Mach Learn (2010) 79: 73–103
DOI 10.1007/s10994-009-5150-6

Composite kernel learning

Marie Szafranski · Yves Grandvalet · Alain Rakotomamonjy

Received: 6 March 2009 / Revised: 22 July 2009 / Accepted: 8 September 2009 /
Published online: 14 October 2009

Abstract The Support Vector Machine 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 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 correspond to channels.

Keywords Supervized learning · Support vector machine · Kernel learning · Structured kernels · Feature selection and sparsity

1 Motivations

Kernel methods are very versatile tools for learning from examples (Schölkopf and Smola 2001). In these models, the observations $\boldsymbol{x}$ belonging to some measurable instance space

Editors: Nicolo Cesa-Bianchi, David R. Hardoon, and Gayle Leen.

M. Szafranski (✉)
CNRS FRE 3190—IBISC, Université d'Évry Val d'Essonne, 91025 Évry Cedex, France
e-mail: marie.szafranski@ibisc.fr

M. Szafranski
CNRS UMR 6166—LIF, Universités d'Aix-Marseille, Marseille, France

Y. Grandvalet
CNRS UMR 6599—Heudiasyc, Université de Technologie de Compiègne, 60205 Compiègne Cedex, France
e-mail: yves.grandvalet@hds.utc.fr

A. Rakotomamonjy
EA 4108—LITIS, Université de Rouen, 76801 Saint-Étienne-du-Rouvray Cedex, France
e-mail: alain.rakotomamonjy@univ-rouen.fr

$\mathcal{X}$ are implicitly mapped in a feature space $\mathcal{H}$ via a mapping $\Phi : \mathcal{X} \to \mathcal{H}$, where $\mathcal{H}$ is a Reproducing Kernel Hilbert Space (RKHS) with reproducing kernel $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$.

When learning from a single source, selecting the right kernel is an essential choice, conditioning the success of the learning method. Indeed, the kernel is crucial in many respects regarding data representation issues. Formally, the primary role of K is to define the evaluation functional in $\mathcal{H}$:

$$\forall \boldsymbol{x} \in \mathcal{X}, K(\boldsymbol{x}, \cdot) \in \mathcal{H} \quad \text{and} \quad \forall f \in \mathcal{H}, f(\boldsymbol{x}) = \langle f, K(\boldsymbol{x}, \cdot) \rangle_{\mathcal{H}},$$

but K also defines

1. $\mathcal{H}$, since $\forall f \in \mathcal{H}, \forall \boldsymbol{x} \in \mathcal{X}, \exists \alpha_i \in \mathbb{R}, i = 1, \ldots, \infty, f(\boldsymbol{x}) = \sum_{i=1}^{\infty} \alpha_i K(\boldsymbol{x}_i, \boldsymbol{x})$;
2. a metric, and hence a smoothness functional in $\mathcal{H}$, where, for f defined above, $\|f\|_{\mathcal{H}}^2 = \sum_{i=1}^{\infty} \sum_{j=1}^{\infty} \alpha_i \alpha_i K(\boldsymbol{x}_i, \boldsymbol{x}_j)$;
3. the mapping $\boldsymbol{\phi}(\boldsymbol{x}) = K(\boldsymbol{x}, \cdot)$ and a scalar product between observations: $\forall (\boldsymbol{x}, \boldsymbol{x}') \in \mathcal{X}^2, \langle \Phi(\boldsymbol{x}), \Phi(\boldsymbol{x}') \rangle_{\mathcal{H}} = K(\boldsymbol{x}, \boldsymbol{x}')$.

In other words, the kernel defines

1. the hypothesis space $\mathcal{H}$;
2. the complexity measure $\|f\|_{\mathcal{H}}^2$ indexing the family of nested functional spaces in the structural risk minimization principle (Vapnik 1995);
3. the representation space of data endowed with a scalar product.

These observations motivate the developments of means to avoid the use of unsupported kernel, which do not represent prior knowledge about the task at hand, and are fixed before observing data. The consequences of the arbitrary choice that may be involved at this level range from interpretability issues to poor performances (see for example Weston et al. 2001; Grandvalet and Canu 2003). "Learning the kernel", aims at alleviating these problems, by adapting the kernel to the problem at hand.

A general model of learning the kernel has two components: (i) a family of kernels, that is, a set $\mathcal{K} = \{K_\theta, \theta \in \Theta\}$, where Θ is a set of parameters and K_θ is the kernel parameterized by θ, and (ii) an empirical functional, whose minimization with respect to θ will be used to choose a kernel in $\mathcal{K}$ that best fits the data according to some empirical criterion.

In this paper, we develop the Composite Kernel Learning (CKL) approach, which is dedicated to learning the kernel when there is a known group structure among a set of candidate kernels. This framework applies to learning problems arising from a single data source when the input variables have a group structure, and it is also particularly well suited to the problem of learning from multiple sources. Then, each source can be represented by a group of kernels, and the algorithm aims at identifying the relevant sources and their apposite kernel representation. Thanks to the notion of source embedded in the kernel parameterization, our framework introduces in the Multiple Kernel Learning framework (Lanckriet et al. 2004) the ability to select sources, or alternatively to ensure the use of all sources.

We briefly review the different means proposed to extend kernel methods beyond the predefined kernel setup in Sect. 2, with an emphasis on Multiple Kernel Learning and the parametric relatives that inspired our approach. In Sect. 3, we formalize the general CKL framework, starting from basic desiderata, and finishing with a general and compact formulation amenable to optimization. The algorithm is provided in Sect. 4, and experiments are reported in Sect. 5. Finally, Sect. 6 summarizes the paper and provides directions for future research. We used the standard notations found in textbooks, such as Schölkopf and Smola (2001); they are introduced when they first appear in the document, and an overview is provided in Appendix C.

2 Flexible kernel methods

From now on, we restrict our discussion to binary classification, where, from n pairs $(\boldsymbol{x}_i, y_i) \in \mathcal{X} \times \{-1, 1\}$ of observations and binary labels, one aims at inferring a decision rule that predicts the class label y of any observation $\boldsymbol{x} \in \mathcal{X}$. However, most of our statements carry on to other settings, such as multiclass classification, regression or clustering with kernel methods. Indeed, the penalties we will propose are learned from data, but they are defined without any interdependence with the data-fitting term.

2.1 Support vector machines

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

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

where $f \in \mathcal{H}$, $b \in \mathbb{R}$ and $\boldsymbol{\xi} \in \mathbb{R}^n$ are the optimization variables, and C is a positive regularization parameter that is the only adjustable parameter in the SVM learning problem once $\mathcal{H}$ has been chosen. Note that, though C and $\mathcal{H}$ are usually tuned in the same outer loop, their role is completely different. While C sets the trade-off between regularity and data-fitting, $\mathcal{H}$, the so-called feature space, defines the embedding of the observations via the mapping Φ. Hence, while choosing C amounts to select a model in a nested family of functional spaces whose size is controlled by $\|f\|_{\mathcal{H}}^2$ (or equivalently by the margin in $\mathcal{H}$), choosing $\mathcal{H}$ boils down to picking a representation (endowed with a metric) for the observations $\boldsymbol{x}$.

Adapting the kernel to data is not representative of model selection strategies that typically balance goodness of fit with simplicity. As a result, Vapnik (1995) did not provide guidelines for choosing the kernel, which was considered to be chosen prior to seeing data when deriving generalization bounds for SVMs. Following these observations, all methods adapting the kernel to data will be here referred to as kernel learning instead of model selection.

Since solving (1) is usually not flexible enough to provide good results when $\mathcal{H}$ is fixed, most applications of SVM incorporate a mechanism for learning the kernel. This mechanism may be as simple as picking a kernel in a finite set, but may also be an elaborate optimization process within a finite or infinite family of kernels. These options are described in more details below.

2.2 Learning the kernel

In our view, kernel learning methods encompass all processes where the kernel K is chosen from a pre-defined set $\mathcal{K}$, by optimizing an empirical functional defined on the training set $\{\boldsymbol{x}_i, y_i\}_{i=1}^n$. With this viewpoint, the most rudimentary, but also the most common way to learn the kernel is cross-validation, that consists here in (i) defining a family of kernels (*e.g.* Gaussian), indexed by one or more parameters (*e.g.* bandwidth), $\mathcal{K} = \{K_m\}_{m=1}^M$, where m indexes the trial values for the kernel parameters, and, (ii) computing a cross-validation score on each hyper-parameter setting, and picking the kernel whose hyper-parameters minimize the cross-validation score. In this example, the empirical functional used for learning the

kernel is the minimum of the cross-validation score with respect to the trial values of the regularization parameter C.

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 in the kernel choice, where cross-validation may still be used, but only for tuning the regularization parameter C.

2.2.1 Filters, wrappers & embedded methods

As already stated, 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,[1] where one encounters filters, wrappers and embedded methods (Guyon and Elisseeff 2003). Some general frameworks, such as *hyperkernels* (Ong et al. 2005) 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 with the objective value of Problem (1). For example, the kernel target alignment of Cristianini et al. (2002) adapts the kernel matrix 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). In all these methods, the set of admissible kernels $\mathcal{K}$ is defined by kernel parameter(s) θ, where θ may be the kernel bandwidth, or a diagonal or a full covariance matrix in Gaussian kernels. The empirical criterion optimized with respect to θ is a generalization bound such as the radius/margin bound (using the actual radius and margin obtained with θ on the training set).

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 and Canu 2003). In this line of research, Argyriou et al. (2006) consider combinations of kernels whose parameters are optimized by a DC (difference of convex functions) program. The present approach builds on the simplest Multiple Kernel Learning (MKL) framework initiated by Lanckriet et al. (2004), which is limited to the combination of prescribed kernels but leads to simpler convex programs.

2.2.2 Multiple kernel learning

In MKL, we are provided with M candidate kernels, $K_1, \ldots, K_M$, and we wish to estimate the parameters of the SVM classifier together with the weights of a convex combination of kernels $K_1, \ldots, K_M$ that defines the *effective kernel* $K_{\boldsymbol{\sigma}}$

$$\mathcal{K} = \left\{ K_{\boldsymbol{\sigma}} = \sum_{m=1}^{M} \sigma_m K_m, \sigma_m \geq 0, \sum_{m=1}^{M} \sigma_m = 1 \right\}. \tag{2}$$

Each kernel K_m is associated to a RKHS $\mathcal{H}_m$ whose elements will be denoted f_m, and $\boldsymbol{\sigma} = (\sigma_1, \ldots, \sigma_M)^\top$ is the vector of coefficients to be learned under the convex combination constraints. The positiveness constraint ensures that K is positive definite when the base kernels

[1] In variable selection, the situation is simpler since selecting variables provides simpler models, so that variable selection or shrinkage may be used for model selection purposes.

K_m are themselves positive definite. The unitary constraint may be seen as a normalization of the effective kernel that is necessary to avoid diverging solutions. In an embedded approach, where the empirical functional used to select $K_{\boldsymbol{\sigma}}$ is the fitting criterion (1), the unitary constraint on $\boldsymbol{\sigma}$ is also important to preserve the role of the SVM regularization parameter C. Furthermore, provided that the individual kernels K_m are properly normalized (with identical trace norm), the norm constraint on $\boldsymbol{\sigma}$ can be motivated by generalization error bounds that are valid for learned kernels. The first works in this direction (Lanckriet et al. 2004; Bousquet and Herrmann 2003) were found to be meaningless, with bounds on the expected error never less than one, but Srebro and Ben-David (2006) provide tighter bounds based on the *pseudodimension* of a family of kernel, which is at most the number of kernels in combination (2).

The original MKL formulation of Lanckriet et al. (2004) was based on the dual of the SVM optimization problem. It was later shown to be equivalent to the following primal problem (Bach et al. 2004)

$$\begin{cases} \min\limits_{\substack{f_1,\ldots,f_M \\ b,\boldsymbol{\xi}}} & \frac{1}{2}\left(\sum\limits_{m=1}^{M}\|f_m\|_{\mathcal{H}_m}\right)^2 + C\sum\limits_{i=1}^{n}\xi_i & \text{(3a)} \\ \text{s.t.} & y_i\left(\sum\limits_{m=1}^{M} f_m(\boldsymbol{x}_i) + b\right) \geq 1-\xi_i, \qquad \xi_i \geq 0, \quad i=1,\ldots,n, & \text{(3b)} \end{cases}$$

whose solution leads to a decision rule of the form $\operatorname{sign}(\sum_{m=1}^{M} f_m^\star(\boldsymbol{x}) + b^\star)$. This expression of the learning problem is remarkable in that it only differs 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.

With this mixed-norm, the objective function is not differentiable at $\|f_m\|_{\mathcal{H}_m} = 0$. This is the cause of a considerable algorithmic burden, which is rewarded by the sparseness of solutions, that is, solutions where some functions f_m have zero norm. As each function f_m is computed from K_m, this results in a sparse kernel expansion in (2).

Looking at Problem (3), one may wonder why a mixed-norm should be more flexible than a squared RKHS norm, and why the former should be considered as a kernel learning technique. These questions are answered with the MKL formulation of Rakotomamonjy et al. (2008), which is a variational form of Problem (3), in the sense that the solution of Problem (3) is defined as the minimizer with respect to the additional variable $\boldsymbol{\sigma}$ of an optimization problem in $f_1,\ldots,f_M,b,\boldsymbol{\xi}$. By introducing the parameters $\sigma_1,\ldots,\sigma_M$ of the combination (2) in the objective function, kernel learning comes explicitly into view. The resulting optimization problem, which is equivalent to Problem (3), circumvents its differentiability issues, as shown below:

$$\begin{cases} \min\limits_{\substack{f_1,\ldots,f_M \\ b,\boldsymbol{\xi},\boldsymbol{\sigma}}} & \frac{1}{2}\sum\limits_{m=1}^{M}\frac{1}{\sigma_m}\|f_m\|_{\mathcal{H}_m}^2 + C\sum\limits_{i=1}^{n}\xi_i & \text{(4a)} \\ \text{s.t.} & y_i\left(\sum\limits_{m=1}^{M} f_m(\boldsymbol{x}_i) + b\right) \geq 1-\xi_i, \qquad \xi_i \geq 0, \quad i=1,\ldots,n & \text{(4b)} \\ & \sum\limits_{m=1}^{M}\sigma_m = 1, \qquad \sigma_m \geq 0, \quad m=1,\ldots,M, & \text{(4c)} \end{cases}$$

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

MKL may be used in different prospects. When the individual kernels K_m represent a series, such as Gaussian kernels with different scale parameters, it constitutes an alternative to cross-validating the kernel parameters. When the input data originates from M differents sources, and that each kernel is affiliated to one group of input variables, it enables to select relevant sources.

However, MKL is not meant to address problems where several kernels pertain to a single source. In this situation, its sparseness mechanism does not account for the structure among kernels. In particular, it cannot favor solutions discarding all the kernels computed from an irrelevant source. Although most of the related coefficients should vanish in combination (2), spurious correlation may cause irrelevant sources to participate to the solution. A single coefficient could be attached for each source, but this solution forbids kernel adaptation within each source, whose equivalent kernel would be clamped to the average kernel. Note also that, in the opposite situation where we want to involve all sources in the solution, with only a few kernels per source, MKL is not guaranteed to provide a solution complying with the requisite.

2.3 Group and composite penalties

The selection/removal of kernels between or within predefined groups relies on the definition of a structure among kernels. This type of hierarchy has been investigated among variables in linear models (Yuan and Lin 2006; Szafranski et al. 2008a; Zhao et al. 2009).

The very general Composite Absolute Penalties (CAP) family of Zhao et al. (2009) considers a linear model with M parameters, $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_M)^T$. Let $\mathcal{I} = \{1, \ldots, M\}$ be a set of index on these parameters, a group structure on the parameters is defined by a series of L subsets $\{\mathcal{G}_\ell\}_{\ell=1}^L$, where $\mathcal{G}_\ell \subseteq \mathcal{I}$. 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

$$\sum_{\ell=1}^{L} \left(\sum_{m \in \mathcal{G}_\ell} |\beta_m|^{\gamma_\ell} \right)^{\gamma_0/\gamma_\ell}.$$

To our knowledge, there is no efficient general purpose algorithm for fitting parametric models with penalties belonging to the CAP family, but for the prominent particular cases listed below, such algorithms exist. They all consider $\gamma_0 = 1$ that enforces sparseness at the group level and identical norms $\{\gamma_\ell\}_{\ell=1}^L$ at the parameter level:

- $\gamma_\ell = 1$ is the LASSO (Tibshirani 1996), which clears the group structure;
- $\gamma_\ell = 4/3$ is the Hierarchical Penalization (Szafranski et al. 2008a), which gives rise to few dominant variables within groups;
- $\gamma_\ell = 2$ is the group-LASSO (Yuan and Lin 2006), which applies a proportional shrinkage to the variables within groups;
- $\gamma_\ell = \infty$ is the iCAP penalty (examined in more details by Zhao et al. 2009), which limits the maximal magnitude of the coefficients within groups.

Mixed-norms correspond to groups defined as a partition of the set of variables. A CAP may also rely on nested groups, $\mathcal{G}_1 \subset \mathcal{G}_2 \subset \cdots \subset \mathcal{G}_L$, 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 $\{\mathcal{I} \setminus \mathcal{G}_L\}, \{\mathcal{G}_L \setminus \mathcal{G}_{L-1}\}, \ldots, \{\mathcal{G}_2 \setminus \mathcal{G}_1\}, \mathcal{G}_1$ according to some heredity principle. This example is provided here to stress that Zhao et al.'s notion of hierarchy differs from the one that will be introduced in Sect. 3.

2.4 Relations between MKL and CAP

CAP and its earlier predecessor LASSO have been initiated in the parametric regression setting. Using the notations introduced for CAP, the LASSO penalty is

$$\sum_{\ell=1}^{L}\Big(\sum_{m\in\mathcal{G}_\ell}|\beta_m|\Big)=\sum_{m=1}^{M}|\beta_m|=\sum_{m=1}^{M}\big(\beta_m^2\big)^{1/2},$$

but the LASSO penalty can take a more general form. In the example of M RKHS $\mathcal{H}_1,\ldots,\mathcal{H}_M$, one may consider the penalty

$$\sum_{m=1}^{M}\|f_m\|_{\mathcal{H}_m}=\sum_{m=1}^{M}(\boldsymbol{\alpha}_m^\top\mathbf{K}_m\boldsymbol{\alpha}_m)^{1/2},$$

where $\boldsymbol{\alpha}_m\in\mathbb{R}^n$, $\mathbf{K}_m$ is the mth kernel matrix $\mathbf{K}_m(i,j)=K_m(\boldsymbol{x}_i,\boldsymbol{x}_j)$ and $f_m(\boldsymbol{x})=\sum_{i=1}^{n}\boldsymbol{\alpha}_m(i)K(\boldsymbol{x}_i,\boldsymbol{x})$.

The representer theorem (Schölkopf and Smola 2001) ensures that the f_m solving the MKL Problem (3a) are of the above form. Hence, MKL may be seen as a kernelization of LASSO, extended to SVM classifiers, whose penalty generalizes the ones proposed in the framework of additive modeling with spline functions (see Grandvalet and Canu 1999) to arbitrary RKHS. In this sense, MKL extends the simplest member of the CAP family to SVM classifiers.

Being a sum of ℓ_2 norms, the MKL penalty is also of the group-LASSO type, but the groups are defined at the level of the expansion coefficients $\boldsymbol{\alpha}_m$.[2] CKL extends the MKL framework by defining groups at a higher level, that is at the kernel level: Composite Kernel Learning is to CAP what Multiple Kernel Learning is to LASSO.

3 Composite kernel learning

The flat combination of kernels in MKL does not include any mechanism to cluster the kernels related to each source. In order to favor the selection/removal of kernels between or within predefined groups, one has to define a structure among kernels, which will guide the selection process. We present here the kernel methods counterpart of the methods surveyed in Sect. 2.3 for parametric models.

3.1 Groups of kernels

We consider problems where we have a set of kernels, partitioned in groups, which may correspond to subsets of inputs, sources, or more generally distinct families of similarity measures between examples. This structure will be represented by a tree, as we envision more complex structures with a hierarchy of nested groups. We index the tree depth by h, with $h=0$ for the root, and $h=2$ for the leaves. The leaf nodes represent the kernels at hand for the classification task; the nodes at depth 1 stand for the *group-kernels* formed by combining the kernels within each group; the root represents the global *effective kernel*

[2]Note that, except for the case where $\mathbf{K}_m$ has a block-diagonal structure, there is no effective grouping in the MKL penalty.

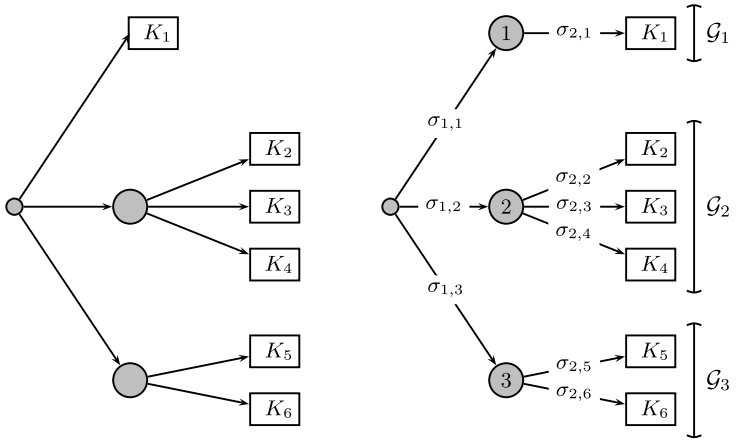


Fig. 1 A tree of height two depicting groups of kernels

merging the group-kernels. Without loss of generality, we consider that all leaves are at depth 2. If not the case, an intermediate node should be inserted at depth 1 between the root and each isolated leaf, as illustrated in Fig. 1.

In CKL, learning the kernel consists in learning the parameters of each combination of kernels. There are $L + 1$ such combinations, one at each group level, and one at the root level. As illustrated in Fig. 1, the weights of these combinations may be thought of as being attached to the branches of the tree: a branch stemming from the root and going to node ℓ is labelled by $\sigma_{1,\ell}$, which is the weight associated to the ℓth group in the effective kernel; a branch stemming from node ℓ at depth 1 and reaching leaf m is labelled by $\sigma_{2,m}$, which is the weight associated to the mth kernel in its group-kernel.

3.2 Kernel selection

In the learning process, we would like to suppress the kernels and/or the groups that are irrelevant for the classification task. In the tree representation, this removal process consists in pruning the tree. When a branch is pruned at the leaf level, a single kernel is removed from the combination. When a subtree is pruned, a group-kernel is removed from the combination, and the corresponding group of kernels has no influence on the classifier. With the branch labeling introduced above and illustrated in Fig. 1, removing kernel m consists in setting $\sigma_{2,m}$ to 0, and removing group ℓ consists in setting $\sigma_{1,\ell}$ to 0.

For the purpose of performing flat kernel selection, $\sigma_{1,\ell}$ is redundant with $\sigma_{2,m}$, but the decomposition proposed here allows to pursue different goals, by constraining the solutions to have a given sparsity pattern induced by the sparseness constraints at each level of the hierarchy: in the example of Fig. 1, though they delete the same number of leaves, we may prefer for a solution with $\sigma_{1,3} = 0$ (that is, the removal of group 3 composed of kernels 5 and 6) to $\sigma_{2,3} = \sigma_{2,4} = 0$ that also removes two kernels, but retains all the groups.

We now elaborate on the notations introduced in Sect. 2.3 for the CAP family. The M kernels situated at the leaves are indexed by $\{1, \ldots, m, \ldots, M\}$, and the group-kernels (at

depth 1) are indexed by $\{1, \ldots, \ell, \ldots, L\}$. The set $\mathcal{G}_\ell$ of cardinality d_ℓ indexes the leaf-kernels belonging to group-kernel ℓ, that is, the children of node ℓ. The groups form a partition of the leaf-kernels, that is, $\bigcup_\ell \mathcal{G}_\ell = \{1, \ldots, m, \ldots, M\}$ and $\sum_\ell d_\ell = M$. Note that, to lighten notations, the range of indexes will often be omitted in summations, in which case: indexes i and j refer to examples and go from 1 to n; index m refers to leaf-kernels and goes from 1 to M; index ℓ refers to group-kernels and goes from 1 to L.

In a hard selection setup, where $\boldsymbol{\sigma}_1 = (\sigma_{1,1} \ldots \sigma_{1,L})^\top$ and $\boldsymbol{\sigma}_2 = (\sigma_{2,1} \ldots \sigma_{2,M})^\top$ are binary vectors, the learning problem is stated as follows

$$
\begin{cases}
\min\limits_{\substack{f_1,\ldots,f_M \\ b,\boldsymbol{\xi},\boldsymbol{\sigma}_1\boldsymbol{\sigma}_2}} & \dfrac{1}{2}\sum\limits_m \|f_m\|^2_{\mathcal{H}_m} + C\sum\limits_i \xi_i & \text{(5a)} \\
\text{s.t.} & y_i\left(\sum\limits_\ell \sigma_{1,\ell} \sum\limits_{m\in\mathcal{G}_\ell} \sigma_{2,m} f_m(\boldsymbol{x}_i) + b\right) \geq 1 - \xi_i, \quad i = 1, \ldots, n & \text{(5b)} \\
& \xi_i \geq 0, \quad i = 1, \ldots, n & \text{(5c)} \\
& \sum\limits_\ell d_\ell \sigma_{1,\ell} \leq s_1, \qquad \sigma_{1,\ell} \in \{0, 1\}, \quad \ell = 1, \ldots, L & \text{(5d)} \\
& \sum\limits_m \sigma_{2,m} \leq s_2, \qquad \sigma_{2,m} \in \{0, 1\}, \quad m = 1, \ldots, M, & \text{(5e)}
\end{cases}
$$

where s_1 and s_2 designate the number of leaves that should be retained after pruning. The constraint (5d) on $\boldsymbol{\sigma}_1$ imposes some pruning at the group level, while the constraint (5e) on $\boldsymbol{\sigma}_2$ imposes some additional pruning at the leaf level. Note that constraint (5e) may only be active if $s_2 \leq s_1$.

Problem (5) has a number of shortcomings. First, it is an inherently combinatorial problem, for which finding a global optimum is challenging even with a small number of kernels. Second, this type of hard selection problem is known to provide unstable solutions (Breiman 1996), especially when the number of kernels is not orders of magnitude lower than the training set size. Unstability refers here to large changes in the overall predictor, in particular via the changes in the set of selected kernels, in response to small perturbations of the training set. Besides having detrimental effects on the variability of model parameters, unstability has been shown to badly affect model selection (Breiman 1996). More recently, stability has been shown to characterize the generalization ability of learning algorithms (Bousquet and Elisseeff 2002).

As the kernel choice is especially decisive for small to moderate sample sizes, we should devise well-behaved algorithms in this setup. Hence, we will consider stable soft-selection techniques, such as the ones based on ℓ_2 or ℓ_1 regularization.

3.3 Soft selection

To convert Problem (5) in a smooth soft-selection problem, we will transform the binary vectors $\boldsymbol{\sigma}_1$ and $\boldsymbol{\sigma}_2$ in continuous positive variables, which may either "choke" some branches or prune them. We also replace the hyper-parameters s_1 and s_2 in constraints (5d) and (5e) by 1, since their role is redundant with the parameters d_ℓ when the latter are not restrained to be equal to the group size. The problem reads

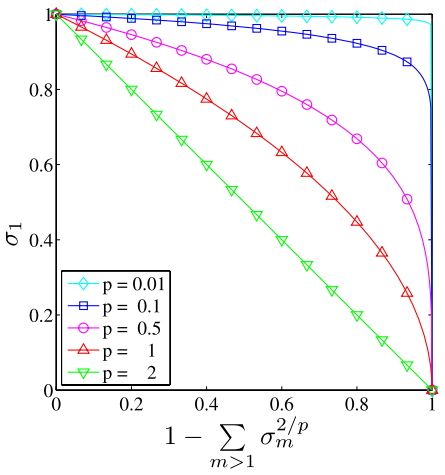


Fig. 2 Graph of σ_1 *vs.* $(1 - \sum_{m>1} \sigma_m^{2/p})$ when $\sum_m \sigma_m^{2/p} = 1$. As p goes to zero, the constraint approaches a hard selection process with $\sigma_m \in \{0, 1\}$

$$
\begin{cases}
\min\limits_{\substack{f_1,\ldots,f_M \\ b,\boldsymbol{\xi},\boldsymbol{\sigma}_1\boldsymbol{\sigma}_2}} & \dfrac{1}{2}\sum\limits_m \|f_m\|^2_{\mathcal{H}_m} + C\sum\limits_i \xi_i & \text{(6a)} \\
\text{s.t.} & y_i\left(\sum\limits_\ell \sigma_{1,\ell} \sum\limits_{m\in\mathcal{G}_\ell} \sigma_{2,m} f_m(\boldsymbol{x}_i) + b\right) \geq 1 - \xi_i, \quad i = 1,\ldots,n & \text{(6b)} \\
& \xi_i \geq 0, \quad i = 1,\ldots,n & \text{(6c)} \\
& \sum\limits_\ell d_\ell \sigma_{1,\ell}^{2/p} \leq 1, \qquad \sigma_{1,\ell} \geq 0, \quad \ell = 1,\ldots,L & \text{(6d)} \\
& \sum\limits_m \sigma_{2,m}^{2/q} \leq 1, \qquad \sigma_{2,m} \geq 0, \quad m = 1,\ldots,M, & \text{(6e)}
\end{cases}
$$

where we incorporated two hyper-parameters p and q appearing respectively in constraints (6d) and (6e), whose roles are to drive these constraint closer or further from their binary counterpart in (5), as illustrated in Fig. 2. These exponents can thus be tuned to implement harder or softer selection strategies, and different values for p and q will lead to more or less emphasis on sparsity within or between groups. Some properties related to the choice of p and q will be discussed in the following section, and the practical outcomes of these choices will be illustrated in Sect. 5.

3.4 Properties

Problem (6) is difficult to analyze and to optimize. We derive here a "flat" equivalent formulation using a single weight per kernel K_m, using the simple fact that the composition of combinations is itself a combination. The kernel group structure will not be lost in the process, it will be transferred to the weights of the combination.

This simplification proceeds in three steps (see details in Appendix A). First, variable $\boldsymbol{\sigma}_2$ disappears in a change of variables where $\boldsymbol{\sigma}$ appears, then, we use a necessary optimality condition that ties $\boldsymbol{\sigma}_1$ with $\boldsymbol{\sigma}$ for all stationnary points, including the global maximum.[3]

[3] A stationnary point is defined as a point satisfying the KKT conditions.

Finally, plugging these optimality conditions into Problem (6), we get

$$\begin{cases} \min\limits_{\substack{f_1,\ldots,f_M \\ b,\boldsymbol{\xi},\boldsymbol{\sigma}}} & \frac{1}{2}\sum\limits_m \frac{1}{\sigma_m}\|f_m\|^2_{\mathcal{H}_m} + C\sum\limits_i \xi_i & \text{(7a)} \\ \text{s.t.} & y_i\Big(\sum\limits_m f_m(\boldsymbol{x}_i) + b\Big) \geq 1-\xi_i, \qquad \xi_i \geq 0, \quad i=1,\ldots,n & \text{(7b)} \\ & \sum\limits_\ell d_\ell^{p/(p+q)} \Big(\sum\limits_{m\in\mathcal{G}_\ell} \sigma_m^{1/q}\Big)^{q/(p+q)} \leq 1, \qquad \sigma_m \geq 0, \quad m=1,\ldots,M, & \text{(7c)} \end{cases}$$

where, here and it what follows $(\sum_m \sigma_m^{1/0})^0$ is defined as the ℓ_∞ norm, with value $\max_m \sigma_m$ since $\sigma_m \geq 0$.

Problem (7) is equivalent to Problem (6) in the sense that its stationnary points correspond to the ones of (6). As the objective function is convex, the stationnary points are minima and multiple (local) minima may only occur if the feasible domain is non-convex.

This flat formulation is more easily amenable to the analysis of convexity, and optimization can be carried out by a simple adaptation of the SimpleMKL algorithm (Rakotomamonjy et al. 2008). Indeed, compared to (4), Problem (7) only differs in constraint (7c) on $\boldsymbol{\sigma}$, where the ℓ_1 norm is replaced by a mixed-norm $\ell_{(1/q,1/(p+q))}$. As a special case, MKL is recovered from CKL for parameters $(p,q)=(0,1)$.

Proposition 1 *Problem* (7) *is convex 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 (7a) is convex (cf. Rakotomamonjy et al. 2008);
- the usual SVM constraints (7b) define convex sets in $(f_1,\ldots,f_M,b,\boldsymbol{\xi})$;
- if $0 \leq q \leq 1$ and $0 \leq p+q \leq 1$, the constraints (7c) defines a convex set in $\boldsymbol{\sigma}$ since
  - $(\sum_{m\in\mathcal{G}_\ell} \sigma_m^{1/q})^q$ is convex;
  - $\sum_\ell t_\ell^{1/(p+q)}$ is convex and non-decreasing in t_ℓ. □

The proposition below generalizes the equivalence between the MKL formulations of Bach et al. (2004) and Rakotomamonjy et al. (2008), that is, between Problems (3) and (4) respectively. If MKL may be seen as the kernelization of the LASSO, CKL can be interpreted as the kernelization of the hierarchical penalizer of Szafranski et al. (2008a) or more generally of the Composite Absolute Penalty (CAP) of Zhao et al. (2009).

Proposition 2 *Problem* (7) *is equivalent to*

$$\begin{cases} \min\limits_{\substack{f_1,\ldots,f_M \\ b,\boldsymbol{\xi}}} & \frac{1}{2}\Big(\sum\limits_\ell d_\ell^t \Big(\sum\limits_{m\in\mathcal{G}_\ell} \|f_m\|^s_{\mathcal{H}_m}\Big)^{r/s}\Big)^{2/r} + C\sum\limits_i \xi_i & \text{(8a)} \\ \textit{s.t.} & y_i\Big(\sum\limits_m f_m(\boldsymbol{x}_i) + b\Big) \geq 1-\xi_i, \qquad \xi_i \geq 0, \quad i=1,\ldots,n, & \text{(8b)} \end{cases}$$

where $s=\frac{2}{q+1}$, $r=\frac{2}{p+q+1}$ *and* $t=1-\frac{r}{s}$, *in the sense that the minima of* (7) *are the minima of* (8).

See proof in Appendix B.

Table 1 Equivalence between mixed-norms in σ_m in Problem (7), and mixed-norms in $\|f_m\|_{\mathcal{H}_m}$ in Problem (8) for some particular (p, q) values

(p, q)	$(0, 1)$	$(1, 0)$	$(-1, 1)$	$(1/2, 1/2)$	$(1,1)$
σ_m	$\ell_{(1,1)}$	$\ell_{(1,\infty)}$	$\ell_{(\infty,1)}$	$\ell_{(1,2)}$	$\ell_{(1/2,1)}$
$\|f_m\|_{\mathcal{H}_m}$	$\ell_{(1,1)}$	$\ell_{(1,2)}$	$\ell_{(2,1)}$	$\ell_{(1,4/3)}$	$\ell_{(2/3,1)}$

Corollary 1 *Problem (7) is sparse at the group level if and only if $p + q \geq 1$. It is sparse at the leaf level if and only if $q \geq 1$ or $p + q \geq 1$.*

Proof This is the direct consequence of the equivalence stated in Proposition 2, since sparsity is obtained if and only if the boundary of the feasible region is nondifferentiable at $f_m = 0$ (Nikolova 2000). The sub-differential at $\|f_m\|_{\mathcal{H}_m} = 0$ is reduced to one point if and only if $s > 1$, that is $q < 1$, and the sub-differential at $\sum_{m\in\mathcal{G}_\ell} \|f_m\|_{\mathcal{H}_m} = 0$ is reduced to one point if and only if $r > 1$, that is $p + q < 1$. □

Note that the external square on the norm of (8) affects the strength of the penalty, but not its type. Hence, CKL penalizes a kernelized mixed-norm $\ell_{(r,s)}$ in $\|f_m\|_{\mathcal{H}_m}$.

Table 1 displays some particular instances of the equivalence given in Proposition 2. Since the latter was obtained from the primal formulation of Problem (7), it also holds for non-convex penalties, such as the one displayed in the last column of the table.

The first column of Table 1 illustrates that CKL indeed generalizes MKL, since it enables to implement a $\ell_{(1,1)}$ mixed-norm, that is the ℓ_1 norm of MKL. The second column leads to a $\ell_{(1,2)}$ mixed-norm, that could also be obtained by an MKL algorithm using the average of leaf-kernels within each group. The third column displays a more interesting result, with the $\ell_{(2,1)}$ that encourages a sparse expansion within each group, and then performs a standard SVM with the kernel formed by summing the group-kernels. This setting corresponds to the situation where we want all sources to participate to the solution, but where the relevant similarities are to be discovered for each source. It has been used in the regression framework for audio signals (Kowalski and Torrésani 2008). The fourth solution, leading to a $\ell_{(1,4/3)}$ norm is the kernelized version of hierarchical penalization (Szafranski et al. 2008a), which takes into account the group structure, provides sparse results at the group-level and approximately sparse ones at the leaf level, with few leading coefficients. Finally, the last column displays a non-convex solution that enables exact sparsity at the group-level and at the leaf-level, with a group-structure that greatly encourages group selection.

Figure 3 illustrates the shape of the feasible region

$$\sum_{\ell} d_\ell^t \Big(\sum_{m\in\mathcal{G}_\ell} \|f_m\|_{\mathcal{H}_m}^s \Big)^{r/s} \leq 1,$$

for the values of (r, s) given in Table 1, in a problem with $M = 3$ kernels.

The left column depicts the 3D-shape in the positive octant, where the two horizontal axes represent the positive quadrant $(\|f_1\|_{\mathcal{H}_1}, \|f_2\|_{\mathcal{H}_2})$ associated to the first group $\mathcal{G}_1$, and the vertical axis represents $\|f_3\|_{\mathcal{H}_3}$ associated to the second group $\mathcal{G}_2$.

The cuts at $\|f_2\|_{\mathcal{H}_2} = 0$ and $\|f_3\|_{\mathcal{H}_3} = 0$ are displayed to provide a between-group plane and the within group view of the feasible region in the center and right column respectively. These plots provide an intuitive way to comprehend the convexity and sparsity issues. Sparsity is related to convexity and the shape of the boundary of the admissible set as $\|f_m\|_{\mathcal{H}_m}$ goes to zero (Nikolova 2000).

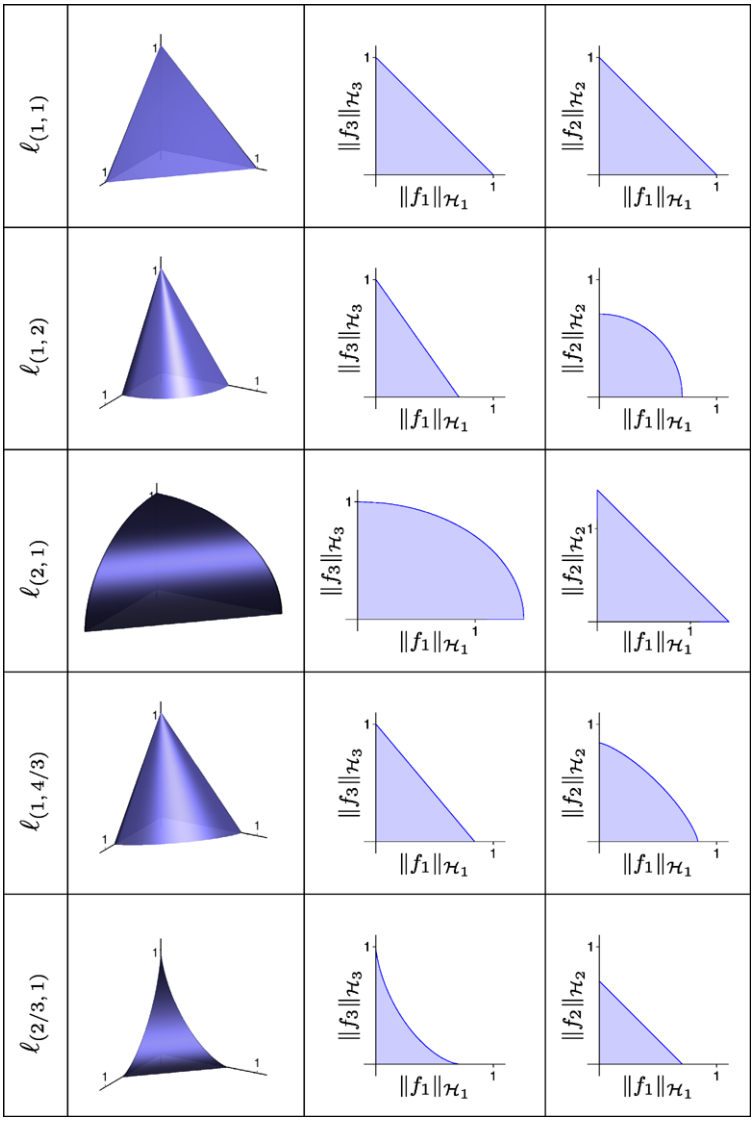


Fig. 3 Feasible regions for the mixed-norm of Table 1 for a problem with three kernels (K_1 and K_2 in the same group, K_3 in the second group): *left*, 3-D representation in $(\|f_1\|_{\mathcal{H}_1}, \|f_2\|_{\mathcal{H}_2}, \|f_3\|_{\mathcal{H}_3})$; *center*: between-group cut at $\|f_2\|_{\mathcal{H}_2} = 0$; *right*: within-group cut at $\|f_3\|_{\mathcal{H}_3} = 0$

4 Solving the problem

Since CKL generalizes MKL, we begin this section by a brief review of the algorithmic developments of MKL dedicated to solve Problem (3) or one of its equivalent forms. The original MKL algorithm of Lanckriet et al. (2004) was based on a quadratically-constrained quadratic program (QCQP) solver that had high computational requirements and was thus

limited to small problems, that is, small numbers of kernels and data points. This restraint motivated the introduction of a smoothing term allowing to use the SMO algorithm (Bach et al. 2004).

The following generation of MKL algorithms was then based on wrapper algorithms, consisting in two nested optimization problems, where the outer loop optimizes the kernel and the inner loop is a standard SVM solver. The outer loop was a cutting plane algorithm for the Semi-Infinite Linear Program (SILP) of Sonnenburg et al. (2006) that optimizes the non-smooth dual of Problem (3); it was later improved by a gradient algorithm addressing Problem (4) in the SimpleMKL of Rakotomamonjy et al. (2008).

The benefit of these approaches is to rely on standard SVM solvers, for which several efficient implementations exist. This type of approach was also used in the multiple task learning framework by Argyriou et al. (2008), and again in some recent developments of MKL (Xu et al. 2009; Bach 2009).

We first chose the gradient-based approach that was demonstrated to be efficient for MKL (Szafranski et al. 2008b). Nevertheless, moving along a curved surface such as the ones illustrated in Fig. 3 may be problematic for some mixed-norms. Hence, we pursue here another wrapper approach, where we will use a fixed point strategy to update the kernels parameters in the outer loop.

4.1 A wrapper approach

Our wrapper scheme extends SimpleMKL by considering the following optimization problem

$$\begin{cases} \min\limits_{\boldsymbol{\sigma}} \quad J(\boldsymbol{\sigma}) & \text{(9a)} \\ \text{s.t.} \quad \sum\limits_{\ell} \left(d_\ell^p \Big(\sum\limits_{m \in \mathcal{G}_\ell} \sigma_m^{1/q} \Big)^q \right)^{1/(p+q)} \leq 1, \qquad \sigma_m \geq 0, \quad m = 1, \ldots, M, & \text{(9b)} \end{cases}$$

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

$$\begin{cases} \min\limits_{\substack{f_1, \ldots, f_M \\ b, \boldsymbol{\xi}}} \quad \frac{1}{2} \sum\limits_m \frac{1}{\sigma_m} \| f_m \|_{\mathcal{H}_m}^2 + C \sum\limits_i \xi_i & \text{(10a)} \\ \text{s.t.} \quad y_i \left(\sum\limits_m f_m(\boldsymbol{x}_i) + b \right) \geq 1 - \xi_i, \qquad \xi_i \geq 0, \quad i = 1, \ldots, n. & \text{(10b)} \end{cases}$$

In the inner loop, the criterion is optimized with respect to $\{f_m\}$, b and $\boldsymbol{\xi}$, considering that the coefficients $\boldsymbol{\sigma}$ are fixed. In the outer loop, $\boldsymbol{\sigma}$ is updated to decrease the criterion, using an expression derived from the optimality conditions, with the dual variables related to $\{f_m\}$, b and $\boldsymbol{\xi}$ being fixed.

4.2 First-order optimality conditions

To lay down the foundations of our algorithm, we derive the first-order optimality conditions for each part of Problem (7). These conditions characterize the global minimizer if Problem (7) is convex, and all local minima otherwise. The Lagrangian reads

$$\mathcal{L} = \frac{1}{2}\sum_m \frac{1}{\sigma_m}\|f_m\|^2_{\mathcal{H}_m} + C\sum_i \xi_i - \sum_i \alpha_i \left[y_i \left(\sum_m f_m(\boldsymbol{x}_i) + b \right) + \xi_i - 1 \right] - \sum_i \eta_i \xi_i$$
$$+ \lambda \left[\sum_\ell \left(d_\ell^p \left(\sum_{m \in \mathcal{G}_\ell} \sigma_m^{1/q} \right)^q \right)^{1/(p+q)} - 1 \right] - \sum_m \mu_m \sigma_m,$$

where α_i and η_i, the usual positive Lagrange multipliers related to the constraints (7b) on the slack variable ξ_i, will be optimized by considering Problem (10), while λ and μ_m are the positive Lagrange multipliers related to constraints (7c) on σ_m, that appear in Problem (9).

4.2.1 *Optimality conditions for f_m, b and $\boldsymbol{\xi}$*

We first focus on the optimality conditions of Problem (10). The derivative of $\mathcal{L}$ with respect to f_m, b and $\boldsymbol{\xi}$ give

$$\frac{\partial \mathcal{L}}{\partial f_m} = 0 \Rightarrow f_m(\cdot) = \sigma_m \sum_i \alpha_i y_i K_m(\boldsymbol{x}_i, \cdot)$$
$$\frac{\partial \mathcal{L}}{\partial b} = 0 \Rightarrow \sum_i \alpha_i y_i = 0$$
$$\frac{\partial \mathcal{L}}{\partial \xi_i} = 0 \Rightarrow 0 \leq \alpha_i \leq C.$$

Hence, the equivalent dual formulation of Problem (10) is a standard SVM problem

$$\begin{cases} \max\limits_{\boldsymbol{\alpha}} & -\frac{1}{2}\sum\limits_{i,j} \alpha_i \alpha_j y_i y_j K_{\boldsymbol{\sigma}}(\boldsymbol{x}_i, \boldsymbol{x}_j) + \sum\limits_i \alpha_i & \text{(11a)} \\ \text{s.t.} & \sum\limits_i \alpha_i y_i = 0 & \text{(11b)} \\ & C \geq \alpha_i \geq 0, \quad i = 1, \ldots, n, & \text{(11c)} \end{cases}$$

where $K_{\boldsymbol{\sigma}}$ is the effective kernel defined in (2). Note that this dual pertains to the sub-problem (10), not to the global problem (7).

4.2.2 *Optimality conditions for σ_m*

The first-order optimality conditions for σ_m, derived in Appendix B, establish the relation between σ_m and $\|f_m\|_{\mathcal{H}_m}$, which is

$$\sigma_m = \|f_m\|^{2q/(q+1)}_{\mathcal{H}_m} \left(d_\ell^{-1} s_\ell\right)^{p/(p+q+1)} \left(\sum_{\ell'} d_{\ell'}^{p/(p+q+1)} s_{\ell'}^{(q+1)/(p+q+1)} \right)^{-(p+q)}, \qquad (12)$$

where $s_\ell = \sum_{m \in \mathcal{G}_\ell} \|f_m\|^{2/(q+1)}_{\mathcal{H}_m}$.

Since $\|f_m\|^2_{\mathcal{H}_m} = \sigma_m^2 \sum_{i,j} \alpha_i \alpha_j y_i y_j K_m(\boldsymbol{x}_i, \boldsymbol{x}_j)$, (12) only provides an implicit definition of the optimal value of σ_m. Let $g_m(\boldsymbol{\sigma})$ denote the right-hand-side of (12), we have that $g(\boldsymbol{\sigma}) = (g_1(\boldsymbol{\sigma}), \ldots, g_M(\boldsymbol{\sigma}))$ is a continuous mapping from the closed unit ball defined by

constraint (9b) to itself. Hence, Brouwer's fixed point theorem applies,[4] and the outer loop of the wrapper can be performed by a fixed point strategy, using the expression (12).

When the values of p and q do not define a convex set in (9b), Brouwer's theorem does not hold anymore. Nevertheless, one can circumvent this problem by considering the optimization with respect to $\boldsymbol{\sigma}_1$ and $\boldsymbol{\sigma}_2$, such as in Problem (6) provided the constraints (6d) and (6e) both span closed unit balls.

4.3 Algorithm

We now have all the ingredients to define our wrapper algorithm (see Algorithm 1). The stopping criterion for assessing the convergence of $\boldsymbol{\sigma}$ can be based on standard criteria for fixed point algorithms, while the one related to the SVM solver can be based on the duality gap. In the following experiments, it is respectively based on the stability of $\boldsymbol{\sigma}$ and $J(\boldsymbol{\sigma})$.

5 Channel selection for brain computer interfaces

We consider here two studies in Brain-Computer Interfaces (BCI). In BCI, one aims at recognizing the cerebral activity of a person subject to a stimulus, thanks to an array of sensors placed on the scalp of the subject that records a set of electroencephalograms (EEG). Here, the EEG signals are collected from 64 electrodes or *channels*, positioned onto the scalp as illustrated in Fig. 4.

Automated channel selection has to be performed for each single subject since it leads to better performances or a substantial reduction of the number of useful channels (Schröder et al. 2005). 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 cheaper, easier to use and to set-up.

In this setup, each electrode may be considered as a source that generates a series of potentials along the experiment. Composite Kernel Learning is well-suited to the identification of a specific behavior in the EEG signals, by its ability to encode the notion of channels.

Algorithm 1: CKL

initialize $\boldsymbol{\sigma}$
solve the SVM problem $\rightarrow J(\boldsymbol{\sigma})$
repeat
  repeat
    $\boldsymbol{\sigma} = g(\boldsymbol{\sigma})$ `// with` $g_m(\boldsymbol{\sigma})$ `defined by the r.h.s of (12)`
  until *convergence*
  solve the SVM problem $\rightarrow J(\boldsymbol{\sigma})$
until *convergence*

[4] Brouwer's fixed point theorem states that, if $\mathcal{B}$ is a closed unit ball, then, any continuous function $g : \mathcal{B} \rightarrow \mathcal{B}$, has at least one fixed point.

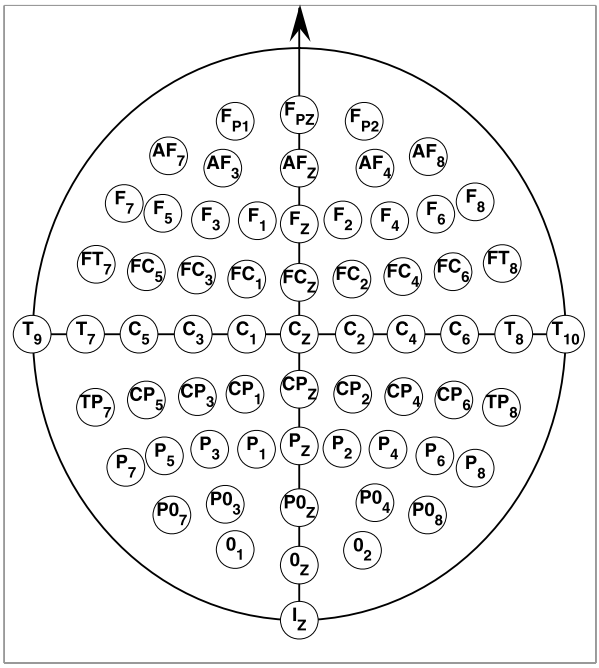


Fig. 4 Positions of the 64 electrodes on the scalp, for the two considered BCI problems. The *arrow* represents the frontal direction

Besides the benefits of potentially reducing the number of channels, CKL may also be beneficial if able to identify the salient features within each channel. Hence, we will experiment with a non-convex parameterization of CKL that encourages sparseness within and between groups, in order to reach a sparse solution at the channel and the feature levels. Note that, for non-convex settings, we have no means to assess the convergence towards a global optimum. Though the SVM solver may return the optimal decision rule for the returned $\boldsymbol{\sigma}$, we have no way to secure global convergence for the outer Problem (9), and no certificate of sub-optimality, such as the one that could be provided by a duality gap.

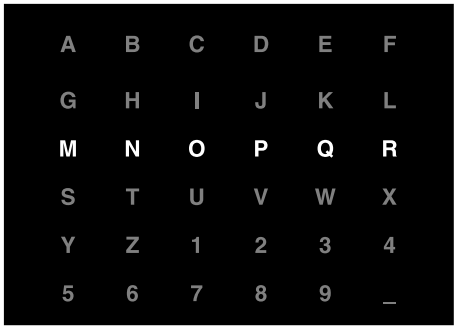
In the following, $\mathrm{CKL}_{1/2}$ stands for a convex version of our algorithm, with $p = q = 1/2$ (a $\ell_{(1,4/3)}$ mixed-norm), CKL_1 is a non-convex version, with $p = q = 1$ (a $\ell_{(2/3,1)}$ dissimilarity, that we will also abusively qualify as a mixed-norm). Note that MKL is also implemented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

5.1 P300 speller paradigm

5.1.1 Protocol

The so-called *oddball* paradigm states that a rare expected stimulus produces a positive deflection in an EEG signal after about 300 ms. The P300 speller interface is based on this paradigm (Farwell and Donchin 1998). Its role is to trigger a related event potential, namely the P300, in response to a visual stimulus. This protocol uses a matrix composed of 6 rows and 6 columns of letters and numbers, as illustrated in Fig. 5. First, the subject chooses a specific character in the matrix. Then, the 12 lines (rows or columns) are intensified in a random order. When an intensified row or column contains the chosen character, the subject is asked to count; this is assumed to generate a P300. Because the signal to noise ratio of a scalp EEG signal is usually low, this process is repeated 15 times per character.

Fig. 5 The spelling matrix



The dataset, collected for a BCI competition (Blankertz et al. 2004), is processed as described in Rakotomamonjy and Guigue (2008). For each channel, 14 time samples (that will be referred as frames), going from the beginning of the stimulus up to 667 ms after, have been extracted from the EEG signals. Frames 7 and 8, respectively centered around 300 and 350 ms, are the most salient ones according to the paradigm.

The dataset is composed of 7560 EEG signals (observations), paired with positive or negative stimuli responses (classes). The 896 features extracted (64 channels $\times 14$ frames) are not transformed. However, to unify the presentation, we will refer to these features as kernels. The kernels related to a given channel form a group of kernels, and we have to learn $M = 896$ coefficients σ_m, divided into $L = 64$ groups. Thus, our goal is to identify the significant channels, and within these channels, the significant frames, which discriminate the positive from the negative signals.

The classification protocol is the following: we have randomly picked 567 training examples from the dataset and used the remaining as testing examples. The parameter C has been selected by 5-fold cross-validation. This overall procedure has been repeated 10 times. Using a small part of the examples for training is motivated by the use of ensemble of SVM (that we do not consider here) at a later stage of the EEG classification procedure (Rakotomamonjy and Guigue 2008). The performance is measured by the AUC, due to the post-processing that is done throughout repetitions in the P300: as the final decision regarding letters is taken after several trials, the correct row and column should receive high scores to identify correctly the letter.

5.1.2 Results

Table 2 summarizes the average performance of SVM, MKL, and CKL, that is, for 4 different penalization terms: quadratic penalization for the classical SVM (that is, trained with the mean of the 896 kernels), ℓ_1 norm for MKL, and mixed-norms for the two versions of CKL assessed here: $\text{CKL}_{1/2}$ and CKL_1. The number of channels and kernels selected by these algorithms and the time needed for the training process are also reported, together with the standard deviations.

The prediction performances of the four algorithms are similar, with an insignificant advantage for MKL. In terms of kernels, MKL is much sparser than $\text{CKL}_{1/2}$, but twice less sparse than CKL_1. Regarding the number of groups, CKL_1 is still the sparsest solution, removing about three quarters of the channels. At this level $\text{CKL}_{1/2}$ is sparser than MKL, although it retained many more kernels: as expected, $\text{CKL}_{1/2}$ favors sparseness among groups rather than sparseness in kernels.

Table 2 Average results and standard deviations, for SVMs with different kernel learning strategies on the BCI dataset (P300 speller paradigm)

Algorithms	AUC	# Channels	# Kernels	Time (s)
SVM	84.6 ± 0.9	64	896	1.9 ± 1.0
$\text{CKL}_{1/2}$	84.9 ± 1.1	40.1 ± 15.2	513.0 ± 224.7	149.1 ± 94.1
CKL_1	84.7 ± 1.1	14.6 ± 13.1	65.8 ± 52.2	64.8 ± 18.5
MKL	85.7 ± 0.9	47.0 ± 7.9	112.6 ± 46.2	60.3 ± 12.1

Insofar as SVM does not require to estimate the coefficients σ_m, the training process is much faster than for other methods. The kernel learning methods training time is however still reasonable, and is rewarded by interpretability and cheaper evaluations in the test phase. $\text{CKL}_{1/2}$ is slower than MKL and CKL_1 on this problem, but this difference is not consistently observed: the orders of magnitude are identical for all versions.

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

$$\frac{d_\ell^t}{Z} \left(\sum_{m \in \mathcal{G}_\ell} \| f_m^\star \|_{\mathcal{H}_m}^s \right)^{1/s}, \tag{13}$$

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

$$\| f_m^\star \|_{\mathcal{H}_m}^2 = {\sigma_m^\star}^2 \sum_{i,j} \alpha_i^\star \alpha_j^\star y_i y_j K_m(\boldsymbol{x}_i, \boldsymbol{x}_j).$$

The results for CKL_1 are particularly neat, with high relevances for the electrodes in the areas of the visual cortex (lateral electrodes PO_7 and PO_8). The scalp maps for MKL and $\text{CKL}_{1/2}$ show the importance of the same region, followed by the primary motor and somatosensory cortex ($\text{C}_\bullet$ and CP_Z).[5] In addition, they also highlight numerous frontal electrodes that are not likely to be relevant for the BCI P300 Speller paradigm. Finally, the plots of relevance through time (not shown) are similar for all kernel learning methods, with a sudden peak at frames 7 and 8 followed by a slow decline.

5.1.3 Sanity check for channel selection

We provide supplementary experiments to support the relevance of the channel selection mechanism of CKL. We first have randomly picked x channels, then randomly selected y kernels among the $x \times 14$ candidates. Variable x (resp. y) has been set so that it corresponds to the average number of channels (resp. kernels) used by $\text{CKL}_{1/2}$ and CKL_1, that is 41 and 15 (resp. 513 and 66).

Table 3 gives the average performances for classical SVMs: SVM_x is trained with a subset of x channels randomly chosen as described above, while SVM_{CV} is trained with the single channel that reaches the highest cross-validation score.

[5] These channels also appear in the third quartile map of CKL_1.

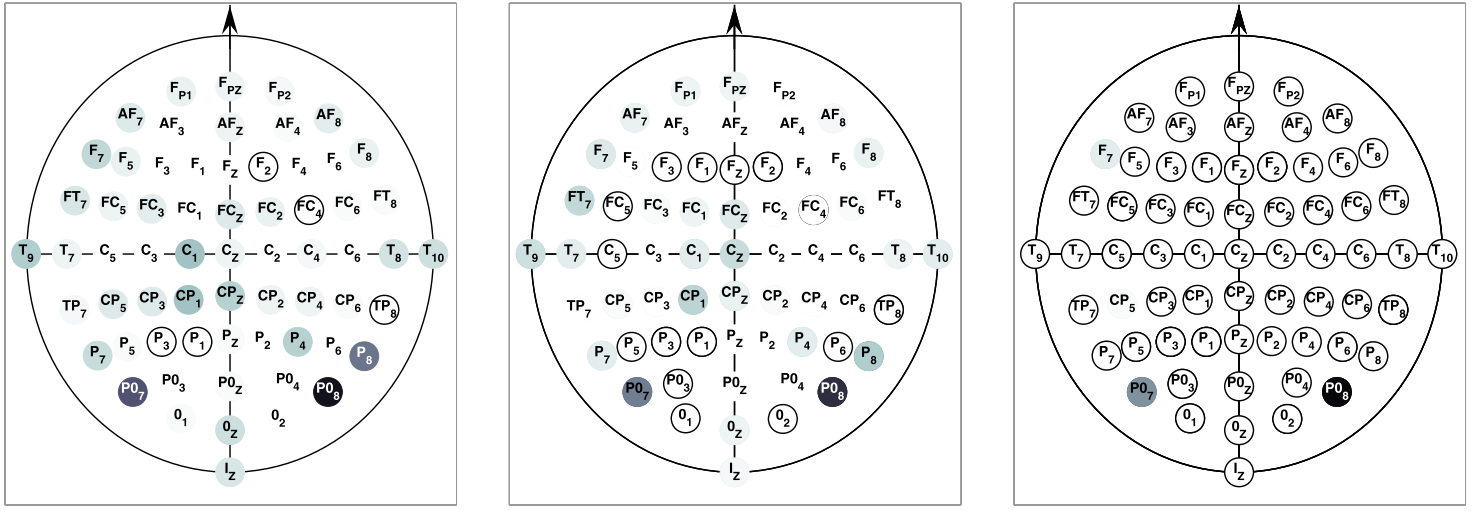


Fig. 6 Electrode median relevance for MKL, $\mathrm{CKL}_{1/2}$ and CKL_1 (P300 speller paradigm). The *darker* the color, the *higher* the relevance. *Electrodes in white with a black circle* are discarded (the relevance is exactly zero)

Table 3 Average results and standard deviations for SVMs (P300 speller paradigm). SVM_{CV} selects the single best channel using a cross-validation procedure, while SVM_x randomly selects a subset of x channels

Algorithms	AUC	# Channels	# Kernels
SVM_{41}	80.7 ± 1.0	41	513
SVM_{15}	76.8 ± 1.7	15	66
SVM_{CV}	68.8 ± 2.0	1	14

With only one channel left, SVM_{CV} performs significantly worse than any other method. Several channels are thus necessary to build accurate SVM classifiers. Note that most of the channels picked out by cross-validation, shown in Fig. 7, are also identified by CKL (see Fig. 6). SVM_{15} behaves poorly compared with CKL_1, highlighting the ability of CKL to identify appropriate channels. The same remark applies to SVM_{41}, where, despite the important number of channels and kernels involved, the average AUC is much lower than for $CKL_{1/2}$ that selected 41 channels. Figure 7 shows that some of the channels assumed to be relevant according to $CKL_{1/2}$ are missing here, especially electrodes PO_8 and P_8 located in the visual cortex, and electrodes CP_Z, CP_1 and C_1 in the somatosensory cortex.

5.2 Contingent negative variation paradigm

5.2.1 Protocol

This new set of BCI experiments aims at detecting some activated regions in the brain when an event is being anticipated (Garipelli et al. 2009).[6] The potentials are here recorded according to the Contingent Negative Variation (CNV) paradigm (Walter et al. 1964). In this paradigm, a warning stimulus predicts the appearance of an imperative stimulus in a predictable inter-stimulus-interval. More precisely, an experiment processes as follows. A subject, looking at a screen, encounters two kinds of events:

1. In "GO" events, a green dot is displayed in the middle of the screen. This signal triggers the anticipation of the subject. Four seconds later, the dot becomes red, prompting the subject to press a button as soon as possible.
2. In "NOGO " events, a yellow dot is displayed in the middle of the screen. The subject has been instructed to do nothing in this situation. When, four seconds later, the dot becomes red, the subject does not react.

The data gather recordings on two subjects, in 20 experimental sessions, each being composed of 10 trials. For each subject, we have thus 200 examples. The 64 EEG signals are available from time 0 to 3.25 s, in the anticipation phase, before the event appears (at 4 s). This results in $64 \times 21 = 1344$ linear kernels.

Available knowledge on the problem identifies the central role of the electrode C_Z. More generally, the channels located in the central region of the scalp are expected to be relevant for classification, contrary to the one at the periphery. Complying with that knowledge, Garipelli et al. (2009) use Linear Discriminant Analysis (LDA) on C_Z to estimate the predictability of anticipation.

5.2.2 Results

We compare the results obtained with LDA to the ones achieved by CKL. The parameter C is estimated by 10-fold cross-validation, which is also used to estimate the test error rate.

[6] We thank the authors for sharing their data with us.

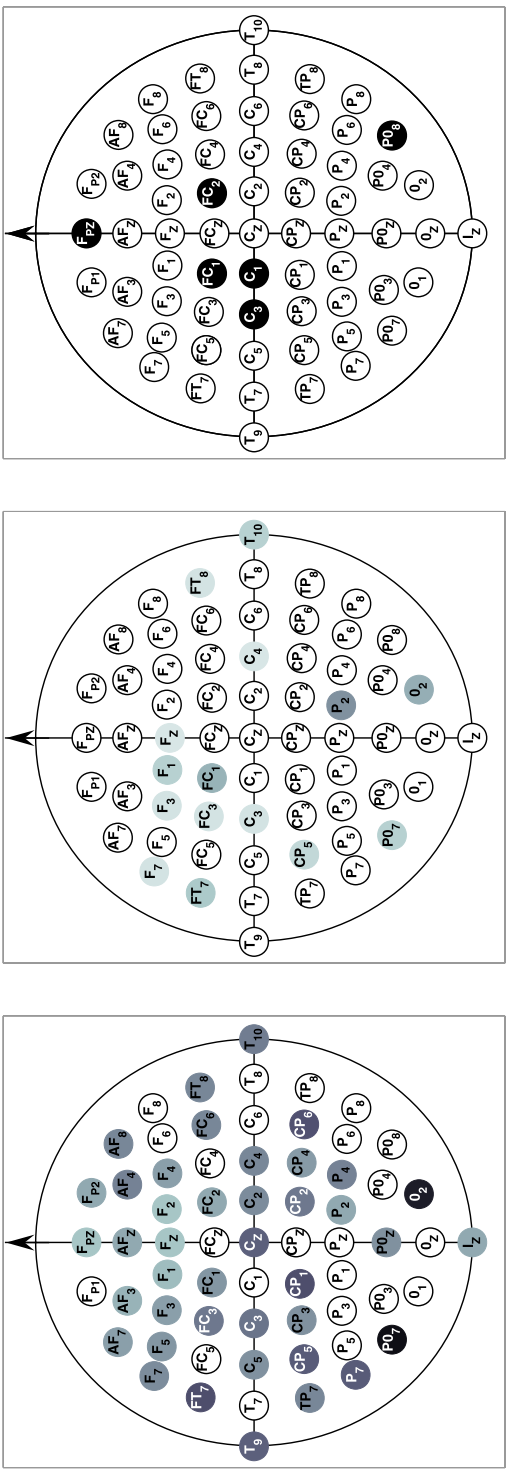


Fig. 7 Electrode median relevance for different SVMs, with channels and kernels randomly selected (P300 speller paradigm). The *darker* the color, the *higher* the relevance. *Electrodes in white with a black circle* are discarded (the relevance is exactly zero). For SVM_{CV}, *electrodes in black* correspond to the best channels identified using a cross-validation procedure (over the 10 repetitions, PO_8 and C_1 have been selected 3 times each)

Table 4 Average cross-validation score with standard deviations for Subject 1, for SVMs with different kernel learning strategies on the BCI dataset (CNV paradigm). The number of channels and kernels correspond to the predictor trained on the whole data set

Subject 1	Error rate (%)	# Channels	# Kernels	Time (s)
LDA	25.0 ± 1.2	C_Z	21	–
SVM	21.0 ± 1.0	64	1344	0.3
$CKL_{1/2}$	22.0 ± 1.0	50	988	20.7
CKL_1	23.0 ± 1.3	9	37	6.24
MKL	24.0 ± 1.5	29	58	23.1

Table 5 Average cross-validation score with standard deviations for Subject 2, for SVMs with different kernel learning strategies on the BCI dataset (CNV paradigm). The number of channels and kernels correspond to the predictor trained on the whole data set

Subject 2	Error rate (%)	# Channels	# Kernels	Time (s)
LDA	36.5 ± 0.9	C_Z	21	–
SVM	29.0 ± 1.3	64	1344	0.4
$CKL_{1/2}$	27.0 ± 1.2	44	800	16.7
CKL_1	23.0 ± 1.1	6	35	8.6
MKL	33.0 ± 1.3	51	112	20.0

This procedure is slightly biased, but since all the methods share this bias, the comparison should be fair. Considering the high variability between folds, we did not go through a thorough double cross-validation procedure. The reported standard deviations are likely be irrepresentative of the variability with respect to changes in the training set, due to the known bias of the variance estimators in K-fold cross-validation (Bengio and Grandvalet 2004).

Tables 4 and 5 reports the average performances for $CKL_{1/2}$, CKL_1 and MKL in terms of accuracy, channel and kernel selection, and training time. The accuracy achieved by a SVM, trained with the mean of the 1344 kernels, is also reported.

Concerning Subject 1, all SVMs perform slightly better than LDA. In this experiment, $CKL_{1/2}$ is much less sparse, in the number of kernels and channels, than MKL or CKL_1. The latter only retains 9 channels for classifying.

For Subject 2, both versions of CKL considerably improve upon LDA. Although $CKL_{1/2}$ selects most of the kernels, it is sparser than MKL in terms of groups. CKL_1, with only 6 channels achieves the lowest error rate.

With regard to training times, the overhead compared to SVMs is comparable to the previous experiment. MKL and $CKL_{1/2}$ require approximately the same time, and CKL_1, which provides very sparse results is about twice faster.

Results concerning interpretation are obtained with the whole dataset. Figure 8 shows the relevance of the electrodes, for both subject, as computed in (13) for the P300 speller problem. The three versions of CKL highlight the central region of the brain. However, CKL_1 discards most peripheric channels, whereas $CKL_{1/2}$ and MKL locate numerous relevant electrodes out of the central area. For the first subject, C_Z is estimated to be relevant by all methods. The results for the second subject are somewhat puzzling, since the contribution of C_Z is much lower than the one of F_Z. This shift may be due to an inappropriate positioning of the measurement device on the scalp.

5.2.3 Sanity check for channel selection

Here also, additional experiments are carried out to support the channel and kernel selection given by CKL, using the scheme described in Sect. 5.1.3. We consider two random draws per subject, that correspond, in terms of number of kernels and channels, to the solutions

Fig. 8 Electrode relevance for Subject 1 (*top*) and Subject 2 (*bottom*), for MKL, $\mathrm{CKL}_{1/2}$ and CKL_1 (CNV paradigm). The *darker* the color, the *higher* the relevance. *Electrodes in white with a black circle* are discarded (the relevance is exactly zero)

Table 6 Average cross-validation score with standard deviations for Subjects 1 and 2, for SVMs (CNV paradigm). SVM_{CV} selects the best channel using a cross-validation procedure, while SVM_x randomly selects a subset of x channels. The results reported for SVM_x are averaged over 10 repetitions

Algorithms		Error rate (%)	# Channels	# Kernels
Subject 1	SVM_{50}	29.1 ± 1.0	50	988
	SVM_9	37.9 ± 1.1	9	37
	SVM_{CV}	25.5 ± 1.2	C_2	21
Subject 2	SVM_{44}	31.2 ± 1.1	44	800
	SVM_6	36.2 ± 0.9	6	35
	SVM_{CV}	27.5 ± 0.7	FC_1	21

produced by $\mathrm{CKL}_{1/2}$ and CKL_1. This process is repeated 10 times. Table 6 summarizes the performances for these SVMs, as for a SVM trained with the channel that reaches the highest cross-validation score. Figure 9 displays the electrodes used for each method.

Concerning Subject 1, the first two versions of SVMs perform badly, especially SVM_9 where C_Z was chosen only once and CP_Z only twice over the 10 repetitions. The error rate for SVM_{CV} is comparable to the one of LDA, and it selects C_2, which is relevant in all versions of CKL. The error rate of SVM_{CV} is slightly greater than the one of $\mathrm{CKL}_{1/2}$ or CKL_1.

For Subject 2, SVM_{CV} fails compared to CKL_1, but reaches the performance of $\mathrm{CKL}_{1/2}$ with the "outsider" FC_1. SVMs with randomly selected kernels behave poorly again, with regard to CKL.

6 Conclusion

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

We provide here 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-norms, 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 kernel expansion to minimize the hinge loss. 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.

Appendix A: Detailed derivation of Problem (7)

We rewrite Problem (6) by applying successively two changes of variable. We first note that, when $\sigma_{1,\ell}$ or $\sigma_{2,m}$ is null, then the optimal f_m is also null. Hence, we may apply

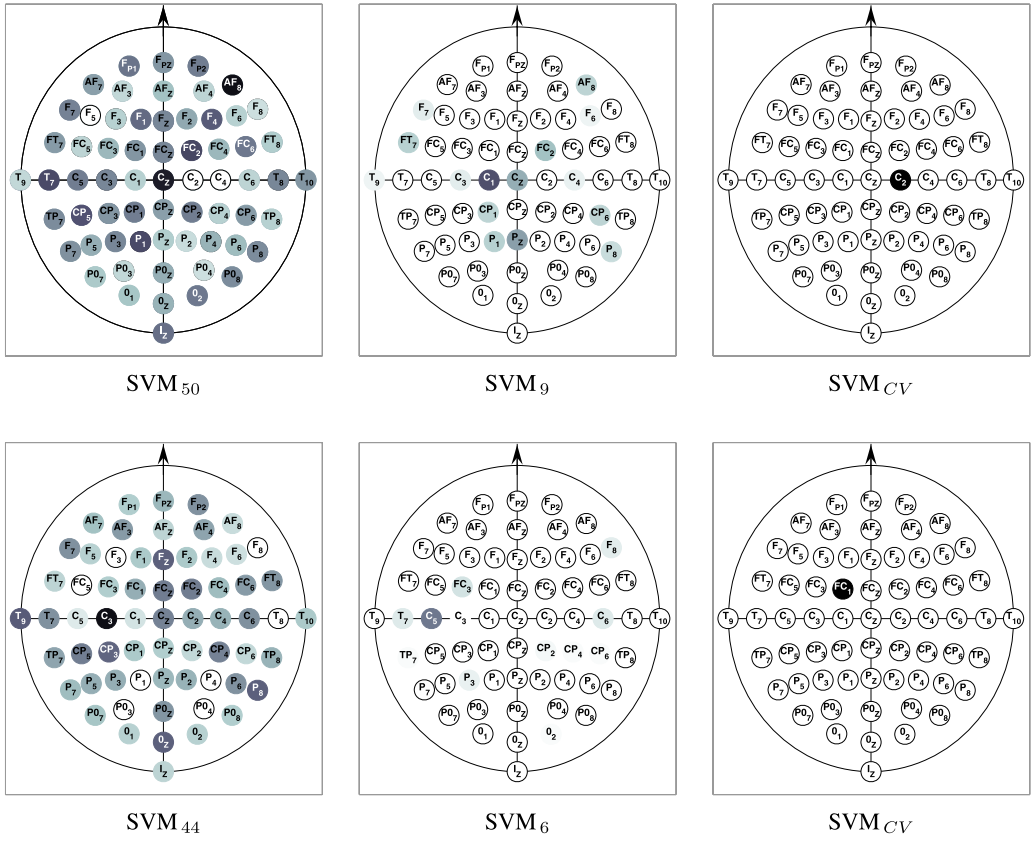


Fig. 9 Electrode median relevance for Subject 1 (*top*) and Subject 2 (*bottom*), for different SVMs, with channels and kernels randomly selected over 10 repetitions (P300 speller paradigm). The *darker* the color, the *higher* the relevance. *Electrodes in white with a black circle* are discarded (the relevance is exactly zero). For SVM_{CV}, *electrodes in black* correspond to the best channels identified using a cross-validation procedure

$f_m \leftarrow \sigma_{1,\ell}\sigma_{2,m} f_m$ since this transformation is one-to-one provided $\sigma_{1,\ell} \neq 0$ and $\sigma_{2,m} \neq 0$. We then follow with, $\sigma_{1,\ell} \leftarrow \sigma_{1,\ell}^{2/p}$, $\sigma_{2,m} \leftarrow \sigma_{2,m}^{2/q}$; this yields:

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

then, we proceed to another change of variable, that is, $\sigma_m = \sigma_{1,\ell}^{p}\sigma_{2,m}^{q}$, and Problem (6) is equivalent to the following optimization problem in $f_1,\ldots,f_M, b, \boldsymbol{\xi}, \boldsymbol{\sigma}_1, \boldsymbol{\sigma}$:

$$\begin{cases} \min\limits_{\substack{f_1,\ldots,f_M \\ b,\boldsymbol{\xi},\boldsymbol{\sigma}_1,\boldsymbol{\sigma}}} & \frac{1}{2}\sum\limits_{m}\frac{1}{\sigma_m}\|f_m\|_{\mathcal{H}_m}^2 + C\sum\limits_i \xi_i & \text{(14a)} \\ \text{s.t.} & y_i\Big(\sum\limits_m f_m(\boldsymbol{x}_i) + b\Big) \geq 1-\xi_i, \qquad \xi_i \geq 0 \quad i=1,\ldots,n & \text{(14b)} \\ & \sum\limits_\ell d_\ell \sigma_{1,\ell} \leq 1, \qquad \sigma_{1,\ell} \geq 0 \quad \ell = 1,\ldots,L & \text{(14c)} \\ & \sum\limits_\ell \sigma_{1,\ell}^{-p/q}\sum\limits_{m\in\mathcal{G}_\ell}\sigma_m^{1/q} \leq 1, \qquad \sigma_m \geq 0 \quad m=1,\ldots,M. & \text{(14d)} \end{cases}$$

We now use the fact that, in the formulation above, the first-order necessary optimality conditions establish a functional link between $\boldsymbol{\sigma}_1$ and $\boldsymbol{\sigma}$. This link is derived from the Karush-Kuhn-Tucker necessary optimality conditions of Problem (14), computed from the associated Lagrange function $\mathcal{L}$:

$$\frac{\partial \mathcal{L}}{\partial \sigma_{1,\ell}} = \lambda_1 d_\ell - \lambda_2 \frac{p}{q}\sigma_{1,\ell}^{-(p+q)/q}\sum_{m\in\mathcal{G}_\ell}\sigma_m^{1/q} - \eta_{1,\ell}, \tag{15}$$

$$\frac{\partial \mathcal{L}}{\partial \sigma_m} = -\frac{\|f_m\|_{\mathcal{H}_m}^2}{\sigma_m^2} + \lambda_2 \frac{1}{q}\sigma_{1,\ell}^{-p/q}\sigma_m^{(1-q)/q} - \eta_{2,m}, \tag{16}$$

where λ_1 and λ_2 are the Lagrange parameters related to the norm constraints (14c) and (14d) respectively while $\eta_{1,\ell}$ and $\eta_{2,m}$ are associated to the positivity of $\sigma_{1,\ell}$ and σ_m.

From (16), one sees that, except for the trivial case where $\sum_m \|f_m\|_{\mathcal{H}_m}^2 = 0$, $\lambda_2 \neq 0$ at the optimum. Then, one easily derives from (15) that, at the optimum, $q\lambda_1 = p\lambda_2$.

Finally, combining (15) and the ones stating that the norm constraints (14c) and (14d) are saturated, after some algebra, we get that the optimal $(\boldsymbol{\sigma}^\star, \boldsymbol{\sigma}_1^\star)$ satisfies

$$\sum_\ell \sigma_{1,\ell}^{\star\,-p/q}\sum_{m\in\mathcal{G}_\ell}\sigma_m^{\star\,1/q} = \sum_\ell d_\ell^{p/(p+q)}\Big(\sum_{m\in\mathcal{G}_\ell}\sigma_m^{\star\,1/q}\Big)^{q/(p+q)}.$$

Plugging this optimality condition into Problem (14), we get Problem (7).

Appendix B: Proof of Proposition 2

The proof of Proposition 2 can be decomposed into three steps. We first derive the optimality conditions for σ_m, from which we express a relationship between σ_m and f_m at stationary points. Since the stationary points are local minima of the convex objective function, the minima of (7) are minima of (8). Finally, this expression in f_m is plugged in the original objective function.

The Lagrangian associated to Problem (7) is

$$\mathcal{L} = \frac{1}{2}\sum_m \frac{1}{\sigma_m}\|f_m\|^2_{\mathcal{H}_m} + C\sum_i \xi_i - \sum_i \alpha_i \left[y_i \left(\sum_m f_m(\boldsymbol{x}_i) + b \right) + \xi_i - 1 \right]$$
$$- \sum_i \eta_i \xi_i + \lambda \left[\sum_\ell \left(d_\ell^p \left(\sum_{m \in \mathcal{G}_\ell} \sigma_m^{1/q} \right)^q \right)^{1/(p+q)} - 1 \right] - \sum_m \mu_m \sigma_m,$$

where η_i and μ_m are the Lagrange parameters respectively related to the positivity of η_i and σ_m, and λ is the Lagrange parameter pertaining to the norm constraint (7c). The first-order necessary optimality condition $\partial\mathcal{L}/\partial\sigma_m = 0$ reads

$$-\frac{\|f_m\|^2_{\mathcal{H}_m}}{2\sigma_m^2} + \frac{\lambda}{p+q}\sigma_m^{(1-q)/q} \left(d_\ell^{-1} \sum_{m \in \mathcal{G}_\ell} \sigma_m^{1/q} \right)^{-p/(p+q)} - \mu_m = 0.$$

As all the Lagrange parameters are non-negative, except for the trivial case where, for all m, $\sigma_m = 0$, the Lagrange parameter λ is non-zero. We then have that, either

$$\begin{aligned} &\sigma_m = 0 \quad \text{and} \quad \|f_m\|_{\mathcal{H}_m} = 0, \quad \text{either} \\ &\sigma_m = \left(\frac{p+q}{2\lambda} \right)^{q/(q+1)} \|f_m\|^{2q/(q+1)}_{\mathcal{H}_m} \left(d_\ell^{-1} \sum_{m \in \mathcal{G}_\ell} \sigma_m^{1/q} \right)^{pq/(p+q)(q+1)}. \end{aligned} \tag{17}$$

To uncover the relationship of σ_m with $\|f_m\|_{\mathcal{H}_m}$ at the stationnary points, we start from (17):

$$\begin{aligned} \sigma_m^{1/q} &= \left(\frac{p+q}{2\lambda} \right)^{1/(q+1)} \|f_m\|^{2/q+1}_{\mathcal{H}_m} \left(d_\ell^{-1} \sum_{m \in \mathcal{G}_\ell} \sigma_m^{1/q} \right)^{p/(p+q)(q+1)}, \\ \left(\sum_{m \in \mathcal{G}_\ell} \sigma_m^{1/q} \right)^{q+1} &= \frac{p+q}{2\lambda} \left(\sum_{m \in \mathcal{G}_\ell} \|f_m\|^{2/q+1}_{\mathcal{H}_m} \right)^{q+1} \left(d_\ell^{-1} \sum_{m \in \mathcal{G}_\ell} \sigma_m^{1/q} \right)^{p/(p+q)}, \\ \left(\sum_{m \in \mathcal{G}_\ell} \sigma_m^{1/q} \right)^{q} &= \left[\frac{p+q}{2\lambda} d_\ell^{-p/(p+q)} \left(\sum_{m \in \mathcal{G}_\ell} \|f_m\|^{2/q+1}_{\mathcal{H}_m} \right)^{(q+1)} \right]^{(p+q)/(p+q+1)}. \end{aligned} \tag{18}$$

As $\lambda \neq 0$, the constraint (7c) is saturated. We use this fact to get rid of λ. Denoting $s_\ell = \sum_{m \in \mathcal{G}_\ell} \|f_m\|^{2/q+1}_{\mathcal{H}_m}$, and summing both sides of (19) over ℓ, we get

$$\frac{2\lambda}{p+q} = \left(\sum_\ell d_\ell^{p/(p+q+1)} s_\ell^{(q+1)/(p+q+1)} \right)^{p+q+1}. \tag{19}$$

Finally, plugging (19) and (19) in (17), we obtain the relationship

$$\sigma_m = \|f_m\|_{\mathcal{H}_m}^{2q/(q+1)} \left(d_\ell^{-1} s_\ell\right)^{p/(p+q+1)} \left(\sum_\ell d_\ell^{p/(p+q+1)} s_\ell^{(q+1)/(p+q+1)}\right)^{-(p+q)}.$$

Note that this equation also holds for $\sigma_m = 0$. It is now sufficient to replace σ_m by this expression in the objective function of Problem (7) to obtain the claimed equivalence with Problem (8) in Proposition 2.

Appendix C: Overview of notations and symbols

Data

$\mathcal{X}$	observation domain
n	number of training examples
i, j	indices, often running over $\{1, \ldots, n\}$
$\boldsymbol{x}_i$	observations in $\mathcal{X}$
y_i	class labels in $\{-1, 1\}$

Kernels

$\mathcal{H}$	feature space
Φ	feature map, $\Phi : \mathcal{X} \to \mathcal{H}$
K	reproducing kernel $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$
$\langle \cdot, \cdot \rangle_{\mathcal{H}}$	scalar product in $\mathcal{H}$; if $f(\cdot) = \sum_{i=1}^{\infty} \alpha_i K(\boldsymbol{x}_i, \cdot)$ and $g(\cdot) = \sum_{j=1}^{\infty} \alpha_j K(\boldsymbol{x}_j, \cdot)$, then $\langle f, g \rangle_{\mathcal{H}} = \sum_{i=1}^{\infty} \sum_{j=1}^{\infty} \alpha_i \alpha_j K(\boldsymbol{x}_i, \boldsymbol{x}_j)$
$\|\cdot\|_{\mathcal{H}}$	norm induced by the scalar product in $\mathcal{H}$, $\|f\|_{\mathcal{H}} = \sqrt{\langle f, f \rangle_{\mathcal{H}}}$
$\mathbf{K}$	kernel matrix $\mathbf{K}_{ij} = K(\boldsymbol{x}_i, \boldsymbol{x}_j)$
α_i	expansion coefficients or Lagrange multipliers

SVM-related

f	function, from $\mathcal{X}$ to $\mathbb{R}$
b	constant offset (or threshold) in $\mathbb{R}$
ξ_i	slack variables in $\mathbb{R}$ (constrained to be non-negative)
$\boldsymbol{\xi}$	vector of all slack variables in $\mathbb{R}^n$
C	regularization parameter in front of the empirical risk term
η_i	Lagrange multiplier related to the positivity of ξ_i

MKL and CKL-related

$\mathcal{K}$	set of admissible kernels
M	number of kernels
m	kernel index, often running over $\{1, \ldots, M\}$
L	number of groups for CKL
ℓ	group index, running over $\{1, \ldots, L\}$
$\mathcal{G}_\ell$	set of indices for group ℓ, $\mathcal{G}_\ell \subseteq \{1, \ldots, M\}$
d_ℓ	cardinality of $\mathcal{G}_\ell$
$\mathcal{H}_m$	mth feature space
K_m	reproducing kernel for the mth feature space
σ_m	weight of the mth kernel in the kernel combination

$\boldsymbol{\sigma}$ vector of kernel weights in $\mathbb{R}^M$
$K_{\boldsymbol{\sigma}}$ equivalent kernel $K_{\boldsymbol{\sigma}} = \sum_{m=1}^{M} \sigma_m K_m$
$\sigma_{1,\ell}$ weight of the ℓth group in the kernel combination
$\boldsymbol{\sigma}_1$ vector of group weights in $\mathbb{R}^L$
$\sigma_{2,m}$ weight of the mth kernel in the group-kernel combination
$\boldsymbol{\sigma}_2$ vector of kernel weights in $\mathbb{R}^M$

Miscellaneous

$\mathbb{R}$ set of reals
$\boldsymbol{A}^\top$ transposed of matrix $\boldsymbol{A}$ (ditto for vectors)
sign sign function, from $\mathbb{R}$ to $\{-1, 1\}$, $\operatorname{sign}(x) = \begin{cases} -1 & \text{if } x < 0 \\ 0 & \text{if } x \geq 0 \end{cases}$
$\ell_{(p,q)}$ mixed (p, q)-norm, the $\ell_{(p,q)}$ norm of $\boldsymbol{\sigma}$ is $(\sum_\ell (\sum_{m \in \mathcal{G}_\ell} \sigma_m^p)^{q/p})^{1/q}$

References

Argyriou, A., Hauser, R., Micchelli, C. A., & Pontil, M. (2006). A dc-programming algorithm for kernel selection. In W. W. Cohen & A. Moore (Eds.), *Proceedings of the twenty-third international conference on machine learning* (pp. 41–48). New York: ACM.

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

Bach, F. (2009). Exploring large feature spaces with hierarchical multiple kernel learning. In *Advances in neural information processing systems 21*. Cambridge: MIT Press.

Bach, F. R., Lanckriet, G. R. G., & Jordan, M. I. (2004). Multiple kernel learning, conic duality, and the SMO algorithm. In *ACM international conference proceeding series*. *Proceedings of the 21th annual international conference on machine learning (ICML 2004)* (pp. 41–48). New York: ACM.

Bengio, Y., & Grandvalet, Y. (2004). No unbiased estimator of the variance of K-fold cross-validation. *Journal of Machine Learning Research (JMLR)*, *5*, 1089–1105.

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 Transactions on Biomedical Engineering*, *51*(6), 1044–1051.

Bousquet, O., & Elisseeff, A. (2002). Stability and generalization. *Journal of Machine Learning Research*, *2*, 499–526.

Bousquet, O., & Herrmann, D. J. L. (2003). On the complexity of learning the kernel matrix. In S. Becker, S. Thrun, & K. Obermayer (Eds.), *Advances in neural information processing systems 15* (pp. 399–406). Cambridge: MIT Press.

Breiman, L. (1996). Heuristics of instability and stabilization in model selection. *Annals of Statistics*, *24*(6), 2350–2383.

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* (pp. 204–210). Cambridge: 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* (pp. 367–373). Cambridge: MIT Press.

Farwell, A., & Donchin, E. (1998). Talking off the top of your head: toward a mental prosthesis utilizing event-related brain potentials. *Electroencephalography and Clinical Neurophysiology*, *70*(6), 510–523.

Garipelli, G., Chavarriaga, R., & del Millán, J. R. (2009). Fast recognition of anticipation related potentials. *IEEE Transactions on Biomedical Engineering*, *56*(4), 1257–1260.

Grandvalet, Y., & Canu, S. (1999). Outcomes of the equivalence of adaptive ridge with least absolute shrinkage. In M. S. Kearns, S. A. Solla, & D. A. Cohn (Eds.), *Advances in neural information processing systems 11 (NIPS 1998)* (pp. 445–451). Cambridge: 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* (pp. 569–576). Cambridge: MIT Press.

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

Kowalski, M., & Torrésani, B. (2008). Sparsity and persistence: mixed norms provide simple signals models with dependent coefficients. *Signal, Image and Video Processing*, 1863–1703.

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.

Nikolova, M. (2000). Local strong homogeneity of a regularized estimator. *SIAM Journal on Applied Mathematics*, *61*(2), 633–658.

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., & Guigue, V. (2008). BCI competition 3: Dataset 2—ensemble of SVM for BCI P300 speller. *IEEE Transactions on Biomedical Engineering*, *55*(3), 1147–1154.

Rakotomamonjy, A., Bach, F. R., Canu, S., & Grandvalet, Y. (2008). SimpleMKL. *Journal of Machine Learning Research (JMLR)*, *9*, 2491–2521.

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

Srebro, N., & Ben-David, S. (2006). Learning bounds for support vector machines with learned kernels. In G. Lugosi & H.-U. Simon (Eds.), *19th annual conference on learning theory* (Vol. 4005, pp. 169–183). Berlin: Springer.

Szafranski, M., Grandvalet, Y., & Morizet-Mahoudeaux, P. (2008a). Hierarchical penalization. In J. C. Platt, D. Koller, Y. Singer, & S. Roweis (Eds.), *Advances in neural information processing systems 20* (pp. 1457–1464). Cambridge: MIT Press.

Szafranski, M., Grandvalet, Y., & Rakotomamonjy, A. (2008b). Composite kernel learning. In A. McCallum & S. Roweis (Eds.), *Proceedings of the 25th annual international conference on machine learning (ICML 2008)* (pp. 1040–1047). Eastbourne: Omnipress.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society, Series B*, *58*(1), 267–288.

Vapnik, V. N. (1995). *The nature of statistical learning theory*. New York: Springer.

Walter, W. G., Cooper, R., Aldridge, V. J., McCallum, W. C., & Winter, A. L. (1964). Contingent negative variation: An electric sign of sensorimotor association and expectancy in the human brain. *Nature*, *203*, 380–384.

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* (pp. 668–674). Cambridge: MIT Press.

Xu, Z., Jin, R., King, I., & Lyu, M. (2009). An extended level method for efficient multiple kernel learning. In D. Koller, D. Schuurmans, Y. Bengio, & L. Bottou (Eds.), *Advances in neural information processing systems 21* (pp. 1825–1832). Cambridge: 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. (2009). The composite absolute penalties family for grouped and hierarchical variable selection. *Annals of Statistics*, *37*(6A), 3468–3497.