SET-MEMBERSHIP KERNEL ADAPTIVE ALGORITHMS

Andre Flores[1] *and Rodrigo C. de Lamare*[1,2]

[1]CETUC, Pontifical Catholic University of Rio de Janeiro, Brazil
[2]Department of Electronics, University of York, UK
Emails: andre.flores@cetuc.puc-rio.br, delamare@cetuc.puc-rio.br

ABSTRACT

Adaptive algorithms based on kernel structures have been a topic of significant research over the past few years. The main advantage is that they form a family of universal approximators, offering an elegant solution to problems with nonlinearities. Nevertheless, these methods deal with kernel expansions, creating a growing structure also known as dictionary, whose size depends on the number of new inputs. In this paper, we derive the set-membership kernel-based normalized least-mean square (SM-NKLMS) algorithm, which is capable of limiting the size of the dictionary created in stationary environments. We also derive as an extension the set-membership kernel-based affine projection (SM-KAP) algorithm. Finally, several experiments are presented to compare the proposed SM-NKLMS and SM-KAP algorithms to existing methods.

Index Terms— Kernel methods, sparsification, set-membership algorithms, kernel adaptive filtering.

1. INTRODUCTION

Adaptive filtering algorithms have been the focus of a great deal of research in the past decades, and the machine learning community has embraced and further advanced the study of these methods. However, conventional adaptive algorithms often work with linear structures, limiting the performance that they can achieve and constraining the number of problems that can be solved. Under this scope a new family of nonlinear adaptive filtering algorithms based on kernels has been developed. A kernel is a function that compares the similarity between two inputs. Kernel adaptive filtering (KAF) algorithms have been tested in many different scenarios and applications [1, 2, 3, 4, 5], showing very good results. One of the main advantages of KAF algorithms is that they are universal approximators [1], which gives them the capability to address complex and nonlinear problems. In other words, they can in principle model any input-output mapping. Most of these algorithms have been designed to solve convex optimization problems, which is also a desirable characteristic. However, the computational complexity is significantly higher than their linear counterparts[6].

One of the first KAF algorithms to appear and that is widely adopted in the KAF family because of its simplicity is the kernel least-mean square (KLMS) algorithm proposed in [7] and extended in [8]. The KLMS algorithm is inspired by the least-mean square algorithm and thanks to its good performance, led many researchers to work in the development of kernel versions of conventional adaptive algorithms. A few years later, a kernel version of the NLMS algorithm was proposed in [5] using a nonlinear regression approach for time series prediction. In [9], the affine projection algorithm (APA) was modified to develop a family of four algorithms known as the kernel affine projection algorithms (KAPA). The recursive least squares algorithm (RLS) was extended in [10], where the kernel recursive least squares (KRLS) was introduced. Later, the authors of [11] proposed an extended version of the KRLS algorithm. Moreover, the use of multiple kernels was studied in [12] and [13].

All the algorithms mentioned before have to deal with kernel expansions. In other words, they create a growing structure, also called dictionary, where they keep every new data input that arrives to compute the estimate of the desired output. The natural problem that arises is that the time and computational cost required to compute a certain output could exceed the tolerable limits . Several criteria were proposed to solve this problem such as algorithms with fixed dictonary size [14, 15, 16]. One of the most simple criteria is the novelty criterion (NC), presented in [17]. Basically it establishes two thresholds to limit the size of the dictionary. Another method, the approximate linear dependency (ALD) was proposed in [10] and verifies if a new input can be expressed as a linear combination of the elements stored before adding this input to the dictionary. The coherence criterion (CC) was introduced in [5] also to limit the size of the dictionary based on the similarity of the inputs. A measure called surprise (SC) was presented in [18] to remove redundant data.

In this work, we present the set-membership normalized kernel least-mean square (SM-NKLMS) and the set-membership kernel affine projection (SM-KAP) adaptive algorithms, which can provide a faster learning than existing kernel-based algorithms and limit the size of the dictionary without compromising performance. Similarly to existing set-membership algorithms [19, 20, 21, 22, 23, 24], the proposed SM-NKLMS and SM-KAP algorithms are equipped with variable step sizes and perform sparse updates. Unlike existing kernel-based adaptive algorithms the proposed SM-NKLMS and SM-KAP algorithms deal with, in a natural way, with the kernel expansion because of the data selectivity based on error bounds that they implement.

This paper is organized as follows. In Section 2, the problem formulation is presented. In Section 3 the SM-NKLMS and the SM-KAP algorithms are derived. Section 4 presents the results of the algorithms developed in an application involving a time series prediction task. Finally, Section 5 presents the conclusions of this work.

2. PROBLEM STATEMENT

Let us consider an adaptive filtering problem with a sequence of training samples given by $\{\boldsymbol{x}\,[i]\,, d\,[i]\}$, where $\boldsymbol{x}\,[i]$ represents the N-dimensional input column vector of the system and $d[i]$ is the desired signal at time instant i. The output of the adaptive filter is given by

$$y\,[i] = \mathbf{w}^T \boldsymbol{x}\,[i]\,, \tag{1}$$

where $\mathbf{w}$ is the column weight vector with length N.

Let us define a non-linear transformation denoted by $\boldsymbol{\varphi} : \mathbb{R} \rightarrow \mathbb{F}$ that maps the input to a high-dimensional feature space. Applying the transformation stated before, we map the input and the weights to a high-dimensional space obtaining:

$$\boldsymbol{\varphi}\,[i] = \boldsymbol{\varphi}\,(\boldsymbol{x}\,[i])\,, \tag{2}$$

Thanks to CNPq and FAPERJ Brazilian agencies for funding.

$$\boldsymbol{\omega}[i] = \boldsymbol{\varphi}(\mathbf{w}[i]), \tag{3}$$

The error generated is given by $e[i] = d[i] - \boldsymbol{\omega}^T[i]\,\boldsymbol{\varphi}[i]$. The main objective of the kernel-based adaptive algorithms is to implement an input-output mapping, such that the mean square error generated by the system is minimized. In addition, we assume that the magnitude of the estimated error is upper bounded by a quantity γ. The idea of using an error bound in system identification was reported in [19] and was used since then to develop different versions of data selective algorithms [20, 21].

3. PROPOSED SET-MEMBERSHIP KERNEL-BASED ALGORITHMS

Assuming that the value of γ is appropriately chosen then there exists several functions that satisfy the error requirement. To summarize, any function leading to an estimation error smaller than the defined threshold is an adequate solution, resulting in a set of filters. Consider a set $\bar{\boldsymbol{S}}$ containing all the possible input-desired pairs of interest $\{\boldsymbol{\varphi}[i], d[i]\}$. Now we can define a set $\boldsymbol{\theta}$ with all the possible functions leading to an estimation error bounded in magnitude by γ. This set is known as the feasibility set and is expressed by

$$\boldsymbol{\theta} = \bigcap_{\{\boldsymbol{\varphi},d\}\in\bar{\boldsymbol{S}}} \left\{\boldsymbol{\omega} \in \mathbb{F} \,/\, |d - \boldsymbol{\omega}^T\boldsymbol{\varphi}| \leq \gamma\right\} \tag{4}$$

Suppose that we are only interested in the case in which only measured data are available. Let us define a new set $\mathcal{H}[i]$ with all the functions such that the estimation error is upper bounded by γ . The set is called constraint set and is mathematically defined by

$$\mathcal{H}[i] \triangleq \left\{\boldsymbol{\omega} \in \mathbb{F} \,/\, |d[i] - \boldsymbol{\omega}^T\boldsymbol{\varphi}[i]| \leq \gamma\right\} \tag{5}$$

It follows that for each data pair there exists an associated constraint set. The set containing the intersection of the constraint sets over all available time instants is called exact membership set and is given by the following equation:

$$\psi[i] = \bigcap_{k=0}^{i} \mathcal{H}[i] \tag{6}$$

The exact membership set,$\psi[i]$, should become small as the data containing new information arrives. This means that at some point the adaptive filter will reach a state where $\psi[i] = \psi[i-1]$, so that there is no need to update $\boldsymbol{\omega}[i]$. This happens because $\psi[i-1]$ is already a subset of $\mathcal{H}[i]$.As a result, the update of any set-membership based algorithm is data dependent, saving resources, a fact that is crucial in kernel-based adaptive filters because of the growing structure that they create.

As a first step we check if the previous estimate is outside the constraint set, i.e., $|d[i] - \boldsymbol{\omega}^T[i-1]\,\boldsymbol{\varphi}[i]| > \gamma$. If the error exceeds the bound established, the algorithm performs an update so that the a posteriori estimated error lies in $\mathcal{H}[i]$ .If the previous case occurs we minimize $||\boldsymbol{\omega}[i+1] - \boldsymbol{\omega}[i]||^2$ subject to $\boldsymbol{\omega}[i+1] \in \mathcal{H}[i]$, which means that the a posteriori error $\xi_{ap}[i]$ is given by

$$\xi_{ap}[i] = d[i] - \boldsymbol{\omega}^T[i+1]\,\boldsymbol{\varphi}[i] = \pm\gamma \tag{7}$$

3.1. Proposed NKLMS Algorithm

The NKLMS update equation presented in [1] is given by

$$\boldsymbol{\omega}[i+1] = \boldsymbol{\omega}[i] + \frac{\mu[i]}{\varepsilon + ||\boldsymbol{\varphi}[i]||^2} e[i]\,\boldsymbol{\varphi}[i], \tag{8}$$

where $\mu[i]$ is the step size that should be chosen to satisfy the constraints of the algorithm and ε is a small constant used to avoid numerical problems. Substituting (8) in (7) and using the kernel trick to replace dot products by kernel evaluations we arrive at:

$$\xi_{ap}[i] = e[i] - \frac{\mu[i]}{\varepsilon + \kappa(\boldsymbol{x}[i], \boldsymbol{x}[i])} e[i]\,\kappa(\boldsymbol{x}[i], \boldsymbol{x}[i]). \tag{9}$$

Assuming that the constant ε is sufficiently small to guarantee that $\frac{\kappa(\boldsymbol{x}[i],\boldsymbol{x}[i])}{\varepsilon+\kappa(\boldsymbol{x}[i],\boldsymbol{x}[i])} \approx 1$ and following the procedure stated in [25] we obtain:

$$\mu[i] = \begin{cases} 1 - \frac{\gamma}{|e[i]|} & |e[i]| > \gamma \\ 0 & \text{otherwise} \end{cases} \tag{10}$$

We can then compute $\boldsymbol{\omega}$ recursively as follows:

$$\begin{aligned} \boldsymbol{\omega}[i+1] &= \boldsymbol{\omega}[i-1] + \frac{\mu[i-1]\,e[i-1]}{\varepsilon + ||\boldsymbol{\varphi}[i-1]||^2}\boldsymbol{\varphi}[i-1] \\ &\quad + \frac{\mu[i]}{\varepsilon + ||\boldsymbol{\varphi}[i]||^2} e[i]\,\boldsymbol{\varphi}[i] \\ &\quad \vdots \\ &= \boldsymbol{\omega}[0] + \sum_{k=1}^{i} \frac{\mu[k]}{\varepsilon + ||\boldsymbol{\varphi}[k]||^2} e[k]\,\boldsymbol{\varphi}[k]. \end{aligned} \tag{11}$$

Setting $\boldsymbol{\omega}[0]$ to zero leads to:

$$\boldsymbol{\omega}[i+1] = \sum_{k=1}^{i} \frac{\mu[k]}{\varepsilon + ||\boldsymbol{\varphi}[k]||^2} e[k]\,\boldsymbol{\varphi}[k]. \tag{12}$$

The output $f(\boldsymbol{\varphi}[i+1]) = \boldsymbol{\omega}^{\boldsymbol{T}}[i+1]\,\boldsymbol{\varphi}[i+1]$ of the filter to a new input $\boldsymbol{\varphi}[i+1]$ can be computed as the following inner product:

$$\begin{aligned} f(\boldsymbol{\varphi}[i+1]) &= \left[\sum_{k=1}^{i} \frac{\mu[k]\,e[k]}{\varepsilon + ||\boldsymbol{\varphi}[k]||^2}\boldsymbol{\varphi}^T[k]\right]\boldsymbol{\varphi}[i+1] \\ &= \sum_{k=1}^{i} \frac{\mu[k]\,e[k]}{\varepsilon + ||\boldsymbol{\varphi}[k]||^2}\boldsymbol{\varphi}^T[k]\,\boldsymbol{\varphi}[i+1]. \end{aligned} \tag{13}$$

Using the kernel trick [1] we obtain that the output is equal to:

$$\sum_{k=1}^{i} \frac{\mu[k]\,e[k]}{\varepsilon + \kappa(\boldsymbol{x}[k], \boldsymbol{x}[k])}\kappa(\boldsymbol{x}[k], \boldsymbol{x}[i+1]), \tag{14}$$

where $\mu[k]$ is given by (10).

Let us now define a coefficient vector $\boldsymbol{a} = \mu[i]\,e[i]$, so that equation (14) becomes:

$$\sum \frac{a_i}{\varepsilon + \kappa(\boldsymbol{x}[k], \boldsymbol{x}[k])}\kappa(\boldsymbol{x}[k], \boldsymbol{x}[i+1]) \tag{15}$$

Equations (10) -(15) summarize the proposed SM-NKLMS algorithm. We set the initial value of $\boldsymbol{\omega}$ to zero as well as the first coefficient. As new inputs arrive we can calculate the output of the system with (15). Then the error may be computed and if it exceeds the bound established we calculate the step size with (10). Finally, we update the coefficients $\boldsymbol{a}_i$. Note that some coefficients may be zero due the data selective characteristic of the algorithm. We do not need to store the zero coefficients as they do not contribute to the output computations, resulting in a saving of resources. This is an important result because it controls in a natural way the growing network created by the algorithm. In stationary environments the algorithm will limit the growing structure.

3.2. Proposed KAP Algorithm

Consider now the KAP algorithm, which uses the last K inputs to update the coefficients. Based on this fact, let us redefine our problem and use the past K constraint sets to perform the update. Under this scope it is also convenient to express the exact membership as follows:

$$\psi[i] = \left(\bigcap_{j=0}^{i-K} \mathcal{H}[j]\right)\left(\bigcap_{l=i-K+1}^{i} \mathcal{H}[l]\right) = \psi^{i-K}[i] \bigcap \psi^{K}[i], \tag{16}$$

where $\psi^K[i]$ designates the use of K constraint sets for updating. This means that the vector $\boldsymbol{\omega}[i]$ should belong to $\psi^K[i]$. In order to develop the SM-KAP algorithm we need to set several bounds $\bar{\gamma}_k[i]$, for $k = 1, \ldots, K$, so that the error magnitudes should satisfy these constraints after updating. It follows that there exists a space $S(i-k+1)$ containing all vectors $\boldsymbol{\omega}$ satisfying $d(i-k+1) - \boldsymbol{\omega}^{\boldsymbol{T}}\boldsymbol{\varphi}(i-k+1) = \bar{\gamma}_k[i]$ for $k = 1, \ldots, K$. The SM-KAPA should perform an update whenever $\boldsymbol{\omega}[i] \notin \psi^K[i]$, so that the equation $\| \boldsymbol{\omega}[i] - \boldsymbol{\omega}[i-1] \|^2$ subject to $\boldsymbol{d}[i] - \boldsymbol{\Phi}^T[i]\boldsymbol{\omega}[i] = \bar{\boldsymbol{\gamma}}[i]$ should be minimized, where $\bar{\boldsymbol{\gamma}}[i]$ is a vector containing all the K bounds. This constraint can also be expressed as $\boldsymbol{d}[i] - \bar{\boldsymbol{\gamma}}[i] = \boldsymbol{\Phi}^T[i]\boldsymbol{\omega}[i]$. Solving the problem with the method of the Lagrange multipliers we obtain:

$$\begin{aligned}\mathcal{L}(\boldsymbol{\omega}[i]) &= \| \boldsymbol{\omega}[i] - \boldsymbol{\omega}[i-1] \|^2 \\ &\quad + \boldsymbol{\lambda}^T[i]\left(\boldsymbol{d}[i] - \boldsymbol{\Phi}^T[i]\boldsymbol{\omega}[i] - \bar{\boldsymbol{\gamma}}[i]\right),\end{aligned} \tag{17}$$

where $\boldsymbol{\lambda}^T[i]$ is the vector of Lagrange multipliers. Now we can compute the gradient of $\mathcal{L}(\boldsymbol{\omega}[i])$ and equate it to a null vector.

$$\frac{\partial \mathcal{L}(\boldsymbol{\omega}[i])}{\partial \boldsymbol{\omega}[i]} = 2\boldsymbol{\omega}[i] - 2\boldsymbol{\omega}[i-1] - \boldsymbol{\lambda}^T[i]\boldsymbol{\Phi}^T[i] = \mathbf{0} \tag{18}$$

$$\boldsymbol{\omega}[i] = \boldsymbol{\omega}[i-1] + \frac{1}{2}\boldsymbol{\Phi}[i]\boldsymbol{\lambda}[i] \tag{19}$$

$$\boldsymbol{d}[i] - \bar{\boldsymbol{\gamma}}[i] = \boldsymbol{\Phi}^T[i]\left(\boldsymbol{\omega}[i-1] + \frac{1}{2}\boldsymbol{\Phi}[i]\boldsymbol{\lambda}[i]\right) \tag{20}$$

$$\boldsymbol{d}[i] - \bar{\boldsymbol{\gamma}}[i] = \boldsymbol{\Phi}^T[i]\boldsymbol{\omega}[i-1] + \boldsymbol{\Phi}^T[i]\boldsymbol{\Phi}[i]\frac{\boldsymbol{\lambda}[i]}{2} \tag{21}$$

$$\frac{\boldsymbol{\lambda}[i]}{2} = \left(\boldsymbol{\Phi}^T[i]\boldsymbol{\Phi}[i]\right)^{-1}(\boldsymbol{e}[i] - \bar{\boldsymbol{\gamma}}[i]), \tag{22}$$

We can now formulate the update equation, which is used as long as the error is greater than the established bound, i.e., $|e[i]| > \bar{\gamma}$

$$\boldsymbol{\omega}[i] = \boldsymbol{\omega}[i-1] + \boldsymbol{\Phi}[i]\left(\boldsymbol{\Phi}^T[i]\boldsymbol{\Phi}[i]\right)^{-1}(\boldsymbol{e}[i] - \bar{\boldsymbol{\gamma}}[i]), \tag{23}$$

where we have to consider that the vector $\boldsymbol{e}[i]$ is composed by the actual error and all $K-1$ a posteriori errors, corresponding to the $K-1$ last inputs used. This means that vector $\boldsymbol{e}[i]$ is expressed by $\begin{bmatrix} e[i] & e_{ap}[i-1] & \cdots & e_{ap}[i-K+1] \end{bmatrix}$, where $e_{ap}[i-k]$ denotes the a posteriori error computed using the coefficients at iteration i. In other words, $e_{ap}[i-k] = d[i-k] - \boldsymbol{\varphi}^T[i-k]\boldsymbol{\omega}[k]$.

Let us now consider a simple choice for vector $\bar{\boldsymbol{\gamma}}[i]$. We can exploit the fact that the a posteriori error was updated to satisfy the constraint $\boldsymbol{d}[i] - \boldsymbol{\Phi}^T[i]\boldsymbol{\omega}[i] = \bar{\boldsymbol{\gamma}}[i]$.That means that we can set the values of $\bar{\gamma}_k[i]$ equal to $e_{ap}[i-k+1]$for $i \neq 1$. Substituting this condition in equation (23), we obtain:

$$\boldsymbol{\omega}[i] = \boldsymbol{\omega}[i-1] + \boldsymbol{\Phi}[i]\left(\boldsymbol{\Phi}^T[i]\boldsymbol{\Phi}[i]\right)^{-1}(e[i] - \bar{\gamma}_1[i])\,\boldsymbol{u}, \tag{24}$$

where $\boldsymbol{u} = \begin{bmatrix} 1 & 0 & \cdots & 0 \end{bmatrix}^T$. We can now select $\bar{\gamma}_1[i]$ as in the SM-NKLMS so that

$$\bar{\gamma}_1[i] = \bar{\gamma}\frac{e[i]}{|e[i]|} \tag{25}$$

$$\boldsymbol{\omega}[i] = \boldsymbol{\omega}[i-1] + \boldsymbol{\Phi}[i]\left(\boldsymbol{\Phi}^T[i]\boldsymbol{\Phi}[i]\right)^{-1}(\mu[i]\,e[i])\,\boldsymbol{u} \tag{26}$$

$$\mu[i] = \begin{cases} 1 - \frac{\bar{\gamma}}{|e[i]|} & |e[i]| > \bar{\gamma} \\ 0 & \text{Other Case} \end{cases} \tag{27}$$

$$\boldsymbol{\omega}[i] = \sum_{j=1}^{i-1} a_j[i-1]\,\boldsymbol{\varphi}[j] + (\mu[i]\,e[i])\,\boldsymbol{\Phi}[i]\,\tilde{\mathbf{A}}[i], \tag{28}$$

where the matrix $\tilde{\mathbf{A}}[i]$ was redefined as

$$\tilde{\mathbf{A}}[i] = \left(\boldsymbol{\Phi}^T[i]\boldsymbol{\Phi}[i] + \epsilon\mathbf{I}\right)^{-1}\boldsymbol{u} \tag{29}$$

$$a_k[i] = \begin{cases} \mu[i]\,e[i]\,\tilde{a}_k[i], & k = i \\ a_k[i-1] + \mu[i]\,e[i]\,\tilde{a}_{K+k-i}[i], & i-K+1 \leq k \\ a_k[i-1] & 1 \leq k < i-K+1 \end{cases} \tag{30}$$

4. SIMULATIONS

In this section we assess the performance of the algorithms proposed for a time series prediction task. We have used two different time series to perform the tests, the Mackey Glass time series and a laser generated time series. First we separate the data into two sets, one for training and the other for testing as suggested in [1]. The time-window was set to seven and the prediction horizon to one, so that the last seven inputs of the time series were used to predict the value one step ahead. Additionally, both time series were corrupted by additive Gaussian noise with zero mean and standard deviation equal to 0.04. The Gaussian kernel was used in all the algorithms to perform all the experiments. Using the silver rule and after several tests, the bandwith of the kernel was set to one.

For the first experiment we analyze the performance of the adaptive algorithms over the Mackey-Glass time series. A total of 1500 sample inputs were used to generate the learning curve and the prediction was performed over 100 test samples. For the KAPA and the SM-KAPA algorithms, K was set to 7 so that the algorithms used the last seven input samples as a single input. For the KLMS algorithm the step size was set to 0.05.The error bound for the SM-NKLMS and the SM-KAPA algorithm was set to $\sqrt{5}\sigma$. The final results of the algorithms tested are shown in Table 1 where the last 100 data points of each learning curve were averaged to obtain the MSE. The learning curves of the algorithms based on kernels is presented in Fig. 1. From the curves, we see that the algorithms proposed outperform conventional algorithms in convergence speed.

Table 1. Performance on Mackey-Glass time series prediction

Algorithm	Test MSE	Standard Deviation
LMS	0.0230680	+/-0.00020388
NLMS	0.0213180	+/-0.00017318
SM-NLMS	0.0202340	+/-0.00084243
APA	0.0208600	+/-0.00231500
SM-APA	0.0204340	+/-0.00228940
KLMS	0.0075596	+/-0.00030344
SM-NKLMS	0.0054699	+/-0.00046209
KAPA2	0.0047812	+/-0.00041816
SM-KAPA	0.0046603	+/-0.00032855

In the second experiment we consider the performance of the proposed algorithms over a laser generated time series. In this case, 3500 sample inputs were used to generate the learning curves and the prediction was performed over 100 test samples. The setup used in the previous experiment was considered. Table 2 summarizes the MSE obtained for every algorithm tested. The final learning curves are showed in Fig. 2.

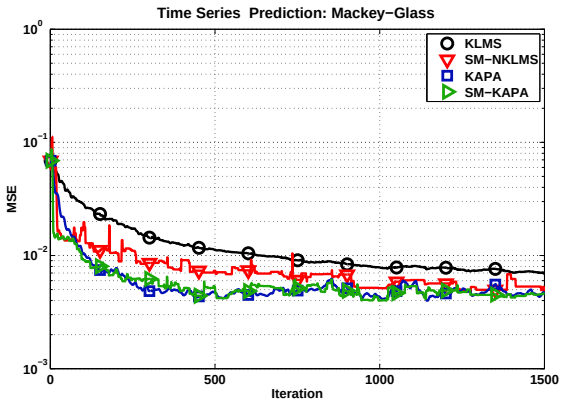

Fig. 1. Learning Curve of the Kernel Adaptive Algorithms for the Mackey-Glass Time Series prediction

Table 2. Performance on laser generated time series prediction

Algorithm	Test MSE	Standard Deviation
LMS	0.0214290	+/-0.00035874
NLMS	0.0197260	+/-0.00101250
SM-NLMS	0.0246950	+/-0.00647190
APA	0.0255460	+/-0.00465890
SM-APA	0.0200020	+/-0.00154490
KLMS	0.0090129	+/-0.00067428
SM-NKLMS	0.0038472	+/-0.00054237
KAPA2	0.0028253	+/-0.00030613
SM-KAPA	0.0029454	+/-0.00019424

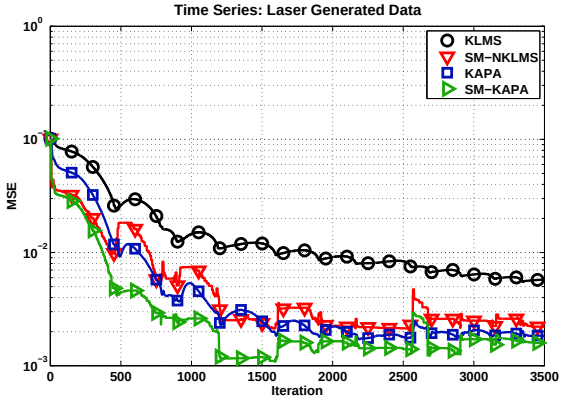

Fig. 2. Learning curve of the SM-KAPA for the Laser Time Series prediction

In the next experiment we study the size of the dictionary generated by the conventional KLMS algorithm using different criteria to limit the size and by the proposed SM-NKLMS algorithm. The result is presented in Fig. 3. We see that the proposed SM-NKLMS algorithm naturaly limits the size of the dictionary.

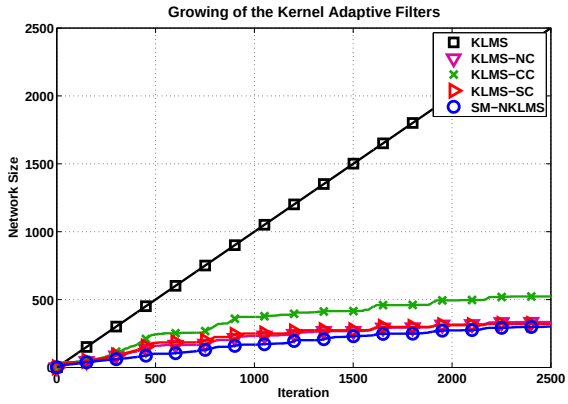

Fig. 3. Dictionary Size vs Iterations

As a final experiment, we analyze and compare the robustness of the algorithms proposed with respect to the conventional algorithms. Fig. 4 shows the results obtained. It is clear that the SM-NKLMS exhibits a better perfomance than the KLMS algorithm. In general, all kernel algorithms overperform their linear counterparts.

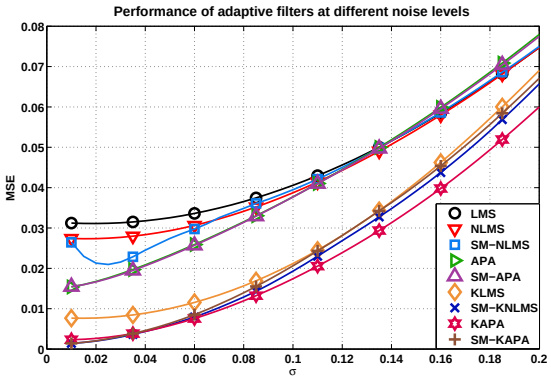

Fig. 4. Robustness

5. CONCLUSIONS

In this paper, we have devised two new data-selective kernel-type algorithms, namely, the SM-KNLMS and the SM-KAP algorithms. The proposed SM-KNLMS and SM-KAP algorithms have a faster convergence speed and a lower computational cost than the existing kernel-type algorithms in the same category. The proposed SM-KNLMS and SM-KAP algorithms also have the advantage of naturally limiting the size of the dictionary created by kernel based algorithms and a satisfactory noise robustness.

6. REFERENCES

[1] W. Liu, J. Príncipe, and S. Haykin, *Kernel Adaptive Filtering: A Comprehensive Introduction*, S. Haykin, Ed. John Wiley & Sons, 2010.

[2] J. M. Gil-Cacho, M. Signoretto, T. van Waterschoot, M. Moonen, and S. Jensen, "Nonlinear acoustic echo cancellation based on a sliding-window leaky kernel affine projection algorithm," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 21, no. 9, pp. 1867 – 1878, April 2013.

[3] J. M. Gil-Cacho, T. van Waterschoot, M. Moonen, and S. Jensen, "Nonlinear acoustic echo cancellation based on a parallel-cascade kernel affine projection algorithm," *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2012.

[4] Y. Nakijama and M. Yukawa, "Nonlinear channel equalization by multi-kernel adaptive filter," *IEEE 13th International Workshop on Signal Processing Advances in Wireless Communications*, 2012.

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

[6] S. Theodoridis, *Machine Learning A Bayesian and Optimization Perspective*, T. Pitts, Ed. Elsevir, 2015.

[7] W. Liu, P. Pokharel, and J. Príncipe, "The kernel least-mean-squares algorithm," *IEEE Transactions on Signal Processing*, vol. 56, no. 2, pp. 543–554, February 2008.

[8] P. Boboulis and S. Theodoridis, "Extension of wirtinger's calculus to reproducing kernel hilbert spaces and the complex kernel lms," *IEEE Transactions on Signal Processing*, vol. 59, no. 3, pp. 964–978, March 2011.

[9] W. Liu and J. Príncipe, "Kernel affine projection algorithms," *EURASIP Journal on Advances in Signal Processing*, vol. 2008, February 2008.

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

[11] W. Liu, Y. Wang, and J. Príncipe, "Extended kernel recursive least squares algorithm," *IEEE Transactions on Signal Processing*, vol. 57, no. 10, pp. 3801–3814, May 2009.

[12] R. Pokharel, S. Seth, and J. Príncipe, "Mixture kernel least mean square," *The 2013 International Joint Conference on Neural Networks*, 2013.

[13] M. Yukawa, "Multikernel adaptive filtering," *IEEE Transactions on Signal Processing*, vol. 60, no. 9, pp. 4672 – 4682, August 2012.

[14] S. Van Vaerenbergh, J. Via, and I. Santamaria, "A sliding-window kernel rls algorithm and its application to nonlinear channel identification," *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2006.

[15] S. Van Vaerenbergh, I. Santamaria, W. Liu, and J. Príncipe, "Fixed-budget kernel recursive least-squares," *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2010.

[16] F. Sheikholeslami, D. Berberidis, and G. B. Giannakis, "Kernel-based low-rank feature extraction on a budget for big data streams," *IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, 2015.

[17] J. Platt, "A resource-allocating network for function interpolation," *Neural Computation*, vol. 3, no. 3, pp. 213–225, 1991.

[18] W. Liu and J. Príncipe, "An information theoretic approach of designing sparse kernel adaptive filters," *IEEE Transactions on Neural Networks*, vol. 20, no. 12, pp. 1950 – 1961, November 2009.

[19] E. Fogel and Y. Huang, "On the value of information in system identification-bounded noise case." *Automatica*, vol. 18, pp. 229–238, March 1982.

[20] S. Gollamudi, S. Nagaraj, and Y. Kapoor, S.and Huang, "Set-membership filtering and a set-membership normalized lms algorithm with an adaptive step size." *IEEE Signal Processing Letters*, vol. 5, no. 5, pp. 111–114, May 1998.

[21] S. Werner and P. Diniz, "Set-membership affine projection algorithm," *IEEE Signal Processing Letters*, vol. 8, no. 8, pp. 231–235, August 2001.

[22] P. Diniz and S. Werner, "Set-membership binormalized data-reusing lms algorithms," *IEEE Transactions on Signal Processing*, vol. 51, no. 1, pp. 124–134, January 2003.

[23] R. C. de Lamare and P. Diniz, "Set-membership adaptive algorithms based on time-varying error bounds for cdma interference suppression," *IEEE Transactions on Vehicular Technology*, vol. 58, no. 2, pp. 644 – 654, February 2009.

[24] R. de Lamare and P. Diniz, "Blind adaptive interference suppression based on set-membership constrained constant-modulus algorithms with dynamic bounds," *IEEE Transactions on Signal Processing*, vol. 61, no. 5, pp. 1288 – 1301, November 2012.

[25] P. Diniz, *Adaptive Filtering: Algorithms and Practical Implementation*, third edition ed. Springer, 2008.