Multiple Indefinite Kernel Learning with Mixed Norm Regularization

Matthieu Kowalski [1] MATTHIEU.KOWALSKI@CMI.UNIV-MRS.FR
Marie Szafranski [2] MARIE.SZAFRANSKI@LIF.UNIV-MRS.FR
Liva Ralaivola [2] LIVA.RALAIVOLA@LIF.UNIV-MRS.FR
[1] LATP – UMR CNRS 6166, [2] LIF – UMR CNRS 6632, Université de Provence, 13453 Marseille Cedex 13, France

Abstract

We address the problem of learning classifiers using several kernel functions. On the contrary to many contributions in the field of learning from different sources of information using kernels, we here do not assume that the kernels used are positive definite. The learning problem that we are interested in involves a misclassification loss term and a regularization term that is expressed by means of a mixed norm. The use of a mixed norm allows us to enforce some sparsity structure, a particular case of which is, for instance, the Group Lasso. We solve the convex problem by employing proximal minimization algorithms, which can be viewed as refined versions of gradient descent procedures capable of naturally dealing with nondifferentiability. A numerical simulation on a UCI dataset shows the modularity of our approach.

1. Introduction

Lately, there has been much attention paid to the problem of learning from multiple sources. This amount of work has been mainly spurred by new problems stemming from, e.g., bioinformatics or multimedia processing. The main line of approaches for this situation of learning is that of Multiple Kernel Learning (MKL) first initiated by Lanckriet et al. (2004), where the information provided by each data source at hand is encoded by means of a *Mercer kernel*.

We address the problem of learning multiple indefinite kernel classifiers, where the kernels used to learn are not necessarily Mercer kernels. Our main motivation is that if Mercer kernels exhibit many interesting mathematical properties that make them particularly suitable to work with, encoding knowledge in terms of a positive definite kernels is not always possible. The idea of making use of several kernels is to take advantage of many sources of information, hoping a reliable algorithm can single out the useful ones.

Being able to identify the relevant information in terms of data or kernels is also very important. To achieve this task, we propose a formulation of the learning problem that makes use of mixed norms as a regularizing tool. Mixed norms allow us to impose some kind of structure on the data and the kernels that we use, and we enforce our objective of automatically selecting the relevant information by using nondifferentiable – but still convex – mixed norms.

Another way of viewing our approach is that of formalizing the problem of learning kernel classifiers as learning a representation of data based on (data-dependent) dictionaries. This is a common approach in the signal processing community, where efficient algorithms exist to handle nondifferentiable minimization problems as those we consider. We note that learning with several kernels is also closely related to the popular idea from signal processing to find representations of data from unions of dictionaries.

The contributions of the present work are: a setting to learn multiple kernel classifiers with mixed norm regularization, a data-dependent bound on the generalization ability of the classifiers learned, a learning algorithm that instantiates the idea of proximal optimization methods, which provides a framework to build refined versions of gradient descent algorithms capable of dealing with nondifferentiability.

The paper is organized as follows. Section 2 introduces the setting of learning multiple kernel classifiers with mixed norm regularization; insights as to why classifiers learned from the proposed setting should generalize well are given. Section 3 recalls the proximal optimization framework and derives the minimization algorithm to solve our learning problem. In Section 4, numerical simulations carried out on a dataset from the UCI repository show how the mixed norms can indeed induce desired sparsity. Section 5 discusses how our approach is related to other MKL strategies.

Appearing in *Proceedings of the 26^{th} International Conference on Machine Learning*, Montreal, Canada, 2009. Copyright 2009 by the author(s)/owner(s).

2. MIKL and Mixed Norms

2.1. Notational Conventions

We use the following notation. Bold letters will usually denote column vectors. Let $L \in \mathbb{N}$ and $M \in \mathbb{N}$; for a doubly indexed set of real coefficients $(\alpha_{\ell m})$, with $1 \leq \ell \leq L$ and $1 \leq m \leq M$, $\boldsymbol{\alpha}_{\bullet m}$ is the column vector $\boldsymbol{\alpha}_{\bullet m} = [\alpha_{1m} \cdots \alpha_{Lm}]^\top$, $\boldsymbol{\alpha}_{\ell \bullet}$ the column vector $\boldsymbol{\alpha}_{\ell \bullet} = [\alpha_{\ell 1} \cdots \alpha_{\ell M}]^\top$ and $\boldsymbol{\alpha}$ is the column vector $\boldsymbol{\alpha} = [\boldsymbol{\alpha}_{\bullet 1}^\top \cdots \boldsymbol{\alpha}_{\bullet M}^\top]^\top$.

$\mathbb{I}(p)$ is such that $\mathbb{I}(p) = 1$ if p is true and $\mathbb{I}(p) = 0$ otherwise. The hinge function is denoted as $|u|_+ = (\max(u, 0))$ and for any real vector $\mathbf{u}$, $[\![\mathbf{u}]\!]_+^2$ stands for $[\![\mathbf{u}]\!]_+^2 = \sum_k |u_k|_+^2$.

2.2. Setting

We focus on the problem of supervised binary classification. We are interested in learning classifiers from a training sample $S = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$ of n labeled pairs $(\mathbf{x}_i, y_i)$ from the product space $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$, where $\mathcal{Y} = \{-1, +1\}$ is the set of labels and $\mathcal{X}$ is the *input space*. These pairs are the realizations of n independent copies $(X_1, Y_1), \ldots, (X_n, Y_n)$ of a random labeled variable (X, Y) distributed according to an unknown and fixed distribution D on $\mathcal{Z}$. With a slight abuse of notation S will also denote the random sample $\{(X_i, Y_i)\}_{i=1}^n$.

The classifiers that we consider are *multiple kernel classifiers* where the kernels used are not necessarily positive definite kernels: we consider the largest possible definition and a kernel k is merely an element of $\mathbb{R}^{\mathcal{X} \times \mathcal{X}}$. Throughout the paper, we consider that we have at hand a set $\mathcal{K} = \{k_1, \ldots, k_\tau\}$ of τ kernels and multiple kernel classifiers are the signed versions of functions f from the sample-dependent family $\mathcal{F}_S$ defined for a training set $S = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$ as [1]

$$\mathcal{F}_S = \left\{ \mathbf{x} \mapsto \sum_{i,t=1}^{n,\tau} \alpha_{it} k_t(\mathbf{x}_i, \mathbf{x}) : \boldsymbol{\alpha} \in \mathbb{R}^{n\tau}, k_t \in \mathcal{K} \right\}. \tag{1}$$

Thus, the output predicted for $\mathbf{x}$ by $f \in \mathcal{F}_S$ is $\text{sign}(f(\mathbf{x}))$.

With this setting, learning a classifier from S comes down to the problem of finding a vector $\boldsymbol{\alpha}$ that entails a low generalization error. To this end, we propose to set $\boldsymbol{\alpha}$ as the solution of the following penalized optimization problem:

$$\min_{\boldsymbol{\alpha} \in \mathbb{R}^{n\kappa}} \sum_{i=1}^n \left|1 - y_i \mathbf{k}_i^\top \boldsymbol{\alpha}\right|_+^2 + \frac{\lambda}{q} \|\boldsymbol{\alpha}\|_{pq;r}^q \tag{2}$$

for $\lambda > 0$, $p, q \in \{1, 2\}$ and $r \in \{1, 2\}$. Here, $\boldsymbol{\alpha}$ is the column vector associated with the doubly indexed set of coefficients $(\alpha_{it})_{\substack{1 \leq i \leq n \\ 1 \leq t \leq \tau}}$ and $\mathbf{k}_i$ is the ith column of

$$K = [K_1 \ldots K_\tau]^\top \in \mathbb{R}^{n\tau \times n}, \tag{3}$$

where K_t is the matrix $K_t = (k_t(\mathbf{x}_i, \mathbf{x}_j))_{1 \leq i,j \leq n}$ associated with kernel k_t. Mixed norm $\| \cdot \|_{pq;1}$ is such that:

$$\|\boldsymbol{\alpha}\|_{pq;1} = \left[\sum_{i=1}^n \left[\sum_{t=1}^\tau |\alpha_{it}|^p \right]^{q/p} \right]^{1/q}, \tag{4}$$

and $\| \cdot \|_{pq;2}$ such that:

$$\|\boldsymbol{\alpha}\|_{pq;2} = \left[\sum_{t=1}^\tau \left[\sum_{i=1}^n |\alpha_{it}|^p \right]^{q/p} \right]^{1/q}, \tag{5}$$

(note that the order of summation has changed). As we discuss below, the choice for the values of p and q, if one is set to 1, induces different sparsity structures for the solution $\boldsymbol{\alpha}$ of (2).

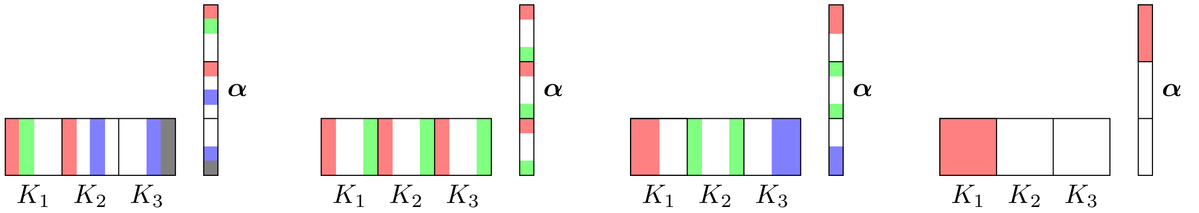
The left-hand side of objective function (2) is the squared hinge loss, which is used by the 2-norm Support Vector Machines (Boser et al., 1992; Cortes & Vapnik, 1995). This loss is differentiable everywhere, a feature that will render some parts of the optimization procedure easy to derive. In addition, it is straightforward to see that the loss part is convex with respect to $\boldsymbol{\alpha}$.

The second term is a regularization part, which allows us to control the capacity of the class of classifiers considered. For the set of values that we consider for p and q, namely $p, q \in \{1, 2\}$, the regularization part, i.e. the mixed norm to the qth is a convex function of $\boldsymbol{\alpha}$; the resulting objective function is henceforth convex in $\boldsymbol{\alpha}$, since $\lambda > 0$. Note however that this objective function becomes nondifferentiable as soon as p or q is equal to 1, which is the situation of interest to us since such a choice induces sparsity on the optimal $\boldsymbol{\alpha}$. This nondifferentiability is nicely handled by the proximal minimization algorithm that we derive.

2.3. Expected Sparsity Structure

The minimization problems (2) that we focus on will use mixed norms such that $p = 1$ or $q = 1$, whichever r. The reason why we retain this setting is because it may induce sparsity on the optimal $\boldsymbol{\alpha}$. Such sparsity is useful from two different points of view. On the one hand, it may help identify which of the different kernels used are indeed informative for the problem at hand by, e.g., setting all the coefficients of $\boldsymbol{\alpha}$ related to a specific kernel k_t to 0, in which case $\boldsymbol{\alpha}_{\bullet t} = 0$, or by setting all the coefficients of $\boldsymbol{\alpha}$ related to one $\mathbf{x}_i$ to 0, which corresponds to $\boldsymbol{\alpha}_{i\bullet} = 0$. On the other

[1] A bias term is taken into account by adding a constant kernel k_0 to $\mathcal{K}$ such that $k_0(\mathbf{x}, \mathbf{x}') = 1, \forall \mathbf{x}, \mathbf{x}' \in \mathcal{X}$.



(a) $\|\cdot\|_{12;1}$ (b) $\|\cdot\|_{21;1}$ (c) $\|\cdot\|_{12;2}$ (d) $\|\cdot\|_{21:2}$

Figure 1. Expected sparsity structure of $\boldsymbol{\alpha}$ for different norms $\|\cdot\|_{pq;r}$ used, where white squares in $\boldsymbol{\alpha}$ correspond to 0 coefficients. When $r = 1$ ((a) and (b)), the sparseness is defined with respect to the $\boldsymbol{\alpha}_{i\bullet}$ vectors, which are indicated with one color; the corresponding column of K that are used by the resulting function are shown using the same colors. When $r = 2$ ((c) and (d)), the sparseness is defined with respect to the kernels. See text for detailed explanations.

hand, sparseness, or more precisely, the use of a ℓ_1-like penalization (which is the case if p or q is equal to 1) is useful to draw generalization bounds as we show just below.

The structure of the sparsity of $\boldsymbol{\alpha}$, depends not only on which of p or q is equal to 1 but also on the value of r. We may summarize the expected pattern of sparsity as follows.

If $p = q = 1$ then $\|\boldsymbol{\alpha}\|_{pq;1} = \|\boldsymbol{\alpha}\|_{pq;2}$, for all $\boldsymbol{\alpha}$. A large number of coefficients α_{it} are expected to be 0, as with the Lasso (Tibshirani, 1996).

If $r = 1$, the sparseness is related to the $\mathbf{x}_i$'s. If $p = 1$ and $q = 2$, each $\mathbf{x}_i$ only uses a limited number of kernels, or in other words, the $\boldsymbol{\alpha}_{i\bullet}$'s are sparse (see Figure 1(a)). If $p = 2$ and $q = 1$, then we may expect several $\boldsymbol{\alpha}_{i\bullet}$ to be zero, meaning that the kernel expansion of the decision function is based on few training vectors $\mathbf{x}_i$ only; these vectors may be thought of as 'support vectors' (see Figure 1(b)).

If $r = 2$, the sparseness is related to the kernels used. If $p = 1$ and $q = 2$, then the vectors $\boldsymbol{\alpha}_{\bullet t}$ are expected to be sparse and only few $\mathbf{x}_i$'s are activated per kernel (see Figure 1(c)). If $p = 2$ and $q = 1$ then some kernels are expected to be discarded and not used in the decision function learned: some vectors $\boldsymbol{\alpha}_{\bullet t}$ are expected to be 0 (see Figure 1(d)).

For $r \in \{1, 2\}$, $\|\cdot\|_{12;r}$ is related to the Elitist-Lasso of Kowalski and Torrésani (2008) while $\|\cdot\|_{21;r}$ is related to the Group-Lasso of (Yuan & Lin, 2006).

2.4. A Data-Dependent Generalization Bound

Here, we give insights as to why a classifier learned by solving (2) may generalize and we provide a data-dependent bound on the generalization error for such a classifier. This bound relies on a recent and elegant result about the generalization ability of classifiers drawn from sample-dependent classes presented by Ben-David et al. (2008), a particular case of which focuses on a generalized notion of Rademacher complexity that we recall here.

Definition 1 (Ben-David et al. (2008))**.** The Rademacher complexity $R_n^*(\mathcal{F}_S)$ of a sample-dependent hypothesis class $\mathcal{F}_S$ is defined as

$$R_n^*(\mathcal{F}_S) = \sup_{S=\{(\mathbf{x}_i,y_i)\}_{i=1}^n} \mathbb{E}_{\boldsymbol{\sigma}} \left[\sup_{f\in\mathcal{F}_S} \frac{1}{n}\sum_{i=1}^n \sigma_i f(\mathbf{x}_i) \right],$$

where $\boldsymbol{\sigma}$ is a vector of n independent Rademacher variables, i.e. $\mathbb{P}(\sigma_i = 1) = \mathbb{P}(\sigma_i = -1) = \frac{1}{2}, i = 1, \dots, n$.

Ben-David et al. (2008) provide the following result.

Theorem 1. *Assume that $\forall S, S' \in \cup_{i=1}^{\infty}\mathcal{Z}^i$, $S \subseteq S' \Rightarrow \mathcal{F}_S \subseteq \mathcal{F}_{S'}$. For all distributions D on $\mathcal{Z}$, $\forall n > 0$, then with probability at least $1-\delta$ over the random draw of $S = \{(X_i, Y_i)\}_{i=1}^n$ the following holds: $\forall f \in \mathcal{F}_S$,*

$$\mathbb{P}_D(Yf(X) \leq 0) \leq \hat{\mathbb{E}}_n\phi(Yf(X)) + 16R_{2n}^*(\mathcal{F}_S) + \sqrt{\frac{\log 1/\delta}{2n}},$$

where $\phi(\gamma) = \min(|1-\gamma|_+^2, 1)$ is the clipped *squared hinge loss and $\hat{\mathbb{E}}_n\phi(Yf(X)) = \frac{1}{n}\sum_{i=1}^n \phi(Yf(X_i))$.*

Note that we have slightly modified the result of Ben-David et al. (2008) so that it takes into account the squared hinge loss. To do that, we have used a structural result on the Rademacher complexity of classes of composite functions given by Bartlett and Mendelson (2002).

Let us now consider that p, q, r and τ are fixed. We define the set $\mathcal{A}_n(\kappa) \subseteq \mathbb{R}^{n\tau}$ as

$$\mathcal{A}_n(\kappa) = \{\boldsymbol{\alpha} : \boldsymbol{\alpha} \in \mathbb{R}^{n\tau}, \|\boldsymbol{\alpha}\|_{pq;r} \leq \kappa\}$$

and the sample-dependent hypothesis class $\mathcal{F}_S(\kappa)$ as, for $S = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$

$$\mathcal{F}_S(\kappa) = \left\{ \mathbf{x} \mapsto \sum_{i,t=1}^{n,\tau} \alpha_{it} k_t(\mathbf{x}_i, \mathbf{x}), \boldsymbol{\alpha} \in \mathcal{A}_n(\kappa) \right\}.$$

Theorem 1 applies as soon as an upper bound on the sample-dependent Rademacher complexity of the hypothesis class under consideration can be computed. A bound on $R_{2n}^*(\mathcal{F}_S(\kappa))$ therefore suffices to bound the generalization error of the classifier learned through (2). The following proposition provides such a bound.

Proposition 1. $\forall \kappa > 0, \forall n \in \mathbb{N}$,

$$R^*_{2n}(\mathcal{F}_S(\kappa)) \le \kappa \sup_{S=\{(\mathbf{x}_i,y_i)\}_{i=1}^{2n}} \mathbb{E}_{\boldsymbol{\sigma}} \frac{1}{2n} \left\|K_S\boldsymbol{\sigma}\right\|_{p'q';r},$$

where K_S is defined as in (3) with respect to the $2n$-sample S and $\frac{1}{p}+\frac{1}{p'}=1$ and $\frac{1}{q}+\frac{1}{q'}=1$.

Proof. Assume that S is a fixed sample of size $2n$ and σ a fixed vector of $\{-1,+1\}^{2n}$. Then

$$\begin{aligned}\sup_{f\in\mathcal{F}_S(\kappa)} \sum_{i=1}^{2n}\sigma_i f(\mathbf{x}_i) &= \sup_{\boldsymbol{\alpha}\in\mathcal{A}_{2n}(\kappa)} \sum_{j=1}^{2n}\sum_{t=1}^{\tau}\alpha_{jt}\sum_{i=1}^{2n}\sigma_i k_t(\mathbf{x}_i,\mathbf{x}_j)\\ &= \sup_{\boldsymbol{\alpha}\in\mathcal{A}_{2n}(\kappa)} \boldsymbol{\alpha}^\top K\boldsymbol{\sigma} = \sup_{\boldsymbol{\alpha}:\|\boldsymbol{\alpha}\|_{pq;r}\le\kappa/n} \boldsymbol{\alpha}^\top(K\boldsymbol{\sigma})\\ &\le \kappa\|K\boldsymbol{\sigma}\|_{p'q';r},\end{aligned}$$

where Holder's inequality has been used twice to get the last line. Taking the expectation with respect to $\boldsymbol{\sigma}$ and then the supremum over S ends the proof. □

This allows us to state the following theorem.

Theorem 2. *For all distributions D on $\mathcal{Z}$, $\forall n > 0$, $\forall \kappa > 0$, with probability at least $1-\delta$ over the random draw of $S = \{(X_i, Y_i)\}_{i=1}^n$, $\forall f \in \mathcal{F}_S(\kappa)$,*

$$\begin{aligned}\mathbb{P}_D(Yf(X)\le 0) &\le \hat{\mathbb{E}}_n\phi(Yf(X))\\ &+ 16\kappa\sup_{S'_{2n}}\mathbb{E}_{\boldsymbol{\sigma}}\frac{1}{2n}\left\|K_{S'_{2n}}\boldsymbol{\sigma}\right\|_{p'q';r} + \sqrt{\frac{\log 1/\delta}{2n}},\end{aligned}$$

where S'_{2n} denotes a sample of size $2n$.

Proof. Straightforward using Theorem 1, Proposition 1 and noting that $S \subseteq S' \Rightarrow \mathcal{F}_S(\kappa) \subseteq \mathcal{F}_{S'}(\kappa)$. □

This theorem is useful as soon as the kernels used imply $\sup_{S'_{2n}}\mathbb{E}_{\boldsymbol{\sigma}}\frac{1}{2n}\left\|K_{S'_{2n}}\boldsymbol{\sigma}\right\|_{p'q';r} = O(n^{-\beta})$ for $\beta > 0$. The following proposition gives an example of a simple condition on the kernels used to be in that situation for some values of p', q' and r.

Proposition 2. *Let D be a distribution on $\mathcal{Z}$. If $\exists K_\infty \in \mathbb{R}$ such that $\mathbb{P}(k_t(X,X') \le K_\infty) = 1, \forall t = 1,\ldots,\tau$, then*

$$\sup_{S_{2n}}\mathbb{E}_{\boldsymbol{\sigma}}\frac{1}{2n}\left\|K_{S_{2n}}\boldsymbol{\sigma}\right\|_{p'q';r} \le \tau K_\infty\sqrt{\frac{\ln 4n}{n}},$$

for $(p',q',r) \in \{(\infty,\infty,1),(\infty,\infty,2),(\infty,2,1),(2,\infty,2)\}$, i.e., for $(p,q,r) \in \{(1,1,1),(1,1,2),(1,2,1),(2,1,2)\}$.

Proof. We just give the proof for $p' = 2, q' = \infty$ and $r = 1$, i.e., $p = 2, q = 1$ and $r = 1$. The other cases can be obtained along the same lines.

Here, S_{2n} denotes an i.i.d sample of size $2n$. The dimensions of $K_{S_{2n}}$ and $\boldsymbol{\sigma}$ follow accordingly. Noting that $\|\boldsymbol{\alpha}\|_{pq;1} = \left[\sum_i \|\boldsymbol{\alpha}_{i\bullet}\|_p^q\right]^{1/q}$ for any vector $\boldsymbol{\alpha}$, we have, dropping the $2n$ subscript for sake of clarity:

$$\begin{aligned}\|K\boldsymbol{\sigma}\|_{2,\infty;1} &= \sup_{1\le i\le 2n}\|[K\boldsymbol{\sigma}]_{i\bullet}\|_2\\ &\le \sup_{1\le i\le 2n}\|[K\boldsymbol{\sigma}]_{i\bullet}\|_1 \qquad (\|\boldsymbol{\alpha}\|_2\le\|\boldsymbol{\alpha}\|_1,\forall\boldsymbol{\alpha})\\ &= \sup_{1\le i\le 2n}\sum_{t=1}^{\tau}|[K\boldsymbol{\sigma}]_{it}|\\ &= \sup_{1\le i\le 2n}\sum_{t=1}^{\tau}\left|\sum_{j=1}^{2n}k_t(\mathbf{x}_i,\mathbf{x}_j)\sigma_j\right|\\ &\le \sum_{t=1}^{\tau}\sup_{1\le i\le 2n}\left|\sum_{j=1}^{2n}k_t(\mathbf{x}_i,\mathbf{x}_j)\sigma_j\right|.\end{aligned}$$

Now, for fixed t, we can apply Massart's finite class lemma (see appendix) to the $2n$ $2n$-dimensional vectors $\mathbf{v}_i = [k_t(\mathbf{x}_i,\mathbf{x}_1)\cdots k_t(\mathbf{x}_i,\mathbf{x}_{2n})]^\top$ of length $\|\mathbf{v}_i\| \le K_\infty\sqrt{2n}$:

$$\mathbb{E}_{\boldsymbol{\sigma}}\sup_i\left|\sum_{j=1}^{2n}k_t(\mathbf{x}_i,\mathbf{x}_j)\sigma_j\right| \le K_\infty\sqrt{\frac{\ln 4n}{n}},$$

which concludes the proof. □

This proposition establishes a (simple) condition so that the bound of Theorem 2 displays a $1/\sqrt{n}$ factor only for specific values of p, q and r. Finding a more general condition for such a factor to be present in the bound for any combination of p, q and r is the subject of ongoing research on our part; Besov spaces are a possible direction.

3. Algorithms

3.1. Proximal algorithms

This section synthetically describes the proximal framework used to solve problem (2). Proximal algorithms deal with general problems taking the form of

$$\min_{\boldsymbol{\alpha}} f_1(\boldsymbol{\alpha}) + f_2(\boldsymbol{\alpha}), \tag{6}$$

where f_1 and f_2 are convex and lower semicontinuous functions. Resolving such kind of problem relies on proximity operators, introduced by Moreau (1965). More details on the proximal framework can be found in the work of Combettes and Pesquet (2007).

Definition 2 (Proximity operator)**.** Let $\varphi : \mathbb{R}^P \to \mathbb{R}$ be a lower semicontinuous, convex function. The proximity operator $\mathrm{prox}_\varphi : \mathbb{R}^P \to \mathbb{R}^P$ associated with φ is given by

$$\mathrm{prox}_\varphi(\mathbf{u}) = \operatorname*{argmin}_{\boldsymbol{\alpha}\in\mathbb{R}^P}\frac{1}{2}\|\mathbf{u}-\boldsymbol{\alpha}\|_2^2 + \varphi(\boldsymbol{\alpha}).$$

Algorithm 1 Forward-backward proximal algorithm

```
input       $K_y, \gamma$, with $\gamma < 2/\beta$
initialize  $\boldsymbol{\alpha}^{(0)} \in \mathbb{R}^{n\tau}$ (for instance $\mathbf{0}$)
repeat
    $\boldsymbol{\alpha}^{(s+1)} = \mathrm{prox}_{\gamma f_1}\left(\boldsymbol{\alpha}^{(s)} - \gamma \nabla_{\boldsymbol{\alpha}} f_2(\boldsymbol{\alpha})\right)$
until convergence
```

When f_1 is convex lower semicontinuous, and f_2 is differentiable with ∇f_2 β-Lipschitz, then Problem (6) can be solved with Algorithm 1. Combettes and Wajs (2005), and more recently Combettes and Pesquet (2007), show that this algorithm converges to a minimum of Problem (6).

3.2. Proximity operators

Here, we are interested in proximity operators related to mixed norms (Kowalski, 2008). In Problem (6), the mixed norm penalty $f_2(\boldsymbol{\alpha}) = q^{-1}\lambda\|\boldsymbol{\alpha}\|_{pq}^q$, with $p, q \geq 1$, is a convex lower semicontinuous function, nondifferentiable in 0. Furthermore, $\nabla_{\boldsymbol{\alpha}} f_2(\boldsymbol{\alpha})$ is only β-Lipschtiz when $p, q \in \{1, 2\}$. We thus limit the study of proximity operators for the norms $\|\cdot\|_1$, $\|\cdot\|_2$, $\|\cdot\|_{12}$, $\|\cdot\|_{21}$.

Proposition 3 (Proximity operators for mixed norms)**.** *Let* $\mathbf{u} \in \mathbb{R}^P$ *be indexed by* (ℓ, m) *and* $\lambda > 0$*. The proximity operators for* $\lambda\|\cdot\|_{pq}$*, with* $p, q \in \{1, 2\}$*, defined by*

$$\hat{\boldsymbol{\alpha}} = \mathrm{prox}_{\lambda\|\cdot\|_{p,q}^q}(\mathbf{u}) = \operatorname*{argmin}_{\boldsymbol{\alpha}\in\mathbb{R}^P} \frac{1}{2}\|\mathbf{u} - \boldsymbol{\alpha}\|_2^2 + \frac{\lambda}{q}\|\boldsymbol{\alpha}\|_{p,q}^q \ ,$$

are given coordinate-wise for each (ℓ, m) *by:*

- **when** $p = q = 1$*,*

  $$\hat{\alpha}_{\ell,m} = \mathrm{sign}(u_{\ell,m}) \left||u_{\ell,m}| - \lambda\right|_+ \ ,$$

  which is the well-known soft-thresholding operator;

- **when** $p = 2$ **and** $q = 2$*,*

  $$\hat{\alpha}_{\ell,m} = \frac{1}{1+\lambda} u_{\ell,m} \ ;$$

- **when** $p = 2$ **and** $q = 1$*,*

  $$\hat{\alpha}_{\ell,m} = u_{\ell,m}\left|1 - \frac{\lambda}{\|\mathbf{u}_{\ell\bullet}\|_2}\right|_+ \ ;$$

- **when** $p = 1$ **and** $q = 2$*,*

  $$\hat{\alpha}_{\ell,m} = \mathrm{sign}(u_{\ell,m})\left||u_{\ell,m}| - \frac{\lambda \sum\limits_{m_\ell=1}^{M_\ell} \breve{u}_{\ell,m_\ell}}{(1+\lambda M_\ell)\|\mathbf{u}_{\ell\bullet}\|_2}\right|_+ \ ,$$

  where $\breve{u}_{\ell,m_\ell}$ *denotes the* $|u_{\ell,m_\ell}|$ *ordered by descending order* for fixed ℓ*, and the quantity* M_ℓ *is the number such that*

  $$\breve{u}_{\ell,M_\ell+1} \leq \lambda \sum_{m_\ell=1}^{M_\ell+1} \left(\breve{u}_{\ell,m_\ell} - \breve{u}_{\ell,M_\ell+1}\right) \ ,$$

  and

  $$\breve{u}_{\ell,M_\ell} > \lambda \sum_{m_\ell=1}^{M_\ell} \left(\breve{u}_{\ell,m_\ell} - \breve{u}_{k,M_\ell}\right) \ .$$

Algorithm 2 Forward-backward for squared hinge loss

```
input       $K_y, \lambda, \gamma$, with $\gamma < 2/\|K_y^T K_y\|$
initialize  $\boldsymbol{\alpha}^{(0)} \in \mathbb{R}^{n\tau}$
repeat
    $\boldsymbol{\alpha}^{(s+1)} = \mathrm{prox}_{\gamma\lambda\|\cdot\|_{pq}^q}\left(\boldsymbol{\alpha}^{(s)} + \gamma K_y^T \left[\mathbf{1} - K_y\boldsymbol{\alpha}\right]_+\right)$
until convergence
```

3.3. Solving Problem (2) with Proximal Optimization

The squared hinge loss can be restated in matrix form as $[\![\mathbf{1} - K_y\boldsymbol{\alpha}]\!]_+^2$, where $K_y = \mathrm{diag}([y_1, \ldots, y_n])\, K^\top$. In the previous section, we have shown how to compute the proximity operators for $\|\cdot\|_{pq;r}$ norms. Let us remind that $f_1(\boldsymbol{\alpha}) = [\![\mathbf{1} - K_y\boldsymbol{\alpha}]\!]_+^2$, is differentiable with gradient β-Lipschitz, while $f_2(\boldsymbol{\alpha}) = q^{-1}\lambda\|\boldsymbol{\alpha}\|_{pq}^q$, with $p, q \in \{1, 2\}$, is a convex lower semicontinuous functions, nondifferentiable in 0. Thus, we can use the forward-backward strategy given in Algorithm 1 to solve Problem (2). To do so, it suffices to compute $\nabla[\![\mathbf{1} - K_y\boldsymbol{\alpha}]\!]_+^2 = -K_y^T[\![\mathbf{1} - K_y\boldsymbol{\alpha}]\!]_+$, which is β-Lipschitz with $\beta = \|K_y^T K_y\|$. The resulting procedure for Problem (6) is given in Algorithm 2.

4. The Good, the Bad, and the Ugly: a Numerical Illustration

In this section, our aim is to exhibit the effects of regularization when using a structure on kernels or data. The structure is introduced in Problem (2) by mixed norms $\|\cdot\|_{pq;r}$, with $p, q \in \{1, 2\}$ as explained in section 2. An in-depth study concerning the predictive performances using actual indefinite kernels will be adressed in a longer version of this paper.

Here, we compare Algorithm 2 for different regularization terms, with regard to the sparsity behavior. The comparison is done on the *Titanic* dataset, provided by Gunnar Rätsch. [2] This binary classification problem consists of 150 training and 2051 test examples.

We have designed a global kernel matrix, composed of three kernels, chosen so that a classifier obtains *Good*, *Bad*,

[2] `http://ida.first.fraunhofer.de/projects/bench/`

and *Ugly* performances, according to the state of the art. More precisely, we have $K = [K_G, K_B, K_U]$, where:

- K_G, a linear kernel, is the Good guy;
- K_B, a Gaussian kernel of width 0.1, is the Bad guy;
- K_U, a Gaussian kernel of width 100, is the Ugly guy.

As baseline performances, the lowest test errors achieved with Algorithm 2, using the $\| \cdot \|_2$ norm, with kernels K_G, K_B and K_U taken separately, are respectively 21.84%, 25.89% and 32.57%

In Figure 2, we compare the influence of the different regularizations, with $K = [K_G, K_B, K_U]$. Here, the parameter λ has been chosen by a 5-fold cross-validation procedure, between logarithmically spaced values varying from 1 to 10^5. The classification error rate, which is 22.92% for the $\| \cdot \|_{21;1}$ norm, and 21.84% for the other norms, was computed on the test set.

- The use of the norm $\| \cdot \|_2$ does not single out either of K_G or K_B. Ugly's coefficients are all smoothed towards zero. According to the nature of the $\| \cdot \|_2$ penalization, there is no sparsity induced.
- Contrary to the $\| \cdot \|_2$ regularization, the $\| \cdot \|_1$ norm introduces sparsity everywhere. The most influent coefficients belong to K_G, and only few of them are nonzero. Even if many coefficients related to the Ugly kernel are nonzero, they still remain small in magnitude.
- The $\| \cdot \|_{21;2}$ norm, which focuses on the kernel structure, identifies quite well the Good kernel, giving to the corresponding coefficients values close to one. Even though it is not discarded, an insignificant relevance is assigned to the Ugly kernel.
- The $\| \cdot \|_{12;2}$ penalization behaves similarly as the $\| \cdot \|_1$ norm. However, one can see in the Bad kernel that some coefficients have a higher importance, which is consistent with the nature of the norm. Indeed, it is expected that within each relevant kernel, the penalization puts more weight on the most relevant elements.
- The $\| \cdot \|_{21;1}$ regularization is supposed to put emphasis on the most relevant observations, whatever the kernel, and to eliminate the others. In that sense, the remaining coefficients can be envisioned as support vectors. This is quite well illustrated on Figure 2.
- Finally, for all data, the $\| \cdot \|_{12;1}$ norm identifies the most significant kernels for the classification task. It is worth noting that there are few contiguous lines: for numerous observations, only one kernel is selected.

One can note that the Ugly kernel is involved in the solutions related to the $\| \cdot \|_1$ and $\| \cdot \|_{21;2}$ norms, which could appear inconsistent. An insight concerning the presence of the Ugly kernel is that λ was chosen through cross-validation based on the generalization error. As Leng et al. (2004) showed for the Lasso, this might be not optimal in terms of kernel selection. A slight increase of λ allowed us to discard the Ugly kernel (when using the $\| \cdot \|_{21;2}$ norm), or to significantly reduce its influence (when using the $\| \cdot \|_1$ norm).

5. Discussion

The formulation of the MKL problem by Bach et al. (2004) may be seen as the kernelization of the Group-Lasso, considering penalties on elements f_t from several RKHS $\mathcal{H}_t$, in a standard SVM problem.Rakotomamonjy et al. (2007) tackled the dual formulation. It consists of explicitly optimizing a convex combination of kernels, which defines the actual SVM kernel $K(\mathbf{x}, \mathbf{x}') = \sum_{t=1}^{\tau} \sigma_t K_t(\mathbf{x}, \mathbf{x}')$, where K_t is the reproducing kernel of $\mathcal{H}_t$, and σ_t the coefficients to be learned under a ℓ_1 constraint.

MKL involves a kernel which is a convex combination of candidate kernels, where the coefficients of the less relevant ones are shrinked towards zero. In that sense, using $\| \cdot \|_{21;2}$ in Problem (2) is closely related to MKL, as it induces sparsity in the kernels. We may note that MKL not only enforces sparsity in kernels but also with respect to data, since it essentially is a SVM problem and thus produces (few) support vectors. To achieve such a joint sparsity in our framework, we would have to sacrifice the convexity of $\| \cdot \|_{pq;r}$, by choosing $p, q \leq 1$.

Another difference with MKL is that we do not have any notion of 'norm' of f; instead we control the (mixed) norm of synthesis coefficients $\alpha_{\ell m}$ in the frame generated by the kernels. This perspective is closely related with the idea of expansion with respect to a dictionary of $\star$-lets (such as wavelets) in signal processing.

6. Conclusion and Further Work

We have proposed a mixed norm setting to learn multiple kernel classifiers. On the contrary to a common assumption on kernel positive definiteness, our framework is still valid when using indefinite kernels. The learning problem that tackle can be formulated as a convex optimization problem and the desired sparsity structure can be enforced by the choice of the mixed norm used, at the price of rendering the optimization problem nondifferentiable. To cope with this nondifferentiability, we derive an optimization algorithm stemming from the proximal framework. Simulations showing the modularity of our approach are provided.

This work raises several open problems. First, we would like to provide more extensive numerical simulations, especially with indefinite or nonsymmetric kernels. The primary application we have in mind is image categorization, where the kernels are based on Kullback Leibler divergences between probability density functions derived from wavelet decompositions (Piro et al., 2008). We also plan to investigate the possibility of analytically computing the regularization path for λ. Finally, we have been working on two extensions of our optimization procedure, namely (a) a chunking strategy (Osuna & Girosi, 1997) to make a better use of the resulting sparseness and (b) the adaptation of latest advances in convex optimization introduced by Nesterov (2007) to our learning problem.

Acknowledgments

MS and LR are partially supported by the IST Program of the European Community, under the FP7 Pascal 2 Network of Excellence ICT-216886-NOE.

Appendix

Theorem 3 (Holder's Inequality)**.** *Let $p, q \geq 1$ and $n \in \mathbb{N}$. If $\frac{1}{p} + \frac{1}{q} = 1$ then*

$$\forall \mathbf{u}, \mathbf{v} \in \mathbb{R}^n, \sum_{i=1}^{n} |u_i v_i| \leq \left[\sum_{i=1}^{n} |u_i|^p\right]^{\frac{1}{p}} \left[\sum_{i=1}^{n} |v_i|^q\right]^{\frac{1}{q}}.$$

Lemma 1 (Massart's finite class lemma)**.** *Let A be a finite subset of $\mathbb{R}^n$ with each vector $\mathbf{x} = [x_1 \cdots x_n]^\top$ in A having norm bounded by $r = \max \|\mathbf{x}\|_2$. If $\boldsymbol{\sigma}$ is an n-dimensional vector of independent Rademacher variables, then*

$$\mathbb{E}_{\boldsymbol{\sigma}} \left[\sup_{\mathbf{x} \in A} \frac{1}{n} \left|\sum_{i=1}^{n} x_i \sigma_i\right|\right] \leq \frac{r\sqrt{2 \ln 2|A|}}{n}.$$

(We have slightly changed the statement from its original form to take the absolute value into account.)

References

Bach, F. R., Lanckriet, G. R. G., & Jordan, M. I. (2004). Multiple kernel learning, conic duality, and the SMO algorithm. *Proc. 21th Int. Conf. Mac. Learn. (ICML 2004)* (pp. 41–48).

Bartlett, P. L., & Mendelson, S. (2002). Rademacher and gaussian complexities: Risk bounds and structural results. *J. Mac. Learn. Res.*, *3*, 463–482.

Ben-David, S., Rahimi, A., & Srebro, N. (2008). Generalization Bounds for Indefinite Kernel Machines. Nips*08 Workshop: New Challenges in Theoretical Machine Learning.

Boser, B., Guyon, I., & Vapnik, V. (1992). A Training Algorithm for Optimal Margin Classifiers. *Proc. 5th Workshop Comp. Learn. Theory*.

Combettes, P. L., & Pesquet, J.-C. (2007). A Douglas-Rachford splitting approach to nonsmooth convex variational signal recovery. *IEEE J. Select. Top. Sig. Proc.*, *1*, 564–574.

Combettes, P. L., & Wajs, V. R. (2005). Signal recovery by proximal forward-backward splitting. *Multiscale Modeling and Simulation*, *4*, 1168–1200.

Cortes, C., & Vapnik, V. (1995). Support Vector Networks. *Machine Learning*, *20*, 1–25.

Kowalski, M. (2008). *Sparse regression using mixed norms* (Technical Report). LATP, Marseille, France. `http://hal.archives-ouvertes.fr/hal-00202904/`.

Kowalski, M., & Torrésani, B. (2008). Sparsity and persistence: mixed norms provide simple signals models with dependent coefficients. *Signal, Image and Video Processing*. doi:10.1007/s11760-008-0076-1.

Lanckriet, G., Cristianini, N., Bartlett, P., El Ghaoui, L., & Jordan, M. (2004). Learning the Kernel Matrix with Semidefinite Programming. *J. Mac. Learn. Res.*, *5*, 27–72.

Leng, C., Lin, Y., & Wahba, G. (2004). A note on lasso and related procedures in model selection. *Statistica Sinica*, *16*, 1273–1284.

Moreau, J.-J. (1965). Proximité et dualité dans un espace hilbertien. *Bull. Soc. Math. France*, *93*, 273–299.

Nesterov, Y. (2007). *Gradient methods for minimizing composite objective function* (Technical Report). Université Catholique de Louvain. CORE discussion paper.

Osuna, E., & Girosi, F. (1997). Improved Training Algorithm for Support Vector Machines. *Neural Networks for Signal Processing* (pp. 276–285). IEEE Press.

Piro, P., Anthoine, S., Debreuve, E., & Barlaud, M. (2008). Sparse multiscale patches for image processing. *Emerging Trends in Visual Computing* (pp. 284–304).

Rakotomamonjy, A., Bach, F., Canu, S., & Grandvalet, Y. (2007). More efficiency in Multiple Kernel Learning. *Proc. 24th Int. Conf. Mac. Learn. (ICML 2007)* (pp. 775–782). Omnipress.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *J. Roy. Stat. Soc. B*, *58*, 267–288.

Yuan, M., & Lin, Y. (2006). Model selection and estimation in regression with grouped variables. *J. Roy. Stat. Soc. B*, *68*, 49–67.

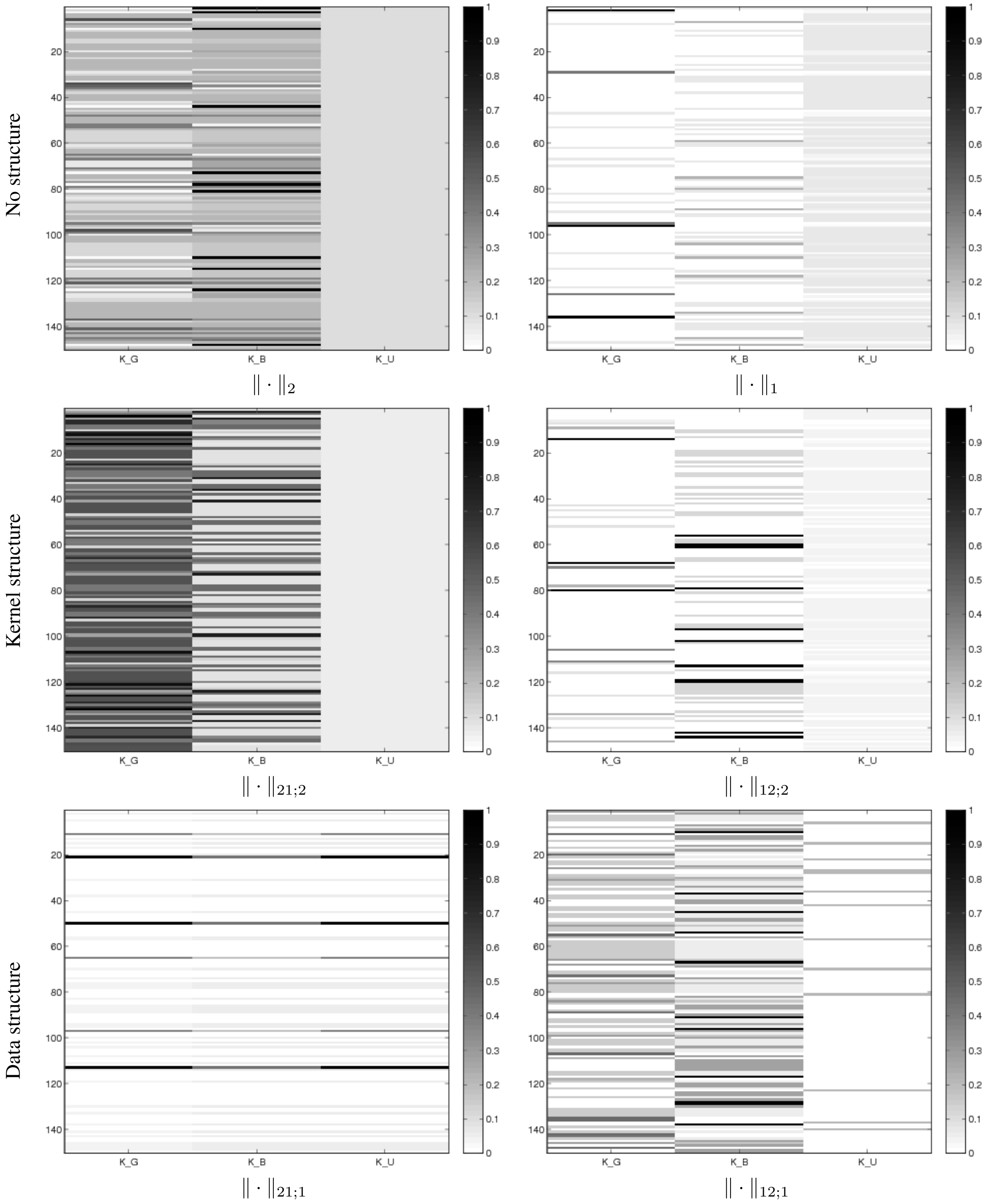


Figure 2. Relevance maps of $\boldsymbol{\alpha}_{\bullet t}$, with $t \in \{G, B, U\}$, for different norms. The coefficients have been normalized, so that $\alpha_{it} \in [0, 1]$. Top: no structure defined. Middle: structure defined with respect to the kernels. Bottom: structure defined with respect to the data.