i) \in \mathcal{X} \times \{\pm 1\}$. However, it should be kept in mind that most of our observations carry on to other settings, such as multiclass classification, clustering or regression with kernel methods.

2.1. Support Vector Machines

A SVM builds the decision rule $\operatorname{sign}(f^\star(\boldsymbol{x}) + b^\star)$, where $f^\star$ and $b^\star$ are defined as the solution of

$$\left\{\begin{array}{lll} \displaystyle\min_{f,b,\boldsymbol{\xi}} & \frac{1}{2}\|f\|_{\mathcal{H}}^2 + C\sum_{i=1}^{n}\xi_i & \\ \text{s. t.} & y_i\big(f(\boldsymbol{x}_i) + b\big) \geq 1 - \xi_i & 1 \leq i \leq n \\ & \xi_i \geq 0 & 1 \leq i \leq n\ . \end{array}\right. \tag{1}$$

The regularization parameter C is the only adjustable parameter in this procedure. This is usually not flexible

Appearing in *Proceedings of the 25^{th} International Conference on Machine Learning*, Helsinki, Finland, 2008.

enough to provide good results when the kernel is chosen prior to seeing data. Hence, most applications of SVM incorporate a mechanism for learning the kernel.

2.2. Learning the Kernel

Cross-validation is the most rudimentary, but also the most common way to learn the kernel. It consists in (i) defining a family of kernels (*e.g.* Gaussian), indexed by one or more parameters (*e.g.* bandwidth), the so-called kernel hyper-parameters, (ii) running the SVM algorithm on each hyper-parameter setting, and (iii) finally choosing the hyper-parameter minimizing a cross-validation score.

A thorough discussion of the pros and cons of cross-validation is out of the scope of this paper, but it is clear that this approach is inherently limited to one or two hyper-parameters and few trial values. This observation led to several proposals allowing for more flexibility.

2.2.1. Filters, Wrappers & Embedded Methods

Learning the kernel amounts to learn the feature mapping. It should thus be of no surprise that the approaches investigated bear some similarities with the ones developed for variable selection, where one encounters filters, wrappers and embedded methods (Guyon & Elisseeff, 2003). Some general frameworks do not belong to a single category but the distinction is appropriate in most cases.

In filter approaches, the kernel is adjusted before building the SVM, with no explicit relationship to the objective value of Problem (1). For example, the kernel target alignment of Cristianini et al. (2002) adapts the kernel to the available data without training any classifier.

In wrapper algorithms, the SVM solver is the inner loop of two nested optimizers, whose outer loop is dedicated to adjust the kernel. This tuning may be guided by various generalization bounds (Cristianini et al., 1999; Weston et al., 2001; Chapelle et al., 2002).

Kernel learning can also be embedded in Problem (1), with the SVM objective value minimized jointly with respect to the SVM parameters and the kernel hyper-parameters (Grandvalet & Canu, 2003). Our approach, which belongs to this family of methods, is based on the Multiple Kernel Learning (MKL) framework (Lanckriet et al., 2004).

2.2.2. Multiple Kernel Learning

MKL is a joint optimization problem of the coefficients of the SVM classifier and a convex combination of kernels that defines the actual SVM kernel

$$K(\boldsymbol{x},\boldsymbol{x}') = \sum_{m=1}^{M} \sigma_m K_m(\boldsymbol{x},\boldsymbol{x}') \ , \tag{2}$$

where each kernel K_m is associated to a RKHS $\mathcal{H}_m$ whose elements will be denoted f_m, and $\{\sigma_m\}_{m=1}^{M}$ are coefficients to be learned under the convex combination constraints

$$\sum_{m=1}^{M} \sigma_m = 1 \ , \ \ \sigma_m \geq 0 \ , 1 \leq m \leq M \ . \tag{3}$$

Bach et al. (2004) proposed the following formulation of MKL [1]:

$$\begin{cases} \displaystyle\min_{\substack{f_1,\ldots,f_M,\\ b,\boldsymbol{\xi}}} \ \frac{1}{2}\Big(\sum_m \|f_m\|_{\mathcal{H}_m}\Big)^2 + C\sum_i \xi_i & \\ \text{s. t.} \ \ y_i\Big(\sum_m f_m(\boldsymbol{x}_i) + b\Big) \geq 1 - \xi_i & 1 \leq i \leq n \\ \qquad \xi_i \geq 0 & 1 \leq i \leq n \,, \end{cases} \tag{4}$$

whose solution leads to a decision rule of the form $\operatorname{sign}\left(\sum_m f_m^\star(\boldsymbol{x}) + b^\star\right)$. This expression of the learning problem is remarkable in that it only deviates slightly from the original SVM problem (1). The squared RKHS norm in $\mathcal{H}$ is simply replaced by a mixed-norm, with the standard RKHS norm within each feature space $\mathcal{H}_m$, and an ℓ_1 norm in $\mathbb{R}^M$ on the vector built by concatenating these norms. This ℓ_1 norm encourages sparse solutions, that is, solutions where some functions f_m have zero norm. In this respect, the MKL problem may be seen as the kernelization of the group-LASSO (Yuan & Lin, 2006).

2.2.3. Composite Kernel Learning

When the individual kernels K_m represent a series, such as Gaussian kernels with different scale parameters, MKL may be used as an alternative to cross-validation. When the input data originates from M differents sources, and that each kernel is affiliated to one input variable, MKL can be used to select relevant input variables.

However, MKL is not meant to address problems where several kernels pertain to one input variable. In this situation, the sparseness mechanism of MKL does not favor solutions discarding all the kernels computed from an irrelevant input. Although most of the related coefficients should vanish in combination (2), spurious correlation may cause irrelevant input variables to participate to the solution.

The flat combination of kernels in MKL does not include a mechanism to cluster the kernels related to one input variable. In order to favor the selection of kernels within predefined groups, one has to define a group structure among kernels, which will guide the selection process through a structured kernel combination. This type of hierarchy among

[1] To lighten notations, the range of indexes is often omitted in summations, in which case: indexes i and j refer to examples and go from 1 to n; index m refers to kernels and goes from 1 to M; index ℓ refers to groups of kernels and goes from 1 to L.

variables has been investigated in linear models (Szafranski et al., 2008; Zhao et al., to appear). We briefly recapitulate the general framework in the following section, before discussing its adaptation to kernel learning in Section 4.

3. Grouped and Hierarchical Selection

The introduction of ℓ_1 penalties, with the seminal paper of Tibshirani (1996) on the LASSO, gave rise to many important theoretical and practical advances in the statistics and machine learning fields. As stated in Section 2.2.2, MKL itself belongs to the series of algorithms affiliated to the LASSO, through its relationship with group-LASSO. In this lineage, Zhao et al. (to appear) defined the very general Composite Absolute Penalties (CAP) family.

3.1. Composite Absolute Penalties

Consider a linear model with M parameters, $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_M)^{\mathrm{t}}$, and let $I = \{1, \ldots, M\}$ be a set of index on these parameters. A group structure on the parameters is defined by a series of L subsets $\{G_\ell\}_{\ell=1}^{L}$, where $G_\ell \subseteq I$. Additionally, let $\{\gamma_\ell\}_{\ell=0}^{L}$ be $L+1$ norm parameters. Then, the member of the CAP family for the chosen groups and norm parameters is

$$\Omega = \sum_{\ell} \Big(\sum_{m \in G_\ell} |\beta_m|^{\gamma_\ell} \Big)^{\gamma_0/\gamma_\ell} . \tag{5}$$

Mixed-norms correspond to groups defined as a partition of the set of variables. A CAP may also rely on nested groups, $G_1 \subset G_2 \subset \ldots \subset G_L$, and $\gamma_0 = 1$, in which case it favors what Zhao et al. call hierarchical selection, that is, the selection of groups of variables in the predefined order $\{I \setminus G_L\}, \{G_L \setminus G_{L-1}\}, \ldots, \{G_2 \setminus G_1\}, G_1$. This example is provided here to stress that Zhao et al.'s notion of hierarchy differs from the one that follows.

3.2. Hierarchical Penalization

Hierarchical penalization uses shrinking coefficients to transform a ridge-like penalty into a sparse penalizer (Szafranski et al., 2008). The model parameterized by $\boldsymbol{\beta}$ is fitted by minimizing a differentiable loss function $J(\cdot)$, subject to a ridge penalty with adaptive coefficients that encourages sparseness among and within groups:

$$\begin{cases} \displaystyle\min_{\boldsymbol{\beta},\boldsymbol{\sigma}_1,\boldsymbol{\sigma}_2} J(\boldsymbol{\beta}) + \lambda \sum_{\ell} \sum_{m \in G_\ell} \frac{\beta_m^2}{\sqrt{\sigma_{1,\ell}\, \sigma_{2,m}}} \\ \text{s. t. } \displaystyle\sum_{\ell} d_\ell\, \sigma_{1,\ell} = 1 \,, \quad \sigma_{1,\ell} \geq 0 \qquad 1 \leq \ell \leq L \\ \qquad \displaystyle\sum_{m} \sigma_{2,m} = 1 \quad , \sigma_{2,m} \geq 0 \quad 1 \leq m \leq M \,. \end{cases} \tag{6}$$

The Lagrange parameter λ controls the amount of shrinkage, and d_ℓ is the size of group ℓ. The constraints expressed on the two last lines encourage sparseness in $\sigma_{1,\ell}$ and $\sigma_{2,m}$, which induces sparseness in β_m.

Here, the groups G_ℓ form a partition of I, and the hierarchy refers to the tree-structure of the shrinking coefficients: $\sigma_{2,m}$ shrinks parameter β_m, while $\sigma_{1,\ell}$ shrinks the parameters for group G_ℓ. In the words of Zhao et al., the objective here is grouped variable selection.

The minimizer of Problem (6) is the minimizer of

$$\min_{\boldsymbol{\beta}} J(\boldsymbol{\beta}) + \lambda \Bigg(\sum_{\ell} d_\ell^{1/4} \Big(\sum_{m \in G_\ell} |\beta_m|^{4/3} \Big)^{3/4} \Bigg)^2 ,$$

which is essentially a CAP estimate, where parameter d_ℓ only accounts for the group sizes (Szafranski et al., 2008). The inner $\ell_{4/3}$ norm and the outer ℓ_1 norm form a mixed-norm penalty that will be denoted $\ell_{(4/3,1)}$. The overall penalizer favors sparse solutions at the group level, with few leading coefficients within the selected groups.

4. From Multiple to Composite Kernels

MKL has been formalized as a quadratically constrained program by Lanckriet et al. (2004), then as a second-order cone program by Bach et al. (2004). More recently, other formulations led to wrapper algorithms, where the optimization with respect to kernel hyper-parameters is still based on the SVM objective value, but is performed in an outer loop that wraps a standard SVM solver. The outer loop is cutting planes for Sonnenburg et al. (2006), and gradient descent for Rakotomamonjy et al. (2007). Wrapper algorithms have appealing features: (i) they benefit from the developments of solvers specifically tailored for the SVM problem in the inner loop; (ii) they allow to address large-scale problems; (iii) they are multipurpose, since the SVM inner loop may be replaced by another algorithm with little or no adjustments.

We chose to build on gradient-based MKL. First, it has been shown to be more efficient than the SILP approach of Sonnenburg et al. (2006), thanks to the stability of the updates performed in the outer loop, which induces good initializations for the inner loop solver (Rakotomamonjy et al., 2007). Second, and even more important for our purpose, gradient-based MKL is amenable to the extension to groups of kernels, thanks to the formulation of hierarchical penalization of Section 3.2.

4.1. Variational Multiple Kernel Learning

Problem (4) is not differentiable at $\|f_m\|_{\mathcal{H}_m} = 0$, a difficulty that causes a considerable algorithmic burden. The MKL formulation of Rakotomamonjy et al. (2007) can be viewed as a variational form of Problem (4), where M new variables $\sigma_1, \ldots, \sigma_M$ are introduced in order to avoid these differentiability issues. The resulting problem, which is equivalent to Problem (4), is stated as:

$$\begin{cases} \min_{\substack{f_1,\dots,f_M,\\ b,\boldsymbol{\xi},\boldsymbol{\sigma}}} \frac{1}{2}\sum_m \frac{1}{\sigma_m}\|f_m\|^2_{\mathcal{H}_m} + C\sum_i \xi_i \\ \text{s. t. } \; y_i\big(\sum_m f_m(\boldsymbol{x}_i)+b\big) \geq 1-\xi_i \quad 1\leq i\leq n \\ \qquad \xi_i \geq 0 \quad 1\leq i\leq n \\ \qquad \sum_m \sigma_m = 1 \quad , \sigma_m \geq 0 \quad 1\leq m\leq M\,. \end{cases} \tag{7}$$

Here and in what follows, u/v is defined by continuation at zero as $u/0=\infty$ if $u\neq 0$ and $0/0=0$.

The constraints expressed on the last line encourage sparseness in σ_m, which induces sparseness in f_m. As already mentioned in Section 2.2.2, the sparseness applies at the kernel level, ignoring the group structure. The latter is taken into account in the formulation proposed in the following section.

4.2. Variational Composite Kernel Learning

Here, we build on the variational form of the composite absolute penalties presented in Section 3.2 to take into account the group structure. Hierarchical penalization can deal with kernel methods if the ridge penalties are replaced by RKHS norms. We first generalize Problem (6) to obtain smooth variational formulations for arbitrary mixed-norm penalties, so that to address a wide variety of problems including MKL:

$$\begin{cases} \min_{\substack{f_1,\dots,f_M,\\ b,\boldsymbol{\xi},\boldsymbol{\sigma}_1,\boldsymbol{\sigma}_2}} \frac{1}{2}\sum_\ell \sigma_{1,\ell}^{-p}\sum_{m\in G_\ell}\sigma_{2,m}^{-q}\|f_m\|^2_{\mathcal{H}_m} + C\sum_i \xi_i \\ \text{s. t. } \; y_i\big(\sum_m f_m(\boldsymbol{x}_i)+b\big) \geq 1-\xi_i \; 1\leq i\leq n \\ \qquad \xi_i \geq 0 \quad 1\leq i\leq n \\ \qquad \sum_\ell d_\ell\,\sigma_{1,\ell} = 1 \quad , \sigma_{1,\ell}\geq 0 \quad 1\leq \ell\leq L \\ \qquad \sum_m \sigma_{2,m} = 1 \quad , \sigma_{2,m} \geq 0 \quad 1\leq m\leq M, \end{cases} \tag{8}$$

where p and q are exponents to be set according to the problem at hand.

This formulation, which is difficult to optimize, is simplified by replacing the two shrinking coefficients $\boldsymbol{\sigma}_1$ and $\boldsymbol{\sigma}_2$ by $\boldsymbol{\sigma}$, defined by $\sigma_m = \sigma_{1,\ell}^p\sigma_{2,m}^q$. In a first step, we consider the change of variable that maps $\boldsymbol{\sigma}_2$ to $\boldsymbol{\sigma}$. When $q\neq 0$, this mapping is one-to-one provided $\sigma_{1,\ell}\neq 0$. Furthermore, if $\sigma^\star_{1,\ell}$ and $\sigma^\star_{2,m}$ denote the optimal $\sigma_{1,\ell}$ and $\sigma_{2,m}$ values for Problem (8), we have that $\sigma^\star_{1,\ell}=0 \Rightarrow \sigma^\star_{2,m}=0$, hence Problem (8) is equivalent to

$$\begin{cases} \min_{\substack{f_1,\dots,f_M,\\ b,\boldsymbol{\xi},\boldsymbol{\sigma}_1,\boldsymbol{\sigma}}} \frac{1}{2}\sum_m \frac{1}{\sigma_m}\|f_m\|^2_{\mathcal{H}_m} + C\sum_i \xi_i \\ \text{s. t. } \; y_i\big(\sum_m f_m(\boldsymbol{x}_i)+b\big) \geq 1-\xi_i \; 1\leq i\leq n \\ \qquad \xi_i \geq 0 \quad 1\leq i\leq n \\ \qquad \sum_\ell d_\ell\,\sigma_{1,\ell} = 1 \quad , \sigma_{1,\ell}\geq 0 \quad 1\leq \ell\leq L \\ \qquad \sum_\ell \sigma_{1,\ell}^{-p/q}\sum_{m\in G_\ell}\sigma_m^{1/q} \leq 1 \\ \qquad \sigma_m \geq 0 \quad 1\leq m\leq M\,. \end{cases} \tag{9}$$

The new problem is simplified further by showing that $\boldsymbol{\sigma}_1$ can be dropped out from the optimization process, leading to the following formulation of Composite Kernel Learning (CKL):

$$\begin{cases} \min_{\substack{f_1,\dots,f_M,\\ b,\boldsymbol{\xi},\boldsymbol{\sigma}}} \frac{1}{2}\sum_m \frac{1}{\sigma_m}\|f_m\|^2_{\mathcal{H}_m} + C\sum_i \xi_i \\ \text{s. t. } \; y_i\big(\sum_m f_m(\boldsymbol{x}_i)+b\big) \geq 1-\xi_i \; 1\leq i\leq n \\ \qquad \xi_i \geq 0 \quad 1\leq i\leq n \\ \qquad \sum_\ell \left(d_\ell^p\Big(\sum_{m\in G_\ell}\sigma_m^{1/q}\Big)^q\right)^{1/(p+q)} \leq 1 \\ \qquad \sigma_m \geq 0 \quad 1\leq m\leq M\,, \end{cases} \tag{10}$$

Before considering particular settings of interest, we state below two helpful propositions. The first one gives a more interpretable formulation of Problem (10); the second one presents the conditions for convexity of formulation (10), that will guaranty the convergence towards the global minimum for the algorithm described in Section 5.

Proposition 1. ***CAP Formulation:*** *Problem (10) is equivalent to the following MKL problem with a CAP-like penalty on the RKHS norms:*

$$\begin{cases} \min_{\substack{f_1,\dots,f_M,\\ b,\boldsymbol{\xi}}} \frac{1}{2}\Big(\sum_\ell d_\ell^{\gamma^*}\big(\sum_{m\in G_\ell}\|f_m\|^\gamma_{\mathcal{H}_m}\big)^{\gamma_0/\gamma}\Big)^{2/\gamma_0} + C\sum_i \xi_i \\ \text{s. t. } \; y_i\big(\sum_m f_m(\boldsymbol{x}_i)+b\big) \geq 1-\xi_i \quad 1\leq i\leq n \\ \qquad \xi_i \geq 0 \quad 1\leq i\leq n, \end{cases} \tag{11}$$

with $\gamma=\frac{2}{q+1}$, $\gamma_0=\frac{2}{p+q+1}$ *and* $\gamma^*=1-\frac{\gamma_0}{\gamma}$.

Sketch of proof. Let $\mathcal{L}$ be the Lagrangian of problem (10). The optimality conditions for σ_m are obtained from the first order optimality conditions for σ_m ($\frac{\partial\mathcal{L}}{\partial\sigma_m}=0$):

$$\sigma_m = \Big(\sum_\ell d_\ell^{\gamma^*}s_\ell^{\gamma_0/\gamma}\Big)^{(\gamma_0-2)/\gamma_0} d_\ell^{-\gamma^*}s_\ell^{\gamma^*}\|f_m\|^{2-\gamma}_{\mathcal{H}_m}\,, \tag{12}$$

where $s_\ell = \sum_{m\in G_\ell}\|f_m\|^\gamma_{\mathcal{H}_m}$. Plugging this expression in Problem (10) yields the claimed result. $\square$

Note that the outer exponent $\frac{2}{\gamma_0}$ only influences the strength of the penalty, not its type. Hence, the penalty in the objective function (11) differs from (5) in the RKHS norms $\|\cdot\|_{\mathcal{H}_m}$ and in the parameters d_ℓ that accommodate for group sizes.

Proposition 2. ***Conditions for Convexity:*** *Problem (10) is convex if and only if* $0 \leq q \leq 1$ *and* $0 \leq p+q \leq 1$.

Proof. A problem minimizing a convex criterion on a convex set is convex. The objective function of Problem (10) is convex (Boyd & Vandenberghe, 2004, p. 89). The first, second and fourth constraints define convex sets, and the third one also provided (i) $\left(\sum_{m \in G_\ell} \sigma_m^{1/q}\right)^q$ is a norm, that is $0 \leq q \leq 1$, and (ii) $\sum_\ell t_\ell^{1/(p+q)}$ is convex in t_ℓ, that is $0 \leq p+q \leq 1$. □

Within the values of p and q ensuring convexity, we pick the following particular cases of interest:

- $p = 0$, $q = 1$ yields a LASSO type penalty on the RKHS norms. It results in the generalization of the group-LASSO known as MKL, as formulated in (4);
- $p = 1$, $q = 0$ yields a group-LASSO type penalty on the RKHS norms. It results in another MKL, with L effective kernels $\overline{K}_\ell$, defined as $\overline{K}_\ell = \sum\limits_{m \in G_\ell} K_m$;
- $p = q = \frac{1}{2}$ yields a hierarchical-penalization type penalty on the RKHS norms. It is a true CKL, where there are M effective kernels, and where the penalty favors sparse solutions at the group level, with few leading kernels within the selected groups.

Hence, when p goes from zero to one, with $q = 1-p$, the penalty gives more and more emphasis to the group structure. For most applications where convexity is a key issue, we recommend the balanced setup $p = q = \frac{1}{2}$.

Note however that convex penalties restrict the sparseness of the solution to either the group level or the kernel level. In Section 6, we will illustrate that giving up convexity may turn out to be an interesting option when considering interpretability issues.

5. Algorithm

Our approach to solve Problem (10) draws on the MKL algorithm of Rakotomamonjy et al. (2007). We use the wrapper scheme described below, where the outer loop is carried out by a projected gradient descent update.

5.1. A Gradient-Based Wrapper

The wrapper scheme considers the following constrained optimization problem:

$$\begin{cases} \min\limits_{\boldsymbol{\sigma}} \ J(\boldsymbol{\sigma}) & \\ \text{s. t.} \sum\limits_\ell \left(d_\ell^p \left(\sum\limits_m \sigma_m^{1/q} \right)^q \right)^{1/(p+q)} & \leq 1 \\ \quad\ \sigma_m \geq 0, & 1 \leq m \leq M \ , \end{cases}$$

where $J(\boldsymbol{\sigma})$ is defined as the objective value of

$$\begin{cases} \min\limits_{\substack{f_1,\ldots,f_M,\\ b,\boldsymbol{\xi}}} \frac{1}{2} \sum\limits_\ell \sum\limits_{m \in G_\ell} \frac{1}{\sigma_m} \|f_m\|^2_{\mathcal{H}_m} + C \sum\limits_i \xi_i \\ \text{s. t.} \quad y_i \big(\sum\limits_m f_m(\boldsymbol{x}_i) + b \big) \geq 1 - \xi_i \ , \ 1 \leq i \leq n \\ \qquad\quad \xi_i \geq 0 \ , \ 1 \leq i \leq n \ . \end{cases} \tag{13}$$

The global optimization problem consists thus of two nested problems. In the inner loop, the criterion is optimized with respect to $f_1, \ldots, f_M$, b and $\boldsymbol{\xi}$, considering that the coefficients $\boldsymbol{\sigma}$ are fixed. In the outer loop, $\boldsymbol{\sigma}$ is updated to decrease the criterion, with f_m, b and $\boldsymbol{\xi}$ being fixed.

Equation (12) may be used to update $\boldsymbol{\sigma}$ in closed form. However, this approach lacks convergence guarantees and may lead to numerical problems, in particular when some elements of $\boldsymbol{\sigma}$ approach zero. Hence, following Rakotomamonjy et al. (2007), we use that the objective function $J(\boldsymbol{\sigma})$ is actually an optimal SVM objective value to update $\boldsymbol{\sigma}$ by an efficient projected gradient descent scheme.

5.2. Computing the Gradient

The dual formulation offers a convenient means to compute the gradient $\nabla J(\boldsymbol{\sigma})$. The derivation of the Lagrangian of Problem (13), which is omitted here for brevity, shows that its dual formulation is identical to the one of a standard SVM using the aggregated kernel $\overline{K}_{\boldsymbol{\sigma}}$ defined in Equation (2). Hence, the dual problem takes the usual form

$$\begin{cases} \max\limits_{\alpha} - \frac{1}{2} \sum\limits_{i,j} \alpha_i \alpha_j y_i y_j \overline{K}_{\boldsymbol{\sigma}}(\boldsymbol{x}_i, \boldsymbol{x}_j) + \sum\limits_i \alpha_i \\ \text{s. t.} \sum\limits_i \alpha_i y_i = 0 \\ \quad\ C \geq \alpha_i \geq 0 \qquad\qquad 1 \leq i \leq n \ , \end{cases} \tag{14}$$

which can be solved by any SVM solver.

As $J(\boldsymbol{\sigma})$ is defined as the optimal objective value of the convex Problem (13) for which strong duality applies, $J(\boldsymbol{\sigma})$ is also the dual objective value:

$$J(\boldsymbol{\sigma}) = -\frac{1}{2} \sum_{i,j} \alpha_i^\star \alpha_j^\star y_i y_j \overline{K}_{\boldsymbol{\sigma}}(\boldsymbol{x}_i, \boldsymbol{x}_j) + \sum_i \alpha_i^\star \ , \tag{15}$$

where $\alpha^\star$ solves Problem (14).

The existence and computation of the derivatives of $J(\cdot)$ follow from general results on optimal values, such as Theorem 4.1 of Bonnans and Shapiro (1998), which, in a nutshell states that the differentiability of $J(\boldsymbol{\sigma})$ is ensured by the unicity of $\alpha^\star$, and by the differentiability of (15). [2] Furthermore, the derivatives of $J(\boldsymbol{\sigma})$ can be computed as if $\alpha^\star$ were not to depend on $\boldsymbol{\sigma}$. Thus, the gradient $\nabla J(\boldsymbol{\sigma})$ is simply

$$\frac{\partial J}{\partial \sigma_m} = -\frac{1}{2} \sum_{i,j} \alpha_i^\star \alpha_j^\star y_i y_j \sum_{\ell} \sum_{m \in G_\ell} K_m(\boldsymbol{x}_i, \boldsymbol{x}_j) \ .$$

5.3. CKL Algorithm

Now, we have all the ingredients to adapt the machinery developed for MKL by Rakotomamonjy et al. (2007). According to the process described in Section 5.1, we propose Algorithm 1.

Algorithm 1 Composite Kernel Learning

initialize $\boldsymbol{\sigma}$
solve the SVM problem $\rightarrow J(\boldsymbol{\sigma})$
repeat
  compute direction $d = -\nabla J(\boldsymbol{\sigma})$
  repeat
    compute d', the projection of d onto the tangent of the surface of the admissible set
    compute the smallest step that nullifies a component of $\boldsymbol{\sigma}$
      $\mathcal{S} = \{ j : d'_j < 0 \text{ and } \sigma_j \neq 0 \}$
      $\nu = \min_{j \in \mathcal{S}} -\frac{\sigma_j}{d'_j} \quad k = \arg\min_{j \in \mathcal{S}} -\frac{\sigma_j}{d'_j} \quad d_k = 0$
    $\boldsymbol{\sigma}^\dagger = \boldsymbol{\sigma} + \nu\, d'$
    project $\boldsymbol{\sigma}^\dagger$ onto the surface of the admissible set
    solve the SVM problem $\rightarrow J(\boldsymbol{\sigma}^\dagger)$
    if $J(\boldsymbol{\sigma}^\dagger) < J(\boldsymbol{\sigma})$ **then** $\boldsymbol{\sigma} = \boldsymbol{\sigma}^\dagger$
  until $J(\boldsymbol{\sigma}^\dagger) \geq J(\boldsymbol{\sigma})$
  compute $\nu^\star = \arg\min_\nu J(\boldsymbol{\sigma} + \nu\, d)$
  $\boldsymbol{\sigma} = \boldsymbol{\sigma} + \nu^\star d$
until convergence

The stopping criterion for assessing the convergence of the outer loop can be based on standard criteria for gradient-based algorithms or on the duality gap. In the following experiments, it is based on the stability of $\boldsymbol{\sigma}$ and $J(\boldsymbol{\sigma})$.

6. Channel Selection for BCI

This experiment deals with single trial classification of EEG signals coming from Brain-Computer Interface (BCI). Depending on each BCI paradigm, these EEG signals are recorded from specific electrode positions. However, as stated by Schröder et al. (2005), automated channel selection should be performed for each single subject since it leads to better performances or a substantial reduction of the number of useful channels. Reducing the number of channels involved in the decision function is of primary importance for BCI real-life applications, since it makes the acquisition system easier to use and to set-up.

We use here the dataset from the BCI 2003 competition for the task of interfacing the P300 Speller (Blankertz et al., 2004). The dataset consists in 7560 EEG signals paired with positive or negative stimuli responses. The signal, processed as in (Rakotomamonjy et al., 2005), leads to 7560 examples of dimension 896 (14 time frames for each of the 64 channels).

The experimental protocol is then the following: we have randomly picked 567 training examples from the datasets and used the remaining as testing examples. For each parameter, C has been selected by retaining a small part of the training set as a validation set, for selecting the parameter which the highest AUC. This overall procedure has been repeated 10 times. Using a small part of the examples for training can be justified by the use of ensemble of SVMs (that we do not consider here) on a latter stage of the EEG classification procedure (Rakotomamonjy et al., 2005), and the AUC performance measure is justified by how the EEG recognition is transformed into selected character in the P300.

The 896 features extracted from the EEG signals are not tranformed before classification: we do not use any kernelization. However, to unify the presentation, we will refer to these features as linear kernels. Hence, in this application where the kernels related to a given channel form a group of kernels, we have to learn $M = 896$ coefficients σ_m, divided into $L = 64$ groups.

CKL is well-suited to the classification objectives, since we aim at classifying the EEG trials with as few channels as possible. Furthermore, it is also likely that some time frames are irrelevant, so that variable selection may be carried out within each channel. To reach a sparse solution at the channel and the time frame levels, we test a non-convex parametrization of CKL that encourages sparseness within and between groups.

In the following, $\text{CKL}_{1/2}$ stands for a convex version of our algorithm, with $p = q = 1/2$ (a $\ell_{(4/3,1)}$ mixed-norm), CKL_1 is a non-convex version, with $p = q = 1$ (a $\ell_{(1,2/3)}$ (pseudo) mixed-norm). Note that MKL is also implemented by our algorithm, with $p = 0$ and $q = 1$.

Table 1 summarizes the average performance of SVM, MKL, and CKL, that is, for 4 different penalization terms: quadratic penalization for the classical SVM (which is

[2] The unicity of $\alpha^\star$ is ensured provided that the Gram matrix built from kernel $\overline{K}_{\boldsymbol{\sigma}}$ is positive-definite. To enforce this property, a small ridge may be added to the diagonal.

trained with the mean of 896 kernels), ℓ_1 norm for MKL, and mixed-norms for the two versions of CKL: $\text{CKL}_{1/2}$ and CKL_1. The number of channels and kernels selected by these algorithms is also reported.

Table 1. Average Results for SVMs with 4 different penalization terms on the BCI datasets.

Algorithms	AUC	# Channels	# Kernels
SVM	83.87 ± 0.8	64	896
MKL	85.43 ± 0.9	62.2 ± 1	255.8 ± 15
$\text{CKL}_{1/2}$	85.49 ± 1.1	62.9 ± 1	835.7 ± 25
CKL_1	84.15 ± 0.8	24.0 ± 4	60.9 ± 10

The prediction performances of the 4 algorithms are similar, with a slight advantage for sparse methods. $\text{CKL}_{1/2}$ is much less sparse than MKL, which itself keeps about four times as much kernels compared to CKL_1. In the number of groups, MKL and $\text{CKL}_{1/2}$ behave similarly, with only one or two channels removed. CKL_1 is much sparser and removes about two thirds of the channels.

Figure 6 represents the median relevance of the electrodes over the 10 experiments. It displays which electrodes have been selected by the different kernel learning methods. For one experiment, the relevance for channel ℓ is computed by the relative contribution of group ℓ to the norm of the solution, that is

$$\frac{1}{Z} \sum_{m \in G_\ell} \frac{1}{\sigma_m^\star} \|f_m^\star\|_{\mathcal{H}_m}^2 \ ,$$

where Z is a normalization factor that sets the sum of relevances to one.

The results for CKL_1 are particularly neat, with high relevances for the electrodes in the areas of the visual cortex (especially the lateral electrodes PO_7 and PO_8), and the primary motor and Somatosensory cortex ($\text{C}_\bullet$ and CP_Z). The scalp maps for MKL and $\text{CKL}_{1/2}$ are very similar and show the importance of the same regions. In addition they also highlight numerous frontal electrodes that are not likely to be relevant for the BCI P300 Speller paradigm.

7. Conclusion and Further Works

This paper is at the crossroad of kernel learning and variable selection. From the former viewpoint, we extended the multiple kernel learning problem to take into account the group structure among kernels. From the latter viewpoint, we generalized the hierarchical penalization framework to kernel classifiers by considering penalties in RKHS instead of parametric function spaces.

As a side contribution, we also provide a smooth variational formulation for arbitrary mixed-norm penalties, enabling to tackle a wide variety of problems. This formulation is not restricted to convex mixed-norm, a property that turns out to be of interest for reaching sparser, hence more interpretable solutions.

Our approach is embedded, in the sense that the kernel hyper-parameters are optimized jointly with the parameters of classifier to minimize the soft-margin criterion. It is however implemented by a simple wrapper algorithm, for which the inner and the outer subproblems have the same objective function, and where the inner loop is a standard SVM problem.

In particular, this implementation allows to use available solvers for kernel machines in the inner loop. Hence, although this paper considered binary classification problems, our approach can be readily extended to other learning problems, such as multiclass classification, clustering, regression or ranking.

Acknowledgments

This work was supported in part by the IST Program of the European Community, under the PASCAL Network of Excellence, IST-2002-506778. The authors would like to thank N. Bourdaud and G. Garipelli for their help in the interpretation of the BCI experiments.

References

Bach, F. R., Lanckriet, G. R. G., & Jordan, M. I. (2004). Multiple kernel learning, conic duality, and the SMO algorithm. *Proceedings of the twenty-first international conference on Machine Learning* (pp. 41–48).

Blankertz, B., Müller, K.-R., Curio, G., Vaughan, T. M., Schalk, G., Wolpaw, J. R., Schlögl, A., Neuper, C., Pfurtscheller, G., Hinterberger, T., Schröder, M., & Birbaumer, N. (2004). The BCI competition 2003: progress and perspectives in detection and discrimination of EEG single trials. *IEEE Trans. Biomed. Eng*, *51*, 1044–1051.

Bonnans, J., & Shapiro, A. (1998). Optimization problems with pertubation: A guided tour. *SIAM Review*, *40*, 228–264.

Boyd, S., & Vandenberghe, L. (2004). *Convex optimization*. Cambridge University Press.

Chapelle, O., Vapnik, V., Bousquet, O., & Mukherjee, S. (2002). Choosing multiple parameters for support vector machines. *Machine Learning*, *46*, 131–159.

Cristianini, N., Campbell, C., & Shawe-Taylor, J. (1999). Dynamically adapting kernels in support vector machines. *Advances in Neural Information Processing Systems 11* (pp. 204–210). MIT Press.

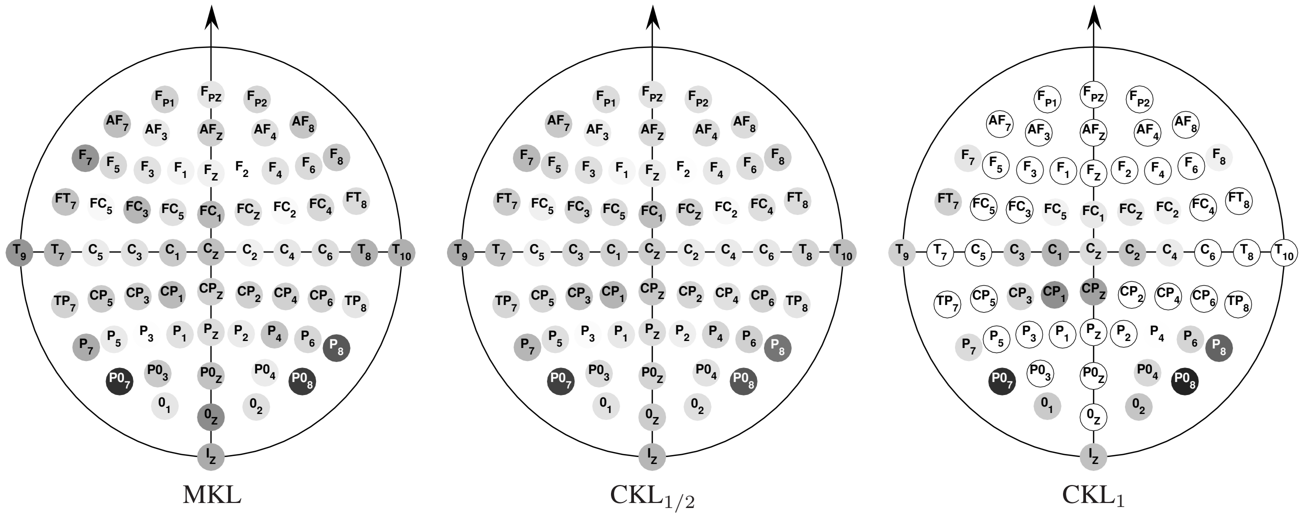

Figure 1. Electrode median relevance for MKL (left), $\text{CKL}_{1/2}$ (center) and CKL_1 (right). The darker the color, the higher the relevance. Electrodes in white with a black circle are discarded (the relevance is exactly zero). The arrow represents the frontal direction.

Cristianini, N., Shawe-Taylor, J., Elisseeff, A., & Kandola, K. (2002). On kernel-target alignment. *Advances in Neural Information Processing Systems 14* (pp. 367–373). MIT Press.

Grandvalet, Y., & Canu, S. (2003). Adaptive scaling for feature selection in SVMs. *Advances in Neural Information Processing Systems 15* (pp. 569–576). MIT Press.

Guyon, I., & Elisseeff, A. (2003). An introduction to variable and feature selection. *Journal of Machine Learning Research*, *3*, 1157–1182.

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

Rakotomamonjy, A., Bach, F., Canu, S., & Grandvalet, Y. (2007). More efficiency in multiple kernel learning. *Proceedings of the 24th International Conference on Machine Learning (ICML 2007)* (pp. 775–782). Omnipress.

Rakotomamonjy, A., Guigue, V., Mallet, G., & Alvarado, V. (2005). Ensemble of SVMs for improving brain-computer interface P300 speller performances. *15th International Conference on Artificial Neural Networks* (pp. 45–50). Springer.

Schölkopf, B., & Smola, A. J. (2001). *Learning with kernels: Support vector machines, regularization, optimization, and beyond*. MIT Press.

Schröder, M., Lal, T. N., Hinterberger, T., Bogdan, M., Hill, J., Birbaumer, N., Rosenstiel, W., & Schölkopf, B. (2005). Robust EEG channel selection across subjects for brain computer interfaces. *EURASIP Journal on Applied Signal Processing*, *19*, 3103–3112.

Sonnenburg, S., Rätsch, G., Schäfer, C., & Schölkopf, B. (2006). Large scale multiple kernel learning. *Journal of Machine Learning Research*, *7*, 1531–1565.

Szafranski, M., Grandvalet, Y., & Morizet-Mahoudeaux, P. (2008). Hierarchical penalization. In J. Platt, D. Koller, Y. Singer and S. Roweis (Eds.), *Advances in neural information processing systems 20*. MIT Press.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B*, *58*, 267–288.

Weston, J., Mukherjee, S., Chapelle, O., Pontil, M., Poggio, T., & Vapnik, V. (2001). Feature selection for SVMs. *Advances in Neural Information Processing Systems 13* (pp. 668–674). MIT Press.

Yuan, M., & Lin, Y. (2006). Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society. Series B*, *68*, 49–67.

Zhao, P., Rocha, G., & Yu, B. (to appear). The composite absolute penalties family for grouped and hierarchical variable selection. *Annals of Statistics*.