OPTIMAL SLIDING WINDOW SPARSIFICATION FOR ONLINE KERNEL-BASED CLASSIFICATION BY PROJECTIONS

Konstantinos Slavakis

University of Peloponnese,
Dept. Telecommunications Science
and Technology,
Tripoli 22100, Greece.
Email: slavakis@uop.gr.

Sergios Theodoridis

University of Athens,
Department of Informatics
& Telecommunications,
Athens 15784, Greece.
Email: stheodor@di.uoa.gr.

ABSTRACT

This paper presents a new sparsification method for a very recently introduced projection-based algorithm for the online classification task in Reproducing Kernel Hilbert Spaces (RKHS). To accommodate limited computational resources, sparsification is achieved by a sequence of finite dimensional subspaces, with dimensions upper bounded by a predefined buffer length. In the case of a buffer overflow, the term that contributes the least to the kernel series expansion is removed. Such a sparsification scheme shows strong similarities with the classical sliding window adaptive schemes. We validate the proposed design by considering the adaptive equalization problem of a nonlinear communication channel. Since the fundamental tool of metric projections is used, and although a classification problem is considered here, the method can be readily extended to regression tasks, and to cost functions that are in general non-differentiable.

1. INTRODUCTION

We consider the *online setting* where *data* $\boldsymbol{x} \in \mathbb{R}^m$ arrive sequentially ($m \in \mathbb{Z}_{>0}$, with $\mathbb{R}$ and $\mathbb{Z}_{>0}$ denoting the set of real numbers and positive integers respectively). Time will be denoted by n, which takes values from the set of all non-negative integers: $n \in \mathbb{Z}_{\geq 0}$. Thus, the sequence of vectors $(\boldsymbol{x}_n)_{n\in\mathbb{Z}_{\geq 0}} \subset \mathbb{R}^m$ is assumed. Each data vector is drawn from two classes and is thus associated to a label $y_n \in \{\pm 1\}$, $n \in \mathbb{Z}_{\geq 0}$. Hence, the training sequence $\mathcal{D} := (\boldsymbol{x}_n, y_n)_{n\in\mathbb{Z}_{\geq 0}}$ is formed. The *classification problem in the RKHS* $\mathcal{H}$ is defined as selecting a point $\hat{f} \in \mathcal{H}$ and an *offset* $\hat{b} \in \mathbb{R}$ such that $y(\hat{f}(\boldsymbol{x}) + \hat{b}) \geq \rho$, $\forall (\boldsymbol{x}, y) \in \mathcal{D}$, for some *margin* $\rho \geq 0$ [1]. Define now the function $g_{f,b}(\boldsymbol{x}) := f(\boldsymbol{x}) + b$, $\forall \boldsymbol{x} \in \mathbb{R}^m$, $\forall (f, b) \in \mathcal{H} \times \mathbb{R}$ (where $\times$ denotes product space). If $(\hat{f}, \hat{b})$ is such that $yg_{\hat{f},\hat{b}}(\boldsymbol{x}) < \rho$, then the *classifier* $(\hat{f}, \hat{b})$ fails to achieve the margin ρ at $(\boldsymbol{x}, y)$ and we say that the classifier committed a *margin error*. A *misclassification* occurs at $(\boldsymbol{x}, y)$ if $yg_{\hat{f},\hat{b}}(\boldsymbol{x}) < 0$.

The study in [2, 3], based on the Adaptive Projected Subgradient Method [4, 5], derived a convex analytic approach [6, 7] for the online kernel-based classification. Given the parameters $(\boldsymbol{x}, y, \rho)$, the set of all classifiers that do not commit a margin error, i.e., $\Pi^+_{\boldsymbol{x},y,\rho} := \{\hat{u} := (\hat{f}, \hat{b}) \in \mathcal{H} \times \mathbb{R} : y(\hat{f}(\boldsymbol{x}) + \hat{b}) \geq \rho\}$, was shown to be a special closed convex set (a closed halfspace) of the Hilbert space $\mathcal{H} \times \mathbb{R}$. As a result, the online classification task was viewed as the problem of finding a point that belongs to the intersection of an infinite sequence of closed halfspaces $\bigcap_{n\geq n_0} \Pi^+_{\boldsymbol{x}_n,y_n,\rho_n}$, for some $n_0 \in \mathbb{Z}_{\geq 0}$ [3].

In short, the algorithmic solution developed in [3] is as follows: for any $n \in \mathbb{Z}_{\geq 0}$, consider the index set $\mathcal{J}_n \subset \overline{0, n}$, such that $n \in \mathcal{J}_n$, and where $\overline{j_1, j_2} := \{j_1, j_1 + 1, \ldots, j_2\}$ for any integers $j_1 \leq j_2$. An example of such an index set, to be used below, is

$$\mathcal{J}_n := \begin{cases} \overline{0, n}, & \text{if } n < q - 1, \\ \overline{n - q + 1, n}, & \text{if } n \geq q - 1, \end{cases} \quad \forall n \in \mathbb{Z}_{\geq 0}, \tag{1}$$

where $q \in \mathbb{Z}_{>0}$ is a predefined constant denoting the number of closed halfspaces to be concurrently processed at each time instant $n \geq q - 1$. In other words, $\mathcal{J}_n$ gives the freedom to process halfspaces corresponding to time instants previous to the current n, showing thus the same principle as in Affine Projection Algorithms (APA) [8]. Hence, for any $j \in \mathcal{J}_n$ and for any $n \in \mathbb{Z}_{\geq 0}$, let $\Pi^+_{j,n} := \{\hat{u} = (\hat{f}, \hat{b}) \in \mathcal{H} \times \mathbb{R} : y_j(\hat{f}(\boldsymbol{x}_j) + \hat{b}) \geq \rho_j^{(n)}\}$. Let also the weight $\omega_j^{(n)} \geq 0$ such that $\sum_{j\in\mathcal{J}_n} \omega_j^{(n)} = 1$. In other words, in order to assign different significance to every closed halfspace, we associate to each one of them a weight $\omega_j^{(n)}$. For an arbitrary initial offset $b_0 \in \mathbb{R}$, consider as an initial classifier the point $u_0 := (0, b_0) \in \mathcal{H} \times \mathbb{R}$ and generate the following point sequence $u_n := (f_n, b_n)$, $n \in \mathbb{Z}_{\geq 0}$, in $\mathcal{H} \times \mathbb{R}$ as follows; $\forall n \in \mathbb{Z}_{\geq 0}$,

$$u_{n+1} := u_n + \mu_n \Big(\sum_{j\in\mathcal{J}_n} \omega_j^{(n)} P_{\Pi^+_{j,n}}(u_n) - u_n \Big), \tag{2a}$$

where the *extrapolation parameter* μ_n lies in the range $\mu_n \in [0, 2\mathcal{M}_n]$, with $\mathcal{M}_n \geq 1$, as can be seen in [3] and (2e) below. If we use the closed form expressions for the projections $P_{\Pi_{j,n}^+}$ [3], (2a) breaks down to the following relations. Define $\forall j \in \mathcal{J}_n, \forall n \in \mathbb{Z}_{\geq 0}$,

$$\beta_j^{(n)} := \omega_j^{(n)} y_j \frac{(\rho_j^{(n)} - y_j g_n(\boldsymbol{x}_j))^+}{1 + \kappa(\boldsymbol{x}_j, \boldsymbol{x}_j)}, \tag{2b}$$

where we let $g_n := g_{f_n, b_n}$, and where $(\cdot)^+ := \max(0, \cdot)$. Then the algorithmic process (2a) can be written equivalently as follows; $\forall n \in \mathbb{Z}_{\geq 0}$,

$$f_{n+1} = f_n + \mu_n \sum_{j \in \mathcal{J}_n} \beta_j^{(n)} \kappa(\boldsymbol{x}_j, \cdot), \tag{2c}$$

$$b_{n+1} = b_n + \mu_n \sum_{j \in \mathcal{J}_n} \beta_j^{(n)}. \tag{2d}$$

The parameter $\mathcal{M}_n$ takes the following form after the proper algebraic manipulations:

$$\mathcal{M}_n := \begin{cases} \frac{\sum_{j \in \mathcal{J}_n} \omega_j^{(n)} \frac{[(\rho_j^{(n)} - y_j g_n(\boldsymbol{x}_j))^+]^2}{1 + \kappa(\boldsymbol{x}_j, \boldsymbol{x}_j)}}{\sum_{i,j \in \mathcal{J}_n} \beta_i^{(n)} \beta_j^{(n)} (1 + \kappa(\boldsymbol{x}_i, \boldsymbol{x}_j))}, & \text{if } u_n \notin \bigcap_{j \in \mathcal{J}_n} \Pi_{j,n}^+, \\ 1, & \text{otherwise.} \end{cases} \tag{2e}$$

An algorithm to dynamically control the margin parameters $(\rho_j^{(n)})$, $j \in \mathcal{J}_n$, $n \in \mathbb{Z}_{\geq 0}$, was also proposed in [3]. This approach will also be followed for the present work, but with the simplification that $\rho_n := \rho_j^{(n)}$, $\forall j \in \mathcal{J}_n$, $\forall n \in \mathbb{Z}_{\geq 0}$, i.e., the margin parameters corresponding to the concurrently processed closed halfspaces are equal to each other.

It was shown in [3] that under mild conditions, the point sequence $(f_n)_n$ associated with recursion (2a), (strongly) converges to an f_* expressed by a kernel series expansion

$$f_* = \sum_{n=1}^{\infty} \gamma_n \kappa(\boldsymbol{x}_n, \cdot) \in \mathcal{H}, \tag{3}$$

where $(\gamma_n)_n$ are real numbers, and $\kappa(\boldsymbol{x}_n, \cdot)$ stands for the kernel function of the RKHS $\mathcal{H}$ parameterized by the data $\boldsymbol{x}_n$. The point sequence in (2a) asymptotically minimizes also the sequence of cost functions $(l_{\boldsymbol{x}_j, y_j, \rho_j^{(n)}})$, $j \in \mathcal{J}_n$, $n \in \mathbb{Z}_{\geq 0}$, where the *soft margin loss function* l is a standard penalty function for classification problems [1]: given a pair $(\boldsymbol{x}, y) \in \mathcal{D}$ and the margin ρ, let

$$l_{\boldsymbol{x}, y, \rho}((f, b)) := (\rho - y g_{f,b}(\boldsymbol{x}))^+, \quad \forall (f, b) \in \mathcal{H} \times \mathbb{R}.$$

Note that the soft margin loss function is a non-negative, convex, and non-differentiable function.

To accommodate sparsification in (3), an additional closed convex constraint (a closed ball) was imposed on the norm of the estimates $(f_n)_n$ [3]. The method in [3] produced superior results, with *linear complexity*, for an adaptive equalization problem of a nonlinear communication channel, when compared to stochastic gradient descent approaches like the classical kernel perceptron method as well as its soft margin generalization NORMA [9]. Since the above approach is based on the fundamental notion of (metric) projections, it can be straightforwardly extended to cover also regression tasks.

This paper introduces a new sparsification method for (3) by the construction of a sequence of subspaces $(M_n)_n$, and by an upper bound L_b on their dimensions $L_n := \dim(M_n)$, i.e., $L_n \leq L_b$, $\forall n \in \mathbb{Z}_{\geq 0}$. It will be seen that such an upper bound is equivalent to a buffer of length L_b which contains the coefficients of the kernel series expansion in (4). Whenever a buffer overflow occurs, the coefficient that contributes the least in (4) is removed. It turns out that such a sparsification scheme shows strong similarities with the classical sliding window adaptive schemes [8].

In this paper we will construct a sequence of bases $\mathcal{B}_n := \{\psi_l^{(n)}\}_{l=1}^{L_n}$, $\forall n \in \mathbb{Z}_{\geq 0}$, which will provide us with a sparsification of the series expansion in (3) as

$$\tilde{f}_n = \sum_{l=1}^{L_{n-1}} \tilde{\gamma}_l^{(n)} \psi_l^{(n-1)} \in M_{n-1}, \quad \forall n \in \mathbb{Z}_{\geq 0}, \tag{4}$$

where, in order to avoid any ambiguity, we set $\mathfrak{B}_{-1} := \{0\}$, $M_{-1} := \{0\}$, and $L_{-1} := 1$.

2. MATHEMATICAL PRELIMINARIES

2.1. Reproducing Kernel Hilbert Space (RKHS).

In this paper, the symbol $\mathcal{H}$ stands for an infinite dimensional, in general, real Hilbert space equipped with an inner product denoted by $\langle \cdot, \cdot \rangle$. The induced norm becomes $\|f\| := \langle f, f \rangle^{1/2}$, $\forall f \in \mathcal{H}$.

Assume a real Hilbert space $\mathcal{H}$ consisting of functions defined on $\mathbb{R}^m$, i.e., $f : \mathbb{R}^m \to \mathbb{R}$, for some $m \in \mathbb{Z}_{>0}$. The function $\kappa(\cdot, \cdot) : \mathbb{R}^m \times \mathbb{R}^m \to \mathbb{R}$ is called a *reproducing kernel* of $\mathcal{H}$ if $\forall \boldsymbol{x} \in \mathbb{R}^m$ and $\forall f \in \mathcal{H}$, $f(\boldsymbol{x}) = \langle f, \kappa(\boldsymbol{x}, \cdot) \rangle$. In this case, $\mathcal{H}$ is called a *Reproducing Kernel Hilbert Space (RKHS)* [1].

Celebrated examples are i) the linear kernel, and ii) the Gaussian kernel $\kappa(\boldsymbol{x}, \boldsymbol{y}) := \exp(-\frac{(\boldsymbol{x}-\boldsymbol{y})^t(\boldsymbol{x}-\boldsymbol{y})}{2\sigma^2})$, $\forall \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^m$, where $\sigma > 0$ (here the associated RKHS is of infinite dimension [1]).

2.2. Metric and orthogonal projection mappings.

Given a point $f \in \mathcal{H}$ and a closed convex set $C \subset \mathcal{H}$, a way to move from f to a point in C is by means of the *metric projection mapping* P_C *onto* C, which is defined as the mapping that takes f to the *uniquely* existing point $P_C(f)$ of C such that $\|f - P_C(f)\| = \inf\{\|f - f'\| : f' \in C\}$ [7] (see Fig. 1).

A well-known example of a closed convex set is a *closed linear subspace* M [7]. The metric projection mapping P_M is

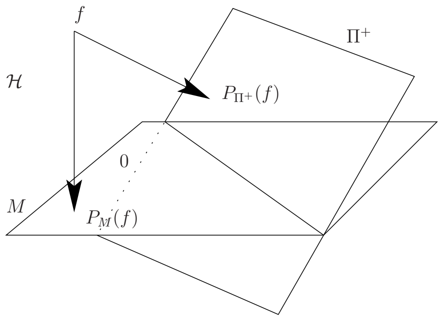

Fig. 1. An illustration of the orthogonal projection mapping P_M onto the closed subspace M of $\mathcal{H}$, and of the metric projection P_{Π^+} onto the closed halfspace $\Pi^+ := \{\hat{f} : \langle \hat{f}, f_0 \rangle \geq \gamma_0\}$, for some given $f_0 \in \mathcal{H}$ and some $\gamma \in \mathbb{R}$. A closed halfspace, thus, stands for all those points of $\mathcal{H}$ that lie in the "nonnegative" side of $\mathcal{H}$ defined by the boundary $\{\hat{f} : \langle \hat{f}, f_0 \rangle = \gamma_0\}$.

called now *orthogonal projection* since $\langle f - P_M(f), \hat{f} \rangle = 0$, $\forall \hat{f} \in M, \forall f \in \mathcal{H}$ [7].

3. SPARSIFICATION BY A SEQUENCE OF FINITE DIMENSIONAL SUBSPACES

The following methodology is in the same line with the study in [10], where a monotonically increasing chain of linear subspaces $(M_n)_n$ is derived: $M_n \subseteq M_{n+1}, \forall n$. If the sequence of data vectors $(\boldsymbol{x}_n)_n$ lies in a compact set of $\mathbb{R}^m$, then it was shown in [10] that the dimensions of $(M_n)_n$ are upper bounded by some positive integer, which is not available a-priori. In what follows, and to accommodate limited computational resources, we a-priori set a bound L_b for the dimensions, leading thus to a sequence $(M_n)_n$ which may not be in general monotonically increasing, i.e., we may have some N_0 such that $M_{N_0} \nsubseteq M_{N_0+1}$.

We will introduce now the sparsification method by using the constructive approach of mathematical induction on the time index $n \in \mathbb{Z}_{\geq 0}$.

3.1. Initialization $(n = 0)$.

At the starting time $n = 0$, our basis $\mathfrak{B}_0$ consists only of the vector $\psi_1^{(0)} := \kappa(\boldsymbol{x}_0, \cdot) \in \mathcal{H}$, i.e., $\mathfrak{B}_0 := \{\psi_1^{(0)}\}$. Define also $M_0 := \mathrm{span}(\mathfrak{B}_0)$, where span stands for the linear span of a collection of vectors. The description of the element $\kappa(\boldsymbol{x}_0, \cdot)$ by the basis $\mathfrak{B}_0$ is obvious here: $\kappa(\boldsymbol{x}_0, \cdot) = 1 \cdot \psi_1^{(0)}$. Hence, we can associate to $\kappa(\boldsymbol{x}_0, \cdot)$ the one-dimensional vector $\boldsymbol{\theta}_{\boldsymbol{x}_0}^{(0)} := 1$. Let also $K_0 := \kappa(\boldsymbol{x}_0, \boldsymbol{x}_0) > 0$, so that $K_0^{-1} = 1/\kappa(\boldsymbol{x}_0, \boldsymbol{x}_0)$.

As for the initial classifier $\tilde{u}_0 := (\tilde{f}_0, \tilde{b}_0)$, we start with $\tilde{f}_0 := 0$, and with an arbitrary offset $\tilde{b}_0 \in \mathbb{R}$. We let thus $\tilde{\gamma}_1^{(0)} := 0$ in (4).

3.2. Time instant $n - 1 \in \mathbb{Z}_{\geq 0}$.

We assume, now, that at time $n-1 \in \mathbb{Z}_{\geq 0}$ the basis $\mathfrak{B}_{n-1} := \{\psi_l^{(n-1)}\}_{l=1}^{L_{n-1}}$ is available, where $L_{n-1} \in \mathbb{Z}_{>0}$. Define also the linear subspace $M_{n-1} := \mathrm{span}(\mathfrak{B}_{n-1})$, which is of dimension L_{n-1}.

At time $n-1$, the index set $\mathcal{J}_{n-1} = \overline{n-q, n-1}$ is available (for simplicity we consider here only the case $n \geq q$ in (1)). Available are also the kernel functions $\{\kappa(\boldsymbol{x}_j, \cdot)\}_{j \in \mathcal{J}_{n-1}}$. From now and on, to each $\kappa(\boldsymbol{x}_j, \cdot)$, we associate a vector $\boldsymbol{\theta}_{\boldsymbol{x}_j}^{(n-1)} \in \mathbb{R}^{L_{n-1}}$, which gives us a point $k_{\boldsymbol{x}_j}^{(n-1)}$ of M_{n-1} that approximates $\kappa(\boldsymbol{x}_j, \cdot)$: $\forall j \in \mathcal{J}_{n-1}$,

$$\kappa(\boldsymbol{x}_j, \cdot) \mapsto k_{\boldsymbol{x}_j}^{(n-1)} := \sum_{l=1}^{L_{n-1}} \theta_{\boldsymbol{x}_j, l}^{(n-1)} \psi_l^{(n-1)} \in M_{n-1}. \quad (5)$$

Since our approach here is inductive, we assume that the vectors $\{\boldsymbol{\theta}_{\boldsymbol{x}_j}^{(n-1)}\}_{j=1}^{\mathcal{J}_{n-1}}$ are also available. Their construction will be made clear in the next section. At time 0, $k_{\boldsymbol{x}_0}^{(0)} := \psi_1^{(0)}$, so that $\boldsymbol{\theta}_{\boldsymbol{x}_0}^{(0)} := 1$.

Consider also the matrix $K_{n-1} \in \mathbb{R}^{L_{n-1} \times L_{n-1}}$ whose (i,j)-th component is $(K_{n-1})_{i,j} := \langle \psi_i^{(n-1)}, \psi_j^{(n-1)} \rangle, \forall i, j \in \overline{1, L_{n-1}}$. By our hypothesis that $\{\psi_l^{(n-1)}\}_{l=1}^{L_{n-1}}$ constitute a basis, and are thus linearly independent, it can be verified that K_{n-1} is a positive definite Gram matrix [11]. As such, the existence of its inverse K_{n-1}^{-1} is guaranteed. The availability of K_{n-1}^{-1} is also assumed here.

We assume, also, the existence of the set of coefficients $\{\tilde{\gamma}_l^{(n)}\}_{l=1}^{L_{n-1}}$ which define our estimate $\tilde{f}_n$ by (4). Available is also the offset $\tilde{b}_n$.

3.3. At time n, a new data vector $\boldsymbol{x}_n$ is available.

At time n, a new element $\kappa(\boldsymbol{x}_n, \cdot)$ of $\mathcal{H}$ becomes available. Since M_{n-1} is a finite dimensional linear subspace of $\mathcal{H}$, the orthogonal projection of $\kappa(\boldsymbol{x}_n, \cdot)$ onto M_{n-1} is well-defined and given by $P_{M_{n-1}}(\kappa(\boldsymbol{x}_n, \cdot)) = \sum_{l=1}^{L_{n-1}} \zeta_{\boldsymbol{x}_n, l}^{(n)} \psi_l^{(n-1)}$, where the vector $\boldsymbol{\zeta}_{\boldsymbol{x}_n}^{(n)} \in \mathbb{R}^{L_{n-1}}$ satisfies the normal equations $K_{n-1} \boldsymbol{\zeta}_{\boldsymbol{x}_n}^{(n)} = \boldsymbol{c}_{\boldsymbol{x}_n}^{(n)}$, with $\boldsymbol{c}_{\boldsymbol{x}_n}^{(n)}$ given by [7]

$$\boldsymbol{c}_{\boldsymbol{x}_n}^{(n)} := [\langle \kappa(\boldsymbol{x}_n, \cdot), \psi_1^{(n-1)} \rangle, \ldots, \langle \kappa(\boldsymbol{x}_n, \cdot), \psi_{L_{n-1}}^{(n-1)} \rangle]^t.$$

Since K_{n-1}^{-1} was assumed available, we can compute

$$\boldsymbol{\zeta}_{\boldsymbol{x}_n}^{(n)} = K_{n-1}^{-1} \boldsymbol{c}_{\boldsymbol{x}_n}^{(n)}. \quad (6)$$

The distance d_n of $\kappa(\boldsymbol{x}_n, \cdot)$ from M_{n-1} can be calculated as follows:

$$\begin{aligned} d_n^2 &:= \| \kappa(\boldsymbol{x}_n, \cdot) - P_{M_{n-1}}(\kappa(\boldsymbol{x}_n, \cdot)) \|^2 \\ &= \kappa(\boldsymbol{x}_n, \boldsymbol{x}_n) - (\boldsymbol{c}_{\boldsymbol{x}_n}^{(n)})^t \boldsymbol{\zeta}_{\boldsymbol{x}_n}^{(n)}, \end{aligned}$$

where the proof of the second equality is omitted due to lack of space. To visualize d_n, refer to Fig. 1 and consider f as

$\kappa(\boldsymbol{x}_n,\cdot)$ and M as M_{n-1}. Then, d_n becomes $\|f - P_M(f)\|$. Recall that if $d_n = 0$, then $\kappa(\boldsymbol{x}_n,\cdot) \in M_{n-1}$, and $\kappa(\boldsymbol{x}_n,\cdot)$ is linearly dependent on $\{\psi_l^{(n-1)}\}_{l=1}^{L_{n-1}}$. Fix, now, an $\alpha \geq 0$.

3.3.1. Approximate linear dependency ($d_n \leq \alpha$).

If the distance satisfies $d_n \leq \alpha$, then we say that $\kappa(\boldsymbol{x}_n,\cdot)$ is *approximately linearly dependent* on $\mathfrak{B}_{n-1} = \{\psi_l^{(n-1)}\}_{l=1}^{L_{n-1}}$, and that it is not necessary to insert $\kappa(\boldsymbol{x}_n,\cdot)$ into the new basis $\mathfrak{B}_n$. That is, we keep $\mathfrak{B}_n := \mathfrak{B}_{n-1}$, which clearly implies that $L_n := L_{n-1}$, and $\psi_l^{(n)} := \psi_l^{(n-1)}$, $\forall l \in \overline{1, L_n}$. Moreover, $M_n := \operatorname{span}(\mathfrak{B}_n) = M_{n-1}$. We also let $K_n := K_{n-1}$, and $K_n^{-1} := K_{n-1}^{-1}$.

Since the current time instant is n, the definition of the index set $\mathcal{J}_n$ in (1) suggests that we need the kernel functions $\{\kappa(\boldsymbol{x}_j,\cdot)\}_{j\in\mathcal{J}_n}$, and the vectors $\{\boldsymbol{\theta}_{\boldsymbol{x}_j}^{(n)}\}_{j=1}^{\mathcal{J}_n}$ by (5). Till now, all of the information was contained in the subspace M_{n-1}. To transfer all of our information to the newly obtained subspace M_n, we notice that $M_n = M_{n+1}$, so we let $\boldsymbol{\theta}_{\boldsymbol{x}_j}^{(n)} := \boldsymbol{\theta}_{\boldsymbol{x}_j}^{(n-1)}$, $\forall j \in \mathcal{J}_n \setminus \{n\}$. This implies that $k_{\boldsymbol{x}_j}^{(n)} = k_{\boldsymbol{x}_j}^{(n-1)}$, $\forall j \in \mathcal{J}_n \setminus \{n\}$. As for $k_{\boldsymbol{x}_n}^{(n)}$ we define $k_{\boldsymbol{x}_n}^{(n)} := P_{M_{n-1}}(\kappa(\boldsymbol{x}_n,\cdot))$, such that the corresponding $\boldsymbol{\theta}_{\boldsymbol{x}_n}^{(n)}$ becomes the vector $\boldsymbol{\zeta}_{\boldsymbol{x}_n}^{(n)}$ given in (6).

As in [3], the coefficients which realize the metric projection mappings onto the closed halfspaces are

$$\tilde{\beta}_j^{(n)} := \omega_j^{(n)} y_j \frac{(\rho_j^{(n)} - y_j \tilde{g}_n(\boldsymbol{x}_j))^+}{1 + \kappa(\boldsymbol{x}_j, \boldsymbol{x}_j)}, \forall j \in \mathcal{J}_n, \forall n \in \mathbb{Z}_{\geq 0}, \tag{7a}$$

with $\tilde{g}_n := g_{\tilde{f}_n, \tilde{b}_n}$, and the function g is defined is Section 1. The extrapolation coefficient $\tilde{\mu}_n \in [0, 2\tilde{\mathcal{M}}_n]$, where

$$\tilde{\mathcal{M}}_n := \begin{cases} \frac{\sum_{j\in\mathcal{J}_n} \omega_j^{(n)} \frac{[(\rho_j^{(n)} - y_j \tilde{g}_n(\boldsymbol{x}_j))^+]^2}{1+\kappa(\boldsymbol{x}_j,\boldsymbol{x}_j)}}{\sum_{i,j\in\mathcal{J}_n} \tilde{\beta}_i^{(n)} \tilde{\beta}_j^{(n)} (1+\kappa(\boldsymbol{x}_i,\boldsymbol{x}_j))}, & \text{if } u_n \notin \bigcap_{j\in\mathcal{J}_n} \Pi_{j,n}^+, \\ 1, & \text{otherwise.} \end{cases} \tag{7b}$$

$$\begin{aligned} \tilde{f}_{n+1} &:= \tilde{f}_n + \tilde{\mu}_n \sum_{j\in\mathcal{J}_n} \tilde{\beta}_j^{(n)} k_{\boldsymbol{x}_j}^{(n)} \\ &= \tilde{f}_n + \tilde{\mu}_n \sum_{j\in\mathcal{J}_n} \tilde{\beta}_j^{(n)} \left(\sum_{l=1}^{L_n} \theta_{\boldsymbol{x}_j,l}^{(n)} \psi_l^{(n)} \right) \\ &= \sum_{l=1}^{L_{n-1}} \tilde{\gamma}_l^{(n)} \psi_l^{(n-1)} + \sum_{l=1}^{L_n} \left(\tilde{\mu}_n \sum_{j\in\mathcal{J}_n} \tilde{\beta}_j^{(n)} \theta_{\boldsymbol{x}_j,l}^{(n)} \right) \psi_l^{(n)} \end{aligned} \tag{7c}$$

where the vectors $(\boldsymbol{\theta}_{\boldsymbol{x}_j}^{(n)})_{j\in\mathcal{J}_n}$, $\forall n \in \mathbb{Z}_{\geq 0}$ are calculated as in the previous section, and the offsets

$$\tilde{b}_{n+1} := \tilde{b}_n + \tilde{\mu}_n \sum_{j\in\mathcal{J}_n} \tilde{\beta}_j^{(n)}, \quad \forall n \in \mathbb{Z}_{\geq 0}. \tag{7d}$$

By (7c), if we define

$$\tilde{\gamma}_l^{(n+1)} := \tilde{\gamma}_l^{(n)} + \tilde{\mu}_n \sum_{j\in\mathcal{J}_n} \tilde{\beta}_j^{(n)} \theta_{\boldsymbol{x}_j,l}^{(n)}, \ \forall l \in \overline{1, L_n}, \tag{8}$$

we finally obtain

$$\tilde{f}_{n+1} = \sum_{l=1}^{L_n} \tilde{\gamma}_l^{(n+1)} \psi_l^{(n)}. \tag{9}$$

3.3.2. Approximate linear independency ($d_n > \alpha$), and no buffer overflow ($L_{n-1} + 1 \leq L_b$).

On the other hand, if $d_n > \alpha$, then $\kappa(\boldsymbol{x}_n,\cdot)$ is declared as *approximately linearly independent* on $\mathfrak{B}_{n-1}$, and we add it to our new basis $\mathfrak{B}_n$. If we also have $L_{n-1} \leq L_b - 1$, then we can increase the dimension of the basis without exceeding the memory of the buffer: $L_n := L_{n-1} + 1$ and $\mathfrak{B}_n := \mathfrak{B}_{n-1} \cup \{\kappa(\boldsymbol{x}_n,\cdot)\}$, such that the elements $\{\psi_l^{(n)}\}_{l=1}^{L_n}$ of $\mathfrak{B}_n$ become $\psi_l^{(n)} := \psi_l^{(n-1)}$, $\forall l \in \overline{1, L_{n-1}}$, and $\psi_{L_n}^{(n)} := \kappa(\boldsymbol{x}_n,\cdot)$.

We also update the Gram matrix by

$$K_n := \begin{bmatrix} K_{n-1} & \boldsymbol{c}_{\boldsymbol{x}_n}^{(n)} \\ (\boldsymbol{c}_{\boldsymbol{x}_n}^{(n)})^t & \kappa(\boldsymbol{x}_n, \boldsymbol{x}_n) \end{bmatrix}. \tag{10}$$

The fact $d_n > \alpha \geq 0$ guarantees that the vectors in $\mathfrak{B}_n$ are linearly independent. In this way, the Gram matrix K_n is positive definite. It can be verified by simple algebraic manipulations that

$$K_n^{-1} = \begin{bmatrix} K_{n-1}^{-1} + \boldsymbol{\zeta}_{\boldsymbol{x}_n}^{(n)} (\boldsymbol{\zeta}_{\boldsymbol{x}_n}^{(n)})^t / d_n^2 & -\boldsymbol{\zeta}_{\boldsymbol{x}_n}^{(n)} / d_n^2 \\ -(\boldsymbol{\zeta}_{\boldsymbol{x}_n}^{(n)})^t / d_n^2 & 1/d_n^2 \end{bmatrix}. \tag{11}$$

Since $\mathfrak{B}_{n-1} \subsetneq \mathfrak{B}_n$, we immediately obtain that $M_{n-1} \subsetneq M_n$. Again the available information has to be transfered to the new subspace $M_n : \forall j \in \mathcal{J}_n \setminus \{n\}, k_{\boldsymbol{x}_j}^{(n)} := k_{\boldsymbol{x}_j}^{(n-1)}$. Since the cardinality of $\mathfrak{B}_n$ is larger than the cardinality of $\mathfrak{B}_{n-1}$ by one, then $\boldsymbol{\theta}_{\boldsymbol{x}_j}^{(n)} = [(\boldsymbol{\theta}_{\boldsymbol{x}_j}^{(n-1)})^t, 0]^t$, for any $j \in \mathcal{J}_n \setminus \{n\}$. The new vector $\kappa(\boldsymbol{x}_n,\cdot)$, being a basis vector itself, satisfies $\kappa(\boldsymbol{x}_n,\cdot) \in M_n$ so that $k_{\boldsymbol{x}_n}^{(n)} := \kappa(\boldsymbol{x}_n,\cdot)$. Hence, it has the following representation with respect to the new basis $\mathfrak{B}_n$: $\boldsymbol{\theta}_{\boldsymbol{x}_n}^{(n)} = [\boldsymbol{0}^t, 1]^t \in \mathbb{R}^{L_n}$.

We reproduce the formulas given in (7), but with

$$\tilde{\gamma}_l^{(n+1)} := \begin{cases} \tilde{\gamma}_l^{(n)} + \tilde{\mu}_n \sum_{j\in\mathcal{J}_n} \tilde{\beta}_j^{(n)} \theta_{\boldsymbol{x}_j,l}^{(n)}, & \forall l \in \overline{1, L_n - 1}, \\ \tilde{\mu}_n \tilde{\beta}_n^{(n)} \theta_{\boldsymbol{x}_n,L_n}^{(n)}, & l = L_n, \end{cases} \tag{12}$$

in order to obtain the estimate $\tilde{f}_{n+1}$ as in (9).

3.3.3. Approximate linear independency ($d_n > \alpha$) and buffer overflow ($L_{n-1} + 1 > L_b$). The sliding window effect.

Now, assume that $d_n > \alpha$ and that $L_{n-1} = L_b$. According to the above methodology, we still need to add $\kappa(\boldsymbol{x}_n,\cdot)$ to our new basis; define $\mathfrak{B}'_n := \{\psi_l^{(n)}\}_{l=1}^{L_b+1} := \mathfrak{B}_{n-1} \cup \{\kappa(\boldsymbol{x}_n,\cdot)\}$. Since we have inserted a new vector in our basis, the Gram matrix and its inverse are updated according to (10) and (11). However, note that the cardinality of the augmented $\mathfrak{B}'_n$ becomes $L_{n-1}+1 = L_b+1$, which exceeds our buffer's memory L_b.

As we did above, let $\forall j \in \mathcal{J}_n \setminus \{n\}$, $k_{\boldsymbol{x}_j}^{(n)} := k_{\boldsymbol{x}_j}^{(n-1)}$, and $k_{\boldsymbol{x}_n}^{(n)} := \kappa(\boldsymbol{x}_n,\cdot)$. Thus, $\boldsymbol{\theta}_{\boldsymbol{x}_j}^{(n)} = [(\boldsymbol{\theta}_{\boldsymbol{x}_j}^{(n-1)})^t, 0]^t$, for any $j \in \mathcal{J}_n \setminus \{n\}$, and $\boldsymbol{\theta}_{\boldsymbol{x}_n}^{(n)} = [\boldsymbol{0}^t, 1]^t$. Form the sum in (7c), and define

$$\tilde{\eta}_l^{(n+1)} := \begin{cases} \tilde{\gamma}_l^{(n)} + \tilde{\mu}_n \sum_{j\in\mathcal{J}_n} \tilde{\beta}_j^{(n)} \theta_{\boldsymbol{x}_j,l}^{(n)}, & \forall l \in \overline{1,L_b},\\ \tilde{\mu}_n \tilde{\beta}_n^{(n)} \theta_{\boldsymbol{x}_n,L_b+1}^{(n)}, & l = L_b+1, \end{cases}$$

Introduce then,

$$\tilde{f}'_{n+1} := \sum_{l=1}^{L_b+1} \tilde{\eta}_l^{(n+1)} \psi_l^{(n)}. \tag{13}$$

Since this expansion has L_b+1 terms, we have to discard one of them in order to comply with the memory limitations, i.e., with the length L_b of the buffer. Note that all the terms in (13) are linearly independent by definition. We decide to remove the term with the smallest contribution to the estimate $\tilde{f}'_{n+1}$. However, to prevent the removal of the term that corresponds to the currently received element $\kappa(\boldsymbol{x}_n,\cdot)$, we exclude the index L_b+1 from our search:

$$\mathcal{L}_* := \arg\min\{|\tilde{\eta}_l^{(n+1)}|\|\psi_l^{(n)}\| : l \in \overline{1,L_b}\}. \tag{14}$$

Notice that since every $\psi_l^{(n)}$ is some point $\kappa(\boldsymbol{x}_{n_l},\cdot)$, the norm $\|\psi_l^{(n)}\|$ above can be easily calculated as $\sqrt{\kappa(\boldsymbol{x}_{n_l},\boldsymbol{x}_{n_l})}$.

Among the indexes $\mathcal{L}_*$, we choose to discard the one that is located the furthest from the current time instant n:

$$l_* := \min\{l : l \in \mathcal{L}_*\}. \tag{15}$$

In such a way, we stay in line with the basic strategy of time-adaptive algorithms, where focus is put on data that describe the recent signal changes. Form then the estimate,

$$\tilde{f}_{n+1} := \tilde{f}'_{n+1} - \tilde{\eta}_{l_*}^{(n+1)} \psi_{l_*}^{(n)} = \sum_{m=1}^{L_b} \tilde{\eta}_{l_m}^{(n+1)} \psi_{l_m}^{(n)}, \tag{16}$$

where the index set $\{l_m\}_{m=1}^{L_b}$ was obtained by removing the index l_* from $\overline{1,L_b+1}$, i.e., $\{l_m\}_{m=1}^{L_b} := \overline{1,L_b+1} \setminus \{l_*\}$. In such a way, we can introduce the coefficients $\{\tilde{\gamma}_{l_m}^{(n+1)}\}_{m=1}^{L_b} := \{\tilde{\eta}_l^{(n+1)}\}_{l=1}^{L_b+1} \setminus \{\tilde{\eta}_{l_*}^{(n+1)}\}$, and the basis $\mathfrak{B}_n := \mathfrak{B}'_n \setminus \{\psi_{l_*}^{(n)}\}$. By a simple re-enumeration of these coefficients and of the new basis vectors, we obtain an estimate as in (9).

It remains to update the inverse of our Gram matrix. First, define the permutation,

$$\begin{aligned} &\pi(1,\ldots,l_*-1,l_*,l_*+1,\ldots,L_b+1) \\ &\qquad := (l_*,1,\ldots,l_*-1,l_*+1\ldots,L_b+1), \end{aligned} \tag{17}$$

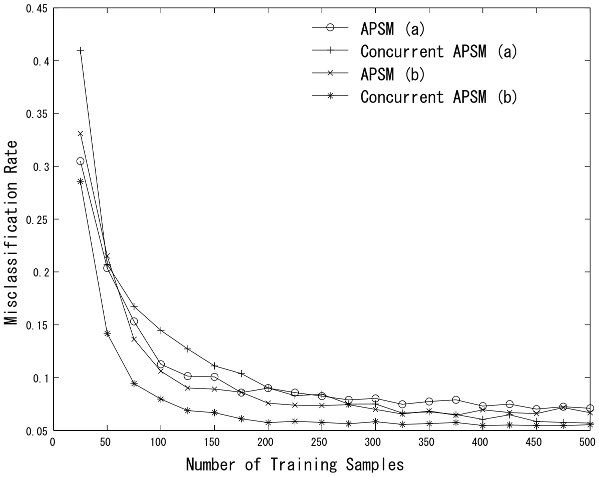

Fig. 2. Tracking performance for the channel when the LTI system is H_1. The variance of the Gaussian kernel takes the value of $\sigma^2 := 0.5$. The APSM(a) refers to (2a) with the closed ball sparsification methodology, i.e., [3], while APSM(b) refers to the present design. The buffer length $L_b := 500$, and $\alpha := 0.5$.

together with its corresponding permutation matrix P_π [11] whose (i,j)-th element is given by $\delta_{\pi(i),j}$, with δ being the Kronecker's delta. Then, let

$$\begin{bmatrix} r_n & \boldsymbol{h}_n^t \\ \boldsymbol{h}_n & H_n \end{bmatrix} := P_\pi^t K_n P_\pi,$$

and since P_π is orthogonal,

$$\begin{bmatrix} s_n & \boldsymbol{p}_n^t \\ \boldsymbol{p}_n & P_n \end{bmatrix} := (P_\pi^t K_n P_\pi)^{-1} = P_\pi^t K_n^{-1} P_\pi.$$

The K_n^{-1} was updated at the beginning of this section by (11), so that the above matrix can be easily computed. Since we remove the term that contributes the least in our estimate, we have to re-update $K_n := H_n$. It can be verified then by some algebra (proof is omitted) that $K_n^{-1} = H_n^{-1} = P_n - \frac{1}{s_n}\boldsymbol{p}_n\boldsymbol{p}_n^t$.

Note that the proposed algorithm shows *quadratic complexity*, with respect to the dimension L_n, due to the calculation of the orthogonal projection onto a subspace M_n in (6). Another quadratic complexity calculation, but with respect to the parameter q, is given in (7b). Since however, L_n is usually considered much larger than q, and since we have upper-bounded $L_n \leq L_b$, $\forall n \in \mathbb{Z}_{\geq 0}$, the overall complexity of our algorithm is bounded by $\mathcal{O}(L_b^2)$.

4. NUMERICAL RESULTS

The adaptive equalization problem for the nonlinear channel in [3] is chosen for validation. The sparsification scheme of Section 3 was applied also to the stochastic gradient descent methods of kernel perceptron and NORMA [9]. The method in [3] will be denoted by APSM(a), while the present one by APSM(b).

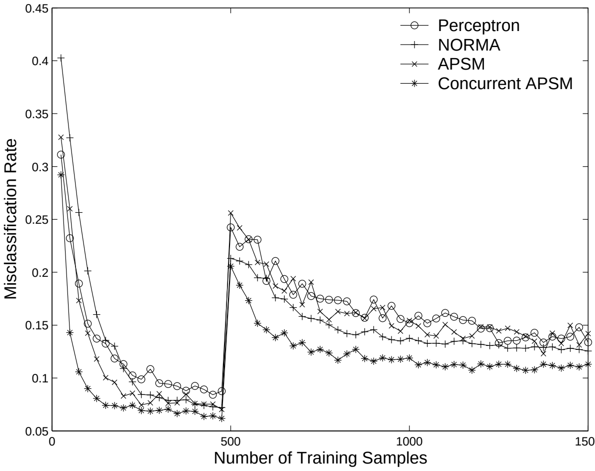

Fig. 3. A channel switch occurs at time $n = 500$, from H_1 to H_2, for the LTI system. The buffer length $L_b := 150$, with $\alpha := 0.9$. The variance of the Gaussian kernel function is $\sigma^2 := 0.5$.

A sequence of numbers taking values from $\{\pm 1\}$ with equal probability is fed into a Linear Time Invariant (LTI) channel producing the signal $(w_n)_n$. Two transfer functions for the LTI channel are available: $H_l(z) := \frac{\sin(\theta_l)}{\sqrt{2}}(1+z^{-2})+\cos(\theta_l)z^{-1}$, $\forall z \in \mathbb{C}$, $l = 1, 2$, where $\theta_1 := 29.5^\circ$ and $\theta_2 := -35^\circ$. In such a way, we can test our design under a sudden system change. We chose this example so that to study not only the convergence properties but also the tracking performance of the algorithm, and this is in line with the set of examples used in adaptive filtering. The transfer functions $H_l(z) := \sum_{i=0}^{2} h_{li} z^{-i}$, $z \in \mathbb{C}$, $l = 1, 2$, were chosen as above in order to simplify computations, since $\sum_{i=0}^{2} h_{li}^2 = 1$, $l = 1, 2$. The nonlinearity is given by $p_n := w_n + 0.2w_n^2 - 0.1w_n^3$, $\forall n$. Gaussian i.i.d. noise with zero mean and SNR $=$ 10dB with respect to $(p_n)_n$, is added to give the received signal $(x_n)_n$. As in [3], the data space is the Euclidean $\mathbb{R}^4$. In order to work in an infinite dimensional RKHS, the Gaussian (RBF) kernel was used (cf. Section 1). A number of 100 test data were used for validation. We performed 100 realizations and uniformly averaged the results.

In Fig. 2, we compare the methods APSM(a) and APSM(b). The parameters were chosen such that corresponding curves produce the same misclassification rate level. For both realizations, the concurrent APSM used a $q = 16$ for the index set $\mathcal{J}_n$, $n \in \mathbb{Z}_{\geq 0}$. The variance of the Gaussian kernel is set to $\sigma^2 := 0.5$, the radius of the closed ball in [3] to $\delta := 2$, the parameter $\alpha := 0.5$, and the buffer length $L_b := 500$. The buffer associated with the sparsification method APSM(a) was set to 500. Also the extrapolation parameters $\mu_n = \tilde{\mu}_n = 1$, $\forall n$, for all the APSM versions. We notice that the concurrent APSM(b) converges faster than the APSM(a). Moreover, we do not notice such big differences between the non-concurrent versions of the APSMs for both types of sparsification.

In Fig. 3, $\alpha := 0.9$, $L_b := 150$, $q := 16$, and $\mu_n = \tilde{\mu}_n = 1$, $\forall n$, for all the proposed here APSM versions. We observe in this figure that the non-concurrent version of the proposed method performs worse than NORMA. However, it is clear that concurrent processing remains by far the most robust approach since it achieves fast convergence as well as low misclassification rate level; something also observed for the sparsification scheme in [3].

5. CONCLUSIONS

A novel sparsification scheme was introduced for a very recently developed projection-based online classification task in Reproducing Kernel Hilbert Spaces. The algorithm is developed by using projection mappings onto special convex sets, namely closed halfspaces and subspaces. Sparsification is achieved via a sequence of finite dimensional subspaces, with dimensions upper bounded by a buffer length. In the case of a buffer overflow, we remove the term that contributes the least in the resulting kernel series expansion. When applied to the adaptive equalization problem of a communication channel, the concurrent proposed scheme exhibited the best performance when compared to its non-concurrent version, as well as to classical and very recently introduced stochastic gradient descent techniques.

6. REFERENCES

[1] S. Theodoridis and K. Koutroumbas, *Pattern Recognition*, Academic Press, Amsterdam, 3rd edition, 2006.

[2] K. Slavakis, S. Theodoridis, and I. Yamada, "Online kernel-based classification by projections," in *Proceedings of ICASSP*, Honolulu, Hawaii, April 2007, IEEE, vol. II, pp. 425–428.

[3] K. Slavakis, S. Theodoridis, and I. Yamada, "Online sparse kernel-based classification by projections," in *Proc. IEEE Machine Learning for Signal Processing (MLSP)*, Thessaloniki: Greece, Aug. 2007, pp. 294–299.

[4] I. Yamada and N. Ogura, "Adaptive Projected Subgradient Method for asymptotic minimization of sequence of nonnegative convex functions," *Numerical Functional Analysis and Optimization*, vol. 25, no. 7&8, pp. 593–617, 2004.

[5] K. Slavakis, I. Yamada, and N. Ogura, "The Adaptive Projected Subgradient Method over the fixed point set of strongly attracting nonexpansive mappings," *Numerical Functional Analysis and Optimization*, vol. 27, no. 7&8, pp. 905–930, 2006.

[6] T. R. Rockafellar, *Convex Analysis*, Princeton University Press, Princeton, NJ, 1970.

[7] D. G. Luenberger, *Optimization by Vector Space Methods*, John Wiley & Sons, New York, 1969.

[8] A. H. Sayed, *Fundamentals of Adaptive Filtering*, John Wiley & Sons, New Jersey, 2003.

[9] J. Kivinen, A. J. Smola, and R. C. Williamson, "Online learning with kernels," *IEEE Trans. Signal Processing*, vol. 52, no. 8, pp. 2165–2176, Aug. 2004.

[10] Y. Engel, S. Mannor, and R. Meir, "The kernel recursive least-squares algorithm," *IEEE Trans. Signal Processing*, vol. 52, no. 8, pp. 2275–2285, Aug. 2004.

[11] A. Ben-Israel and T. N. E. Greville, *Generalized Inverses: Theory and Applications*, Springer-Verlag, New York, 2nd edition, 2003.