Large Margin Filtering

Rémi Flamary, Devis Tuia, Benjamin Labbé, Gustavo Camps-Valls, Alain Rakotomamonjy

▶ **To cite this version:**

Rémi Flamary, Devis Tuia, Benjamin Labbé, Gustavo Camps-Valls, Alain Rakotomamonjy. Large Margin Filtering. IEEE Transactions on Signal Processing, 2012, 60 (2), pp.648 - 659. 10.1109/TSP.2011.2173685 . hal-00528917v2

HAL Id: hal-00528917
https://hal.science/hal-00528917v2

Submitted on 13 Oct 2011

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Large Margin Filtering

Rémi Flamary, Devis Tuia, *Member, IEEE*, Benjamin Labbé,
Gustavo Camps-Valls, *Senior Member, IEEE* and Alain Rakotomamonjy

***Abstract*—Many signal processing problems are tackled by filtering the signal for subsequent feature classification or regression. Both steps are critical and need to be designed carefully to deal with the particular statistical characteristics of both signal and noise. Optimal design of the filter and the classifier are typically aborded in a separated way, thus leading to suboptimal classification schemes. This paper proposes an efficient methodology to learn an optimal signal filter and a support vector machine (SVM) classifier jointly. In particular, we derive algorithms to solve the optimization problem, prove its theoretical convergence, and discuss different filter regularizers for automated scaling and selection of the feature channels. The latter gives rise to different formulations with the appealing properties of sparseness and noise-robustness. We illustrate the performance of the method in several problems. First, linear and nonlinear toy classification examples, under the presence of both Gaussian and convolutional noise, show the robustness of the proposed methods. The approach is then evaluated on two challenging real life datasets: BCI time series classification and multispectral image segmentation. In all the examples, large margin filtering shows competitive classification performances while offering the advantage of interpretability of the filtered channels retrieved.**

***Index Terms*—Sequence labeling, time series classification, large margin methods, support vector machine (SVM).**

I. Introduction

Sequence labeling is a classical pattern recognition problem in which the goal is to assign a label for every sample of a signal (or pixel of an image) while taking into account the sequentiality (or vicinity) of the samples. The field is very vast and typically arises in many signal recognition problems, such as Automatic Speech Recognition (ASR) [1], Brain Computer Interfaces (BCI) [2], or pathology discrimination from biosignals [3]. For instance, speaker diarization aims at recognizing which speaker is talking along time. Another example is the recognition of mental states from Electro-Encephalographic (EEG) signals. These mental states are then mapped into commands for a computer (virtual keyboard, mouse) or a mobile robot, thus creating the need for sample labeling algorithms [2], [4]. Electrocardiographic (ECG) signals are used to diagnose the presence or absence of a given pathology in advance, such as particular arrhythmia or fibrillation episodes [5]. Signal sequence labeling is sometimes referred to as time series (predictive) classification [6].

A widely used approach for performing sequence labeling consists in using Hidden Markov Models (HMMs) [7]. HMMs are probabilistic models that may be used for sequence decoding of discrete state observations. In the case of continuous observations such as signal samples or vector features extracted from the signal, Continuous Density HMMs are considered [7]. When using HMM for sequence decoding, one needs to know the conditional probability of the observations *per* hidden state (class), which is usually obtained through Gaussian Mixtures (GM) [7]. However, this kind of model leads to poor discrimination in high dimensional spaces, and recent works have shown that decoding accuracy may be improved by using discriminative models [8], [9]. Note that HMM require the use of the Viterbi algorithm in order to obtain the optimal sequence. However, such an off-line decoding supposes that the complete sequence of observations is available, situation that seldom occurs. For instance in BCI applications, a real time decision is often needed [2], [4], which precludes the use of standard Viterbi decoding. In these cases, a strategy which considers local Viterbi algorithm may be used for online decision [10]. Another possible online approach is to classify the current sample and the preceding decoded labels directly. These methods are defined in [11] as greedy decoding and permit the use of higher-order HMM taking into account several preceding states.

When the sequence labeling has to be performed on a measured signal, the efficiency of the classifier model highly depends on the type of noise induced by the measurement. This is why in most applications, the acquired signal is first preprocessed by filtering before being fed to a classifier. Even though this approach to sequence labeling typically yields good results, the crucial step of selecting and designing the filter is very often time consuming, needs prior knowledge and is scenario-dependent. Often, an optimal filter in the least-squares sense may not be optimal in terms of classification accuracy. Moreover, in many applications the filter is restricted to particular noise sources (typically Gaussian), while the classifier is not commonly adapted to the non-$i.i.d.$ nature of the signals. HMMs for instance adapt well to additive noise such as Gaussian noise, but they cannot take into account a time-lag between the labels and the discriminative features. If the labels and the features are not re-synchronized, some of the learning observations are mislabeled, leading to a biased density estimation *per* class. This kind of dephasing is a classical simple case of convolutional noise (e.g convolution by a delayed Dirac's delta). This is a problem in BCI applications where the interesting information is not always synchronized with the labels. For instance, since the neural activity precedes

Manuscript received October 2010;

RF, BL and AR are with Laboratoire LITIS - EA 4108 Université de Rouen Avenue de l'Université, 76801 Saint-Étienne-du-Rouvray. E-mail: {remi.flamary,benjamin.labbe,alain.rakoto}@insa-rouen.fr, http://remi.flamary.com

DT was with the Image Processing Laboratory (IPL), Universitat de Valéncia, Spain. He is now with the Laboratoire des Systèmes d'Information Géographiques, Lausanne Institute of Technology (EPFL), 1015 Lausanne, Switzerland. E-mail: devis.tuia@epfl.ch

GCV is with the Image Processing Laboratory (IPL). Universitat de València. C/ Catedrático Escardino, Paterna (València) Spain. E-mail: gustavo.camps@uv.es, http://www.uv.es/gcamps

the actual movement, authors in [12] showed the need for applying delays to the signal. These delays are typically selected heuristically or through cross-validation strategies. Another example is found in the interaction with a computer using multi-modal acquisitions (e.g. EEG and EMG). Since each modality has its own time-lag with respect to neural activity [13], it may be difficult to manually synchronize them. Better adaptation could be obtained by learning the "best" time-lag to apply to each channel. Note that correcting time-lags boils down to applying filters in the same way as denoising a signal by an *ad hoc* filtering.

In the general case, the labeling problem is not restricted to unidimensional signals. A typical multidimensional problem that involves signal sequence labeling is segmentation by pixel labeling [14]–[16]. Images, like time series and data sequences, are not $i.i.d.$ data. Natural images are smooth, autocorrelation functions are broad, and have a $1/f$ band-limited spectrum. In the case of color images, the correlation between the tristimulus values of the natural colors is high. Such a characterization is more difficult in the case of multi- and hyper-spectral images acquired by satellite sensors. Although images are not $i.i.d.$ data, image segmentation algorithms are commonly applied either to single pixels (hence obviating the spatial correlation), to low-pass filtered pixels (imposing an *ad hoc* spatial arrangement), or to small patches (assuming a spatial extent of pixel relations). In remote sensing image processing, spatial filters for taking into account neighboring relations have been addressed through textural [17] and morphological filters [18]–[20]. The extracted features are then fed to the classifier. Again, both processes are optimized separately, and then no guarantee of optimal performance is attained.

In this paper, we propose to learn the filter directly from samples, instead of using a fixed filter as a preprocessing stage. This approach may help in adapting to signal and noise characteristics of each channel in addition to alleviate the time-lag misadjustment. The idea of jointly optimizing a filter and a classifier dates back to the nineties within the field of artificial neural networks for time series processing. Two methods are worth mentioning. The convolutional neural networks [21] are particular multilayer perceptrons for sequential classification of handwritten and machine-printed characters. Another example is the focused-gamma neural network [22], [23], which includes in the first input layer a linear gamma-filter, that is an infinite impulse response (IIR) filter with controlled stability and memory depth. In both cases, network and filter weights are adjusted by standard backpropagation algorithms. In [24], we proposed a method to learn a large margin filtering for linear SVM classification. Here, we extend this preliminary work by formalizing the problem of *large margin filter learning*. The idea is to learn a Finite Impulse Response (FIR) filter for each channel of the signal jointly with a classifier. This approach can adapt to different properties in the channels and the learned filter corresponds to a convolution maximizing the margin between classes. Since the filter is accessible and visualizable, it can be interpreted in both the temporal and the frequency domains. We also extend the method to the non-linear case and propose different regularization types to promote channel scaling or selection.

The remainder of the paper is organized as follows. In Section II, we formalize the problem and review traditional approaches such as filteredsample classification and time-window classification. In Section III, we introduce the problem of large margin filtering to deal with the limitations observed when considering non-$i.i.d.$ signals. We also discuss its theoretical convergence properties, as well as its computational complexity and the effect of different regularization for the filtering matrix. In Section IV, the different proposed approaches are tested on three classification scenarios. First, a toy dataset accounting for both additive and convolutional noise. Then, a real-life BCI classification problem and a multispectral image segmentation problem are considered. Section V concludes the paper.

II. Sample Labeling Problem

In this section, we formally state the problem of sample labeling. Then, we define the filtering of a multi-dimensional signal and the SVM classifier for filtered samples. The problem is stated for the general case using mapping functions in appropriate reproducing kernel Hilbert spaces and both linear and nonlinear problems are discussed.

A. Problem definition

We want to predict a sequence of labels either from a multi-channel signal or from multi-channel features extracted from that signal by learning from examples. We consider that the training samples are gathered in a matrix $\mathbf{X} \in \mathbb{R}^{n \times d}$ containing d channels and n samples. $\mathbf{X}_{i,j}$ is the value of channel j for the i^{th} sample ($\mathbf{X}_{i,\cdot}$). The vector $\mathbf{y} \in \{-1, 1\}^n$ contains the class for each sample. Later on, multiclass problems will be handled by means of pairwise binary classifiers.

In order to reduce noise in the samples or variability in the features, a usual approach is to filter $\mathbf{X}$ before learning the classifier. In the literature, a single filter is typically used for all channels although there is no reason for believing that such a single filter will lead to an optimal classification performance. Moreover, assuming an explicit filter structure may not fit the underlying system that generated the data. The Savitzky-Golay [12] or the gamma filters [25] are examples of structures commonly used for noise reduction before classification. Let us define the filter applied to $\mathbf{X}$ by the matrix $\mathbf{F} \in \mathbb{R}^{f \times d}$. Each column of $\mathbf{F}$ is a filter for the corresponding channel in $\mathbf{X}$ and f is the size of the Finite Impulse Response (FIR) filters.

We define the filtered data matrix $\widetilde{\mathbf{X}}$ by:

$$\widetilde{\mathbf{X}}_{i,j} = \sum_{m=1}^{f} \mathbf{F}_{m,j}\ \mathbf{X}_{i+1-m+n_0,j} = \mathbf{X}_{i,j} \otimes \mathbf{F}_{\cdot,j}$$

where the sum is a uni-dimensional convolution ($\otimes$) of each channel by the filter in the appropriate column of $\mathbf{F}$. Here, n_0 is the delay of the filter: for instance $n_0 = 0$ corresponds to a causal filter and $n_0 = f/2$ corresponds to a non-causal filter centered on the current sample. Figure 1 presents an example of signal $\mathbf{X}$ and filtered signal $\widetilde{\mathbf{X}}$.

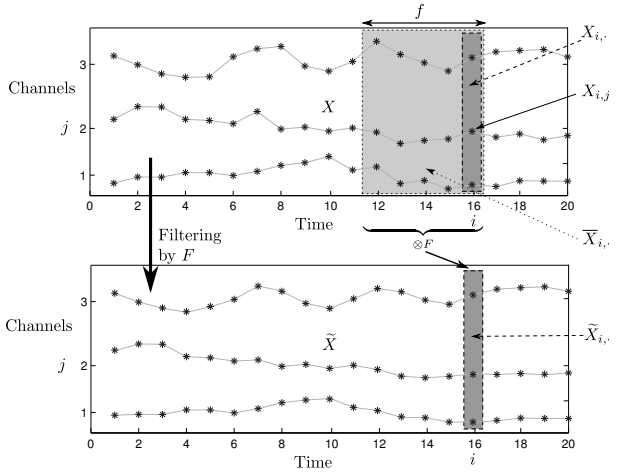

Fig. 1. Data matrix $\mathbf{X}$ (top), filtered matrix $\widetilde{\mathbf{X}}$ (bottom) and time-window $\overline{\mathbf{X}}$ (light gray) with $n_0 = 0, d = 3$ and $f = 5$.

In this work, we make the hypothesis that the class labels of the samples vary at a slow rate. This means that the signal is supposed to be composed of several large segments of signals from the same class. In this context, we aim at learning a filter that enhances the discrimination between the training examples. The filter size does not need to be accurately chosen, since the regularization term used in our optimization framework tends to downweight irrelevant filter coefficients. We will discuss this property later in Section III-E and in the experimental Section IV-A.

B. SVM for filtered samples

To improve the classification rate, one may filter the channels in $\mathbf{X}$ in order to reduce noise pertubation. The usual filter in the case of high frequency noise is the averaging filter defined by $\mathbf{F}_{i,j} = 1/f, \forall i \in \{1, \ldots, f\}$ and $j \in \{1, \ldots, d\}$, while n_0 is selected depending on the problem at hand (i.e. $n_0 = 0$ for a causal filtering or $n_0 > 0$ for a non-causal filtering). In the following, the method which considers a moving average filter for signal preprocessing followed by a SVM classifier is denoted as *Avg-SVM*.

Once the filtering is chosen, we can learn a SVM sample classifier on the filtered samples by solving the optimization problem:

$$\min_{g} \left\{ \frac{1}{2}\|g\|^2_{\mathcal{H}} + \frac{C}{n}\sum_{i=1}^{n} H(\mathbf{y}_i, g(\widetilde{\mathbf{X}}_{i,\cdot})) \right\} \tag{1}$$

where C is a regularization parameter, $g(\cdot) \in \mathcal{H}$ is the decision function in a Reproducing Kernel Hilbert Space $\mathcal{H}$, $H(y, g(x)) = \max(0, 1 - y \cdot g(x))^p$ is the Hinge loss to the power of p ($p = 1$ corresponds to ℓ_1-SVM and $p = 2$ to ℓ_2-SVM [26]). For the ℓ_1-SVM, one can solve the dual of this problem:

$$\max_{\boldsymbol{\alpha}} \left\{ J_{SVM}(\boldsymbol{\alpha}, \mathbf{F}) = -\frac{1}{2}\sum_{i=1,j=1}^{n,n} \mathbf{y}_i\mathbf{y}_j\boldsymbol{\alpha}_i\boldsymbol{\alpha}_j\widetilde{\mathbf{K}}_{i,j} + \sum_{i=1}^{n}\boldsymbol{\alpha}_i \right\} \tag{2}$$

$$\text{s.t.} \quad \frac{C}{n} \geq \boldsymbol{\alpha}_i \geq 0 \quad \forall i \quad \text{and} \quad \sum_{i=1}^{n}\boldsymbol{\alpha}_i\mathbf{y}_i = 0$$

where $\boldsymbol{\alpha}_i$ are the dual variables and $\widetilde{\mathbf{K}}$ is the kernel matrix for filtered samples. For reproducing kernel Hilbert space related to the Gaussian kernel, $\widetilde{\mathbf{K}}$ is defined by:

$$\widetilde{\mathbf{K}}_{i,j} = k(\widetilde{\mathbf{X}}_{i,\cdot}, \widetilde{\mathbf{X}}_{j,\cdot}) = \exp\left(-\frac{\|\widetilde{\mathbf{X}}_{i,\cdot} - \widetilde{\mathbf{X}}_{j,\cdot}\|^2}{2\sigma^2}\right) \tag{3}$$
$$= \exp\left(-\frac{\sum_{m=1}^{d}\|(\mathbf{X}_{i,m} - \mathbf{X}_{j,m}) \otimes \mathbf{F}_{\cdot,m}\|^2}{2\sigma^2}\right)$$

where σ is the kernel bandwidth. Note that for any FIR filter, the resulting matrix $\widetilde{\mathbf{K}}$ is always positive definite as long as the kernel $k(\cdot,\cdot)$ is positive definite. Indeed if $k(\cdot,\cdot)$ is a kernel from $\mathcal{X} \times \mathcal{X}$ to $\mathbb{R}$ and ϕ is a mapping from any $\mathcal{X}'$ to $\mathcal{X}$ then $k'(\cdot,\cdot) = k(\phi(\cdot), \phi(\cdot))$ is a positive definite kernel [26]. In our case, since a FIR filter computes a linear combination of $\mathbb{R}^d$ elements, and the Gaussian kernel takes elements from $\mathbb{R}^d$, the kernel defined equation (3) is positive definite

Once the classifier is learned, the decision function for a new filtered (test) signal $\widetilde{\mathbf{X}'}$ at sample i is:

$$g(\widetilde{\mathbf{X}'}_{i,\cdot}) = \sum_{j=1}^{n}\boldsymbol{\alpha}_j\mathbf{y}_j k(\widetilde{\mathbf{X}'}_{i,\cdot}, \widetilde{\mathbf{X}}_{j,\cdot}) + b \tag{4}$$

with $\boldsymbol{\alpha}_j$ are the dual variables learned by solving (2) and b represents the bias term. We show in the experiments section that this approach may lead to improved performance over the non-filtered approach. However, the method relies on the (critical) choice of a filter structure which in turn depends on prior information or user's knowledge. We will show latter that the filters learned when optimizing a large margin criterion will naturally lead to better discriminative power.

C. Time-Window Classification

Another way for taking into account the sequentiality of the samples, *i.e.* for handling the non-*i.i.d.* characteristics of the time-series, is to classify time-windows of samples. Let us define $\{\overline{\mathbf{X}}_{i,\cdot}\}_{i=1}^{n}$ the set of samples obtained from a complete time window of length f with n_0 delay with each $\overline{\mathbf{X}}_{i,\cdot} \in \mathbb{R}^{f \times d}$ being built by concatenating all samples $\mathbf{X}_{i,\cdot}$ in the window (see Fig. 1). This approach leads to the classification of data in high dimension $f \times d$, and one can learn a SVM classifier on samples $\overline{\mathbf{X}}_{i,\cdot}$ using Equation (1). This method will be called *Win-SVM* hereafter.

1) Linear Win-SVM: For learning a linear classifier on a window of samples, the problem may be expressed as the minimization of:

$$J_W(\mathbf{W}, w_0) = \frac{1}{2}\|\mathbf{W}\|^2_F + \frac{C}{n}\sum_{i=1}^{n} H(\mathbf{y}, g_W(\overline{\mathbf{X}}_{i,\cdot})) \tag{5}$$

where $\|\mathbf{W}\|_F^2 = \sum_{i,j} \mathbf{W}_{i,j}^2$ is the squared Frobenius norm of $\mathbf{W}$, C is a regularization term to be tuned and $g_W(\overline{\mathbf{X}}_{i,\cdot})$ is the decision function defined for the i^{th} time-window sample as:

$$g_W(\overline{\mathbf{X}}_{i,\cdot}) = \sum_{m=1}^{f}\sum_{j=1}^{d} \mathbf{W}_{m,j}\mathbf{X}_{i+1-m+n_0,j} + b$$

where $\mathbf{W} \in \mathbb{R}^{f\times d}$ and $b \in \mathbb{R}$ are the classification parameters. In a nutshell, the problem defined in Equation (5) is simply a SVM problem where examples are matrices instead of vectors. Since we deal with a linear decision function, the problem can be vectorized without loss of generality and linear SVM solvers can be used for its resolution. Furthermore, by setting $p = 2$, the objective function becomes differentiable so that efficient algorithms such as the one proposed by Chapelle [27] can be considered. Using that solver, *Win-SVM* complexity is about $\mathcal{O}(nf^2d^2)$ which scales quadratically with the filter dimension.

One of the interests of this time-windowing approach is that the matrix $\mathbf{W}$ can be also interpreted as a large margin filter. Indeed, the columns of the $\mathbf{W}$ matrix may be viewed as a temporal filtering whereas the rows correspond to a spatial filtering. However, this approach may still lead to sub-optimal classification performance as previously reported for high-dimensional signals [24]. As a matter of fact, the Frobenius norm presents several shortcomings. The first one is that it does not take into account the signal structure of the problem, which means that elements of matrix $\overline{\mathbf{X}}$ are considered independently to each other. Secondly, the Frobenius norm does not promote sparsity which, in high-dimensional noisy problems, may help selecting relevant coefficients of $\overline{\mathbf{X}}$.

While we have essentially focused on linear time-window classifier, it is worthwhile to note that non-linear kernel based SVM can also be used for classifying time-window $\{\overline{\mathbf{X}}_{i,\cdot}\}_i$ examples, as we will do in the experimental section. However, in such a situation, we lose the interpretability of $\mathbf{W}$ as a temporal/spatial filter.

2) Channel selection for linear Win-SVM: In some applications, only a subset of the acquired channels may be useful for the classification task. This situation occurs for instance in BCI problems, where discriminative features are usually spatially localized. In these cases, selecting the relevant channels leads to better interpretability and discrimination of the model. To include an automated channel selection procedure in the time-window classification problem given in (5), we propose to consider a $\ell_1 - \ell_2$ mixed norm as a regularizer instead of the Frobenius norm:

$$\Omega_{1-2}(\mathbf{W}) = \sum_{j=1}^{d}\left(\sum_{i=1}^{f}\mathbf{W}_{i,j}^2\right)^{\frac{1}{2}} = \sum_{j=1}^{d} h\left(\|\mathbf{W}_{\cdot,j}\|^2\right) \quad (6)$$

where $h(u) = u^{\frac{1}{2}}$ is the square root function. This mixed-norm acts as an ℓ_2-norm on each single channel filter, while the ℓ_1-norm of each channel filter energy will induce sparsity over channels. The resulting optimization problem

$$\min_{\mathbf{W},b}\left\{\Omega_{1-2}(\mathbf{W}) + \frac{C}{n}\sum_{i=1}^{n} H(\mathbf{y}, g_W(\overline{\mathbf{X}}_{i,\cdot}))\right\}$$

has a non-differentiable objective function even when $p = 2$, which may pose some numerical difficulties. However, recent research has considered this problem of hybrid objective functions, where one part has a Lipschitz gradient and the other is convex but non-differentiable [28], [29]. Here we straighforwardly applied the accelerated gradient method (AGP) proposed by Chen et al. [30] since the squared hinge loss is known to have Lipschitz gradient . For more details about the algorithm, the reader is referred to [30].

III. Large Margin Filtering

The classification of a time window is a way to handle temporal information, but the classifier model still considers all the time samples independently without taking into account the signal structure. Performing a per-channel convolution is sensible here as it will take into account the channel structure and extract the discriminative information spread along time. In this section, we present the proposed optimization problem for learning a large margin filtering as well as a general algorithm to solve general-purpose sequence labeling problems. Then we detail the implementation of the method, named KF-SVM and provide some insights on the convergence properties of the algorithm. Finally, we discuss the use of different regularizers and the related works.

A. Optimization problem

Jointly learning the filtering matrix $\mathbf{F}$ and the classifier leads to a filter maximizing the margin between the classes in the feature space. The problem we want to solve is:

$$\min_{g,\mathbf{F}}\left\{\frac{1}{2}\|g\|_{\mathcal{H}}^2 + \frac{C}{n}\sum_{i=1}^{n} H(\mathbf{y}_i, g(\widetilde{\mathbf{X}}_{i,\cdot})) + \lambda\Omega(\mathbf{F})\right\} \quad (7)$$

where λ is a regularization parameter and $\Omega(\cdot)$ represents a differentiable regularization function of $\mathbf{F}$. Note that the first two terms of (7) reduces to a standard SVM for filtered samples $\widetilde{\mathbf{X}}$ as defined in Equation (1). However, here $\mathbf{F}$ is a variable to be minimized instead of being a fixed filter structure. When jointly optimizing over the decision function g and the filter $\mathbf{F}$, the objective function is typically non-convex, for instance when the kernel of the RKHS $\mathcal{H}$ is a Gaussian kernel. Even in very simple situations, for instance the linear case with $j = 1$ and $f = 1$, it can be shown that the problem is non-convex.

However, the problem defined by (7) is convex w.r.t. $g(\cdot)$ for any fixed filter $\mathbf{F}$ and in such a case, it boils down to solving the SVM problem. Therefore, in order to take into account this specific structure of the problem, we propose to solve the problem through the following 2-stage approach :

$$\min_{\mathbf{F}}\{J(\mathbf{F})\} = \min_{\mathbf{F}}\{J'(\mathbf{F}) + \lambda\Omega(\mathbf{F})\} \quad (8)$$

where $J'(\mathbf{F})$ is the objective value of the following primal problem

$$J'(\mathbf{F}) = \min_{g}\left\{\frac{1}{2}\|g\|_{\mathcal{H}}^2 + \frac{C}{n}\sum_{i=1}^{n} H(\mathbf{y}_i, g(\widetilde{\mathbf{X}}_{i,\cdot}))\right\} \quad (9)$$

Algorithm 1 KF-SVM CG solver

Set $\mathbf{F}_{l,k} = 1/f$ for $k = 1, \cdots, d$ and $l = 1, \cdots, f$
Set $i = 0$, Set $D_F^0 = 0$
repeat
  $i = i + 1$
  $G_{\mathbf{F}}^i \leftarrow$ gradient of $J'(\mathbf{F}) + \lambda\Omega(\mathbf{F})$ w.r.t. $\mathbf{F}$
  $\beta \leftarrow \frac{\|G_{\mathbf{F}}^i\|^2}{\|G_{\mathbf{F}}^{i-1}\|^2}$ (Fletcher and Reeves)
  $D_{\mathbf{F}}^i \leftarrow -G_{\mathbf{F}}^i + \beta D_{\mathbf{F}}^{i-1}$
  $(\mathbf{F}^i, g*) \leftarrow$ Line-Search along $D_{\mathbf{F}}^i$
until Stopping criterion is reached

and its corresponding dual problem is

$$J'(\mathbf{F}) = \max_{C/n \geq \boldsymbol{\alpha} \geq 0, \sum_i \boldsymbol{\alpha}_i \mathbf{y}_i = 0} \left\{ J_{SVM}(\boldsymbol{\alpha}, \mathbf{F}) \right\} \qquad (10)$$

where J_{SVM} is defined by Equation (2) and $g(\cdot)$ by Equation (4). Due to the strong duality of the SVM problem, $J'(\cdot)$ can be expressed in either his primal or dual form (see (9) and (10)). The objective function $J(\cdot)$ defined in (8) is nonlinear and non-convex. Nevertheless, Since, for any filter $\mathbf{F}$, the related SVM problem given by Equation (9) has an unique solution $g^*(\cdot)$, according to [31], $J'(\cdot)$ is differentiable w.r.t. $\mathbf{F}$. The gradient of $J(\cdot)$ can thus be computed in closed-form as detailed in the sequel. Hence, since we have the gradient of the nonlinear and non-convex objective function $J(\mathbf{F})$, we propose to solve problem (8) by means of a gradient-descent method for unconstrained optimization.

B. KF-SVM Solver

For solving the optimization problem, we propose a conjugate gradient (CG) descent algorithm along $\mathbf{F}$ with a line search method satisfying the Wolfe's condition [32]. The method is detailed in Algorithm 1, where β is the CG update parameter and $D_{\mathbf{F}}^i$ the descent direction for the i^{th} iteration. For the experimental results, we used the β proposed by Fletcher and Reeves (see [32], [33] for more information). The iterations in the algorithm may be stopped by two stopping criteria: a threshold on the relative variation of $J(\mathbf{F})$ or on the norm of the variation of $\mathbf{F}$.

Note that for each computation of $J(\mathbf{F})$ in the line search, the optimal g^* is found by solving a SVM. A similar approach has been considered for solving multiple kernel problems [34], [35]. In these works, an objective function was minimized with respect to kernel parameters (kernel weight in [34] or bandwidth in [35]) using a gradient descent algorithm.

Instead of solving problem (7) through a min-max approach, we could have considered a gradient descent approach on joint parameters $\mathbf{F}$ and $g(\cdot)$. However, such an approach presents several disadvantages over the chosen one. First of all, it does not take into account the structure of the problem which is the well-studied SVM optimization problem for a fixed $\mathbf{F}$. Hence, by separating the optimization over $\mathbf{F}$ and over $g(\cdot)$, we are able to take advantage of the SVM optimization framework and any improvements made to SVM solvers. Furthermore, as stated by Chapelle et al. [27], addressing the nonlinear SVM problem directly in the primal does not lead to improved computational efficiency: therefore, no speed gain should be expected by solving problem (7) directly.

C. Convergence to a local minimum

In this section, we discuss the global convergence of the proposed Conjugate Gradient algorithm for solving the KF-SVM problem. The main technical difficulty of this proof of convergence is the fact that our objective function is itself the minimum of another optimization problem. For a sake of clarity and simplicity, we suppose that :

(a) the SVM problem (9) is solved exactly to obtain $0 \leq \boldsymbol{\alpha}^* \leq C/n$. This hypothesis ensures us that the gradient $\nabla J(\cdot)$ is exact.
(b) $\mathbf{X} \in \mathcal{X}$, with $\mathcal{X}$ being a compact set of $\mathbb{R}^{n \times d}$. For the data that we consider, *e.g* numerical signals and images, this hypothesis is guaranteed by the data acquisition process which ensures us that the signal and image values are bounded.
(c) the regularization parameter λ is strictly positive.
(d) the kernel considered is continuous and twice differentiable over the set $\mathcal{X}$. For instance the Gaussian kernel given in Equation (3) satisfies this condition.
(e) the Frobenius norm is considered as the regularization term hence, $\Omega(\mathbf{F}) = ||\mathbf{F}||_F^2$.

Now, given these hypotheses, the convergence of our algorithm depends on standard convergence conditions of conjugate gradients algorithms [32]. Hence, if

1) the level set $\mathcal{L} := \{\mathbf{F} | J(\mathbf{F}) \leq J(\mathbf{F}^0)\}$ is bounded for the starting point $\mathbf{F}^0$.
2) in some open neighborhood $\mathcal{N}$ of $\mathcal{L}$, the objective function J is Lipschitz continuously differentiable, which means that if J is in addition twice differentiable :

$$\exists L \in \mathbb{R}^+, \|\nabla^2 J(\mathbf{F})\|_2 \leq L \quad \forall \mathbf{F} \in \mathcal{N}$$

where $\|\cdot\|_2$ represents the matrix norm induced by the ℓ_2 norm.
3) a line-search satisfying the strong Wolfe's conditions is used,

then the conjugate gradient in Algorithm 1 converges globally [32], *i.e.* it converges to a local minimum of our objective function.

The proof that the two conditions hold can be found in Appendix A. Note that for the sake of simplicity, we have restricted ourselves to the use of the Frobenious regularizer and some specific kernels, however this technical result can be readily extended to other regularizers.

Furthermore, the proven Lipschitz gradient property of $J(\cdot)$ with respects to kernel parameters (Condition 2) is also of interest in other problems such as multiple kernel learning [34] as it allows the use of efficient algorithms such as the fast iterative shrinkage thresholding algorithms [36] for solving the convex MKL problem.

D. Complexity in the linear and nonlinear cases

At each iteration of the algorithm, the gradient of $J'(\mathbf{F}) + \lambda\Omega(\mathbf{F})$ is computed and a SVM is solved at each function

evaluation in the line-search. In this paragraph, we discuss the complexity of these tasks in both the linear and nonlinear cases.

In the linear case, the optimization problem can be more efficiently solved in the primal when the dimension is lower than the number of training examples, which is the most frequent situation in sequence labeling. In this case, the SVM decision function is a separation hyperplane defined by $d+1$ parameters while in the dual, one needs $n+1$ parameters to express the decision function. Several efficient solvers have been proposed in the literature [27], [37] for solving the SVM problem in the linear case with $p = \{1, 2\}$. In order to have a differentiable objective function, we have set $p = 2$. For computing $J'(\mathbf{F})$, we used the method proposed by Chapelle et al. [27], which learns the SVM classifier by using a CG descent or a Newton descent algorithm. For this linear case, it can be shown that the gradient of $J'(\cdot)$ at point $\mathbf{F}$ is:

$$\nabla J'(\mathbf{F})_{i,j} = -\frac{2C}{n}\sum_{o=1}^{n}\mathbf{y}_o(\mathbf{X}_{o-i+1+n_0,j}) \times H(\mathbf{y}_o, g^*(\widetilde{\mathbf{X}}_{o,.}))$$

where g^* is the optimal linear function for the fixed filtering matrix $\mathbf{F}$. The complexity of computing the gradient is $\mathcal{O}(n \times f \times d)$.

In the non linear case, with a Gaussian kernel for instance, the problem has to be solved in its dual form. In the dual, the choice of p does not impact on the differentiability of $J'(\mathbf{F})$ as long as the optimization problem related to $J'(\mathbf{F})$ is strictly convex [31]. The gradient of $J'(\cdot)$ at a given point $\mathbf{F}$ for a Gaussian kernel is obtained easily by considering all the parameters related to the SVM as constant w.r.t. to $\mathbf{F}$. Thus, the gradient becomes

$$\nabla J'(\mathbf{F})_{i,j} = \frac{1}{2\sigma}\sum_{o=1,p=1}^{n,n}(\mathbf{X}_{o+1-i,j} - \mathbf{X}_{p+1-i,j}) \times (\widetilde{\mathbf{X}}_{o,j} - \widetilde{\mathbf{X}}_{p,j})\widetilde{\mathbf{K}}_{o,p}\mathbf{y}_o\mathbf{y}_p\boldsymbol{\alpha}_o^*\boldsymbol{\alpha}_p^* \quad (11)$$

where $\boldsymbol{\alpha}^*$ are the optimal Lagrangian dual variables of the SVM solution for $\widetilde{\mathbf{X}}$ signal given the filter $\mathbf{F}$ and $\widetilde{\mathbf{K}}$ is the kernel matrix of the filtered samples $\widetilde{\mathbf{X}}_{i,.}$. The complexity of computing this gradient is $\mathcal{O}(n^2 \times f \times d)$. In practice, since SVMs have a sparse representation, the gradient computation reduces to $\mathcal{O}(n_s^2 \times f \times d)$ with n_s being the number of support vectors.

Due to the non-convexity of the objective function, it is difficult to provide an exact evaluation of the algorithm complexity. However, we know the complexity of the gradient computation in the linear and non-linear cases. Moreover, for each evaluation of $J(\mathbf{F})$ in the line search, a $\mathcal{O}(n \times f \times d)$ filtering is applied, and a SVM has to be solved (n parameters in the non linear case and d in the linear case). For further speed-up, one may use previous result of the SVM solver as starting point for the new problem and then iterate.

E. Filter regularization

In this section, we discuss the choice of the filter regularization term $\Omega(\mathbf{F})$ in Equation (7). This choice is crucial because learning the FIR filters adds parameters to the learning problem and regularization is essential in order to avoid overfitting. The first regularization term for the filter that we consider and use in our KF-SVM framework is the Frobenius norm:

$$\Omega_2(\mathbf{F}) = \sum_{i=1,j=1}^{f,d}\mathbf{F}_{i,j}^2$$

This regularization term is differentiable and its gradient is easy to compute. Minimizing this regularization term corresponds to minimizing the filter energy , i.e. to maximize the attenuation of the filters. This attenuation may be seen as a scaling for each signal in the Gaussian kernel (See Equation (3)) and minimizing this scaling will lead to a Gaussian kernel with larger bandwidth, hence to a smooth decision function. In this sense, the filter matrix can be seen as a kernel parameter that weights delayed samples and scales channels. For a given channel, such a sequential weighting is related to a phase/delay and cut-off frequency of the filter. The intuition of how this regularization term influences the filter learning is the following. Suppose we learn our decision function $g(\cdot)$ by minimizing only $J'(.)$. Then, the learned filter matrix will maximize the margin between classes. Adding the Frobenius regularizer will force non-discriminative filter coefficients to shrink to zero thus yielding to a reduced impact on the kernel of the related delayed samples.

Using this regularizer, all filter coefficients are treated independently, and even if it tends to down-weight some non-relevant channels, the resulting filter coefficients are not sparse. If we want to perform a channel selection while learning the filter $\mathbf{F}$, we have to force some columns of $\mathbf{F}$ to be zero. For that, we can use the $\ell_1-\ell_2$ mixed-norm defined in Equation (6) as a regularizer. However, this regularization term is not differentiable and the solver proposed in Algorithm 1 cannot be used. The AGP methods proposed in Section II-C2 cannot be used either due to the non convexity of the objective function $J'(\cdot)$. In order to use the $\ell_1 - \ell_2$ mixed-norm, we address the problem through a Majorization-Minimization algorithm [38] that enables to take advantage of the KF-SVM solver proposed above. The idea here is to iteratively replace the function $h(\cdot)$ defined in (6) by a majorization and then to minimize the resulting objective function. Since $h(u)$ is concave in its positive orthant, we consider the following linear majorization of $h(\cdot)$ at a given point u_0:

$$\forall u > 0, \qquad h(u) \leq u_0^{\frac{1}{2}} + \frac{1}{2}u_0^{-\frac{1}{2}}(u - u_0)$$

The main advantage of a linear majorization is that we can reuse the KF-SVM algorithm. Indeed, at iteration $k+1$, applying this linear majorization of $h(\|\mathbf{F}_{.,j}\|_2)$ around a $\|\mathbf{F}_{.,j}^{(k)}\|_2$ yields to a Majorization-Minimization algorithm for sparse filter learning, which consists in solving:

$$\min_{\mathbf{F}^{(k+1)}}\left\{J'(\mathbf{F}) + \lambda\Omega_{\mathbf{d}}(\mathbf{F})\right\}$$

$$\text{with } \Omega_{\mathbf{d}}(\mathbf{F}) = \sum_{j=1}^{d} d_j \sum_{i=1}^{f}\mathbf{F}_{i,j}^2 \text{ and } d_j = \frac{1}{\|\mathbf{F}_{.,j}^{(k)}\|_2}$$

where $\mathbf{F}^{(k)}$ represents the solution at iteration k, and $\Omega_{\mathbf{d}}$ is a weighted Frobenius norm. Note that this regularization term is

Algorithm 2 SKF-SVM solver

Set $\mathbf{F}_{i,j} = 1/f$ for $i = 1, \cdots, f$ and $j = 1, \cdots, d$
Set $\mathbf{d}_j = 1$ for $j = 1, \cdots, d$
repeat
  $(\mathbf{F}, \boldsymbol{\alpha}) \leftarrow$ Solve KF-SVM with $\mathbf{d}$ column weights
  $\mathbf{d}_j \leftarrow \frac{1}{\|\mathbf{F}_{:,j}\|_2}$ for $j = 1 \cdots d$
until Stopping criterion is reached

differentiable and the KF-SVM solver can then be used. We call this method Sparse KF-SVM (SKF-SVM) and the solver is detailed in Algorithm 2. We use here a similar stopping criteria to that in Algorithm 1.

Globally, the use of regularizers such as the two presented above attenuates the effect of samples or channels that are not discriminative. Hence, these regularizers and the regularization parameter λ define an implicit way for selecting the size f of the filter. If we set f to a sufficiently large value and select appropriately the regularization parameter λ, for instance by cross-validation, then the filter coefficients related to irrelevant samples or channels will tend to shrink towards zero. Nevertheless, the size of the filter should not be too large either due to the non-convexity of the problem. Indeed the proposed initialization, average filtering, might be really far from the optimal value and get stuck in a local minimum when the length of the initial average filtering is too important. In practice, we suggest to select f either by a coarse validation method or by setting it as the filter length that maximizes the performance for an averaging filter approach coupled with an SVM. Our numerical experiments show that these two approaches usually lead to good classification accuracy.

F. Related works

Works on Common Spatio-Spectral Patterns (CSSSP) [39] are probably the most similar to the ones proposed in this paper. In these works, the aim is to learn a linear combination of channels and samples that optimizes a separability criterion. But the criterion optimized by CSSSP and KF-SVM are different: CSSSP aims at maximizing the variance of the samples for the positive class while minimizing the variance for the negative class, whereas KF-SVM aims at maximizing the margin between classes in the feature space. Furthermore, CSSSP is a feature extraction algorithm that is independent of the classifier used, while in our case we learn a filter that is tailored to the (nonlinear) classification algorithm criterion. This is a similar situation with the recently presented kernel signal-to-noise ratio [40], in which one maximizes the ratio between the signal and the noise variances in a kernel feature space. Furthermore, the filter used in KF-SVM is not restricted to signal time samples but can also be applied to complex sequential features extracted from the signal (*e.g.*, PSD). An application to this kind of complex data is provided in the experimental section.

KF-SVM can also be seen as a kernel learning method. The filter coefficients can be interpreted as kernel parameters despite the fact that samples are non-*i.i.d.*. Learning such kernel parameters is now a common approach introduced by [41]. While Chapelle et al. minimize a bound on the generalization error by gradient descent, in our case we simply minimize the SVM objective function. Also, it is worth noting that the influence on the parameters differs in both approaches. More precisely, if we focus on the columns of $\mathbf{F}$, we notice that the coefficients of these columns act as a scaling of the channels. For a filter of size 1, our approach would correspond to adaptive scaling as proposed by [42]. In their work, the authors jointly learn the classifier and the Gaussian kernel parameter σ in a SVM framework together with a sparsity constraint on σ leading thus to automated feature selection. KF-SVM can thus be seen as a generalization of this approach, which takes into account sample sequentiality as well.

In addition to being a kernel learning method, KF-SVM can address the problem of optimal filter design. Efficient solutions are nowadays available for simple cases such as the maximization of signal-to-noise ratio (denoising task). As a consequence, more recent researches focus on the design of optimal filters for specific situations [43], [44]. However, most of these works deal with signal denoising, source separation or frequency estimation. In this work, we focus on the problem of optimal filter design for sample labeling which, to the best of our knowledge, has attracted few attentions in recent literature. Moreover, with respect to previous works on discriminative filter learning [21]–[23], large margin filtering differs in two essential aspects: Firstly, we do not consider the full signal as a single training example but instead consider each sample as an example. Hence the two problems are rather different, and the one we address is more related to *sample classification* than to *full signal classification*. Secondly, we do not optimize the empirical but the structural risk. To the best of our knowledge, the idea of using a large margin criterion for optimal filter design is novel and brings several additional advantages compared to other discriminative criteria. It has been shown to provide a good predictive generalization and it implies a convex problem for a fixed filtering. Even if the filter learning is non-convex, as shown in the sequel, the sample discrimination –which is the final purpose of the method– is done by a unique optimal, in structural risk minimization sense, pair of filter and classifier.

IV. Experimental results

This section presents the numerical experiments comparing the different proposed approaches. First we consider numerical results on a toy dataset containing both convolutional and additive noise. Then we test the methods on a real life BCI dataset from the *BCI Competition III* [4]. Finally, we extend the methods to a 2-dimensional problem of multi-spectral remote sensing image segmentation. The Matlab™ code for all the methods tested in this paper is available in http://remi.flamary.com/code for the interested reader.

A. Experiment 1: Toy Dataset

This first experiment is designed to provide some insight on the capabilities of each method to handle feature and channel weighting/selection. To do this, we use a toy dataset corrupted by both convolutional and additive noise. Within the data,

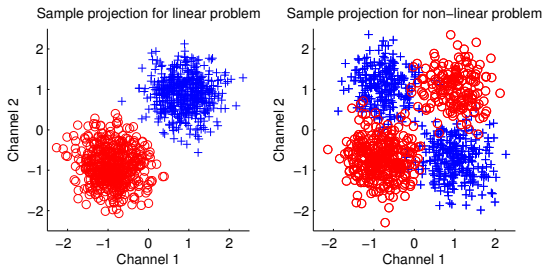

Fig. 2. Projection of the sample on the two discriminative channels for the linear case (left) and the nonlinear case (Right). Here there is no convolutional noise to illustrate the original shape of the data.

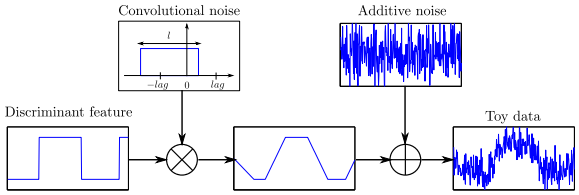

Fig. 3. Toy dataset generation scheme: First a convolutional noise is applied to the discriminative feature and then a Gaussian additive noise is added.

TABLE I
LIST OF THE METHODS COMPARED IN OUR EXPERIMENTS.

Method	Definition
SVM	Classical SVM on the samples.
Avg-SVM	SVM on samples filtered by an average filter to limit the impact of the Gaussian noise (see II-B).
GMM	Gaussian Mixture Model classification learned with an EM algorithm.
WinSVM	Classification of a window of samples (see II-C).
SWinSVM*	Classification of a window of samples with channel selection (see II-C2).
KF-SVM	Kernel FilterSVM, Large Margin Filtering (see III).
SKF-SVM	Kernel FilterSVM with channel selection (See III-E).
KF-GMM	GMM classifiers on the pre-filtered samples. The filter is the one obtained by KF-SVM
WinGMKL**	Multiple kernel learning proposed by [35] for feature selection applied on a window of samples.

* only in the linear case. ** only in the nonlinear case.

discriminative and non-discriminative channels are present. We investigate the linear and the nonlinear cases separately, as some of the proposed methods are limited to the linear case.

The generation of the dataset is done in several steps: first a sequence of labels is created. The length of the regions with constant label in this sequence follows a uniform distribution between 30 and 40 samples. This sequence is used to create the discriminative channels in the signal. Every signal in the toy dataset contains d channels, among which two are informative and the others are corrupted by Gaussian noise only. Depending on their complexity, the discriminative channels cast linear and nonlinear problems, as shown in Fig. 2. Convolutional noise is added to the discriminative channels in two ways: first, a different delay drawn from a uniform distribution on $[-\tau, \tau]$ is applied to every channel and then a moving-average filtering of size l is applied. Finally, additive Gaussian noise of standard deviation σ_n is added to all channels. Figure 3 summarizes how the noise is applied to the discriminant features.

Table I summarizes the methods used in the experiments.

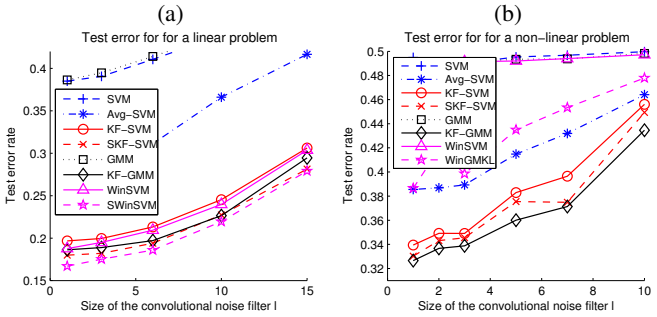

Fig. 4. Classification error rate for different convolutional noises in the (a) linear and (b) nonlinear cases.

The size of the signal is of 1000 samples for both the learning (training) and the validation sets and of 10000 samples for the test set. To allow a fair comparison with Avg-SVM, we selected $f = 11$ and $n_0 = 6$ for the nonlinear case and $f = 15$ and $n_0 = 8$ for the linear case. These values correspond to a good average filtering centered on the current sample. We fixed the additive noise value at $\sigma_n = 3$ and the possible delay at $\tau = 5$ samples. The regularization parameters of all the methods are selected by assessing performance on the validation set. Experiments were repeated 10 times, and the test error is the average over the runs. A Wilcoxon's signed rank test with a risk of 5% was applied to the results in order to check the statistical differences between the methods. The test error used is the number of misclassified samples divided by the total number of samples in the sequence.

The results are shown in Fig. 4 for both the linear and nonlinear problems. For the linear problem (Fig. 4(a)), we can see that all the windowing methods perform well. The best method is SWinSVM closely followed by the SKF-SVM, but no statistical differences were observed by applying the Wilcoxon's test. These two approaches performing channel selection lead to a better generalization. Note that WinSVM performs similarly as KF-SVM which is consistent with the results in [24] for small dimensional problems. Due to the Gaussian nature of this dataset, KF-GMM outperforms KF-SVM.

For the nonlinear problem in Fig. 4(b), statistical differences were observed between methods, suggesting that the large margin filtering methods (KF-SVM, SKF-SVM and KF-GMM) outperformed the rest. Note that the channel selection shows an improvement when the noise is high (long convolutional noise filter). The best results are obtained here by KF-GMM as it uses the best model for the data after denoising. We also report the poor behavior of WinGMKL, which gives worse results than a simple average filtering.

The filter length is an important parameter to be selected. One approach would be to choose the filter length using prior knowledge about the dataset at hand. For instance, a long filter might work well for classes changing slowly. Another possible approach is to select f using standard cross-validation but at a the expense of a higher computational cost. In Figure 5, we investigate the impact of the length of the filter parameter for the above described non-linear toy problem with a size of the noise convolutional filter of length 5. The test error for a

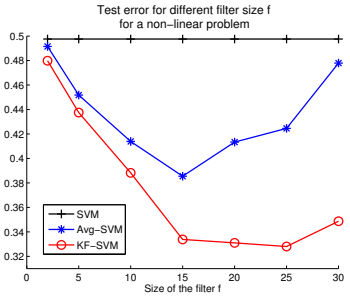

Fig. 5. Classification error rate for different filter size in the nonlinear case. The SVM curve is flat as the samples are not filtered, hence performance does not depend on the filter size.

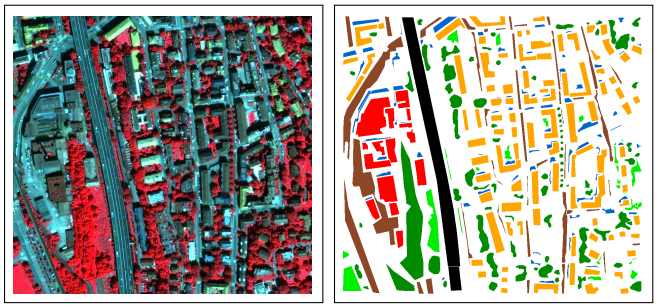

Fig. 6. QuickBird scene of suburbs of Zurich (left) and labeled pixels (right). Legend: *dark green = trees; light green = meadows; black = speedway; brown = roads; orange = residential buildings; red = commercial buildings; blue = shadows.*

varying size of filter f is shown for AVG-SVM and KF-SVM. We can see that KF-SVM outperforms the other method. In addition, it exhibits a wider optimal region so it is less sensitive than the Avg-SVM to the choice of a suitable f value. Note that AVG-SVM has a unique optimal filter length. Indeed, for a moving average the filter size adjusts the bandwidth. Hence, AVG-SVM fits to the data at a specific filter length. In the same way KF-SVM performance also depends on the filter size. However it is less sensitive to this parameter than AVG-SVM because the filter coefficients are optimized in the process. From this figure, we also notice that selecting the filter length for our KF-SVM as the one that minimizes the AVG-SVM performance also leads to nearly optimal classification error rate.

B. Experiment 2: BCI Dataset

The BCI Dataset considered is one of the problems presented in *BCI Competition III* [4]. The objective is to obtain a sequence of labels out of brain activity signals for three human subjects. The data consists of 96 channels containing Power Spectral Density (PSD) features for different band-pass (three training sessions and one test session, with $n \simeq 3000$ samples *per* session) and three possible labels (left arm, right arm or a word). We deal with the several classes in the dataset through a classical One-Against-All strategy. For the non-linear approaches, in order to make the problem tractable despite the large number of samples, we use as a training set only a randomly selected subset of the available dataset (of about 30%). The regularization parameters are tuned using a grid search validation strategy on the third training session. In these experiments, n_0 has been set to 0, we want to predict the current mental task with no delay.

Our method is compared to the best BCI competition results and to the SVM without filtering. Here we do not provide performances for GMM and FilterGMM due to their poor performances that probably comes from the high dimensionality of the problem. We could not obtain WinGMKL results in a reasonable time so this approach has not been reported either.

The test error for different methods and filter lengths is given in Table II. For the linear models, the best methods for all tested filter sizes are KF-SVM, SKF-SVM and SWinSVM. This shows the advantage of taking into account the neighborhood of the samples for decision and the importance of a proper regularization. Longer filtering provides the best results, especially in conjunction with regularization that helps to avoid over-fitting (indeed, for $f = 50$, approximately 5000 filter coefficients are learned based on approximately 10000 sample). The best overall results are obtained by KF-SVM and SKF-SVM with the filter length $f = 50$.

The results follow the same trends for the nonlinear models, showing that for this task a linear classifier is sufficient. However, one should keep in mind that, in these cases, the decision functions are learned from only 30% of the samples. In this case, the Avg-SVM performs well, since the noise is in the high frequencies and the non linearities that can be induced by over-filtering are handled by the Gaussian kernel.

C. Experiment 3: Multispectral Image Segmentation

The method we promote for learning a large margin filtering may be easily extended to the 2-dimensional case. In this experiment, we apply it to the segmentation of remotely-sensed multispectral images. Nowadays, sensors mounted on satellite or airborne platforms may acquire the reflected energy by the Earth with high spatial detail and in several wavelengths or spectral channels. This allows the detection and classification of the pixels in the scene. The obtained classification maps are then used for management, policy making and monitoring. In multispectral imagery, the pixels are multidimensional (RGB and near-infrared bands) and hence the filtering is a 2-dimensional convolution of the image band-by-band. We tested our approach on a Very High Resolution (VHR) image acquired by the sensor QuickBird (spatial detail of 0.6m) over the city of Zürich, Switzerland (see Fig. 6). The considered dataset represents a residential area in the South-West part of the city. Seven classes were labeled by photo-interpretation. The main challenge is to distinguish between the two classes of buildings and the two classes of roads by applying spatial filtering, because the spectral difference between these couples of classes is low.

Classification results when using a Gaussian kernel are shown in Table III. SKF-SVM is not applied to this dataset, since sparse selection is not necessary for such small dimensional data ($d = 4$). We computed the test error rate One-Against-All and the estimated Cohen's kappa coefficient, which is a more appropriate measure to evaluate the classification accuracy in unbalanced class problems (best when 1).

TABLE II
CLASSIFICATION ERROR RATE FOR THE BCI DATASET FOR DIFFERENT METHODS, AND FILTER LENGTH f. RESULTS ARE GIVEN FOR THE LINEAR MODEL (TOP) AND FOR THE NONLINEAR MODEL (BOTTOM). THE THREE BEST METHODS ARE IN BOLD. FOR THE SAKE OF PROPER COMPARISON, NOTE THAT THE BEST COMPETITION RESULTS ARE (0.2040, 0.2969, 0.4398 AND 0.3135 FOR THE AVERAGE).

Method	$f=10$				$f=20$				$f=50$			
	S1	S2	S3	Avg	S1	S2	S3	Avg	S1	S2	S3	Avg
	Linear model											
SVM	0.254	0.377	0.553	0.395	0.254	0.377	0.553	0.395	0.254	0.377	0.553	0.395
Avg-SVM	0.228	0.342	0.534	0.368	0.193	0.298	0.530	0.340	0.133	0.236	0.475	0.282
KF-SVM	0.205	0.304	0.512	**0.340**	0.185	0.269	0.429	**0.294**	0.126	0.231	0.423	**0.260**
SKF-SVM	0.205	0.294	0.473	**0.324**	0.182	0.262	0.481	**0.308**	0.128	0.222	0.438	**0.263**
WinSVM	0.214	0.316	0.540	0.357	0.196	0.280	0.534	0.337	0.146	0.223	0.482	0.284
SWinSVM	0.215	0.314	0.470	**0.333**	0.196	0.264	0.428	**0.296**	0.146	0.218	0.460	**0.274**
	nonlinear model (Gaussian kernel)											
SVM	0.239	0.357	0.481	0.359	0.239	0.357	0.481	0.359	0.239	0.357	0.481	0.359
Avg-SVM	0.217	0.331	0.470	0.340	0.197	0.295	0.448	0.313	0.128	0.234	0.450	**0.271**
KF-SVM	0.205	0.300	0.489	**0.331**	0.173	0.266	0.482	**0.307**	0.158	0.227	0.445	0.277
SKF-SVM	0.206	0.307	0.489	**0.334**	0.174	0.260	0.446	**0.293**	0.114	0.232	0.471	**0.273**
WinSVM	0.210	0.324	0.477	**0.337**	0.174	0.281	0.448	**0.301**	0.134	0.232	0.440	**0.269**

Two configurations are tested: 7 classes and 6 classes. For the last configuration, class 'Residential' and 'Commercial' are merged as 'Building' (see Figure 6 for the list of classes). For the 7–classes setting, the inclusion of spatial information strongly improves the results of the SVM. In this case, learning the filter provides better results in comparison with other approaches. Regarding the 6–classes setting, WinSVM gives slightly better results than KF-SVM. Note that the interest of KF-SVM lies in the learned filters that can be interpreted or used as pre-processing for other classifications, whereas Win-SVM with a Gaussian kernel gives rise to a black-box approach.

These results show the interest of learning a large margin filtering when the overlap between the classes is important. But the most important aspect of our approach is the fact that the filters are interpretable. For instance, it is possible to compute the Fourier transform of the learned filters. Figure 7 shows the magnitude of the Fourier transform of the red component filter for classes 'Residential' and 'Commercial'. First, the algorithm nicely learns low-pass filters, which is due to the fact that the noise is mainly in the high spatial frequencies. Besides, we can see that the cut-off frequency is different for each class. The filter for houses cuts at 5 m (0.2 m^{-1}) whereas for commercial buildings, the cut-off frequency is 10 m (0.1 m^{-1}). This will promote larger spatial filtering for commercial buildings than for the residential ones, as one intuitively would expect: commercial buildings are usually bigger than residential ones, and by learning the filtering we automatically find this discriminant feature from the data.

TABLE III
RESULTS IN IMAGE SEGMENTATION WITH A GAUSSIAN KERNEL. ONE-AGAINT-ALL ACCURACY AND KAPPA COEFFICIENT.

Method	Classes	Filter size	Training Pixels	Error rate	Kappa
SVM	7	9	~ 5000	0.249	0.685
AvgSVM				0.163	0.796
WinSVM				0.170	0.785
KF-SVM				**0.147**	**0.816**
SVM	6*	9	~ 5000	0.170	0.772
AvgSVM				0.105	0.860
WinSVM				**0.083**	**0.889**
KF-SVM				0.085	0.885

* 'Residential' and 'Commercial' are merged into one 'Building' class.

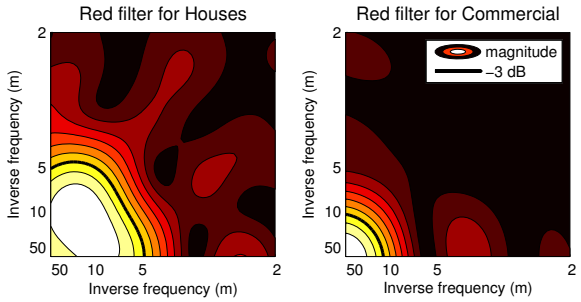

Fig. 7. Magnitude of the spatial Fourier transform on the Red component large margin filtering for the 'House' and 'Commercial' classes. The bold black lines correspond to the −3dB attenuation.

V. CONCLUSION

In this work, we addressed the problem of multi-channel signal sequence labeling in the presence of additive and convolutional noise. At first, several methods based on filtering preprocessing and time-window classification have been reviewed. Afterwards, we introduced a general framework for learning large-margin filtering jointly with a sample classifier. Depending on the regularization term used, this framework allows one to achieve adaptive scaling of the channels or channel selection. For solving the optimization problem yielded by the proposed framework, we have considered a conjugate gradient algorithm which provably converges towards a local minimum of the problem. We empirically compared the different approaches on a non-linear toy example and on a real life BCI classification problem, and these experiments showed the benefits of learning a large margin filtering. Finally, we extended our approach to a multidimensional image segmentation problem and the interpretability of the learned filters have been evaluated by visualizing theirs Fourier transforms.

In future work, we plan to propose new regularization terms that can bring prior information to the problem. For instance, since noise typically appears in the high frequency range, one could design regularizers that promote learning low pass filters. Another interesting problem is the one of large scale

learning. The fact that we have to iteratively solve a SVM makes large-scale problem hardly tractable. We will investigate the use of a one-pass SVM solver such as the one of Bordes al. [45] instead of an exact SVM solver.

APPENDIX
GLOBAL CONVERGENCE OF THE CG ALGORITHM

In this appendix, we prove the conditions 1 and 2 for conjugate gradient algorithm convergence as described in section III-B. Remind that some hypotheses and the convergence conditions are given in section III-C.

First, we address the condition 1 and we prove that the level set $\mathcal{L} := \{\mathbf{F}|J(\mathbf{F}) \leq J(\mathbf{F}^0)\}$ for a fixed initial value $\mathbf{F}^0$ is bounded.

Proof: First, let $\mathbf{F}^0$ be some fixed initial value and $\mathcal{L} := \{\mathbf{F}|J(\mathbf{F}) \leq J(\mathbf{F}^0)\}$ a level set. We know that $\forall o,p$, $|\widetilde{\mathbf{K}}_{o,p}| \leq 1$, $|\boldsymbol{\alpha}_p^*| \leq \frac{C}{n}$ due to the SVM formulation in Equation (10) and $|\mathbf{y}_p| = 1$. We can derive that $|\widetilde{\mathbf{K}}_{o,p}\mathbf{y}_o\mathbf{y}_p\boldsymbol{\alpha}_o^*\boldsymbol{\alpha}_p^*| \leq \frac{C^2}{n^2}$ and that $\sum_{o,p}|\widetilde{\mathbf{K}}_{o,p}\mathbf{y}_o\mathbf{y}_p\boldsymbol{\alpha}_o^*\boldsymbol{\alpha}_p^*| \leq C^2$. Then $|J'(\mathbf{F})| = |-\frac{1}{2}\sum_{o,p}\widetilde{\mathbf{K}}_{o,p}\mathbf{y}_o\mathbf{y}_p\boldsymbol{\alpha}_o^*\boldsymbol{\alpha}_p^* + \sum_i \boldsymbol{\alpha}_i^*| \leq \frac{C^2}{2} + C$ that is $|J'(\mathbf{F})|$ is bounded. We know that $J(\mathbf{F}) = J'(\mathbf{F}) + \lambda||\mathbf{F}||_F^2$ so if $\mathbf{F} \in \mathcal{L}$, we can infer that $-\frac{C^2}{2} - C + \lambda||\mathbf{F}||_F^2 \leq J(\mathbf{F}) \leq J(\mathbf{F}^0) \leq \frac{C^2}{2} + C + \lambda||\mathbf{F}^0||_F^2$ hence $\lambda||\mathbf{F}||_F^2 \leq C^2 + 2C + \lambda||\mathbf{F}^0||_F^2$. This means for all $\forall\mathbf{F}^0$ such that $\|\mathbf{F}^0\|_F^2 < \infty$, the norm of $\mathbf{F} \in \mathcal{L}$ is bounded, so the level set $\mathcal{L}$ is bounded and Condition 1 holds. ∎

Secondly we prove that the norm of the Hessian matrix of $J(\mathbf{F})$ is bounded with $\mathbf{F} \in \mathcal{N}$ with $\mathcal{N}$ an open neighborhood of $\mathcal{L}$. In a nutshell, the proof proceeds by showing that for all possible $\mathbf{F}$, the components of the Hessian are also bounded.

Proof: Firstly, let us note that if we choose $\mathcal{N} := \{\mathbf{F}|J(\mathbf{F}) < J(\mathbf{F}^0) + \epsilon\}$ with $\epsilon > 0$ then $\mathcal{N}$ is an open neighborhood of $\mathcal{L}$. Note that similarly we can define $\mathcal{M} := \{\mathbf{F}|J(\mathbf{F}) \leq J(\mathbf{F}^0) + \epsilon\}$ the closure of $\mathcal{N}$ so that $\mathcal{L} \subset \mathcal{N} \subset \mathcal{M}$. Using a similar approach than in the preceding proof, one can show that for $\mathbf{F} \in \mathcal{M}$, $\lambda||\mathbf{F}||_F^2 \leq C^2 + 2C + \lambda||\mathbf{F}^0||_F^2 + \epsilon$ which implies that $\mathcal{M}$ is a bounded set. Finally $\mathbf{F} \in \mathcal{N}$ lies in a closed and bounded set of the metric space $\mathbb{R}^{f\times d}$ and we can use the Bolzano-Weierstrass theorem to conclude that $\mathbf{F} \in \mathcal{N}$ lies in a compact.

J is twice differentiable because both J' and $||\cdot||^2$ are. Indeed differentiability of $\nabla_{\mathbf{F}}J'$ given equation (11) w.r.t. $\mathbf{F}$ comes from (i) $\alpha^*(\mathbf{F})$ is differentiable [41] and (ii) it is a sum of differentiable terms. Then we express the components of this matrix $\frac{\partial^2 J(\mathbf{F})}{\partial\mathbf{F}_{i,j}\partial\mathbf{F}_{i',j'}} = 2\lambda + \frac{1}{2\sigma}\mathbf{Q}_{(i,j),(i',j')}$ with $\mathbf{Q}_{(i,j),(i',j')}$ equal to:

$$\begin{aligned}
&\sum_{o=1,p=1}^{n,n} \Delta\mathbf{x}_{o,p}^{i,j}\Delta\mathbf{x}_{o,p}^{i',j'}(1 - \frac{1}{2\sigma}\Delta\tilde{\mathbf{x}}_{o,p}^{j}\Delta\tilde{\mathbf{x}}_{o,p}^{j'})\widetilde{\mathbf{K}}_{o,p}\mathbf{y}_o\mathbf{y}_p\boldsymbol{\alpha}_o^*\boldsymbol{\alpha}_p^* \\
&\quad + \mathbf{x}_{o,p}^{i,j}\mathbf{x}_{o,p}^{i,j}\widetilde{\mathbf{K}}_{o,p}\mathbf{y}_o\mathbf{y}_p\left(\boldsymbol{\alpha}_p^*\frac{\partial\boldsymbol{\alpha}_o^*}{\partial\mathbf{F}_{i',j'}} + \boldsymbol{\alpha}_o^*\frac{\partial\boldsymbol{\alpha}_p^*}{\partial\mathbf{F}_{i',j'}}\right) \\
&\text{with } \Delta\mathbf{x}_{o,p}^{i,j} = \mathbf{X}_{o+1-i,j} - \mathbf{X}_{p+1-i,j} \qquad (12)\\
&\text{and } \Delta\tilde{\mathbf{x}}_{o,p}^{j} = \widetilde{\mathbf{X}}_{o,j} - \widetilde{\mathbf{X}}_{p,j}
\end{aligned}$$

Now, by using results from [46] we know that

$$\left.\frac{\partial\boldsymbol{\alpha}^*}{\partial\mathbf{F}_{i,j}}\right|_{\mathbf{F}} = -\bar{\mathbf{A}}\frac{\partial\widetilde{\mathbf{K}}_{sv}}{\partial\mathbf{F}_{i,j}}\boldsymbol{\alpha}_{sv}(\mathbf{F}) \qquad (13)$$

is one column of the Jacobian matrix corresponding to $\mathbf{F}_{i,j}$ where $\bar{\mathbf{A}}$ is the inverse of a matrix continuously dependent on $\mathbf{F}$ through the positive definite Gram matrix. We aim at showing that each component of this Jacobian matrix defined by Equation (13) is bounded. For this purpose, we first remark that $\bar{\mathbf{A}}$ is continuous with respect to $\mathbf{F}$ as the inverse of a matrix continuous with $\mathbf{F}$. Then, $\frac{\partial\widetilde{\mathbf{K}}_{sv}}{\partial\mathbf{F}_{i,j}}$ is the differentiate of the kernel matrix restricted to the support vectors sv ($\boldsymbol{\alpha}^*$ coefficients such that $0 < \boldsymbol{\alpha}_i^* < C/n$) and is continuous by hypothesis. The function $\boldsymbol{\alpha}_{sv}(\mathbf{F})$ represents the values of the support vectors for the SVM problem with a given $\mathbf{F}$, and is continuous with respect to $\mathbf{F}$ as it is differentiable. This means that the components of the Jacobian matrix $\mathbf{J}_{\boldsymbol{\alpha}}$ are continuous functions with respect to $\mathbf{F}$. Then, since $\mathbf{F}$ lies in a compact set, the components $\frac{\partial\boldsymbol{\alpha}_o^*}{\partial\mathbf{F}_{i,j}}$ of the Jacobian matrix $\mathbf{J}_{\boldsymbol{\alpha}}$ are bounded. Moreover, $\widetilde{\mathbf{X}}$ is bounded as it is a finite sum and product of bounded terms. Then we can see in Equation (12) that the components of the Hessian matrix are composed of sums and multiplication of bounded terms, so we can conclude that these components are bounded. Thus the Frobenius norm of the Hessian $||\nabla^2 J(\mathbf{F})||_F$ is itself bounded because it is a finite sum of bounded terms. Using results from [47] we can conclude that:

$$\exists L \quad \text{such that} \quad ||\nabla^2 J(\mathbf{F})||_2 \leq ||\nabla^2 J(\mathbf{F})||_F \leq L.$$

Finally, the second condition holds as the objective function J is continuously differentiable and his Hessian matrix has bounded eigenvalues. ∎

REFERENCES

[1] A. Ganapathiraju, J. Hamaker, and J. Picone, "Applications of support vector machines to speech recognition," *IEEE Transactions on Signal Processing*, vol. 52, no. 8, pp. 2348–2355, 2004.

[2] J. d. R. Millán, "On the need for on-line learning in brain-computer interfaces," in *Proc. Int. Joint Conf. on Neural Networks*, 2004.

[3] A. Kampouraki, G. Manis, and C. Nikou, "Heartbeat time series classification with support vector machines," *Information Technology in Biomedicine, IEEE Transactions on*, vol. 13, no. 4, pp. 512 –518, jul. 2009.

[4] B. Blankertz *et al.*, "The BCI competition 2003: progress and perspectives in detection and discrimination of EEG single trials," *IEEE Transactions on Biomedical Engineering*, vol. 51, no. 6, pp. 1044–1051, 2004.

[5] B. Asl, S. Setarehdan, and M. Mohebbi, "Support vector machine-based arrhythmia classification using reduced features of heart rate variability signal," *Artificial Intelligence in Medicine*, vol. 44, no. 1, pp. 51 – 64, 2008.

[6] S. Lenser and M. Veloso, "Non-parametric time series classification," in *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*. IEEE, 2006, pp. 3918–3923.

[7] O. Cappé, E. Moulines, and T. Rydèn, *Inference in Hidden Markov Models*. Springer, 2005.

[8] Y. Altun, I. Tsochantaridis, T. Hofmann *et al.*, "Hidden Markov support vector machines," in *International Conference in Machine Learning*, vol. 20, 2003, p. 3.

[9] A. Sloin and D. Burshtein, "Support vector machine training for improved hidden Markov modeling," *IEEE Transactions on Signal Processing*, vol. 56, no. 1, p. 172, 2008.

[10] J. Bloit and X. Rodet, "Short-time Viterbi for online HMM decoding: Evaluation on a real-time phone recognition task," in *ICASSP*, 2008.

[11] A. Bordes, N. Usunier, and L. Bottou, "Sequence labelling svms trained in one pass," in *Machine Learning and Knowledge Discovery in Databases: ECML PKDD 2008*, ser. Lecture Notes in Computer Science, LNCS 5211, W. Daelemans, B. Goethals, and K. Morik, Eds. Springer, 2008, pp. 146–161. [Online]. Available: http://leon.bottou.org/papers/bordes-usunier-bottou-2008

[12] T. Pistohl, T. Ball, A. Schulze-Bonhage, A. Aertsen, and C. Mehring, "Prediction of arm movement trajectories from ecog-recordings in humans," *Journal of Neuroscience Methods*, vol. 167, no. 1, pp. 105–114, Jan. 2008.

[13] S. Salenius, R. Salmelin, C. Neuper, G. Pfurtscheller, and R. Hari, "Human cortical 40 Hz rhythm is closely related to EMG rhythmicity," *Neuroscience letters*, vol. 213, no. 2, pp. 75–78, 1996.

[14] X.-Y. Wang, T. Wang, and J. Bu, "Color image segmentation using pixel wise support vector machine classification," *Pattern Recognition*, vol. 44, no. 4, pp. 777–787, 2011.

[15] D. Chai, H. Lin, and Q. Peng, "Bisection approach for pixel labelling problem," *Pattern Recognition*, vol. 43, no. 5, pp. 1826–1834, 2010.

[16] D. Puig and M. Garcia, "Automatic texture feature selection for image pixel classification," *Pattern Recognition*, vol. 39, no. 11, pp. 1996–2009, 2006.

[17] F. Pacifici, M. Chini, and W. Emery, "A neural network approach using multi-scale textural metrics from very high-resolution panchromatic imagery for urban land-use classification," *Remote Sens. Environ.*, vol. 113, no. 6, pp. 1276–1292, 2009.

[18] J. Benediktsson, J. A. Palmason, and J. R. Sveinsson, "Classification of hyperspectral data from urban areas based on extended morphological profiles," *IEEE Trans. Geosci. Remote Sens.*, vol. 43, no. 3, pp. 480–490, 2005.

[19] M. Fauvel, J. A. Benediktsson, J. Chanussot, and J. R. Sveinsson, "Spectral and spatial classification of hyperspectral data using SVMs and morphological profiles," *IEEE Trans. Geosci. Remote Sens.*, vol. 46, no. 11, pp. 3804 – 3814, 2008.

[20] D. Tuia, F. Pacifici, M. Kanevski, and W. Emery, "Classification of very high spatial resolution imagery using mathematical morphology and support vector machines," *IEEE Trans. Geosci. Remote Sens.*, vol. 47, no. 11, pp. 3866–3879, 2009.

[21] Y. LeCun and Y. Bengio, "Convolutional networks for images, speech, and time series," *The handbook of brain theory and neural networks*, pp. 255–258, 1995.

[22] B. de Vries and J. C. Principe, "The Gamma model – a new neural model for temporal processing," *Neural Networks*, vol. 5, no. 4, pp. 565–576, 1992.

[23] S. Lawrence and A. C. Tsoi, "The Gamma MLP for speech phoneme recognition," in *IEEE Workshop on Neural Networks for Signal Processing VII*. MIT Press, 1996, pp. 785–791.

[24] R. Flamary, B. Labbé, and A. Rakotomamonjy, "Large margin filtering for signal sequence labeling," in *International Conference on Acoustic, Speech and Signal Processing 2010*, 2010.

[25] J. C. Principe, B. de Vries, and P. G. de Oliveira, "The gamma filter – a new class of adaptive IIR filters with restricted feedback," *IEEE Transactions on Signal Processing*, vol. 41, no. 2, pp. 649–656, 1993.

[26] J. Shawe-Taylor and N. Cristianini, *Kernel methods for pattern analysis*. Cambridge Univ Pr, 2004.

[27] O. Chapelle, "Training a support vector machine in the primal," *Neural Comput.*, vol. 19, no. 5, pp. 1155–1178, 2007.

[28] Y. Nesterov, "Smooth minimization of non-smooth functions," *Mathematical Programming*, vol. 100, pp. 127–152, 2005.

[29] P. Tseng, "On accelerated proximal gradient methods for convex-concave optimization," *submitted to SIAM Journal on Optimization*, 2008.

[30] X. Chen, W. Pan, J. Kwok, and J. Carbonell, "Accelerated gradient method for multi-task sparse learning problem," in *Proceedings of the International Conference on Data Mining*, 2009.

[31] J. Bonnans and A. Shapiro, "Optimization problems with pertubation: A guided tour," *SIAM Review*, vol. 40, no. 2, pp. 202–227, 1998.

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

[33] W. Hager and H. Zhang, "A survey of nonlinear conjugate gradient methods," *Pacific journal of Optimization*, vol. 2, no. 1, pp. 35–58, 2006.

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

[35] M. Varma and B. Babu, "More generality in efficient multiple kernel learning," in *Proceedings of the 26th Annual International Conference on Machine Learning*. ACM, 2009, pp. 1065–1072.

[36] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM Journal on Imaging Sciences*, vol. 2, pp. 183–202, 2009.

[37] S. Shalev-Shwartz, Y. Singer, and N. Srebro, "Pegasos: Primal estimated sub-gradient solver for svm," in *Proceedings of the 24th international conference on Machine learning*. ACM, 2007, p. 814.

[38] D. Hunter and K. Lange, "A Tutorial on MM Algorithms." *The American Statistician*, vol. 58, no. 1, pp. 30–38, 2004.

[39] G. Dornhege, B. Blankertz, M. Krauledat, F. Losch, G. Curio, and K. Muller, "Optimizing spatio-temporal filters for improving brain-computer interfacing," *Advances in Neural Information Processing Systems*, vol. 18, p. 315, 2006.

[40] L. Gómez-Chova, A. A. Nielsen, and G. Camps-Valls, "Explicit signal to noise ratio in reproducing kernel Hilbert spaces," in *IEEE International Geoscience and Remote Sensing Symposium*, Vancouver, Canada, July 2011.

[41] O. Chapelle, V. Vapnik, O. Bousquet, and S. Mukerjhee, "Choosing multiple parameters for SVM," *Machine Learning*, vol. 46, no. 1-3, pp. 131–159, 2002.

[42] Y. Grandvalet and S. Canu, "Adaptive scaling for feature selection in SVMs," in *Advances in Neural Information Processing Systems*, vol. 15. MIT Press, 2003.

[43] M. Christensen and A. Jakobsson, "Optimal filter designs for separating and enhancing periodic signals," *IEEE Trans. on Signal Processing*, vol. 58, no. 12, pp. 5969–5983, 2010.

[44] M. Christensen, J. Hensen, A. Jakobsson, and S. Jensen, "On optimal filter designs for fundamental frequence estimation," *IEEE Trans. on Signal Processing Letters*, vol. 15, pp. 745–748, 2008.

[45] A. Bordes, S. Ertekin, J. Weston, and L. Bottou, "Fast kernel classifiers with online and active learning," *Journal of Machine Learning Research*, vol. 6, pp. 1579–1619, 2005.

[46] O. Chapelle and A. Rakotomamonjy, "Second order optimization of kernel parameters," in *NIPS Workshop on Automatic Selection of Optimal Kernels*, 2008.

[47] G. Golub and C. Van Loan, *Matrix computations*. Johns Hopkins Univ Pr, 1996.