Learning with Groups of Kernels

Marie Szafranski[1], Yves Grandvalet[1,2], Alain Rakotomamonjy[3]

[1] Heudiasyc, UMR CNRS 6599
Université de Technologie de Compiègne
BP 20529, 60205 Compiègne Cedex, France
marie.szafranski@hds.utc.fr

[2] IDIAP
Centre du Parc, Av. des Prés-Beudin 20
Case Postale 592, CH-1920 Martigny, Switzerland
yves.grandvalet@idiap.ch

[3] Laboratoire I.T.I.S, EA 4108
Université de Rouen
Avenue de l'Université
76801 St Etienne du Rouvray Cedex, France
alain.rakotomamonjy@insa-rouen.fr

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 build on MKL to address the situations where there is a group structure among kernels that is believed to be relevant for the classification task. We develop the theoretical and the algorithmic aspects of learning with groups of kernels. Our formulation of the learning problem encompasses several setups, including MKL, where more or less emphasis is given to the group structure. We characterize the convexity of the learning problem, and provide a general wrapper algorithm for computing solutions. Finally, some experiments illustrate the behavior of several instances of our method.

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 $\Phi : \mathcal{X} \to \mathcal{H}$, where $\mathcal{H}$ is a Reproducing Kernel Hilbert Space (RKHS) with reproducing kernel $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$.

We address here the problem of learning the kernel in Support Vector Machines (SVM) and related methods. Indeed, the kernel is crucial in many respects, and its appropriate choice 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 it should be kept in mind that K also defines

- $\mathcal{H}$ itself, since $\forall f \in \mathcal{H}$, $f(\boldsymbol{x}) = \sum_{i=1}^{\infty} \alpha_i K(\boldsymbol{x}_i, \boldsymbol{x})$;
- a metric, and hence a smoothness functional in $\mathcal{H}$: $||f||^2_{\mathcal{H}} = \sum_{i=1}^{\infty}\sum_{j=1}^{\infty} \alpha_i\alpha_i K(\boldsymbol{x}_i, \boldsymbol{x}_j)$;
- a similarity between pairs of observations, via the mapping Φ: $K(\boldsymbol{x}, \boldsymbol{x}') = \langle\Phi(\boldsymbol{x}), \Phi(\boldsymbol{x}')\rangle_{\mathcal{H}}$.

In this paper, we devise a framework where kernels are learned in a way to favor the selection of variables, or groups of variables. Section 2 motivates our approach while briefly reviewing the different advances in extending 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 our framework. The associated algorithm is detailed in Section 5, and is tested in Section 6. We then conclude the paper in Section 7, which describes possible extensions left for future work.

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

SVM build the decision rule $\operatorname{sign}(f^\star(\boldsymbol{x}) + b^\star)$, where the function $f^\star$ and the offset $b^\star$ are defined as the solution of [1]

$$\begin{cases} \min\limits_{f,b,\boldsymbol{\xi}} & \frac{1}{2}||f||_{\mathcal{H}}^2 + C\sum\limits_i \xi_i \\ \text{s. t.} & y_i(f(\boldsymbol{x}_i) + b) \geq 1 - \xi_i \ , \qquad \xi_i \geq 0 \ , \qquad 1 \leq i \leq n \ . \end{cases} \tag{1}$$

The regularization parameter C is the only adjustable parameter in this procedure. This is usually not flexible 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, then (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 (Ong *et al.*, 2005), 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).

[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.

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 inspired by the Multiple Kernel Learning (MKL) framework originally developed by 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, defining the new SVM kernel

$$K(\boldsymbol{x}, \boldsymbol{x}') = \sum_m \sigma_m K_m(\boldsymbol{x}, \boldsymbol{x}') \ , \tag{2}$$

where $\sigma_1, \ldots, \sigma_M$ are coefficients to be learned under the convex combination constraints

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

Bach *et al.* (2004) proposed an interesting formulation of the MKL problem:

$$\begin{cases} \min\limits_{\substack{f_1,\ldots,f_M,\\ b,\boldsymbol{\xi}}} & \dfrac{1}{2}\left(\sum\limits_m \|f_m\|_{\mathcal{H}_m}\right)^2 + C\sum\limits_i \xi_i \\ \text{s. t.} & y_i\left(\sum\limits_m f_m(\boldsymbol{x}_i) + b\right) \geq 1 - \xi_i \ , \quad \xi_i \geq 0 \ , \quad 1 \leq i \leq n \ , \end{cases} \tag{4}$$

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

MKL may be used in different prospects. 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, each kernel may be affiliated to one input variable, and the goal may be to select relevant input variables.

However, MKL should not be expected to provide a fully satisfactory answer when 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. Hence, although most of the related coefficients should vanish in combination (2), spurious correlation may cause all 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 direct the selection of kernels towards predefined groups, one has to define a group structure among kernels, which will guide the selection process through the organization of the kernel combination. This type of hierarchy among variables has been investigated for subset selection methods in linear models (Szafranski *et al.*, 2008; Zhao *et al.*, to appear). We briefly recapitulate the general framework in the following section, before going into more technical details and 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, whose definition is given below.

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

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

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.* (to appear) 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 this notion of hierarchy differs from the one used by Szafranski *et al.* (2008), which is recalled below.

3.2 Hierarchical Penalization

Hierarchical penalization was devised for the same type of model than the ones for CAP. The model parameterized by $\boldsymbol{\beta}$ is fitted by minimizing a differentiable loss function

$J(\cdot)$, subject to sparseness constraints among and within groups:

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

where λ is a Lagrange parameter that controls the amount of shrinkage, and d_ℓ is the size of group ℓ. Here, the groups partition the set of variables, 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.* (to appear), the objective here is grouped variable selection.

One can show that the minimizer of Problem (6) is the minimizer of

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

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

4 Putting Things Together

The MKL problem 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 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: they benefit from the developments of solvers specifically tailored for the SVM problem in the inner loop; they allow to address large-scale problems; they are multipurpose, since the SVM inner loop may be replaced by another algorithm with little or no adjustments.

We chose to build on the 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, the gradient-based MKL relies on a formulation that is amenable to the extension to groups of kernels, thanks to the smooth formulation of hierarchical penalization (6).

4.1 Gradient-Based Multiple Kernel Learning

Problem (4) is not differentiable at $||f_m||_{\mathcal{H}_m} = 0$, a difficulty that involves a considerable algorithmic burden. The MKL formulation of Rakotomamonjy *et al.* (2007) can be considered 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: [2]

$$\left\{\begin{array}{lll} \min\limits_{\substack{f_1,\ldots,f_M,\\ b,\boldsymbol{\xi},\boldsymbol{\sigma}}} & \frac{1}{2}\sum\limits_m \frac{1}{\sigma_m}||f_m||^2_{\mathcal{H}_m} + C\sum\limits_i \xi_i & \\ \text{s. t.} & y_i\left(\sum\limits_m f_m(\boldsymbol{x}_i) + b\right) \geq 1-\xi_i \ , \quad \xi_i \geq 0 \ , & 1 \leq i \leq n \\ & \sum\limits_m \sigma_m = 1 \ , \quad \sigma_m \geq 0 \ , & 1 \leq m \leq M \ . \end{array}\right. \tag{7}$$

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 Learning with Groups of Kernels

We now generalize hierarchical penalization to formulate a MKL problem, taking into account the group structure. We build on hierarchical penalization by addressing kernel methods that consider penalties in RKHS instead of parametric function spaces. We furthermore provide a smooth variational formulation for arbitrary mixed-norm penalties $\ell(p,q)$, enabling to consider a wide variety of problems subsuming MKL. Our formulation of the Group Kernel Learning (GKL) is as follows:

$$\left\{\begin{array}{lll} \min\limits_{\substack{f_1,\ldots,f_M,\\ b,\boldsymbol{\xi},\boldsymbol{\sigma}_1,\boldsymbol{\sigma}_2}} & \frac{1}{2}\sum\limits_\ell \sigma_{1,\ell}^{-p} \sum\limits_{m\in G_\ell} \sigma_{2,m}^{-q}||f_m||^2_{\mathcal{H}_m} + & C\sum\limits_i \xi_i \\ \text{s. t.} & y_i\left(\sum\limits_m f_m(\boldsymbol{x}_i) + b\right) \geq 1-\xi_i \ , \quad \xi_i \geq 0 \ , & 1 \leq i \leq n \\ & \sum\limits_\ell d_\ell\, \sigma_{1,\ell} = 1 \ , \quad \sigma_{1,\ell} \geq 0 \ , & 1 \leq \ell \leq L \\ & \sum\limits_m \sigma_{2,m} = 1 \ , \quad \sigma_{2,m} \geq 0 \ , & 1 \leq m \leq M, \end{array}\right. \tag{8}$$

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

Before considering particular settings of interest, we state below three helpful propositions. The first one gives a more interpretable formulation of Problem (8); the second

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

one presents necessary conditions for convexity, based on the latter formulation; finally, the third one provides sufficient conditions for the convexity of formulation (8), that will guaranty the convergence towards the global minimum for the algorithm described in Section 5.

Proposition 1
CAP Formulation: *Problem (8) is equivalent to the following MKL problem with a CAP-like penalty on the RKHS norms:*

$$\begin{cases} \min\limits_{\substack{f_1,\dots,f_M,\\ b,\boldsymbol{\xi}}} & \frac{1}{2}\left(\sum\limits_{\ell} d_\ell^{\gamma^*}\left(\sum\limits_{m\in G_\ell} ||f_m||_{\mathcal{H}_m}^{\gamma}\right)^{\frac{\gamma_0}{\gamma}}\right)^{\frac{2}{\gamma_0}} + \; C\sum\limits_i \xi_i \\ \text{s. t.} & y_i\left(\sum\limits_m f_m(\boldsymbol{x}_i)+b\right) \geq 1-\xi_i \;\;, \qquad \xi_i \geq 0 \;\;, \quad 1\leq i\leq n \;\;, \end{cases} \tag{9}$$

with $\gamma = \frac{2}{q+1}$, $\gamma_0 = \frac{2}{p+q+1}$ *and* $\gamma^* = 1-\frac{\gamma_0}{\gamma}$. *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 (9) differs from (5) in the RKHS norms* $||\cdot||_{\mathcal{H}_m}$ *and in the parameters* d_ℓ *that accommodate for group sizes.*

Sketch of proof: *Let* $\mathcal{L}$ *be the Lagrangian of problem (8), the first order optimality conditions for* $\sigma_{1,\ell}$ *and* $\sigma_{2,m}$ *are* $\frac{\partial\mathcal{L}}{\partial\sigma_{1,\ell}} = 0$ *and* $\frac{\partial\mathcal{L}}{\partial\sigma_{2,m}} = 0$, *that is:*

$$\begin{aligned} -\frac{p}{2}\,\sigma_{1,\ell}^{-(p+1)}\sum_{m\in G_\ell}\sigma_{2,m}^{-q}||f_m||_{\mathcal{H}_m}^2 + \lambda_1\, d_\ell - \eta_{1,\ell} &= 0 \\ -\frac{q}{2}\,\sigma_{1,\ell}^{-p}\,\sigma_{2,m}^{-(q+1)}\,||f_m||_{\mathcal{H}_m}^2 + \lambda_2 - \eta_{2,m} &= 0 \;\;, \end{aligned}$$

where λ_1 *and* λ_2 *are the Lagrange multipliers corresponding to the equality constraints on* $\boldsymbol{\sigma}_1$ *and* $\boldsymbol{\sigma}_2$ *respectively;* $\eta_{1,\ell}$ *and* $\eta_{2,m}$ *are the Lagrange multipliers corresponding to the inequality constraints on* $\sigma_{1,\ell}$ *and* $\sigma_{2,m}$ *respectively.*

After some tedious algebra, we obtain the optimality conditions for $\sigma_{1,\ell}$ *and* $\sigma_{2,m}$

$$\sigma_{1,\ell} = \frac{\left(d_\ell^{-1}s_\ell\right)^{\frac{q+1}{p+q+1}}}{\sum\limits_\ell d_\ell^{\frac{p}{p+q+1}}\left(s_\ell\right)^{\frac{q+1}{p+q+1}}} \;\;, \tag{10}$$

$$\sigma_{2,m} = \frac{||f_m||_{\mathcal{H}_m}^{\frac{2}{q+1}}\left(d_\ell^{-1}s_\ell\right)^{-\frac{p}{p+q+1}}}{\sum\limits_\ell d_\ell^{\frac{p}{p+q+1}}\left(s_\ell\right)^{\frac{q+1}{p+q+1}}} \quad \text{for } m\in G_\ell \;\;, \tag{11}$$

where $s_\ell = \sum\limits_{m\in G_\ell} ||f_m||_{\mathcal{H}_m}^{\frac{2}{q+1}}$.

Plugging these conditions in Problem (8) yields the claimed result. □

Proposition 2
Necessary Conditions for Convexity: *Problem (8) is not convex if* $|q| > 1$ *or* $|p+q| > 1$.

Proof: *Proposition 1 shows that Problem (8) can be reduced to Problem (9), which is convex when the CAP penalty (5) is convex. The conditions of convexity for this penalty are* $\gamma \geq 1$ *and* $\gamma_0 \geq 1$ *(Zhao* et al.*, to appear), yielding the claimed condition.* □

Proposition 3
Sufficient Conditions for Convexity: *Problem (8) is convex if* $0 \leq q \leq 1$ *and* $p + q = 1$.

Proof: *Problem (8) is convex if* $J(x, y, z) = \dfrac{x^2}{y^p z^{(1-p)}}$ *is convex for positive* y *and* z. *To show this, we show that its Hessian matrix* H *is positive-definite by computing its decomposition in the sum of two positive-definite matrices:*

$$y^p z^{(1-p)} H = 2 \begin{bmatrix} 1 \\ -\frac{xp}{y} \\ \frac{x(p-1)}{z} \end{bmatrix} \begin{bmatrix} 1 \\ -\frac{xp}{y} \\ \frac{x(p-1)}{z} \end{bmatrix}^{\mathrm{t}} + x^2(1-p) \begin{bmatrix} 0 \\ \sqrt{\frac{p}{y}} \\ -\frac{\sqrt{p}}{z} \end{bmatrix} \begin{bmatrix} 0 \\ \sqrt{\frac{p}{y}} \\ -\frac{\sqrt{p}}{z} \end{bmatrix}^{\mathrm{t}} .$$

□

Regarding the values of p and q ensuring the 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 $\overline{K}_\ell = \sum_{m \in G_\ell} K_m$;
- $p = q = \frac{1}{2}$ yields a hierarchical-penalization type penalty on the RKHS norms. It is a true GKL, 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

5.1 A Gradient-Based Wrapper

To address Problem (8), we opt for a wrapper scheme, by considering the following constrained optimization problem:

$$\begin{cases} \min\limits_{\boldsymbol{\sigma}_1,\boldsymbol{\sigma}_2} & J(\boldsymbol{\sigma}_1,\boldsymbol{\sigma}_2) \\ \text{s. t.} & \sum\limits_{\ell} d_\ell\,\sigma_{1,\ell} = 1 \ , \quad \sigma_{1,\ell} \geq 0 \ , \quad 1 \leq \ell \leq L \\ & \sum\limits_{m} \sigma_{2,m} = 1 \ , \quad \sigma_{2,m} \geq 0 \ , \quad 1 \leq m \leq M \ , \end{cases}$$

where $J(\boldsymbol{\sigma}_1,\boldsymbol{\sigma}_2)$ is defined as the objective value of

$$\begin{cases} \min\limits_{\substack{f_1,\ldots,f_M,\\ b,\boldsymbol{\xi}}} & \frac{1}{2}\sum\limits_{\ell}\sigma_{1,\ell}^{-p}\sum\limits_{m\in G_\ell}\sigma_{2,m}^{-q}||f_m||^2_{\mathcal{H}_m} + \quad C\sum\limits_i \xi_i \\ \text{s. t.} & y_i\left(\sum\limits_m f_m(\boldsymbol{x}_i)+b\right) \geq 1-\xi_i \ , \quad \xi_i \geq 0 \ , \quad 1\leq i\leq n \ . \end{cases} \tag{12}$$

In the inner loop, the problem is optimized with respect to $f_1,\ldots,f_M$, b and $\boldsymbol{\xi}$, considering that $(\boldsymbol{\sigma}_1,\boldsymbol{\sigma}_2)$ are fixed. In the outer loop, $(\boldsymbol{\sigma}_1,\boldsymbol{\sigma}_2)$ are updated to decrease the objective function of Problem (8), with f_m, b and $\boldsymbol{\xi}$ being fixed.

From Equations (10) and (11), the outer loop can be carried out in closed form. However, this approach lacks convergence guarantees and may lead to numerical problems, in particular when some elements of $\boldsymbol{\sigma}_1$ or $\boldsymbol{\sigma}_2$ approach zero. These updates should thus be reserved for initializing the algorithm, so as to provide a rapid decrease of the objective function.

After the initialization phase, our approach to solve Problem (8) draws on the gradient-based MKL algorithm of Rakotomamonjy *et al.* (2007). We still have the wrapper scheme described above, except that the outer loop is a simple projected gradient descent update, which can be computed using that the objective function $J(\boldsymbol{\sigma}_1,\boldsymbol{\sigma}_2)$ is actually an optimal SVM objective value.

5.2 Computing the Gradient

The dual formulation offers a convenient means to compute the gradient $\nabla J(\boldsymbol{\sigma}_1,\boldsymbol{\sigma}_2)$. The derivation of the Lagrangian of Problem (12), which is omitted here for brevity, shows that its dual formulation is identical to the one of a standard SVM using the aggregated kernel $\overline{K}_{\boldsymbol{\sigma}_1,\boldsymbol{\sigma}_2}$ defined as

$$\overline{K}_{\boldsymbol{\sigma}_1,\boldsymbol{\sigma}_2}(\boldsymbol{x},\boldsymbol{x}') = \sum_{\ell}\sigma_{1,\ell}^{p}\sum_{m\in G_\ell}\sigma_{2,m}^{q}K_m(\boldsymbol{x},\boldsymbol{x}') \ .$$

Hence, the dual problem takes the usual form

$$\begin{cases} \max\limits_{\alpha} & -\frac{1}{2}\sum\limits_{i,j}\alpha_i\alpha_j y_i y_j \overline{K}_{\boldsymbol{\sigma}_1,\boldsymbol{\sigma}_2}(\boldsymbol{x}_i,\boldsymbol{x}_j) \; + \sum\limits_i \alpha_i & (13) \\ \text{s. t.} & \sum\limits_i \alpha_i y_i = 0 \;\;, \quad C \geq \alpha_i \geq 0 \;\;, \quad 1 \leq i \leq n \;\;, & \\ & & (14) \end{cases}$$

which can be solved by any SVM solver.

As $J(\boldsymbol{\sigma}_1, \boldsymbol{\sigma}_2)$ is defined as the optimal objective value of the convex Problem (12), strong duality applies, and $J(\boldsymbol{\sigma}_1, \boldsymbol{\sigma}_2)$ is also the dual objective value, that is

$$-\frac{1}{2}\sum_{i,j}\alpha_i^\star\alpha_j^\star y_i y_j \overline{K}_{\boldsymbol{\sigma}_1,\boldsymbol{\sigma}_2}(\boldsymbol{x}_i,\boldsymbol{x}_j) + \sum_i \alpha_i^\star \;\;, \tag{15}$$

where $\alpha^\star$ solves Problem (13).

The existence and computation of the derivatives of optimal value functions such as $J(\cdot)$ have been largely discussed in the literature. For our purpose, an appropriate reference is (Bonnans & Shapiro, 1998, Theorem 4.1), which, in a nutshell, states here that the differentiability of $J(\boldsymbol{\sigma}_1, \boldsymbol{\sigma}_2)$ is ensured by the unicity of $\alpha^\star$, and by the differentiability of (15). [3] Furthermore, the derivatives of $J(\boldsymbol{\sigma}_1, \boldsymbol{\sigma}_2)$ can be computed as if $\alpha^\star$ were not to depend on $(\boldsymbol{\sigma}_1, \boldsymbol{\sigma}_2)$.

Thus, the gradient $\nabla J(\boldsymbol{\sigma}_1, \boldsymbol{\sigma}_2)$ is simply the gradient of the dual function (15), where we take into consideration the dependence of $\overline{K}_{\boldsymbol{\sigma}_1,\boldsymbol{\sigma}_2}$ in $(\boldsymbol{\sigma}_1, \boldsymbol{\sigma}_2)$:

$$\begin{aligned} \frac{\partial J}{\partial \sigma_{1,\ell}} &= -\frac{p\,\sigma_{1,\ell}^{(p-1)}}{2}\sum_{i,j}\alpha_i^\star\alpha_j^\star y_i y_j \sum_{m\in G_\ell}\sigma_{2,m}^q K_m(\boldsymbol{x}_i,\boldsymbol{x}_j) \\ \frac{\partial J}{\partial \sigma_{2,m}} &= -\frac{q\,\sigma_{1,\ell}^{p}\sigma_{2,m}^{(q-1)}}{2}\sum_{i,j}\alpha_i^\star\alpha_j^\star y_i y_j K_m(\boldsymbol{x}_i,\boldsymbol{x}_j) \;\;, \end{aligned}$$

Then, we have all the ingredients to apply the machinery developed for MKL by Rakotomamonjy *et al.* (2007).

6 Experiments

This section presents a set of experiments on two datasets. In the following, MKL stands for the MKL algorithm (Rakotomamonjy *et al.*, 2007), $\text{GKL}_{\frac{1}{2}}$ is a convex version of our algorithm, with $p = q = \frac{1}{2}$ (that is a $\ell(\frac{4}{3}, 1)$ mixed-norm), and GKL_1 is a non-convex version of our algorithm, with $p = q = 1$ (that is a $\ell(1, \frac{2}{3})$ mixed-norm).

[3] The unicity of $\alpha^\star$ is ensured provided that the Gram matrix built from kernel $\overline{K}_{\boldsymbol{\sigma}_1,\boldsymbol{\sigma}_2}$ is positive-definite. To enforce this property, a small ridge may be added to the diagonal.

6.1 Spambase

The spambase problem, from UCI machine learning repository (Asuncion & Newman, 2007), consists in predicting whether an email is a spam or not. The dataset is composed of 57 continuous attributes, divided in 4 groups. The first group is composed of 42 attributes concerning word frequencies; the second group is composed of 6 attributes concerning specific numbers frequencies; the third group is composed of 6 attributes concerning punctuation frequencies; the fourth group is composed of 3 attributes characterizing the distribution of capital letters.

We have randomly picked 30 % of the dataset (1381 examples) for testing, and then divided the 3220 remaining examples in 10 distinct training sets. The parameter C has been tested for 7 logarithmically spaced values, varying from 1 to 10^6, and has been selected by 5-fold cross-validation. The classification accuracy was computed on the test set.

The different kernels have been structured according to the 4 groups of variables. For all groups, multivariate and univariate gaussian kernels have been built. The bandwidths of the gaussian kernels go from 10^{-2} to 10^1. Thus, 244 kernels have been used to solve this classification problem. Each training set has been standardized before constructing the kernels, and each kernel has been weighted so that the trace is equal to 1.

Table 6.1 reports the prediction accuracy, the number of selected kernels, and the running time for a classical SVM, MKL and GKL$_{\frac{1}{2}}$. Note that the SVM has been trained with the mean of the 244 kernels built. The results have been averaged, using the estimates obtained from the 10 training sets on the testing set. For fair time comparison, we have initialized the MKL algorithm with the procedure described in section 5. For MKL and GKL$_{\frac{1}{2}}$ algorithms, we have used the same termination criterion. It is based both on the variation of the objective value $J(\boldsymbol{\sigma})$ $\left(\frac{J(\boldsymbol{\sigma})^{t-1}-J(\boldsymbol{\sigma})^{t}}{1+J(\boldsymbol{\sigma})^{t}} \leq \epsilon\right)$, and on the variation of the $\boldsymbol{\sigma}$ parameters $\left(\max_m \left|\frac{\sigma_m^{t-1}-\sigma_m^{t}}{1+\sigma_m^{t}}\right| \leq \sqrt{\epsilon}\right)$, where $\epsilon = 10^{-12}$, and $\boldsymbol{\sigma} = (\boldsymbol{\sigma}_1, \boldsymbol{\sigma}_2)$ for GKL.

Algorithms	Accuracy	# Kernels	Time (s)
SVM	93.33 ± 0.73	244	1.13 ± 0.33
MKL	91.19 ± 3.31	131.20 ± 43.33	58.37 ± 16.44
GKL$_{\frac{1}{2}}$	93.09 ± 0.84	234.20 ± 5.59	46.02 ± 5.85

Table 1: Average performances for 3 different algorithms on the spambase dataset. The prediction accuracy, the number of selected kernels, and the running time are reported.

In terms of accuracy, SVM and GKL$_{\frac{1}{2}}$ perform slightly better than MKL, with an insignificant advantage for SVM. Concerning sparseness, MKL uses fewer kernels than GKL$_{\frac{1}{2}}$, for which no groups are eliminated. Note that previous analyzes have shown

that all groups of variables are relevant for classification (Hastie *et al.*, 2001). Applying a $\ell(1, q)$ mixed-norm with $q > 1$ would have been more appropriate on this dataset, and will be investigated in further experiments. Indeed, it would achieved a better selection: with GKL$_{\frac{1}{2}}$, the number of kernels remains important because the $\ell_{\frac{4}{3}}$ norm applied within groups is not of a sparse nature, contrary to the ℓ_1 norm of the MKL. Finally, the running time of GKL$_{\frac{1}{2}}$ is slightly better than that of MKL. The SVM is much faster since there is no weighting parameter to optimize.

6.2 Channel Selection for Brain-Computer Interface

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).

Here, GKL is particularly relevant for 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 hould also be carried out within each channel. In order to reach a sparse solution at the channel and the time frame levels, we choose to explore a non-convex parametrization of GKL implementing the $\ell(1, \frac{2}{3})$ mixed-(pseudo)norm penalty, (that is $p = 1$ and $q = 1$), which enables to get sparseness within and between groups.

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 $\sigma_{1,\ell}, \ell = 1, \cdots, 64$ and $\sigma_{2,m}, m = 1, \cdots, 896$.

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. Then the parameter which leads to the highest AUC has been selected. This overall procedure has been repeated 10 times.

Using a small part of the examples for training can be justified by the use of ensem-

ble of SVM (that we do not consider here) on a latter stage of the EEG classification procedure (see for instance Rakotomamonjy *et al.* (2005)), whereas the AUC as a performance measure is justified by how the EEG recognition is transformed into selected character in the P300.

Table 6.2 summarizes the average performance of 3 different algorithms: a classical SVM, a MKL SVM, and our group kernel learning. The number of channels and kernels selected by these algorithms have also been reported. Note that the classical SVM have been trained with the mean of 896 kernels we deal with. Results show that performances of the 3 algorithms are similar. However, we also note that the number of channels selected by our algorithm is far smaller than the ones selected by MKL while the number of kernels is larger.

Algorithms	AUC	# Channels	# Kernels
SVM	83.87 ± 0.8	64	896
MKL	82.11 ± 1.7	48.5 ± 3	105.3 ± 14
GKL_1	84.29 ± 1.2	24.2 ± 8	129.3 ± 41

Table 2: Average AUC performances of 3 different algorithms on the BCI datasets. The number of channels and kernels involved in the decision function is also related.

Interpreting these results tells us that using GKL successfully leads to structured variable selection. Interestingly, structuring the selection also yields to slight performance improvements. These results corroborate prior knowledge and findings about BCI P300 Speller paradigms that is: only a subset of the 64 channels and time samples corresponding to $300ms$ after the visual stimuli are mainly related to the class labels of the EEG signals.

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 arbritrary 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.

References

ASUNCION A. & NEWMAN D. J. (2007). UCI machine learning repository.

BACH F. R., LANCKRIET G. R. G. & JORDAN M. I. (2004). Multiple kernel learning, conic duality, and the SMO algorithm. In *Proceedings of the twenty-first international conference on Machine Learning*, p. 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**(6), 1044–1051.

BONNANS J. & SHAPIRO A. (1998). Optimization problems with pertubation: A guided tour. *SIAM Review*, **40**(2), 228–264.

CHAPELLE O., VAPNIK V., BOUSQUET O. & MUKHERJEE S. (2002). Choosing multiple parameters for support vector machines. *Machine Learning*, **46**(1), 131–159.

CRISTIANINI N., CAMPBELL C. & SHAWE-TAYLOR J. (1999). Dynamically adapting kernels in support vector machines. In M. S. KEARNS, S. A. SOLLA & D. A. COHN, Eds., *Advances in Neural Information Processing Systems 11*, p. 204–210: MIT Press.

CRISTIANINI N., SHAWE-TAYLOR J., ELISSEEFF A. & KANDOLA K. (2002). On kernel-target alignment. In T. G. DIETTERICH, S. BECKER & Z. GHAHRAMANI, Eds., *Advances in Neural Information Processing Systems 14*, p. 367–373: MIT Press.

GRANDVALET Y. & CANU S. (2003). Adaptive scaling for feature selection in SVMs. In S. BECKER, S. THRUN & K. OBERMAYER, Eds., *Advances in Neural Information Processing Systems 15*, p. 569–576: MIT Press.

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

HASTIE T., TIBSHIRANI R. & FRIEDMAN J. H. (2001). *The Elements of Statistical Learning*. Springer.

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.

ONG C. S., SMOLA A. J. & WILLIAMSON R. C. (2005). Learning the kernel with hyperkernels. *Journal of Machine Learning Research*, **6**, 1043–1071.

RAKOTOMAMONJY A., BACH F., CANU S. & GRANDVALET Y. (2007). More efficiency in multiple kernel learning. In Z. GHAHRAMANI, Ed., *Proceedings of the 24th International Conference on Machine Learning (ICML 2007)*, p. 775–782: Omnipress.

RAKOTOMAMONJY A., GUIGUE V., MALLET G. & ALVARADO V. (2005). Ensemble of SVMs for improving brain-computer interface P300 speller performances. In W. DUCH, J. KACPRZYK, E. OJA & S. ZADROZNY, Eds., *15th International Conference on Artificial Neural Networks*, volume 3696 of *Lecture Notes in Computer Science*, p. 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 & 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**(1), 267–288.

WESTON J., MUKHERJEE S., CHAPELLE O., PONTIL M., POGGIO T. & VAPNIK V. (2001). Feature selection for SVMs. In T. K. LEEN, T. G. DIETTERICH & V. TRESP, Eds., *Advances in Neural Information Processing Systems 13*, p. 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**(1), 49–67.

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