Large Margin Filtering

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

Large Margin Filtering

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

***Abstract*—We address in this paper the problem of multi-channel signal sequence labeling. In particular, we consider the problem where the signals are contaminated by both additive and convolutional noise. We investigate several approaches based on windowing and propose to learn a support vector machine (SVM) classifier and a signal filter jointly. We derive algorithms to solve the optimization problem and discuss different filter regularizers for automated scaling or selection of channels. After considering its properties on a toy dataset, the approach is tested on two challenging real life datasets: BCI time series and 2-dimensional image segmentation. Results show the interest of large margin filtering in terms of performance and interpretability.**

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

I. Introduction

Signal sequence labeling is a classical machine learning problem in which the goal is to assign a label for every sample of a signal while taking into account the sequentiality of the samples. The field is very vast and typically arises in many signal processing problems, such as Automatic Speech Recognition (ASR) [1], Brain Computer Interfaces (BCI) [2], or pathology discrimination from biosignals [3]. For instance, in speaker diarization, the aim is to recognize 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, hence the need for sample labeling [2], [4]. Electrocardiographic (ECG) signals are used to predict the presence or absence of a given pathology in advance, such as particular arrythmias 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 is 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 have the conditional probability of the observations *per* hidden states (classes), 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]. One simple approach for using discriminative classifiers in the HMM framework has been proposed in [1]. It consists in learning support vector machine (SVM) classifiers known for their better robustness in high dimension problems and to transform their outputs into probabilities using Platt's method [10], [11], leading to better performances after Viterbi decoding. This approach supposes that the complete sequence of observations is available, which corresponds to an off-line decoding, which is not often the case. In the case of BCI applications, for example, a real time decision is often needed [2], [4], which restricts the use of Viterbi decoding. Note that local Viterbi decoding may be used for online decoding [12]. Another online decoding approach is to directly classify the current sample and the preceding decoded labels. These methods are defined in [13] as greedy decoding and permit the use of higher-order HMM taking into account several preceding states.

Besides the online processing problem, when the sequence labeling has to be performed on a measured signal, the efficiency of the 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 and then a classifier is applied. 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. Besides, we should note that the approach will only work if the peculiarities and particular statistical properties of the signal are taken into account in both the filter and the classifier. In many applications, the filter is restricted to deal with particular noise sources (typically Gaussian), while the classifier is not commonly adapted to the non-$i.i.d.$ nature of the signals.

HMMs 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. Indeed 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). This is a problem in BCI applications where the interesting information is not always synchronized with the labels. For instance, authors in [14] showed the need for applying delays to the signal, since the neural activity precedes the actual movement. Note that they selected the delay through validation. Another illustration of the need for time-lag automated handling is the following. Suppose we want to interact with a computer using multi-modal acquisitions (e.g. EEG or EMG). Then, since each modality has its own time-lag with respect to neural activity [15], it may be difficult to manually synchronize

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

all modalities while 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, labeling is not restricted to unidimensional signals. A paradigmatic multidimensional problem that involves signal sequence labeling is that of image segmentation. Images, like time series and data sequences, are not $i.i.d.$ data. The statistics of grayscale and natural colors has been studied extensively. 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 still 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 [16], [17] and morphological [18]–[20] filters.

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. In [21], 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. Moreover, we have access to the filtering and it can be interpreted in both the temporal and the frequency domains. Here, we 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 first formalize the problem and review traditional approaches such as filtered-sample classification and time-window classification. In Section III, we introduce the problem of large margin filtering to deal with the limitation observed when considering non-$i.i.d.$ signals. We discuss the complexity of our approach and the effect of different regularization for the filtering matrix. Finally, the algorithmic complexity and the effect of different regularizers are discussed. 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 dataset and 2-dimensional image segmentation problem are considered. Section V concludes the paper.

II. The Sample Labeling Problem

In this section, we formally define 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 reproducing kernel Hilbert spaces and both linear and nonlinear problems are discussed.

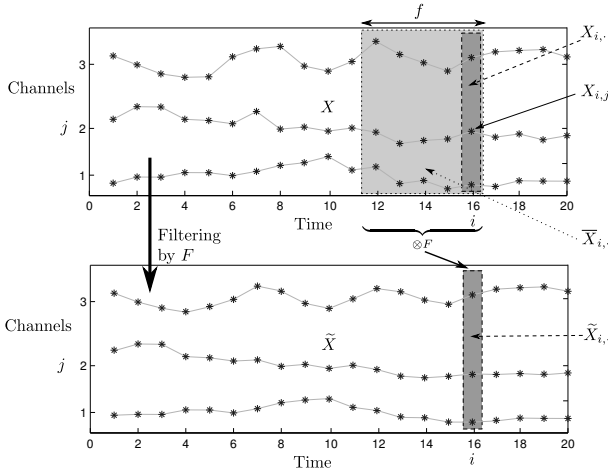

Fig. 1. Data matrix $\mathbf{X}$ (top), filtered matrix $\widetilde{\mathbf{X}}$ (bottom) and time-window $\overline{\mathbf{X}}$ (light gray) with $n_0 = 0, d = 3$ and $f = 5$.

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 suppose 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 of each sample.

In order to reduce noise in the samples or variability in the features, an usual approach is to filter $\mathbf{X}$ before learning the classifier. In literature, all channels are usually filtered similarly although there is no reason for believing that a single filter will be optimal for every channel. Moreover, assuming an explicit filter structure may not fit the underlying system that generated the data. The Savitzky-Golay [14] or the gamma [22] filters are structures commonly used to this purpose. 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 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 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 filter centered on the current sample. Figure 1 presents an example of signal $\mathbf{X}$ and filtered signal $\widetilde{\mathbf{X}}$.

B. SVM for filtered samples

To improve the classification rate one may filter the channels in $\mathbf{X}$ in order to reduce the impact of the noise. The usual filter

in the case of high frequency noise is the average filter defined by $\mathbf{F}_{i,j} = 1/f, \forall i \in \{1,\dots,f\}$ and $j \in \{1,\dots,d\}$, 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 using a moving average filter as preprocessing on the signal and a SVM classifier will be called Avg-SVM.

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

$$\min_g \left\{ \frac{1}{2}||g||^2_{\mathcal{H}} + \frac{C}{n}\sum_i^n H(\mathbf{y}_i, g(\widetilde{\mathbf{X}}_{i,\cdot}))^p \right\} \quad (1)$$

where C is the 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 [23]). For the ℓ_1-SVM, one can solve the dual form of this problem w.r.t. g:

$$\max_\alpha \left\{ J_{SVM}(\alpha, \mathbf{F}) = -\frac{1}{2}\sum_{i,j}^{n,n} \mathbf{y}_i\mathbf{y}_j\alpha_i\alpha_j\widetilde{\mathbf{K}}_{i,j} + \sum_i^n \alpha_i \right\}$$
$$s.t. \quad \frac{C}{n} \geq \alpha_i \geq 0 \quad \forall i \quad and \quad \sum_i^n \alpha_i\mathbf{y}_i = 0$$

where α_i are the dual variables and $\widetilde{\mathbf{K}}$ is the kernel matrix for filtered samples. In the Gaussian case, $\widetilde{\mathbf{K}}$ is defined by:

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

where σ is the kernel bandwidth. Note that for any FIR filter, the resulting matrix $\widetilde{\mathbf{K}}$ is always positive definite if $k(\cdot,\cdot)$ is definite positive. To confirm that, suppose a kernel $k(\cdot,\cdot)$ from $\mathcal{X}^2$ to $\mathbb{R}$ and ϕ a mapping from any $\mathcal{X}'$ to $\mathcal{X}$, then $k'(\cdot,\cdot) = k(\phi(\cdot), \phi(\cdot))$ is a positive definite kernel [23]. Since the proposed filter is a linear combination of $\mathbb{R}^d$ elements the constructed kernel is always definite positive.

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

$$g(\widetilde{\mathbf{X}'}_{i,\cdot}) = \sum_j^n \alpha_j\mathbf{y}_j k(\widetilde{\mathbf{X}'}_{i,\cdot}, \widetilde{\mathbf{X}}_{j,\cdot}) + b \quad (3)$$

with α_j the dual variables learned by solving (2) and b the bias term. We show in the experiment section that this approach may lead to improved performance over the non-filtered approach. However, the method relies on the (possibly critical) choice of a filter structure which in turn depends on prior information or user's knowledge. There is no evidence whatsoever that the user-selected filter will be optimal for the given classification task.

C. Time-Window Classification

Another way for taking into account the sequentiality of the samples, *i.e.* for handling the non-*i.i.d.* caracteristics of the temporal samples, is to classify time-windows of samples. We define $\overline{\mathbf{X}} \in \mathbb{R}^{n\times(d\times f)}$ that stacks for every sample $\overline{\mathbf{X}}_{i,\cdot}$ the samples in the complete time window of length f with n_0 delay (see Fig. 1). Note that this approach leads to the classification of data in high dimension ($f \times d$). Finally, one can learn a SVM classifier on samples $\overline{\mathbf{X}}_{i,\cdot}$ using Equation (1). This method will be called Win-SVM in the following.

1) Linear Win-SVM: When the model is linear, it is often more efficient to solve the SVM in the primal (see Equation (8)). Indeed, the problem has $d + 1$ parameters to learn (separating hyperplane in $\mathbb{R}^d$) instead of $n + 1$.

If we want to learn a linear classifier on a window of samples, the problem may be expressed as the minimization of:

$$J_W(W, w_0) = \frac{1}{2}||\mathbf{W}||^2_F + \frac{C}{n}\sum_i^n H(\mathbf{y}, g_W(i, \mathbf{X}))^p, \quad (4)$$

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

$$g_W(i, \mathbf{X}) = \sum_{m,j}^{f,d} \mathbf{W}_{m,j}\mathbf{X}_{i+1-m+n_0,j} + w_0$$

where $\mathbf{W} \in \mathbb{R}^{f\times d}$ and $w_0 \in \mathbb{R}$ are the classification parameters. Note that we choose $p = 2$ for the linear case as it allows the objective function to be differentiable. Thus, we can take advantage of many linear SVM solvers existing in the literature such as the one proposed by Chapelle [24]. Using that solver, Win-SVM complexity is about $\mathcal{O}(n \times (f \times d)^2)$ which scales quadratically with the filter dimension.

One of the interests of this approach is that W may be seen as a large margin filtering. Indeed the columns of the W matrix correspond to a temporal filtering whereas the rows correspond to a spatial filtering. However, previous experiments have shown a decrease in performance for high dimensional signal [21]. This comes from the fact that the Frobenius norm does not promote sparsity and treats all the coefficients independently.

2) Channel selection for linear Win-SVM: In some applications, among all channels that have been acquired some may be more informative than others and thus it may be useful, in terms of prediction performance and interpretability, to perform channel selection. This situation occurs for instance in BCI problems, where discriminative features are usually well-localized. To include an automated channel selection procedure in the time-window classification problen given in (4), we propose to consider a $\ell_1 - \ell_2$ mixed norm as a regularizer instead of the Frobenius norm :

$$\Omega_{1-2}(\mathbf{W}) = \sum_j^d \left(\sum_i^f \mathbf{W}^2_{i,j}\right)^{\frac{1}{2}} = \sum_j^d h\left(||\mathbf{W}_{\cdot,j}||^2\right) \quad (5)$$

with $h(u) = u^{\frac{1}{2}}$ 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 optimization problem when using (5) for regularization is a problem with non-differentiable objective function which may pose difficulties. However, recently, there has been a lot of works which aim at solving this kind of problem where one part of the objective function has a Lipschitz gradient and the other part is convex but non differentiable [25], [26]. Here we straighforwadly applied the accelerated gradient method proposed (AGP) by Chen et al. [27] since the squared hinge loss is known to be Lipschitz gradient. For more details about the algorithm, the reader is referred to the work of Chen et al.

III. Large Margin Filtering

The classification of a time window is a way to handle temporal information, but the classification approach used treats all the coefficients independently. Doing a per-channel convolution is sensible here as it will take into account the channel structure and extract the discriminative information spread along time. But in order to adapt to the problem at hand, we propose to learn that filtering jointly with the sample classifier. In this section, we present the optimization problem to learn a large margin filtering along with a general algorithm to solve it. Then we discuss the implementation of this method, named KF-SVM, in the linear and non-linear case. 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:

$$\min_{g,\mathbf{F}} \left\{ \frac{1}{2}||g||^2_{\mathcal{H}} + \frac{C}{n}\sum_{i=1}^{n} H(\mathbf{y}_i, \widetilde{\mathbf{X}}_{i,\cdot}, g)^p + \lambda\Omega(\mathbf{F}) \right\}, \quad (6)$$

where λ is a regularization parameter and $\Omega(\cdot)$ represents a differentiable regularization function of $\mathbf{F}$. Note that the left part of (6) is a standard SVM for filtered samples $\widetilde{\mathbf{X}}$, but here $\mathbf{F}$ is a variable to be minimized instead of a fixed one. This objective function is non-convex. However, the optimization problem w.r.t. $g(\cdot)$ is convex for a fixed $\mathbf{F}$ and boils down to a SVM problem. Therefore we propose to solve (6) by minimizing the following objective function with respect to $\mathbf{F}$:

$$J(F) = J'(\mathbf{F}) + \lambda\Omega(\mathbf{F}) \quad (7)$$

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

$$J'(\mathbf{F}) = \min_{g} \left\{ \frac{1}{2}||g||^2_{\mathcal{H}} + \frac{C}{n}\sum_{i}^{n} H(\mathbf{y}_i, \widetilde{\mathbf{X}}_{i,\cdot}, g)^p \right\} \quad (8)$$

and its corresponding dual problem is

$$J'(\mathbf{F}) = \max_{C/n \geq \alpha \geq 0, \sum_i \alpha_i \mathbf{y}_i = 0} \left\{ J_{SVM}(\alpha, \mathbf{F}) \right\} \quad (9)$$

where J_{SVM} is defined by (2) and $g(\cdot)$ by (3). Due to the strong duality of the SVM problem, $J'(\cdot)$ can be expressed in his primal or dual form (see (8) and (9)). The objective function J defined in (7) is non-convex. But, according to [28], for a given $\mathbf{F}^*$ with $g^*(\cdot)$ the solution of problem (8), $J'(\cdot)$ is differentiable w.r.t. $\mathbf{F}$. At the point $\mathbf{F}^*$, the gradient of $J(\cdot)$ can be computed. Finally we can solve the problem in (7) by doing a gradient descent on $J(\mathbf{F})$ along $\mathbf{F}$.

Note that due to the non-convexity of the objective functions, solving problems (6) and (7) may lead to different results. However, it is advantageous to consider (7) because it can be solved using standard SVM solvers [29], [30] and our method would benefit from any improvement in this domain.

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 for finding the optimal step. The method is detailed in Algorithm 1, where β is the CG update parameter and $D^i_{\mathbf{F}}$ the descent direction for the ith iteration. For the experimental results we used the β proposed by Fletcher and Reeves, see [31] for more information. The iterations in the algorithm may be stopped by two stopping criteria: a threshold on the relative variation of $J(F)$ or on the norm of the variation of $\mathbf{F}$.

Algorithm 1 KF-SVM solver

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

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 used to solve the multiple-kernel problem in [32], [33]. In these works, an objective function was minimized with respect to kernel parameters (kernel weight in [32] or bandwidth in [33]) using a gradient descent algorithm.

C. Complexity in the linear and non-linear 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 cost computation in the line-search. In this section, we discuss the complexity of these tasks in both the linear and non-linear cases.

In the linear case, the optimization problem can be solved in the primal. 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 [24], [34]. In this paper, we used the solver proposed by Chapelle et al., which learns the SVM classifier by using a CG descent or a Newton descent algorithm. In order to make both our algorithm and the SVM solver work, we need the

objective function to be differentiable, so we used a ℓ_2-SVM in the linear case. The gradient of $J'(\cdot)$ at point $\mathbf{F}$ is:

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

where g_i^* is the ith component of the optimal separating hyperplane g^* when using the matrix $\mathbf{F}$ for filtering. The complexity of this 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 the dual space. In this case, in order to obtain a sparse representation (in terms of support vectors), we solve the ℓ_1-SVM ($p = 1$). We compute the gradient of $J'(\cdot)$ for a Gaussian kernel using the dual formulation of equation (9). The gradient of $J'(\cdot)$ at a given point $\mathbf{F}$ is:

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

where α^* is the SVM solution for $\widetilde{\mathbf{X}}$ signal filtered by $\mathbf{F}$ and $\widetilde{\mathbf{K}}$ is the kernel matrix of the filtered samples $\widetilde{\mathbf{X}}_{i,.}$. The complexity of this gradient is $\mathcal{O}(n^2 \times f \times d)$ but, in practice, SVMs have a sparse representation. So in fact the gradient computation is $\mathcal{O}(n_s^2 \times f \times d)$ with n_s being the number of support vectors.

In conclusion, due to the non-convexity of the objective function, it is difficult to provide an exact evaluation of the solution complexity. However, we know that the gradient computation is $\mathcal{O}(n_s^2 \times f \times d)$ in the non-linear case and $\mathcal{O}(n \times f \times d)$ in the linear case. Moreover, when $J(F)$ is computed 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). Note that a warm-start trick may be used when using iteratively the SVM solver in order to speed up the method. It consists in using the last results returned by the SVM solver as starting point for the new problem to solve.

D. Filter regularization

In this section, we discuss the choice of the filter regularization term. This choice is important because learning the FIR filters adds parameters to the problem and regularization is essential in order to avoid over-fitting.

The first regularization term for the filter that we consider and use in our KF-SVM framework is the Frobenius norm:

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

This regularization term is differentiable and the gradient is easy to compute. Minimizing this regularization term corresponds to minimizing the filter energy. In terms of classification, the filter matrix can be seen as a kernel parameter that weights delayed samples. For a given channel, such a sequential weighting is related to a phase/delay and cut-off frequency of the filter. Moreover, the Gaussian kernel defined in II-B shows that the column-wise convolution can be seen as a scaling of the channels before the kernel computation. 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 vanish thus yielding to reduced impact on the kernel of some delayed samples.

Algorithm 2 SKF-SVM solver

Set $\mathbf{F}_{l,k} = 1/f$ for $k = 1 \cdots d$ and $l = 1 \cdots f$
Set $\mathbf{d}_k = 1$ for $k = 1 \cdots d$
repeat
  $(\mathbf{F}, \alpha) \leftarrow$ Solve KF-SVM with $\mathbf{d}$ column weights
  $\mathbf{d}_k \leftarrow \frac{1}{||\mathbf{F}_{.,k}||}$ for $k = 1 \cdots d$
until Stopping criterion is reached

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 (5) as a regularizer. However, this regularization term is not differentiable and the solver proposed in Algorithm 1 can not 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 [35] 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 (5) 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 re-use the KF-SVM algorithm. Indeed, at iteration $k+1$, applying this linear majorization of $h(\|\mathbf{F}_{.,j}\|)$ around a $\|\mathbf{F}_{.,j}^{(k)}\|$ yields to a Majorization-Minimization algorithm for sparse filter learning, which consists in solving:

$$\min_{\mathbf{F}^{(k+1)}} \left\{ J'(\mathbf{F}) + \lambda \Omega_{\mathbf{d}}(\mathbf{F}) \right\}$$
$$\text{with } \Omega_{\mathbf{d}}(\mathbf{F}) = \sum_{j}^{d} d_j \sum_{i}^{f} \mathbf{F}_{i,j}^2 \text{ and } d_i = \frac{1}{\|\mathbf{F}_{.,i}^{(k)}\|}$$

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 differentiable and the KF-SVM solver can 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 criterion as in Algorithm 1.

E. Related works

To the best of our knowledge, there has been few works dealing with the joint learning of a temporal filter and a decision function. The first one addressing such a problem is our work [21] that solves the problem for linear decision functions. Here, we have extended this approach to the non-linear case and we have also investigated the utility of different regularizers on the filter coefficients. Notably, we have introduced regularizers that help in performing channel selection.

Works on Common Spatio-Spectral Patterns (CSSSP) [36] 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 (non-linear) classification algorithm criterion. 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 [37]. 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 see that the coefficients of these columns act as a scaling of the channels. Finally note that, for a filter of size 1, our approach would correspond to adaptive scaling as proposed by [38]. In their work, the authors jointly learn the classifier and the Gaussian kernel parameter σ with a sparsity constraint on the dimensions of σ leading to automated feature selection. KF-SVM can thus be seen as a generalization of this approach, which takes into account sample sequentiality as well.

IV. Experimental results

This section contains the numerical experiments comparing the different approaches proposed. First we present numerical results on a toy dataset containing both convolutional and additive noise. Then we test the methods on a real life BCI dataset from *BCI Competition III* [4]. Finally, we extend the methods to a 2-dimensional problem of multispectral remote sensing image segmentation. The Matlab code for all the methods tested in this paper is available as Open Source Software[1].

[1]http://remi.flamary.com/

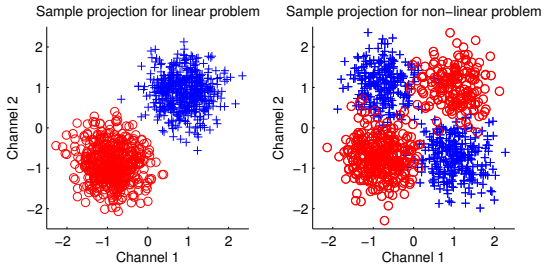

Fig. 2. Projection of the sample on the 2 discriminative channels for the linear case (left) and the non-linear case (Right). Here there is no convolutional noise to illustrate the original shape of the data.

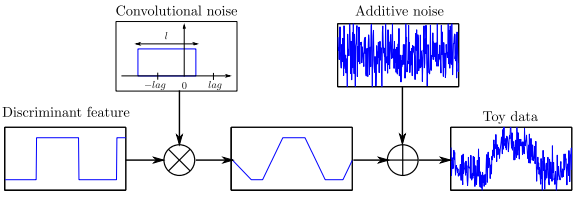

Fig. 3. Toy dataset creation: First a convolutional noise is applied to the discriminative feature and then a gaussian additive noise is added.

A. Experiment 1: Toy Dataset

This first experiment is intended to provide some insight on the capabilities of each method to handle feature weighting/selection. To do this, we use a toy dataset corrupted by both convolutional and additive noise. Within the data, discriminative and non-discriminative channels are present. We investigate the linear and the non-linear 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 contains 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 delay drawn from a uniform distribution on $[-\tau, \tau]$ is applied to the channel, then a moving-average filtering of size l is applied. Finally, additive Gaussian noise of standard deviation σ_n is added. Figure 3 summarizes how the noise is applied to the discriminant features.

Table I sums up the methods used in the experiments reported. 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 non-linear 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 in the validation set. All the processes are run 10 times, the test

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-D).
KF-GMM	GMM classifiers on the pre-filtered samples. The filter is the one obtained by KF-SVM
WinGMKL**	Multiple kernel learning proposed by [33] for feature selection applied on a window of samples. Publicly available code .

* only in the linear case. ** only in the non-linear case.

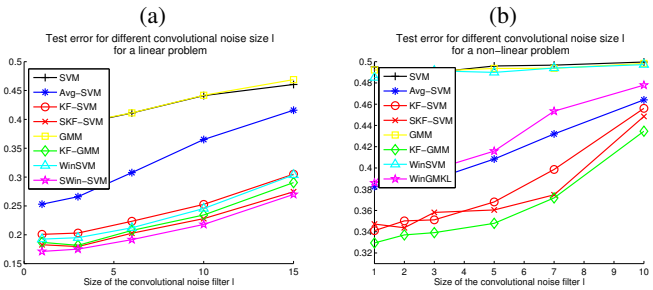

Fig. 4. Test error for different convolutional noise in the (a) linear and (b) non-linear cases.

error is then the average over the runs. A Wilcoxon's signrank test with a risk of 5% was applied to the results in order to check the statistical differences between the methods.

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 work well. The best method is SWinSVM followed closely by SKF-SVM (both are proven statistically equivalent by the Wilcoxon test). Those two approaches perform a channel selection leading to a better generalization. Note that WinSVM performs similarly as KF-SVM which is consistent with the results in [21] for small dimensional problems. Due to the Gaussian nature of this dataset, KF-GMM outperforms KF-SVM.

For the non-linear problem in Fig. 4(b), the statistically best methods are the large margin filtering methods (KF-SVM, SKF-SVM and KF-GMM). Note that the channel selection seems to improve the results when the noise is high. The best results are obtained here by KF-GMM as it uses the best model for the data after denoising. Note here the poor behavior of WinGMKL, that gives worse results than a simple average filtering.

B. Experiment 2: BCI Dataset

The BCI Dataset considered is one of the problems presented in *BCI Competition III* [4]. The problem is to obtain a sequence of labels out of brain activity signals for three human subjects. The data consists of 96 channels containing PSD features (three training sessions and one test session, with $n \approx 3000$ *per* session) and three possible labels (left arm, right arm or a word). To handle the large number of samples in the non-linear case, we use only a randomly selected subset of the available dataset (of about 30%). The regularization parameters are tuned using a grid search validation on the third training session.

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 bad performances on this high dimensional 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 ($\approx$ 500 filter coefficients are learned on $\approx$ 10000 samples for $f = 50$). The best overall results are obtained by KF-SVM and SKF-SVM with the filter length $f = 50$.

The results are similar for the non-linear models, showing that for this task a linear classifier is sufficient. But also it is important to keep in mind that, in this case, the parameters are learned on only 30% of the samples. The windowing methods are then more prone to over-fitting. 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 proposed to learn a large margin filtering may be easily extended to the 2-dimensional case. In this case 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 us 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 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. 5). 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 of roads by applying spatial filtering, because the spectral difference between these couples of classes is low.

Classification results 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 overall accuracy in One-Against-All and the estimated Cohen's kappa coefficient, which is a more appropriate

TABLE II
TEST ERROR FOR THE BCI DATASET FOR DIFFERENT METHODS, AND FILTER LENGTH f. RESULTS ARE GIVEN FOR THE LINEAR MODEL (TOP) AND FOR THE NON-LINEAR MODEL (BOTTOM).

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
Non-linear 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**

Note: best competition results are (0.2040, 0.2969, 0.4398 and 0.3135 for the average).

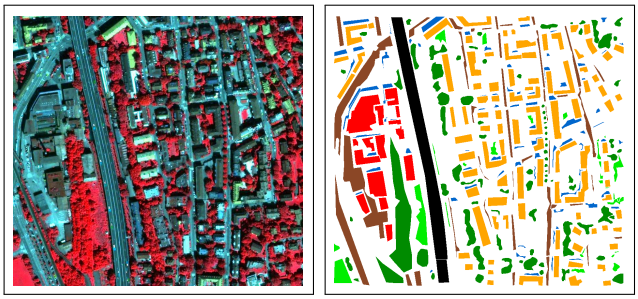

Fig. 5. QuickBird scene of suburbs of Zurich (left) and labeled pixels (right). Legend: *dark green = trees; light green = meadows; black = speedway; brown = roads; orange = residential buildings; red = commercial buildings; blue = shadows*.

TABLE III
RESULTS IN IMAGE SEGMENTATION. ONE-AGAINT-ALL ACCURACY AND KAPPA COEFFICIENT.

Method	Classes	Filter size	Training Pixels	[%]OA	Kappa
SVM	7	9	~ 5000	75.11	0.685
AvgSVM				83.68	0.796
WinSVM				82.98	0.785
KF-SVM				**85.32**	**0.816**
SVM	6*	9	~ 5000	83.04	0.772
AvgSVM				89.48	0.860
WinSVM				**91.71**	**0.889**
KF-SVM				91.45	0.885

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

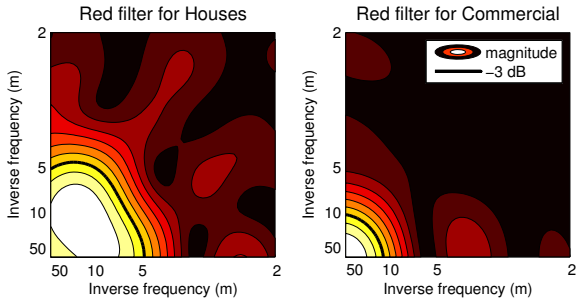

Fig. 6. Magnitude of the spatial Fourier transform on the Red component large margin filtering for the 'House' and 'Commercial' classes. The bold black lines correspond to the -3dB attenuation.

measure to evaluate the classification accuracy in imbalanced class problems. Two configurations are tested: 7 classes and 6 classes. For the last configuration, class 'Residential' and 'Commercial' are merged as 'Building' (see Figure 5 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 6 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.

V. CONCLUSION

In this work, we addressed the problem of multi-channel signal sequence labeling in the presence of additive and convolutional noise. We defined several methods based on filtering preprocessing and time-window classification. We proposed a framework for learning large-margin filtering jointly with a sample classifier. Depending on the regularization term used, we can do either adaptive scaling of the channels or channel selection. We proposed a conjugate gradient algorithm to solve

the minimization problem and we discussed the complexity of the method. We compared the different approaches on a non-linear toy example and on a real life BCI dataset, and these experiments showed the interest of learning a large margin filtering. Finally, we extended our approach to a 2-dimensional image segmentation problem and the interpretability of the results have been shown by visualizing the Fourier transform of the learned filters.

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 limits the size of the problem. We will investigate the use of a one-pass SVM solver such as [39] instead of an exact SVM solver.

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. M. Asl, S. K. 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. [Online]. Available: http://www.sciencedirect.com/science/article/B6T4K-4SVM0YC-1/2/6eda983faadf944cc5cd6aac47dc722c

[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] H. Lin, C. Lin, and R. Weng, "A note on Platt's probabilistic outputs for support vector machines," *Machine Learning*, vol. 68, no. 3, pp. 267–276, 2007.

[11] J. Platt, "Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods," *Advances in large margin classifiers*, 1999.

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

[13] 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

[14] 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.

[15] 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.

[16] J. Inglada, "Automatic recognition of man-made objects in high resolution optical remote sensing images by SVM classification of geometric image features," *Isprs J. Photogramm. Rem. Sens.*, vol. 62, no. 3, pp. 236–248, 2007.

[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] 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.

[22] 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.

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

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

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

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

[27] 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.

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

[29] S. Canu, Y. Grandvalet, V. Guigue, and A. Rakotomamonjy, "SVM and kernel methods Matlab toolbox," LITIS EA4108, INSA de Rouen, Rouen, France, 2003. [Online]. Available: http://asi.insa-rouen.fr/enseignants/~arakotom/toolbox/index.html

[30] C.-C. Chang and C.-J. Lin, *LIBSVM: a library for support vector machines*, 2001, software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

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

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

[33] 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.

[34] 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.

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

[36] 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.

[37] 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.

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

[39] 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.