EUSIPCO 2013 1569743433

ONLINE MODEL SELECTION AND LEARNING BY MULTIKERNEL ADAPTIVE FILTERING

*Masahiro Yukawa**

Dept. Electronics and Electrical Engineering
Keio University, JAPAN

Ryu-ichiro Ishii

Dept. Electrical and Electronic Engineering
Niigata University, JAPAN

ABSTRACT

We propose an efficient multikernel adaptive filtering algorithm with double regularizers, providing a novel pathway towards online model selection and learning. The task is the challenging nonlinear adaptive filtering under no knowledge about a suitable kernel. Under this limited-knowledge assumption on an underlying model of a system of interest, many possible kernels are employed and one of the regularizers, *a block ℓ_1 norm for kernel groups*, contributes to selecting a proper model (relevant kernels) in online and adaptive fashion, preventing a nonlinear filter from overfitting to noisy data. The other regularizer is the block ℓ_1 norm for data groups, contributing to updating the dictionary adaptively. As the resulting cost function contains two nonsmooth (but *proximable*) terms, we approximate the latter regularizer by its Moreau envelope and apply the adaptive proximal forward-backward splitting method to the approximated cost function. Numerical examples show the efficacy of the proposed algorithm.

***Index Terms*—** kernel adaptive filter, proximity operator, multiple kernels

1. INTRODUCTION

We address the challenging task of online model selection and learning using multikernel adaptive filtering [1]. The challenge is that model selection, as well as estimation of the unknown nonlinear system, needs to be made online and also adaptively. The key assumption is that no adequate kernel is available unlike the prior works on kernel adaptive filtering [2–5]. Under this assumption, the existing kernel adaptive filtering algorithms are not expected to work well.

The multikernel adaptive filtering technique models an *estimandum* (a system to be estimated) as a function in the sum of multiple *reproducing kernel Hilbert spaces (RKHSs)* associated with multiple positive definite kernels [6,7]. Therefore, it has higher degrees of freedom compared to the multiple kernel learning (MKL) approaches [8,9] which model the *estimandum* as a function in a *single* RKHS associated with the best kernel. See [1] for more details about the relation-to/differences-from the MKL approaches. One may apply the multikernel adaptive filtering technique with many possible kernels such as Gaussian kernels with a wide range of kernel parameters. This approach however carries a significant risk of overfitting to noisy data due to the high degrees of freedom together with the use of narrow Gaussian kernels. The overfitting issue becomes more serious as the noise magnitude becomes larger (as will be shown in Section 4), resulting in failure to estimate the unknown system accurately.

In this paper, we present an efficient multikernel adaptive filtering algorithm with double regularizers for online model selection and learning. One of the regularizers is the block ℓ_1 norm for kernel groups, which contributes to nulling the coefficients of such kernels that are unsuitable for the learning task. A proper model is thus selected, alleviating the overfitting problem. The other regularizer is the block ℓ_1 norm for data groups, which contributes to nulling the coefficients of such dictionary data that are less relevant to the learning task than the others. The dictionary data are thus updated in an adaptive manner. The time-dependent cost function then becomes a sum of a smooth convex function (a data fidelity term) and a pair of nonsmooth (but *proximable*) convex functions. We approximate the data-selective regularizer by its *Moreau envelope*, and this approximation makes the cost function into a sum of smooth functions and a single nonsmooth function to which the adaptive proximal forward-backward splitting method [10] can be applied. The dictionary is constructed as follows: a new datum is selectively added into the dictionary based on the *coherence criterion* [3], which reduces the risks of overfitting impulsive noise, and the dictionary data with minor contributions are discarded by the combination of soft-shrinkage and hard-thresholding. Numerical examples show that the proposed algorithm selects a proper model online, thereby alleviating the overfitting problem and leading to better estimation performance than the existing multikernel adaptive filtering algorithms, and also that it adapts to an abrupt change of nonlinear systems.

2. MULTIKERNEL ADAPTIVE FILTERING

Throughout the paper, let $\mathbb{R}$, $\mathbb{N}$, and $\mathbb{N}^*$ denote the sets of all real numbers, nonnegative integers, and positive integers, respectively. Define an inner product between two matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ by $\langle \boldsymbol{A}, \boldsymbol{B} \rangle := \mathrm{tr}(\boldsymbol{A}^\mathsf{T}\boldsymbol{B})$, where $(\cdot)^\mathsf{T}$ and $\mathrm{tr}(\cdot)$ stand for *transpose* and *trace*, respectively. Its induced norm is defined as $\|\boldsymbol{A}\| := \sqrt{\langle \boldsymbol{A}, \boldsymbol{A} \rangle}$ for any matrix $\boldsymbol{A}$. Note that, in the particular case that $\boldsymbol{A}$ is a vector, these are reduced to the standard inner product and the Euclidean norm, respectively.

Let $\mathcal{U}$ denote the input space which is a compact (i.e., bounded and closed) subset of the L dimensional Euclidean space $\mathbb{R}^L$. (L is the only knowledge about the estimandum that is assumed known *a priori* excluding the algorithm parameters.) We consider online scenarios in which input vectors $(\boldsymbol{u}_n)_{n\in\mathbb{N}} \subset \mathcal{U}$ arrive sequentially and the response $d_n \in \mathbb{R}$, $n \in \mathbb{N}$, to each $\boldsymbol{u}_n$ is a nonlinear function of $\boldsymbol{u}_n$. The task of nonlinear adaptive filtering is to find and/or track the time-variable nonlinear function (the *estimandum*) in an online fashion with the sequentially arriving measurements $(\boldsymbol{u}_n, d_n)_{n\in\mathbb{N}}$.

*This work was supported by KDDI Foundation. Masahiro Yukawa thanks Dr. Sohan Seth of Aalto University for pointing out the article [9] during his short visit in Finland in summer 2012, which touched off the present study. He also thanks Shunsuke Ono of Tokyo Institute of Technology for pointing out the articles [15–17] which allowed to clarify the motivation for introducing the Moreau envelope approximation.

We consider the case that a proper model for the *estimandum* is unknown. A practical approach in this case is to use many possible kernels under the multikernel adaptive filtering framework [1]. Let $\kappa_m : \mathcal{U} \times \mathcal{U} \to \mathbb{R}$, $m \in \mathcal{M} := \{1, 2, \cdots, M\}$, denote the set of positive definite kernels to be used. Let $\{\kappa_m(\cdot, \boldsymbol{u}_j)\}_{m\in\mathcal{M}, j\in\mathcal{J}_n}$ be the dictionary indicated by the dictionary index set $\mathcal{J}_n := \{j_1^{(n)}, j_2^{(n)}, \cdots, j_{r_n}^{(n)}\} \subset \{0, 1, \cdots, n-1\}$, where $r_n \in \mathbb{N}^*$ is the size of the dictionary index set $\mathcal{J}_n$. A multikernel adaptive filter is then given by

$$\phi_n(\boldsymbol{u}) := \sum_{m\in\mathcal{M}} \underbrace{\sum_{j\in\mathcal{J}_n} h_{j,n}^{(m)} \kappa_m(\boldsymbol{u}, \boldsymbol{u}_j)}_{\text{the } m\text{th model}}, \ \boldsymbol{u} \in \mathcal{U} \tag{1}$$

where $h_{j,n}^{(m)} \in \mathbb{R}$, $m \in \mathcal{M}$, $j \in \mathcal{J}_n$. An estimate of d_n is given by

$$\widehat{d}_n := \phi_n(\boldsymbol{u}_n) = \langle \boldsymbol{H}_n, \boldsymbol{K}_n \rangle \tag{2}$$

where the (m, i) entries of the matrices $\boldsymbol{H}_n \in \mathbb{R}^{M\times r_n}$ and $\boldsymbol{K}_n \in \mathbb{R}^{M\times r_n}$ are given by $[\boldsymbol{H}_n]_{m,i} := h_{j_i^{(n)},n}^{(m)}$ and $[\boldsymbol{K}_n]_{m,i} := \kappa_m\left(\boldsymbol{u}_n, \boldsymbol{u}_{j_i^{(n)}}\right)$, respectively.

3. ONLINE MODEL SELECTION AND LEARNING SCHEME

This section presents the proposed scheme for online model selection and learning. The scheme selects a proper model by making many coefficients $h_{j,n}^{(m)}$ for all $j \in \mathcal{J}_n$ and for some $m \in \mathcal{M}$ associated with those kernels which are irrelevant to the nonlinear system. Thus, improper models (irrelevant kernels) are automatically excluded from the expansion in (1), and this avoids the overfitting problems.

3.1. Cost Function with Double Regularizers

The size and associated data indices of the coefficient matrix $\boldsymbol{H}_n \in \mathbb{R}^{M\times r_n}$ depend on the dictionary index set $\mathcal{J}_n$ and are therefore time dependent. The cost function to be considered is thus a function of a matrix in $\mathbb{R}^{M\times r_{n+1}}$ (not in $\mathbb{R}^{M\times r_n}$). We start by considering the following cost function:

$$\Theta_n(\boldsymbol{X}) := \varphi_n(\boldsymbol{X}) + \psi_n^{(1)}(\boldsymbol{X}) + \psi_n^{(2)}(\boldsymbol{X}), \quad n \in \mathbb{N}, \tag{3}$$

for $\boldsymbol{X} := \left[\boldsymbol{x}_1\ \boldsymbol{x}_2 \cdots \boldsymbol{x}_{r_{n+1}}\right] := \left[\boldsymbol{\xi}_1\ \boldsymbol{\xi}_2 \cdots \boldsymbol{\xi}_M\right]^{\mathsf{T}} \in \mathbb{R}^{M\times r_{n+1}}$, where

$$\varphi_n(\boldsymbol{X}) := \frac{1}{2} d^2(\boldsymbol{X}, C_n) \text{ (data fidelity term)}$$

$$\psi_n^{(1)}(\boldsymbol{X}) := \lambda_1 \sum_{i=1}^{r_{n+1}} w_{i,n} \|\boldsymbol{x}_i\| \text{ (block } \ell_1 \text{ for data groups)}$$

$$\psi_n^{(2)}(\boldsymbol{X}) := \lambda_2 \sum_{m=1}^{M} \nu_{m,n} \|\boldsymbol{\xi}_m\| \text{ (block } \ell_1 \text{ for kernel groups).}$$

Here, $\lambda_1, \lambda_2 \geq 0$ are the regularization parameters, $w_{i,n}, \nu_{m,n} > 0$ are the weights, and $d(\boldsymbol{X}, C_n) := \min_{\boldsymbol{Y}\in C_n} \|\boldsymbol{X} - \boldsymbol{Y}\|$ is the metric distance between a point $\boldsymbol{X} \in \mathbb{R}^{M\times r_{n+1}}$ and the set

$$C_n := \left\{\boldsymbol{X} \in \mathbb{R}^{M\times r_{n+1}} : |\varepsilon_n(\boldsymbol{X})| \leq \rho\right\}. \tag{4}$$

The set C_n consists of the parameter matrices which provide the magnitude of the instantaneous estimation error $\varepsilon_n(\boldsymbol{X}) := \left\langle \boldsymbol{X}, \widetilde{\boldsymbol{K}}_n \right\rangle - d_n$ bounded by $\rho \geq 0$. Here, $\widetilde{\boldsymbol{K}}_n \in \mathbb{R}^{M\times r_{n+1}}$ consists of a submatrix of $\boldsymbol{K}_n$ and possibly a new vector $\boldsymbol{k}_{n,n}$ at the rightmost column according to the update of the dictionary index set $\mathcal{J}_n$ into the new one $\mathcal{J}_{n+1}$. (The formal definition of $\widetilde{\boldsymbol{K}}_n$ is given in Section 3.2.) The role of each term is given as follows: (i) $\varphi_n(\boldsymbol{X})$ is for reducing empirical risks (estimation errors for observed data), and (ii) $\psi_n^{(1)}$ and $\psi_n^{(2)}$ are regularizers for reducing generalization errors. Indeed, $\psi_n^{(1)}$ promotes *column-wise sparsity*, whereas $\psi_n^{(2)}$ promotes *row-wise sparsity*; i.e., those regularizers contribute to selecting data and kernels which are most relevant to the estimation. In the present study, the row-wise sparsity is of particular importance because it is supposed that some of the kernels are not relevant to the estimation and the parameters associated with such irrelevant kernels should be zero to prevent the nonlinear filter from overfitting noisy data.

Now the question is how to suppress the cost function Θ_n which is time-varying. The point is that φ_n is a differentiable convex function having a Lipschitz continuous gradient while the regularizers $\psi_n^{(1)}$ and $\psi_n^{(2)}$ are nondifferentiable but *convex and proximable*. Here, *proximable* means that the proximity operator can be computed easily. Our approach is to apply the adaptive proximal forward-backward splitting algorithm [10] to the following approximate cost function:

$$\widetilde{\Theta}_n(\boldsymbol{X}) := \underbrace{\varphi_n(\boldsymbol{X}) + {}^{\gamma}\psi_n^{(1)}(\boldsymbol{X})}_{\text{smooth}} + \underbrace{\psi_n^{(2)}(\boldsymbol{X})}_{\text{proximable}}$$

where ${}^{\gamma}\psi_n^{(1)}(\boldsymbol{X})$ denotes the Moreau envelope of $\psi_n^{(1)}(\boldsymbol{X})$ of index $\gamma \in (0, \infty)$ defined as

$${}^{\gamma}\psi_n^{(1)}(\boldsymbol{X}) := \min_{\boldsymbol{Y}\in\mathbb{R}^{M\times r_{n+1}}} \psi_n^{(1)}(\boldsymbol{Y}) + \frac{1}{2\gamma} \|\boldsymbol{X} - \boldsymbol{Y}\|^2. \tag{5}$$

See [11, 12] for details about proximity operators and Moreau envelopes.

3.2. Adaptive Algorithm

We define the modified matrices $\widetilde{\boldsymbol{H}}_n \in \mathbb{R}^{M\times r_{n+1}}$ and $\widetilde{\boldsymbol{K}}_n \in \mathbb{R}^{M\times r_{n+1}}$ with their (m, i) entries given by $[\widetilde{\boldsymbol{H}}_n]_{m,i} := h_{j_i^{(n+1)},n}^{(m)}$ and $[\widetilde{\boldsymbol{K}}_n]_{m,i} := \kappa_m\left(\boldsymbol{u}_n, \boldsymbol{u}_{j_i^{(n+1)}}\right)$, respectively. The modified matrix $\widetilde{\boldsymbol{H}}_n$ consists of a submatrix of $\boldsymbol{H}_n$ eliminating some columns with minor contributions and possibly a new entry $\boldsymbol{h}_{n,n} := \boldsymbol{0}$ at the rightmost column if $n \in \mathcal{J}_{n+1}$.

The dictionary is initialized as $\mathcal{J}_0 := \{0\}$. Let $\widetilde{\boldsymbol{H}}_0 := \boldsymbol{h}_{0,0} = \boldsymbol{0}$. The proposed algorithm is then given by

$$\boldsymbol{H}_{n+1} := \operatorname{prox}_{\eta\psi_n^{(2)}}\left[\widetilde{\boldsymbol{H}}_n - \eta\left(\boldsymbol{\nabla}\varphi_n(\widetilde{\boldsymbol{H}}_n) + \boldsymbol{\nabla}{}^{\gamma}\psi_n^{(1)}(\widetilde{\boldsymbol{H}}_n)\right)\right], \quad n \in \mathbb{N}, \tag{6}$$

where $\eta \in (0, 2/\alpha)$ is the step size. Here, $\alpha := 1 + \frac{1}{\gamma}$ is the Lipschitz constant[1] of the mapping $T : \mathbb{R}^{M\times r_{n+1}} \to \mathbb{R}^{M\times r_{n+1}}$, $\boldsymbol{X} \mapsto \boldsymbol{\nabla}\varphi_n(\boldsymbol{X}) + \boldsymbol{\nabla}^\gamma\psi_n^{(1)}(\boldsymbol{X})$. The gradients $\boldsymbol{\nabla}\varphi_n(\widetilde{\boldsymbol{H}}_n)$ and $\boldsymbol{\nabla}^\gamma\psi_n^{(1)}(\widetilde{\boldsymbol{H}})$ in (6) are given respectively by

$$\boldsymbol{\nabla}\varphi_n(\widetilde{\boldsymbol{H}}_n) = \widetilde{\boldsymbol{H}}_n - P_{C_n}(\widetilde{\boldsymbol{H}}_n) \tag{7}$$

$$\boldsymbol{\nabla}^\gamma\psi_n^{(1)}(\widetilde{\boldsymbol{H}}_n) = \frac{\widetilde{\boldsymbol{H}}_{\boldsymbol{n}} - \mathrm{prox}_{\gamma\psi_n^{(1)}}(\widetilde{\boldsymbol{H}}_n)}{\gamma}. \tag{8}$$

Here, $P_{C_n}(\widetilde{\boldsymbol{H}}_n)$ denotes the projection[2] of $\widetilde{\boldsymbol{H}}_n$ onto the hyperslab C_n defined in (4) and has a closed-form expression:

$$P_{C_n}(\widetilde{\boldsymbol{H}}_n) = \widetilde{\boldsymbol{H}}_n - \mathrm{sign}(\varepsilon_n(\widetilde{\boldsymbol{H}}_n))\frac{\max\{|\varepsilon_n(\widetilde{\boldsymbol{H}}_n)| - \rho, 0\}}{\|\widetilde{\boldsymbol{K}}_n\|^2}\widetilde{\boldsymbol{K}}_n.$$

Finally, $\mathrm{prox}_{\gamma\psi_n^{(1)}}$ in (8) and $\mathrm{prox}_{\gamma\psi_n^{(2)}}$ in (6) are the proximity operators of $\psi_n^{(1)}$ and $\psi_n^{(2)}$, respectively, of the index γ, and are given respectively by

$$\begin{aligned}
\mathrm{prox}_{\gamma\psi_n^{(1)}}(\boldsymbol{X}) &:= \operatorname*{argmin}_{\boldsymbol{Y}\in\mathbb{R}^{M\times r_{n+1}}} \psi_n^{(1)}(\boldsymbol{Y}) + \frac{1}{2\gamma}\|\boldsymbol{X} - \boldsymbol{Y}\|^2 \\
&= \sum_{i=1}^{r_{n+1}} \max\left\{1 - \frac{\lambda_1\gamma w_{i,n}}{\|\boldsymbol{x}_i\|}, 0\right\} \boldsymbol{x}_i \boldsymbol{e}_{i,r_{n+1}}^\mathsf{T}, \\
\mathrm{prox}_{\eta\psi_n^{(2)}}(\boldsymbol{X}) &= \sum_{m=1}^{M} \max\left\{1 - \frac{\lambda_2\eta\nu_{m,n}}{\|\boldsymbol{\xi}_m\|}, 0\right\} \boldsymbol{e}_{m,M}\boldsymbol{\xi}_m^\mathsf{T}, \\
\boldsymbol{X} &:= [\boldsymbol{x}_1\ \boldsymbol{x}_2\ \cdots\ \boldsymbol{x}_{r_{n+1}}] := [\boldsymbol{\xi}_1\ \boldsymbol{\xi}_2 \cdots \boldsymbol{\xi}_M]^\mathsf{T} \in \mathbb{R}^{M\times r_{n+1}}.
\end{aligned}$$

Here, $\boldsymbol{e}_{p,q}$, $p, q \in \mathbb{N}^*$, is a length-q unit vector that has one at the pth entry and zeros elsewhere. The proximity operator $\mathrm{prox}_{\gamma\psi_n^{(1)}}$ plays a role in selecting some data groups (some column vectors of $\widetilde{\boldsymbol{H}}_n$), while $\mathrm{prox}_{\eta\psi_n^{(2)}}$ selects some kernel groups (some row vectors of $\widetilde{\boldsymbol{H}}_n$). The operators are specifically referred to as *block soft-thresholding*.

3.3. Sparsification

Our sparsification is based on the following basic ideas: (i) add a new datum into the dictionary only if it is sufficiently novel, and (ii) discard those data which are irrelevant to estimation. To be precise, the dictionary index set is updated as follows:

$$\mathcal{J}_{n+1} := \begin{cases} \mathcal{J}_{\geq\epsilon}^n \cup \{n\}, & \text{if } c(\boldsymbol{u}_n, \mathcal{J}_n) \leq \delta, \\ \mathcal{J}_{\geq\epsilon}^n, & \text{otherwise,} \end{cases} \qquad n \in \mathbb{N}, \tag{9}$$

where $\delta \in (0, 1]$, $\epsilon \geq 0$, and

$$\mathcal{J}_{\geq\epsilon}^n := \{j \in \mathcal{J}_n : \|\boldsymbol{h}_{j,n}\| \geq \epsilon\},$$

$$c(\boldsymbol{u}_n, \mathcal{J}_n) := \max_{j\in\mathcal{J}_n,\, m\in\mathcal{M}} \frac{|\kappa_m(\boldsymbol{u}_n, \boldsymbol{u}_j)|}{\sqrt{\kappa_m(\boldsymbol{u}_n, \boldsymbol{u}_n)\kappa_m(\boldsymbol{u}_j, \boldsymbol{u}_j)}} \in [0, 1].$$

If $\kappa_m(\boldsymbol{u}_n, \boldsymbol{u}_n) = 0$ for some $m \in \mathcal{M}$, we define $c(\boldsymbol{u}_n, \mathcal{J}_n) := 1$. The coherence [3] $c(\boldsymbol{u}_n, \mathcal{J}_n)$ is used as a novelty criterion for simplicity. The proximity operators $\mathrm{prox}_{\gamma\psi_n^{(1)}}$ and $\mathrm{prox}_{\gamma\psi_n^{(2)}}$ shrink those column and row vectors of $\widetilde{\boldsymbol{H}}_n$ which have minor contributions in estimation. While such row vectors (corresponding to kernel groups) tend to become exactly zero, such column vectors (corresponding to data groups) tend *not* to become zero due to the use of the Moreau envelope ${}^\gamma\psi_n^{(1)}$ rather than the $\psi_n^{(1)}$ itself. Note here that $\eta\boldsymbol{\nabla}^\gamma\psi_n^{(1)}(\widetilde{\boldsymbol{H}}_n) = \frac{\eta}{\gamma}\left(\widetilde{\boldsymbol{H}}_n - \mathrm{prox}_{\gamma\psi_n^{(1)}}(\widetilde{\boldsymbol{H}}_n)\right)$ and $\frac{\eta}{\gamma} < \frac{2}{\alpha\gamma} < 2$.

3.4. Remarks

Relation to MKNLMS-CS and MKNLMS-BT: If we let $\lambda_1 = \lambda_2 = \epsilon = 0$, the proposed algorithm is reduced to the MKNLMS-CS (multikernel normalized least mean square with coherence-based sparsification) algorithm [1]; i.e., the proposed algorithm is a generalization of MKNLMS-CS. On the other hand, if we let $\lambda_1 > 0$, $\lambda_2 = 0$, $\delta = 1$, and choose the ϵ value appropriately, the proposed algorithm would behave similarly to the MKNLMS-BT (multikernel normalized least mean square with block soft-thresholding) algorithm [1], but it is not a generalization of MKNLMS-BT. It might be possible to switch the roles of $\psi_n^{(1)}$ and $\psi_n^{(2)}$ with an appropriate modification. This approach however did not work well in our experiments, because those row vectors with minor contributions remain and yield extra errors.

Computational complexity: The number of multiplications required for each update in the proposed algorithm is approximately $(L + 4M + 3\hat{M}_n)r_n$, where $\hat{M}_n$ denotes the number of 'active' kernels at the nth iteration, whereas those for MKNLMS-CS and MKNLMS-BT are $(L+3M)r_n$ and $(L+5M)r_n$, respectively. Here, we say that a kernel is active if its associated row in $\widetilde{\boldsymbol{H}}_n$ is a nonzero vector. The number of exponential calculations and memory requirements are the same as MKNLMS-CS and MKNLMS-BT and are given by Mr_n and $(L + M)r_n$, respectively. A remarkable feature of the proposed algorithm is that the number of active kernels is much less than the number M of the kernels employed, as will be seen in Section 4. This is advantageous because a proper model is naturally obtained as a consequence of adaptation. Also this can be used to reduce the complexity and memory usages.

On the parameter selection: The regularization parameter λ_1 for data groups can be set to zero in stationary environments because there is no particular need to discard data from the dictionary, provided that the coherence threshold δ is chosen adequately. Therefore, a practical strategy is to set it to zero at the beginning and activate it when the dictionary size exceeds some prespecified value. The λ_2 for kernel groups is of great importance to alleviate the noise sensitivity, and therefore its appropriate value depends on noise characteristics. To cope with large noise, the λ_2 should take a large value. The weights $w_{i,n}$ and $\nu_{m,n}$ of the block ℓ_1 norms should be designed in such a way that those column and row vectors with relatively small norms diminish swiftly and, at the same time, that those column and row vectors with large norms shrink gradually to avoid extra bias in estimation. In the numerical examples, the weights are set to some small constants $0 < \epsilon_w, \epsilon_\nu \ll 1$, if each associated column or row vector of $\widetilde{\boldsymbol{H}}_n$ has its norm greater than the thirty percent of the largest norm among the column or row vectors, and set to one other-

[1] A mapping $T : \mathbb{R}^{M\times r_{n+1}} \to \mathbb{R}^{M\times r_{n+1}}$ is said to be Lipschitz continuous if $\|T(\boldsymbol{X}) - T(\boldsymbol{Y})\| \leq \alpha\|\boldsymbol{X} - \boldsymbol{Y}\|, \forall \boldsymbol{X}, \boldsymbol{Y} \in \mathbb{R}^{M\times r_{n+1}}$, for some constant $\alpha > 0$, and the minimum α is called the Lipschitz constant.

[2] Given any closed convex subset $C \subset \mathbb{R}^{M\times r_{n+1}}$, the closest point of an $\boldsymbol{X} \in \mathbb{R}^{M\times r_{n+1}}$ in C is called the metric projection of $\boldsymbol{X}$ onto C and is denoted by $P_C(\boldsymbol{X}) := \mathrm{argmin}_{\boldsymbol{Y}\in C}\|\boldsymbol{X} - \boldsymbol{Y}\|$.

wise (see [1] for more discussion). The error bound ρ in (4) is determined based on noise statistics; typical values, under the assumption that $n_k \sim \mathcal{N}(0, \sigma^2)$, include (i) the mean value plus standard deviation $\rho_1 := (1+\sqrt{2})\sigma^2$, (ii) the mean value plus standard deviation $\rho_2 := \sigma^2$, and (iii) the peak value $\rho_3 := 0$ of the random variable $v_k := n_k^2$ [13]. The index γ of the Moreau envelope in (5) governs the range of the step size η as well as the accuracy of the approximation (i.e., the gap between $\psi_n^{(1)}$ and ${}^{\gamma}\psi_n^{(1)}$). The smaller the γ is, the better the approximation but the smaller the upper bound of η, causing slow convergence. Our recommendation is to set the step size η to some value around 0.1, as in the case of the normalized least mean square (NLMS) algorithm for linear adaptive filters, and let $\gamma = ((2/(\eta + \epsilon_\gamma) - 1)^{-1} > 0$ for some small constant $\epsilon_\gamma \in (0, 2-\eta)$ to ensure $\eta = 2/\alpha - \epsilon_\gamma \in (0, 2/\alpha)$.

4. NUMERICAL EXAMPLES

We conduct simulations in an estimation task of nonlinear function with an abrupt change for $L = 2$ to show that the proposed algorithm (i) selects a proper model online, (ii) alleviates the overfitting issue, (iii) adapts to the abrupt change of nonlinear functions, and (iv) achieves better estimation performance than the existing multikernel adaptive filtering algorithms (MKNLMS-BT and MKNLMS-CS) [1]. We test 300 independent trials and, at each trial $t = 1, 2, \cdots, 300$, the data is generated as $d_n^{(t)} := \psi_n(\boldsymbol{u}_n^{(t)}) + v_n^{(t)}, n \in \mathbb{N}$, with $\psi_n(\boldsymbol{x}) := \exp\left(-\|\boldsymbol{x} - \boldsymbol{c}_1\|^2\right) + 1.2\exp\left(-5\|\boldsymbol{x} - \boldsymbol{c}_2\|^2\right)$ for $n \leq 2000$, where $\boldsymbol{c}_1 := [0.2, 0.2]^\mathsf{T}$ and $\boldsymbol{c}_2 := [0.7, 0.7]^\mathsf{T}$, and $\psi_n(\boldsymbol{x}) := 1.3\exp\left(-10\|\boldsymbol{x} - \tilde{\boldsymbol{c}}_1\|^2\right) + 1.5\exp\left(-2\|\boldsymbol{x} - \tilde{\boldsymbol{c}}_2\|^2\right)$ for $n > 2000$, where $\tilde{\boldsymbol{c}}_1 := [0.1, 0.9]^\mathsf{T}$ and $\tilde{\boldsymbol{c}}_2 := [0.9, 0.1]^\mathsf{T}$. Here, each component of the input vector $\boldsymbol{u}_n^{(t)}$ obeys the i.i.d. uniform distribution between 0 and 1. It is supposed that the data are contaminated by impulsive noise, 12 times between iterations 100 and 135, of amplitude 50, and by Gaussian noise obeying $\mathcal{N}(0, 0.1)$ at the other iterations. In addition to the mean squared error (MSE), we also evaluate the peak squared error (PSE) defined as $\mathrm{PSE}(n) := \max_{t=1,2,\cdots,300}(\phi_n^{(t)}(u_n^{(t)}) - d_n^{(t)})^2$ for measuring how the impact of impulsive noise remains on nonlinear filters during the adaptation. Totally $M = 63$ Gaussian kernels are employed with the kernel parameters $a \times 10^b, a \in \{1, 2, \cdots, 9\}$, $b \in \{-1, 0, 1, 2, 3, 4, 5\}$. To be precise, $\kappa_m(\boldsymbol{x}, \boldsymbol{y}) := \exp\left(-\zeta_m \|\boldsymbol{x} - \boldsymbol{y}\|^2\right), \boldsymbol{x}, \boldsymbol{y} \in \mathcal{U}, m \in \mathcal{M}$, where $\zeta_1 = 0.1$, $\zeta_2 = 0.2, \cdots, \zeta_{63} = 9 \times 10^5$.

The parameters for the proposed algorithm are set to $\lambda_1 = 2.0 \times 10^{-3}$, $\lambda_2 = 5.0 \times 10^{-4}$, $\epsilon_w = 1.0 \times 10^{-6}$, $\epsilon_\nu = 1.0 \times 10^{-6}$, $\rho = 0$, $\eta = 0.2$, $\epsilon_\gamma = 1.0 \times 10^{-5}$, $\delta = 0.9995$, $\epsilon = 5.0 \times 10^{-5}$. The parameters for MKNLMS-CS are set to $\eta = 0.2, \delta = 0.9995, \rho = 0$. The parameters for MKNLMS-BT are set to $\mu = 0.2$, $\lambda = 1.0 \times 10^{-2}$, $\epsilon_w = 1.0 \times 10^{-6}$. The learning curves in PSE and MSE are shown in Fig. 1. Table 1 presents (i) PSE and MSE averaged over the last 1000 iterations and (ii) the number of active kernels and the dictionary size $\bar{r}_n$ averaged over the 5000 iterations. It is seen that the proposed algorithm outperforms the other algorithms both in PSE and MSE and exploits slightly less than a half of the 63 kernels actively on average, while the dictionary sizes of all algorithms are adjusted to be the same approximately.

Table 1. Comparisons of the MKNLMS-BT, MKNLMS-CS, and proposed algorithms.

	MKNLMS-BT	MKNLMS-CS	Proposed
PSE ($\times 10^{-1}$)	5.66	3.94	1.80
MSE ($\times 10^{-2}$)	3.21	4.09	1.77
# active kernels	63	63	31.1
$\bar{r}_n$	97.6	97.7	96.9

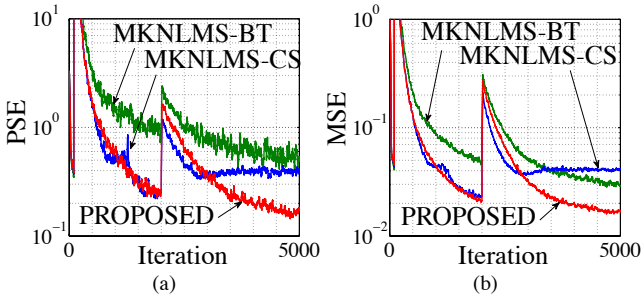

Fig. 1. PSE and MSE learning curves.

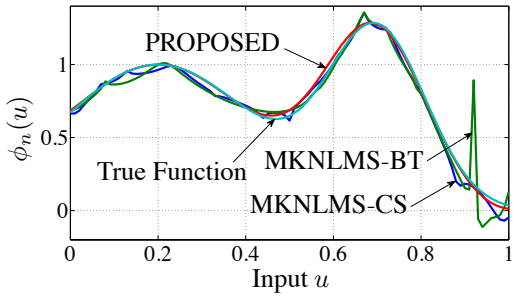

Fig. 2. Typical estimation results in the case of $L = 1$.

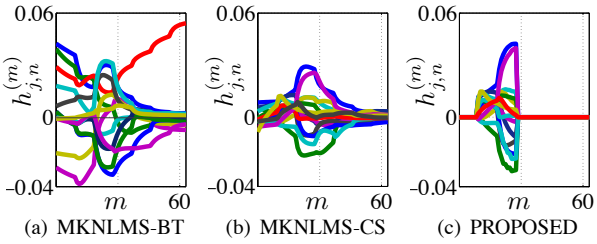

Fig. 3. Typical examples of the coefficients for each kernel in the case of $L = 1$. Each curve corresponds to each dictionary datum.

This is thanks to the use of the double regularizers and better understood by Figs. 2 and 3.

Figs. 2 and 3 show typical estimation results and filter coefficients, respectively, obtained after 5000 iterations in another simpler experiment for $L = 1$ for an illustration purpose. An impulsive noise contaminates the output of a nonlinear system for an input signal around $u = 0.92$. In Fig. 3(a), the red curve rises from $m = 37$ up to $m = 63$ and it depicts the coefficients for the data contaminated by the impulsive noise. This causes the notable overfitting of MKNLMS-BT

observed in Fig. 2. In Fig. 3(b), the coefficients for the large kernel parameters are small but not exactly zero. This causes the slight overfitting of MKNLMS-CS observed in Fig. 2. In Fig. 3(c), the coefficients for irrelevant kernels are exactly zero, implying that a proper model is selected online. This yields the good estimation result of the proposed algorithm observed in Fig. 2.

In our experiments, the MKNLMS-CS and proposed algorithms avoid overfitting impulsive noise with high probability since the algorithms suffer from notable overfitting only when impulsive noise coincidentally happens together with a new datum entering the dictionary. In case that the coincidence happens, the proposed algorithm alleviates it gradually due to the double regularizers as time goes by, while MKNLMS-CS has no such capability. Note that the performance deterioration of MKNLMS-CS stems from the fact that the dictionary size tends to increase in nonstationary environments and a newly entering dictionary datum pushes out the oldest one everytime the dictionary size exceeds the upper bound $r_{\max} := 96$.

Final remarks are given below. We conducted additional experiments to see how the proposed scheme competes with a single kernel approach that is supposed to be able to exploit the best kernel parameter. Under the present experimental conditions, we observed that the baseline algorithm called quantized kernel least mean squares (QKLMS) [5] outperformed the proposed scheme because of the use of extra information on the best kernel parameter. For nonstationary data, on the other hand, we observed that the proposed scheme outperformed QKLMS significantly, as will be reported in a conference [14]. We emphasize that this is because the proposed scheme can automatically track a suitable model adaptively while the single kernel approach cannot straightforwardly.

5. CONCLUDING REMARKS

Under the no-knowledge assumption on a suitable model, we have presented an efficient multikernel adaptive filtering algorithm with double regularizers for online model selection and learning. The proposed algorithm employs many possible kernels and selects relevant ones based on the block ℓ_1 norm regularizer for kernel groups. The other block ℓ_1 norm regularizer for data groups contributes to updating the dictionary adaptively. We have approximated the second regularizer by its Moreau envelope and applied the adaptive proximal forward-backward splitting method. Numerical examples have shown that the proposed algorithm selects a proper model (i.e., relevant kernels), alleviating the overfitting problem significantly, and identifies the *estimandum* with high accuracy. We stress here that the model selection and learning are made in online and adaptive fashion. Remarkably, all of these are made under the framework of adaptive proximal forward-backward splitting method and no separate procedure is required for model selection. The present study indicates that the multikernel adaptive filtering provides an attractive approach to the online model selection and learning problem.

There exist online/adaptive algorithms which can be applied directly to Θ_n in (3), including [15–17]. It is not trivial though that those algorithms completely nullify any columns and/or rows of the coefficient matrix $\boldsymbol{H}_n$ in a finite number of iterations. In contrast, the proposed scheme is naturally expected to make some rows with minor contributions be zero exactly during adaptation (which has been shown by simulations), thereby enabling online model selection. It will be an interesting future work to explore the possibility of employing such direct approaches.

6. REFERENCES

[1] M. Yukawa, "Multikernel adaptive filtering," *IEEE Trans. Signal Processing*, vol. 60, no. 9, pp. 4672–4682, Sept. 2012.

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

[3] C. Richard, J. Bermudez, and P. Honeine, "Online prediction of time series data with kernels," *IEEE Trans. Signal Process.*, vol. 57, no. 3, pp. 1058–1067, Mar. 2009.

[4] K. Slavakis, S. Theodoridis, and I. Yamada, "Adaptive constrained learning in reproducing kernel Hilbert spaces: the robust beamforming case," *IEEE Trans. Signal Process.*, vol. 57, no. 12, pp. 4744–4764, Dec. 2009.

[5] B. Chen, S. Zhao, P. Zhu, and J. C. Príncipe, "Quantized kernel least mean square algorithm," *IEEE Trans. Neural Networks and Learning Systems*, vol. 23, no. 1, pp. 22–32, 2012.

[6] B. Schölkopf and A. J. Smola, *Learning with Kernels*, MIT Press, Cambridge, MA, 2001.

[7] S. Theodoridis and K. Koutroumbas, *Pattern Recognition*, Academic, New York, 4th edition, 2008.

[8] M. Kloft, U. Brefeld, S. Sonnenburg, and A. Zien, "ℓ_p-norm multiple kernel learning," *J. Mach. Learn. Res.*, vol. 12, pp. 953–997, 2011.

[9] M. Gönen, "Bayesian efficient multiple kernel learning," in *Proc. ICML*, 2012.

[10] Y. Murakami, M. Yamagishi, M. Yukawa, and I. Yamada, "A sparse adaptive filtering using time-varying soft-thresholding techniques," in *Proc. IEEE ICASSP*, 2010, pp. 3734–3737.

[11] H. H. Bauschke and P. L. Combettes, *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*, Springer, New York: NY, 1st edition, 2011.

[12] I. Yamada, M. Yukawa, and M. Yamagishi, *Fixed-Point Algorithms for Inverse Problems in Science and Engineering*, vol. 49 of *Optimization and Its Applications*, chapter 17, pp. 345–390, Springer, New York, 2011.

[13] I. Yamada, K. Slavakis, and K. Yamada, "An efficient robust adaptive filtering algorithm based on parallel subgradient projection techniques," *IEEE Trans. Signal Processing*, vol. 50, no. 5, pp. 1091–1101, May 2002.

[14] M. Yukawa and R. Ishii, "On adaptivity of online model selection method based on multikernel adaptive filtering," in *Proc. APSIPA Annual Summit and Conference*, 2013, submitted.

[15] I. Yamada, S. Gandy, and M. Yamagishi, "Sparsity-aware adaptive filtering based on a Douglas-Rachford splitting," in *Proc. EUSIPCO*, 2011, pp. 1929–1933.

[16] H. Wang and A. Banerjee, "Online alternating direction method," in *Proc. ICMP*, 2012.

[17] S. Ono, M. Yamagishi, and I. Yamada, "A sparse system identification by using adaptively-weighted total variation via a primal-dual splitting approach," in *Proc. IEEE ICASSP*, 2013, pp. 6029–6033.