FILTER BANK KERNEL LEARNING FOR NONSTATIONARY SIGNAL CLASSIFICATION

Maxime Sangnier[⋆†] *Jérôme Gauthier*[†] *Alain Rakotomamonjy*[⋆]

[⋆] Université de Rouen, LITIS EA 4108, 76800 Saint-Etienne du Rouvray, France
[†]CEA, LIST, 91191 Gif-sur-Yvette CEDEX, France

ABSTRACT

This paper addresses the problem of automatic feature extraction for signal classification. In order to handle non-stationarity, features are designed in the time-frequency domain using a Filter Bank as the mapping function, which enables an easy interpretation for practitioners. The strategy adopted is to jointly learn a Filter Bank with a Support Vector Machine by casting the optimization program as a Multiple Kernel Learning problem. This solves the program for a finite set of filters. Thus, in order to handle an infinite number of filters, a novel active constraint algorithm is proposed based on the latest breakthroughs. Our method has been tested on a toy dataset and compared to classical methods with competitive results.

Index Terms**—** Time-frequency representation, Filter Bank, signal classification, Multiple Kernel Learning, SVM

1. INTRODUCTION

In recent years, Signal Processing has broadened its focus towards Machine Learning methods. In this scope, features are needed to characterize similarities within a class and disparities between classes to be distinguished. This property, called discrimination, obviously affects the classifier accuracy and thus features should be more elaborated than a simple time representation. There is a lot of possible features for signal classification rocking from physical perceptions (loudness), statistical moments (covariance matrix), spectral characterization (Fourier transform) and time-frequency representations (spectrograms, wavelet decompositions). However, the features extractor of this preprocessing step is usually arbitrarily chosen and abusively considered independently from the choice of the pattern recognition algorithm, so that there is no guarantee of any classification efficiency.

This problem of finding discriminative features with respect to the chosen classifier became a field of interest in the 1990's. It appeared in different areas for different categories of features (e.g. [1–9]). Particularly in signal classification, discriminative features learning was especially developed for time-frequency representations (TFR), which may contain more information than physical and statistical descriptors and present a real advantage for non-stationary signals. The bulk of the scientific contributions deals with wavelets learning combined with a Support Vector Machine (SVM) [10–14]. Moreover, optimization methods usually boil down to an exhaustive search or an evolutionary algorithm. More recently, Yger et al. have proposed an efficient method to learn a mixture of wavelet transforms based on the Kernel Learning theory [15]. Besides, let us mention that dictionaries [2, 16, 17] and Cohen's class TFRs [18] have also been studied in the discriminative approach.

In [19, 20], the authors chose the Filter Bank (FB) model [21] to extract features from signals and tried to learn it in unison with a Hidden Markov Chain based classifier through an evolutionary algorithm. In the past couple of decades, FBs have been well studied in the reconstructive approach for denoising and compression [21–25] and appeared with these studies to be adequate for discriminative feature extraction.

In order to fill the gap between the choice of the features extractor among the various Signal Processing tools and the classifier learning, we investigate, in this work, the way to jointly learn a FB with a SVM. FBs have been chosen for their ability to model a wide class of atomic decompositions (cosine transform, short-time Fourier transform, wavelet transform, etc.). Moreover the very definition of the FB [21] ensures a direct time-frequency interpretation. However, while most of the aforementioned approaches achieve a posterior optimization (the cost function is a misclassification rate), we prefer the more theoretical way of the prior optimization using the SVM objective as the cost function. For this purpose, we follow the same technique as [15], casting the problem as a Multiple Kernel Learning problem [26] in which each kernel is associated to a parametrized filter.

To explain our approach, we first remind of the kernelized framework for classification, especially the SVM and the Multiple Kernel Learning (MKL). Then we expose basics on FBs and introduce the Filter-MKL algorithm, which enables to jointly learn a FB with a SVM. In order to tackle the infinite amount of filters, we propose an active constraint algorithm based on the Karush–Kuhn–Tucker (KKT) conditions. This one extends the work by Varma and Babu [27] to an endless number of kernels. Finally, our method is evaluated on a toy dataset and compared to classical Signal Processing transforms combined with a SVM.

2. KERNELIZED FRAMEWORK FOR CLASSIFICATION

2.1. Support Vector Machine

Let $\mathcal{X}$ be an arbitrary compact input space, $S_{\mathcal{X}}^{+}$ the set of symmetric positive definite kernels on $\mathcal{X}$ and $\mathcal{O}_\infty$ the set of all training datasets: $\mathcal{O}_\infty = \bigcup_{N=1}^{+\infty}(\mathcal{X}^N \times \mathcal{Y}^N)$ (where $\mathcal{Y}$ denotes the set of labels; here, $\mathcal{Y} = \{-1, 1\}$).

SVM is a well known algorithm that, given a training set $((x_i, y_i))_{1\leq i\leq N}$ from $\mathcal{O}_\infty$ (here, x_i are signals), a kernel k from $S_{\mathcal{X}}^{+}$ and a trade-off parameter C ($\frac{1}{N} \leq C$ [28]), returns the optimal linear classifier f^* in the feature space $\mathbb{H}_k$ defined by k [29]. SVM tackles the problem of minimizing the structural risk $R\left((x_i)_{1\leq i\leq N}, k, f\right)$ with respect to f, where, if w_f denotes the unique normal vector of $\mathbb{H}_k$ that defines f,

$$R\left((x_i)_{1\leq i\leq N}, k, f\right) = \frac{1}{2}\|w_f\|^2 + C\sum_{i=1}^{N}\max(0, 1 - y_i f(x_i)).$$

Let $Y = \mathrm{diag}(y)$ and K be the positive definite kernel matrix de-

Thanks to the Direction Générale de l'Armement for funding.

fined by $K = (k(x_i, x_j))_{1 \le i,j \le N}$. In practice SVM solves a dual form (1) of the previous problem (in the forthcoming sections, we note $J\left((x_i)_{1\le i\le N}, k\right)$ the optimal value of (1)), where $\mathbb{1}$ stands for the indicator vector and $\preccurlyeq$ for a pointwise inequality.

$$\begin{aligned} &\underset{\alpha\in\mathbb{R}^N}{\texttt{minimize}}\ \tfrac{1}{2}\alpha^T Y K Y \alpha - \mathbb{1}^T\alpha \\ &\texttt{subject to}\ 0 \preccurlyeq \alpha \preccurlyeq C\mathbb{1},\ y^T\alpha = 0 \end{aligned} \tag{1}$$

Then the KKT conditions give a parametrization (including the bias b from $\mathbb{R}$) of the optimal linear functional f^*:

$$\forall x \in \mathcal{X},\ f^*(x) = \sum_{i=1}^{N} \alpha_i y_i k(x_i, x) + b.$$

2.2. Multiple Kernel Learning

Multiple Kernel Learning is an algorithm that jointly learns a sparse convex combination of kernels with the associated optimal SVM classifier. Thus, let $(s_i)_{1\le i\le d}$ be a set of kernels from $S^+_{\mathcal{X}}$, called *spanning kernels*. In practice, MKL solves the convex problem (2) [26], which boils down to minimize the SVM risk $R\left((x_i)_{1\le i\le N}, k, f\right)$ with respect to f and k, lying in the set of the convex combinations of the spanning kernels $(s_i)_{1\le i\le d}$ with a normalized weights sum.

$$\begin{aligned} &\underset{\mu\in\mathbb{R}^d}{\texttt{minimize}}\ J\left((x_i)_{1\le i\le N}, \textstyle\sum_{i=1}^d \mu_i s_i\right) \\ &\texttt{subject to}\ \mathbb{1}^T\mu = 1,\ \mu \succcurlyeq 0 \end{aligned} \tag{2}$$

In 2009, Varma and Babu proposed a more general framework considering non-convex combinations of kernels [27]. In this paper, we will be interested in the Hadamard product of kernels. Then the Generalized MKL (GMKL) algorithm solves the non-convex problem (3) by finding a local minimum (with σ the coefficient of the sparsity penalty). Even though the solution is not global, this may be more efficient than MKL in some cases.

$$\begin{aligned} &\underset{\mu\in\mathbb{R}^d}{\texttt{minimize}}\ J\left((x_i)_{1\le i\le N}, \textstyle\prod_{i=1}^d s_i^{\mu_i}\right) + \sigma\mathbb{1}^T\mu \\ &\texttt{subject to}\ \mu \succcurlyeq 0 \end{aligned} \tag{3}$$

3. AUTOMATED TFR DESIGN

3.1. Filter Bank Kernel Learning

Let $\mathcal{F}$ be the set of FBs and $\star : \mathcal{F} \times \bigcup_{N=1}^{+\infty} \mathcal{X}^N \to \bigcup_{N=1}^{+\infty} \mathcal{X}^N$ the filtering operator. In practice, a FB is a set of linear filters followed by downsampling operators (called *decimation*) [30]. Note that for the purpose of classification, outputs of the filters are concatenated in a single vector.

The problem studied in this article is to design an algorithm

$$\mathcal{B} : \mathcal{O}_\infty \to \mathcal{F} \times \mathcal{L}_k,$$

where $\mathcal{L}_k$ is the set of linear functionals of $\mathbb{H}_k$, such that

$$\mathcal{B}\left((x_i, y_i)_{1\le i\le N}\right) = \underset{H\in\mathcal{F},\ f\in\mathcal{L}_k}{\operatorname{argmin}}\ R\left((H\star x_i)_{1\le i\le N}, k, f\right). \tag{4}$$

In other words, the algorithm $\mathcal{B}$ jointly learns the optimal TFR (among FBs) and the optimal SVM classifier in the time-frequency domain mapped by the resulting TFR.

The idea of the proposed method (given in algorithm 1) is to take advantage of the Multiple Kernel Learning framework by associating one normalized filter and its decimation factor to each kernel. Let k be either the linear or a Gaussian kernel. The set $(s_i)_{1\le i\le d}$ of MKL spanning kernels is made as follow:

$$\forall i \in [\![1, d]\!],\ s_i = k(\ell_i \star \cdot, \ell_i \star \cdot), \tag{5}$$

where ℓ_i ($i \in [\![1, d]\!]$) is a FB composed by a single filter and its associated decimation factor.

```
  Data: training dataset $(x_i, y_i)_{1\le i\le N}$
  Result: FB $F$ and linear functional $f_k^*$
1 $(\ell_i)_{1\le i\le d} \leftarrow$ candidate filters and their decimation factors;
2 $(s_i)_{1\le i\le d} \leftarrow$ kernels from $(\ell_i)_{1\le i\le d}$ {formula (5)};
3 $(\mu, f_q^*) \leftarrow$ Solve (G)MKL with kernels $(s_i)_{1\le i\le d}$;
4 $F \leftarrow$ FB based on $\{\sqrt{\mu_i}\ell_i,\ i \in [\![1, d]\!] \wedge \mu_i \neq 0\}$;
5 $f_k^* \leftarrow f_q^*$;
```

Algorithm 1: Restricted Filter-MKL algorithm.

Consider the notations of the algorithm 1 and let us prove that the algorithm 1 solves the problem (4) if $\mathcal{F}$ is restricted to the set $\widetilde{\mathcal{F}_\ell}$ of FBs based on a given set of spanning filters $(\ell_i)_{1\le i\le d}$ and weighted such that the weights ℓ^2–norm equals 1. Suppose that k is the linear kernel: $k = \langle\cdot|\cdot\rangle_{\mathcal{X}}$. Then the resulting kernel from MKL is $q = \sum_{i=1}^d \mu_i s_i$ and

$$\begin{aligned} &\forall (x_1, x_2) \in \mathcal{X}^2, \\ &k(F\star x_1, F\star x_2) \\ &=k\left(\begin{bmatrix} (\sqrt{\mu_1}\ell_1)\star x_1 \\ \vdots \\ (\sqrt{\mu_d}\ell_d)\star x_1 \end{bmatrix}, \begin{bmatrix} (\sqrt{\mu_1}\ell_1)\star x_2 \\ \vdots \\ (\sqrt{\mu_d}\ell_d)\star x_2 \end{bmatrix}\right) \\ &=\textstyle\sum_{i=1}^d k\left((\sqrt{\mu_i}\ell_i)\star x_1, (\sqrt{\mu_i}\ell_i)\star x_2\right) \\ &=\textstyle\sum_{i=1}^d \mu_i\, k(\ell_i\star x_1, \ell_i\star x_2) \\ &=q(x_1, x_2). \end{aligned} \tag{6}$$

The equality (6) proves that f_k^* is well defined (since f_q^* is a linear functional of the feature space induced by the kernel k) and that the couple (F, f_k^*) minimizes $R\left((\cdot\star x_i)_{1\le i\le N}, k, \cdot\right)$ on $\widetilde{\mathcal{F}_\ell}\times\mathcal{L}_k$. Indeed, for all (H, g_k) in $\widetilde{\mathcal{F}_\ell}\times\mathcal{L}_k$, with $H = (\sqrt{\lambda_i}\ell_i)_{1\le i\le d}$, let

$$q' = \sum_{i=1}^d \lambda_i k(\ell_i\star\cdot, \ell_i\star\cdot),$$

then, as λ is admissible for the MKL problem ($\mathbb{1}^T\lambda = 1$ and $\lambda \succcurlyeq 0$),

$$\begin{aligned} R\left((H\star x_i)_{1\le i\le N}, k, g_k\right) &= R\left((x_i)_{1\le i\le N}, q', g_k\right) \\ &\ge R\left((x_i)_{1\le i\le N}, q, f_q^*\right) \\ &= R\left((F\star x_i)_{1\le i\le N}, k, f_k^*\right). \end{aligned}$$

Suppose now that k is a Gaussian kernel. Then the proof is rigorously identical to the previous one, substituting the MKL solver by the GMKL one in the algorithm 1 (line 3) and kernels sums by kernels Hadamard products in the equality (6). Obviously, as GMKL, only achieves a local minimum, our algorithm also finds a suboptimal solution.

3.2. Solving the main problem with infinitely many filters

In the previous section, the algorithm 1 solves the problem (4) for a finite number of spanning filters. In this section, we explain how to solve the problem (4) with an endless number of spanning filters, which leads us to a variant of the known IKL algorithm [31] (for a linear kernel) and to the proposed Generalized IKL (GIKL) algorithm (for a Gaussian kernel). The general algorithm to jointly learn a FB with a SVM, which leans on both IKL and GIKL, is the algorithm 2.

Our approach restricts $\mathcal{F}$ to the set $\widetilde{\mathcal{F}}$ of FBs based on filters that are spectrally shifted and scaled version of the low-pass filter whose finite impulse response (FIR) is $\mathbb{1}$. Thus every filter is parametrized by its spectral mode f_m, its spectral width f_w ($f_m, f_w \in [0, \frac{1}{2}]$) and its magnitude. The FIR h of a filter of length f is [32]

$$\forall k \in [\![1, f]\!],\ h_k = \frac{\sin(\pi k f_w)\sin(2\pi(k-1)f_m)}{kQ},$$

where Q is a normalization factor giving the desired magnitude. The previous formula is obtained through the window method [32] with a rectangular window but other types of windows (like Hann or Blackman) can be used.

The proposed method (described in the algorithm 2 and explained thereafter) applies the principle of active set [33]. The idea of active set lies upon the notion of *active kernels* (which results from the sparsity of the MKL solution): the kernels for which the weights are positive. The other kernels have no effect on the solution for their weights are null. They can be neglected beforehand without changing the solution of the problem.

```
   **Data**: training dataset $(x_i, y_i)_{1\leq i\leq N}$
   **Result**: FB and linear functional
1  $A \leftarrow$ grid of filter spectral positions and widths;
2  $(s_t)_{t\in A} \leftarrow$ kernels from filters based on $A$ {formula (5)};
3  **while** *not suboptimal* **do**
4      $\mu \leftarrow$ Solve (G)MKL with kernels $(s_t)_{t\in A}$;
5      $A \leftarrow A\backslash\{t,\ (t \in A) \wedge (\mu_t = 0)\}$;
6      $\theta \leftarrow \operatorname{argmax}_{t\in\mathcal{P}} T_{\text{(G)MKL}}(t)$ {definitions in section 3.3)};
7      **if** $T_{\textit{(G)MKL}}(\theta) > 0$ **then** {optimality condition violated}
8          $A \leftarrow A \cup \{\theta\}$;
9      **else**
10         Subptimality is reached;
11 Deduce from $A$ the suboptimal FB and from the MKL output
   the suboptimal linear functional;
```

Algorithm 2: Filter-MKL algorithm.

Let the algorithm start from a guess on the active set and solve the MKL problem. As the solution is sparse, some kernels have positive weights (the active kernels) while others got null weights. Then if a kernel (out of the active set) violates the optimality conditions (embodied by $T_{\text{(G)MKL}}(\theta) > 0$ in the algorithm 2), it means that the guess on the active set was wrong and that this kernel was missing (otherwise, it would not violate the optimality conditions). So, let us add it to the set of spanning kernels (keeping only active kernels at the current iteration) and iterate alternatively the MKL step and the update of the spanning kernels set until optimality. In this context, the Filter-MKL algorithm solves the MKL problem for a given set of filters, then removes non-active filters and adds the filter that most violates the optimality conditions.

3.3. Checking the optimality

As explained earlier, the main point of the method is to check the optimality of a solution (algorithm 2, line 7), which should be an easy step in order to get a workable algorithm. In the case of a linear kernel (MKL), the work by Gehler et al. [31] that investigates a way to learn a convex combination of an endless number of kernels (Infinite Kernel Learning algorithm) gives the the optimality condition.

To extend MKL to IKL, let us replace the finite set of kernels $(s_i)_{1\leq i\leq d}$ by an infinite parametrized set $(s_\theta)_{\theta\in\mathcal{P}}$, where $\mathcal{P}$ is the set of kernel parameters. Here, parameters are f_m and f_w thus $\mathcal{P} = [0, \frac{1}{2}]^2$. Let J_{MKL} be the objective value of the problem (2) for an infinite set of kernels, i.e. $J_{\text{MKL}}\ :\ \mu \in \mathbb{R}_+^{\mathcal{P}} \mapsto J\left((x_i)_{1\leq i\leq N}, \sum_{\theta\in\mathcal{P}} \mu_\theta s_\theta\right)$ with μ sparse and $\mathbb{1}^T\mu = 1$. In this framework the KKT optimality conditions give (see theorem 4.1 from [34] for the differentiation): $T_{\text{MKL}} \leq 0$ with

$$\forall\theta \in \mathcal{P},\ T_{\text{MKL}}(\theta) = -\frac{\partial J_{\text{MKL}}}{\partial\mu_\theta}(\mu) - \lambda = \frac{1}{2}\alpha^{*T} Y K_\theta Y \alpha^* - \lambda,$$

where $Y = \operatorname{diag}(y)$, K_θ is the positive definite kernel matrix defined by $K_\theta = (s_\theta(x_i, x_j))_{1\leq i,j\leq N}$, α^* is the optimal dual variable of the SVM subproblem (1) and λ is the Lagrange multiplier associated to the sparsity constraint of the MKL problem (2).

In the case of a Gaussian kernel (GMKL), let us derive T_{GMKL}. Let $J_{\text{GMKL}}\ :\ \mu \in \mathbb{R}_+^{\mathcal{P}} \mapsto J\left((x_i)_{1\leq i\leq N}, \prod_{\theta\in\mathcal{P}} s_\theta^{\mu_\theta}\right) + \sigma\mathbb{1}^T\mu$, with μ sparse. If a point μ is suboptimal (i.e. J_{GMKL} achieved a local minimum), then the KKT conditions are satisfied (they are necessary conditions for the suboptimality):

$$\exists\lambda \in \mathbb{R}_+^d,\ \begin{cases} \nabla J_{\text{GMKL}}(\mu) + \sigma\mathbb{1} - \lambda = 0 \\ \mu \succcurlyeq 0 \\ \forall i \in [\![1, d]\!],\ \lambda_i\mu_i = 0. \end{cases}$$

Then, in any case,

$$\nabla J_{\text{GMKL}}(\mu) + \sigma\mathbb{1} \succcurlyeq 0.$$

Considering that the SVM kernel matrix can be formulated as $Q = \exp(-\gamma\sum_{\theta\in\mathcal{P}} \mu_\theta D_\theta)$, where γ is a positive number defining the Gaussian kernel and $(D_\theta)_{\theta\in\mathcal{P}}$ are the distance matrices in the time-frequency domain, then the suboptimality condition can be expressed with the optimal dual variable α^* of the SVM problem (1): $T_{\text{GMKL}} \leq 0$ with

$$\forall\theta \in \mathcal{P},$$
$$T_{\text{GMKL}}(\theta) = -\frac{\partial J_{\text{GMKL}}}{\partial\mu_\theta}(\mu) - \sigma = \frac{\gamma}{2}\alpha^{*T} Y(D_\theta \circ K)Y\alpha^* - \sigma,$$

where $\circ$ is the Hadamard product. Then, the scheme to derive this Generalized Infinite Kernel Learning algorithm, which is the extension of GMKL to an infinite number of kernels, is the same as the one exposed by Gehler et al. [31]. A major point is thus to solve the non-convex subproblem (7), as written in the algorithm 2 (line 6).

$$\underset{\theta\in\mathcal{P}}{\mathtt{maximize}}\ T_{\text{GMKL}}(\theta). \tag{7}$$

Two approaches are implemented in the proposed Filter-MKL (algorithm 2) in order to maximize $T_{\text{(G)MKL}}$: in the first one, a grid search is performed, then a possible gradient ascent can be applied in order to refine the solution. In the second approach, a random search supplants the grid search.

For an insight of the convergence property, consider that, at each iteration, non-active kernels are removed and a new one is added with a null weight. These new weights form a feasible point of the new MKL problem with an objective equals to the optimal objective of the previous MKL problem. Moreover, as the added kernel violates the KKT conditions, then the current point is not optimal. Thus the objective value is compelled to improve.

4. EXPERIMENTAL RESULTS

In this section, we present some results on a toy problem. Without any prior knowledge, we have chosen quite a basic set of parameters. The decimation factor is considered constant for every filter and equal to 8, while filters can have three different lengths: 32, 64 and 128. Besides, several methods are confronted: i) DFT: magnitude of the Discrete Fourier Transform and Gaussian SVM, ii) Wavelet: 4-Daubechies wavelet decomposition and Gaussian SVM, iii) WKL: full stochastic Wavelet Kernel Learning with a RBF kernel [15], iv) IKL: Filter-MKL with a linear kernel, v) GIKL: Filter-MKL with a Gaussian kernel. Methods iv and v are those proposed in this paper.

The toy dataset is a binary problem based on two signals from the Wavelab toolbox: Blocks and HeaviSine. A non-stationary Gaussian colored noise (of standard deviation σ_{noise}) is added to each signal in order to form both classes. The figures 1 and 2 present the classification accuracies of the several methods. The accuracy is measured with the kappa statistic, which is as close to 1 as the classification is accurate [35]. These results have been averaged out over 10 runs. The methods have been trained with N_{train} signals and assessed with 1000 signals from the toy dataset. At each run, methods parameters have been chosen through a 5-fold cross-validation.

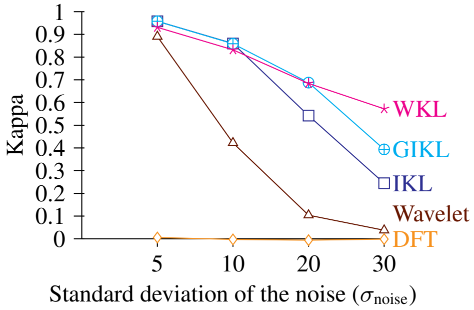

Fig. 1. Classification accuracy on the test dataset ($N_{\text{train}} = 30$).

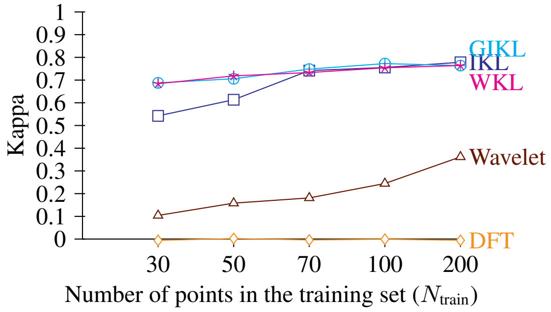

Fig. 2. Learning curve on the test dataset ($\sigma_{\text{noise}} = 20$).

Results point out the importance of data-driven TFRs (WKL and Filter-MKL) for non-stationary signals classification, compared to the use of a fixed wavelet [1] and of a DFT (which is as accurate as a random choice). Moreover it has to be noticed that even though the Gaussian version of Filter-MKL (based on the proposed GIKL algorithm) is non-convex, it achieves better accuracies than the convex and linear version of Filter-MKL for small training sets and very noisy data. However the latter seems preferable for large training sets since it achieves the same accuracy and is faster than the Gaussian version. Besides, our methods are quite competitive with WKL from [15]. The Wilcoxon signed rand tests performed to assess the difference between accuracy results, demonstrate that in most cases, at least a method of ours is significantly better than WKL (with a significance level at 5%). At last, the example of resulting FB (figure 3) underlines that the normalized frequencies 0.025 (filter 1) and 0.134 (filter 2) are of particular interest for this classification task. Note that the filter 1 is bimodal because of the chosen window.

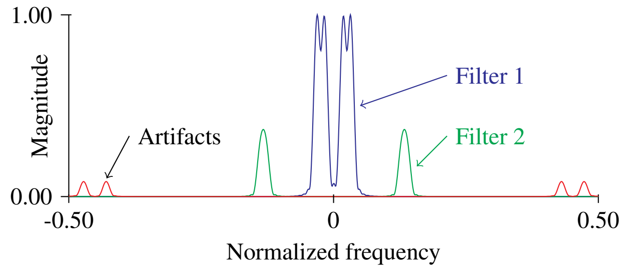

Fig. 3. Example of resulting FB in the Fourier domain. Each color embodies a filter.

The Matlab code used to produce these results is provided on the authors' website for the sake of reproducibility.

5. DISCUSSION ON THE CONTRIBUTION

In this paper, we proposed a novel approach to learn a discriminative FB, that unlike [19, 20] minimizes a structural risk of classification. First results on a toy dataset are promising even though for now, the proposed method is quite time-consuming because of the cross-validation step. The choice of the parameters will be optimized on short terms. Besides in the forthcoming work, other classes of filters will be included in order to widen the subset $\widetilde{\mathcal{F}}$ of FBs. Last but not least, we plan to investigate the way to make our method robust to signal translations to handle real signals, which are usually randomly time shifted.

An implicit viewpoint developed all along this paper is to consider that the problem (4) boils down to learn the SVM kernel (parametrized by a FB). The result is that this study lies in the Kernel Learning field and has the flavor of the work by Gehler et al. [31] that extends the MKL to the Infinite Kernel Learning. The implementation of the IKL algorithm in [31] already uses the concept of active kernels and of violation of the optimality conditions. Thus, using the same principle, we proposed a novel Kernel Learning algorithm, called GIKL, that extends the Generalized MKL by Varma and Babu [27] to an infinite number of kernels. Moreover, our method stands out from [31] in the way to solve the subproblem (algorithm 2, line 6) as we apply a random search, which presents the advantage to be quick compared to the kernel gradient computation.

Besides, the work presented here is close to the study by Yger et al. [15], for it takes advantage of the same IKL framework. The distinction lies in the fact that we proposed the GIKL algorithm and in the goal: we chose to supply practitioners with interpretable tools, using Machine Learning to tune a well known TFR along with a SVM using a classical kernel. On the contrary, WKL learns a kernel parametrized by parts of wavelet transforms.

[1] Note that, as Daubechies wavelet transform is isometric, classifying in the wavelet domain boils down to classify in the Shannon (temporal) domain.

6. REFERENCES

[1] N. Saito and R.R. Coifman, "Local discriminant bases and their applications," *Journal of Mathematical Imaging and Vision*, vol. 5, pp. 337–358, 1995.

[2] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and Zisserman, "Supervised dictionary learning," in *Advances in Neural Information Processing Systems (NIPS)*, 2008.

[3] J. Winn, A. Criminisi, and T. Minka, "Object categorization by learned universal visual dictionary," in *IEEE International Conference on Computer Vision*, 2005.

[4] D. Blei and J. McAuliffe, "Supervised topic models," in *Advances in Neural Information Processing Systems (NIPS)*. MIT Press, 2008.

[5] A. Holub and P. Perona, "A discriminative framework for modelling object classes," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2005.

[6] J.A. Lasserre, C.M. Bishop, and T.P. Minka, "Principled hybrids of generative and discriminative models," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2006.

[7] R. Raina, Y. Shen, A.Y. Ng, and A. McCallum, "Classification with hybrid generative/discriminative models," in *Advances in Neural Information Processing Systems (NIPS)*. 2004.

[8] R. Salakhutdinov and G. Hinton, "Learning a nonlinear embedding by preserving class neighbourhood structure," in *Proceedings of the International Conference on Artificial Intelligence and Statistics*, 2007.

[9] H. Larochelle and Y. Bengio, "Classification using discriminative restricted boltzmann machines," in *Proceedings of the 25th International Conference on Machine Learning*, 2008.

[10] E. Jones, P. Runkle, N. Dasgupta, L. Couchman, and L. Carin, "Genetic algorithm wavelet design for signal classification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, pp. 890–895, 2001.

[11] D. J Strauss and G. Steidl, "Hybrid wavelet-support vector classification of waveforms," *Journal of Computational and Applied Mathematics*, vol. 148, pp. 375–400, 2002.

[12] D. J. Strauss, G. Steidl, and W. Delb, "Feature extraction by shape-adapted local discriminant bases," *IEEE Transactions on Signal Processing*, vol. 83, pp. 359–376, . 2003.

[13] D. Farina, O.F. do Nascimento, M.F. Lucas, and C. Doncarli, "Optimization of wavelets for classification of movement-related cortical potentials generated by variation of force-related parameters," *Journal of Neuroscience Methods*, vol. 162, pp. 357–363, 2007.

[14] D. Vautrin, X. Artusi, M.-F. Lucas, and D. Farina, "A novel criterion of wavelet packet best basis selection for signal classification with application to brain-computer interfaces," *IEEE Transactions on Biomedical Engineering*, vol. 56, pp. 2734–2738, 2009.

[15] F. Yger and A. Rakotomamonjy, "Wavelet kernel learning," *Pattern Recognition*, vol. 44, pp. 2614–2629, 2011.

[16] K. Huang and S. Aviyente, "Sparse representation for signal classification," in *Advances in Neural Information Processing Systems (NIPS)*, 2006.

[17] F. Rodriguez and G. Sapiro, "Sparse representation for image classification: Learning discriminative and reconstructive non-parametric dictionaries," Tech. Rep., University of Minnesota, 2008.

[18] M. Davy, A. Gretton, A. Doucet, and P. J. W. Rayner, "Optimized support vector machines for nonstationary signal classification," *IEEE Signal Processing Letters*, vol. 9, pp. 442–445, 2002.

[19] L. D. Vignolo, H. L. Rufiner, D. H. Milone, and J. Goddard, "Evolutionary cepstral coefficients," *Applied Soft Computing*, vol. 11, pp. 3419–3428, 2011.

[20] L. D. Vignolo, H. L. Rufiner, D. H. Milone, and J. Goddard, "Evolutionary splines for cepstral filterbank optimization in phoneme classification," *EURASIP Journal on Advances in Signal Processing. Biologically Inspired Signal Processing: Analysis, Algorithms and Applications*, vol. 2011, pp. 1–14, 2011.

[21] P.P. Vaidyanathan, *Multirate systems and filter banks*, Prentice Hall, 1993.

[22] H.H. Kha, H.D. Tuan, B.-N. Vo, and T. Q. Nguyen, "Symmetric orthogonal complex-valued filter bank design by semidefinite programming," *IEEE Transactions on Signal Processing*, vol. 55, pp. 4405–4414, 2007.

[23] H.H. Kha, H.D. Tuan, and T.Q. Nguyen, "Efficient design of cosine-modulated filter banks via convex optimization," *IEEE Transactions on Signal Processing*, vol. 57, pp. 966–976, 2009.

[24] S. Akkarakaran and P.P. Vaidyanathan, "Filterbank optimization with convex objectives and the optimality of principal component forms," *IEEE Transactions on Signal Processing*, vol. 49, pp. 100–114, 2001.

[25] J. Gauthier, L. Duval, and J.-C. Pesquet, "Optimization of synthesis oversampled complex filter banks," *IEEE Transactions on Signal Processing*, vol. 57, pp. 3827–3843, 2009.

[26] A. Rakotomamonjy, F. Bach, S. Canu, and Y. Grandvalet, "Simplemkl," *Journal of Machine Learning Research*, vol. 9, pp. 2491–2521, 2008.

[27] M. Varma and B.R. Babu, "More generality in efficient multiple kernel learning," in *Proceedings of the 26th International Conference on Machine Learning*, 2009.

[28] J. Shawe-Taylor and N. Cristianini, *Kernel methods for pattern analysis*, Cambridge University Press, 2004.

[29] V.N. Vapnik, *Statistical learning theory*, Wiley, 1998.

[30] G. Strang and T. Nguyen, *Wavelets and filter banks*, Wellesley-Cambridge Press, 1996.

[31] P.V. Gehler and S. Nowozin, "Infinite kernel learning," in *Advances in Neural Information Processing Systems (NIPS)*, 2008.

[32] T.W. Parks and C.S. Burrus, *Digital filter design*, Wiley, 1987.

[33] J. Nocedal and S.J. Wright, *Numerical optimization*, Springer, 2000.

[34] F. Bonnans and A. Shapiro, "Optimization problems with perturbations, a guided tour," *SIAM Journal on Scientific Computing*, vol. 40, pp. 228–264, 1996.

[35] R.G. Congalton, "A review of assessing the accuracy of classifications of remotely sensed data," *Remote Sensing of Environment*, vol. 37, no. 1, pp. 35 – 46, 1991.