TIME SERIES PREDICTION FOR KERNEL-BASED ADAPTIVE FILTERS USING VARIABLE BANDWIDTH, ADAPTIVE LEARNING-RATE, AND DIMENSIONALITY REDUCTION

S. Garcia-Vega, E. A. León-Gómez, G. Castellanos-Dominguez

Universidad Nacional de Colombia, Signal Processing and Recognition Group,
Campus La Nubia, 170003, Manizales, Colombia.

ABSTRACT

Kernel-based adaptive filters are sequential learning algorithms, operating on reproducing kernel Hilbert spaces. Their learning performance is susceptible to the selection of appropriate values for kernel bandwidth and learning-rate parameters. Additionally, as these algorithms train the model using a sequence of input vectors, their computation scales with the number of samples. We propose a framework that addresses the previous open challenges of kernel-based adaptive filters. In contrast to similar methods, our proposal sequentially optimizes the bandwidth and learning-rate parameters using stochastic gradient algorithms that maximize the correntropy function. To remove redundant samples, a sparsification approach based on dimensionality reduction is introduced. The framework is validated on both synthetic and real-world data sets. Results show that our proposal converges to relatively low values of mean-square-error while provides stable solutions in real-world applications.

Index Terms— Sequential learning, Adaptive learning-rate, Kernel adaptive filters, Correntropy.

1. IMPLEMENTATION OF KERNEL-BASED ADAPTIVE FILTERS

Time series prediction techniques have been used in a wide variety of real-world applications, e.g., financial markets, electric utility load, weather state, and human emotions, among others. In practice, the underlying system models and data generating processes are usually not known, resulting in a challenge to build accurate and unbiased estimation of time-series data. The baseline solutions to perform prediction tasks are the statistical methods, mostly employing some improved versions of regressive models. However, their imposed analytic models frequently face numerous restrictions when dealing with non-stationarities and nonlinearities of data. To overcome nonlinearities, data-driven approaches are widely used like Neural Networks (NN), employing one or more layers of non-linear units to predict outputs. Nonetheless, NN algorithms tend to demand long training time and may get stuck in local minima.

In this paper, the goal is to learn a continuous input-output mapping $f{:}\mathcal{U}\to\mathbb{R}$ based on a paired sequence of input-output examples $\{\boldsymbol{u}_1, y_1\}, \ldots, \{\boldsymbol{u}_t, y_t\}$, where $\boldsymbol{u}_t$ is an m-dimensional input vector that belongs to the input set $\mathcal{U}\subset\mathbb{R}^m$, and $y_t\in\mathbb{R}$ is the output time series over the time domain $t\in N$. Because its ability to model non-linear systems, the input-output mapping function f can be learned using a kernel-based adaptive filter, yielding the following sequential rule through the time domain [1]:

$$f_t = \begin{cases} f_{t-1} + \eta\epsilon_t\kappa_\sigma(\boldsymbol{u}_t, \cdot), & \forall t \neq 0 \\ 0, & t = 0 \end{cases} \tag{1a}$$

$$\epsilon_t = y_t - f_{t-1}(\boldsymbol{u}_t) \tag{1b}$$

where $\eta\in\mathbb{R}^+$ is the learning-rate, $\kappa_\sigma(\cdot,\cdot)\in\mathbb{R}^+$ is a Mercer kernel with a bandwidth $\sigma\in\mathbb{R}^+$ that controls the mapping smoothness. We propose to optimize both η and σ by minimizing the prediction error $\epsilon_t\in\mathbb{R}$, using the following stages of adaptive filter implementation: $\eta\in\mathbb{R}^+$ is the learning-rate, $\kappa_\sigma(\cdot,\cdot)\in\mathbb{R}^+$ is a Mercer kernel with a bandwidth $\sigma\in\mathbb{R}^+$ that controls the mapping smoothness. We propose to optimize both η and σ by minimizing the prediction error $\epsilon_t\in\mathbb{R}$, using the following stages of adaptive filter implementation.

Kernel bandwidth optimization using correntropy: Based on nonlinear similarity measures, the adaptive filter parameters are proposed to be optimized using the correntropy cost function expressed over time as follows [2]:

$$J_t = \arg\max_{\forall\sigma,\eta}\{\exp\left(-\epsilon_t^2(\sigma_t,\eta_t)/2\lambda^2\right)\} \tag{2}$$

where $\lambda\in\mathbb{R}^+$ is the correntropy bandwidth that rules similarity between data points. Correntropy generalizes the conventional correlation function to nonlinear spaces, which has proven useful in many areas such as regression [3], adaptive filtering [4], classification [5], and spectral characterization [6]. The primary rationale behind the suggested strategy in Eq. (2) is to extract more information from the data structure for the adaptation process, yielding solutions that are more accurate for non-Gaussian processes [7]. In the first optimizing value, we perform the Kernel bandwidth estimation in Eq. (2) using the gradient descent method, yielding the

learning rule given as:

$$\sigma_t = \sigma_{t-1} + \beta \partial J_t / \partial \sigma_{t-1} \quad (3)$$

where σ_t is the bandwidth at iteration t and $\beta \in \mathbb{R}^+$ is the step-size parameter. Thus, using Eqs. (1a), (2) and (3), the kernel bandwidth estimation results as below:

$$\sigma_t = \sigma_{t-1} + \alpha \eta \epsilon_t \epsilon_{t-1} ||\boldsymbol{u}_t - \boldsymbol{u}_{t-1}||^2 \kappa_{\sigma_{t-1}}(\boldsymbol{u}_t, \boldsymbol{u}_{t-1}) \quad (4)$$

where $\alpha = J_t \beta / \lambda^2 \sigma_{t-1}^3$, and notation $||\cdot||$ stands for ℓ_2 norm.

Learning-rate estimation based on correntropy: Likewise in Eq. (3), the gradient-descent estimation yields the following learning-rate update at iteration t:

$$\eta_t = \eta_{t-1} + \beta \partial J_t / \partial \eta_{t-1} \quad (5)$$

where $\beta \in \mathbb{R}^+$ is the step-size parameter. Then, considering Eqs. (1a), (2) and (5), the learning-rate update results in the following rule:

$$\eta_t = \eta_{t-1} + \beta'' \epsilon_t \epsilon_{t-1} \kappa_\sigma(\boldsymbol{u}_t, \boldsymbol{u}_{t-1}) \quad (6)$$

being $\beta'' = \beta \exp(-\epsilon_t^2 / 2\lambda^2)$.

Dimensionality reduction through a sparsification strategy: In dimensionality reduction tasks, a low-dimensional representation, $\boldsymbol{V} = \{\boldsymbol{v}_i \in \mathbb{R}^n : i \in [1, t-1]\}$, must be obtained from a provided high-dimensional finite set $\boldsymbol{U} = \{\boldsymbol{u}_i \in \mathbb{R}^m : i \in [1, t-1]\}$ that holds m features extracted at $t-1$ samples, under the dimensionality restriction $n < m$. To this end, given a training pair $\{\boldsymbol{u}_t, y_t\}$ fed at the kernel-based adaptive filter input, sparsification methods can be employed to decide whether a new sample $\boldsymbol{u}_t$ should be added to a dictionary (that is, a reduced set of input samples) used to estimate nonlinear models [8], decreasing the computational complexity.

For encoding all non-linear data relationships within spaces, therefore, a couple of kernel matrices are introduced: *i*) input kernel matrix, $\boldsymbol{P} \in \mathbb{R}^{t-1 \times t-1}$ that holds elements $p_{ij} = \kappa_{\sigma_U}(\boldsymbol{u}_i, \boldsymbol{u}_j)$, with $\kappa_{\sigma_U} : \mathbb{R}^m \times \mathbb{R}^m \to \mathbb{R}^+$; *ii*) output kernel, $\boldsymbol{Q} \in \mathbb{R}^{t-1 \times t-1}$ with elements $q_{ij} = \kappa_{\sigma_V}(\boldsymbol{v}_i, \boldsymbol{v}_j)$, $\kappa_{\sigma_V} : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^+$. Both real-valued kernels are assumed to be Gaussian due to their universal approximating capability, desirable smoothness, and numeric stability [1]. So, the similarity measures of high and low dimensional spaces are respectively as below:

$$p_{ij} = \exp\left(-\|\boldsymbol{u}_i - \boldsymbol{u}_j\|^2 / 2\sigma_U^2\right) \quad (7a)$$

$$q_{ij} = \exp\left(-\|\boldsymbol{v}_i - \boldsymbol{v}_j\|^2 / 2\sigma_V^2\right) \quad (7b)$$

where $\sigma_U, \sigma_V \in \mathbb{R}^+$ are the corresponding kernel sizes.

Therefore, the kernel-based framework is devised so that the more correctly the points ($\boldsymbol{v}_i$ and $\boldsymbol{v}_j$) explain the similarity between the high-dimensional data points ($\boldsymbol{u}_i$ and $\boldsymbol{u}_j$), the more alike the kernel values p_{ij} and q_{ij} become. Thus, the principal rationale behind the suggested similarity framework in Eqs. (7a) and (7b) is to find a low-dimensional data representation $\boldsymbol{V}$ so that the mismatch between p_{ij} and q_{ij} can be minimized. So, the following cost function is proposed:

$$C = \frac{1}{(t-1)-1} \mathbb{E}\left\{|p_{ij} - q_{ij}| / p_{ij} : \forall i, j \in t-1, j \neq i\right\}, \; C \in \mathbb{R} \quad (8)$$

where notation $\mathbb{E}\{\cdot\}$ denotes the expectation operator.

In particular, we suggest to perform the cost function minimization using a gradient descent method, yielding the learning rule described as below:

$$\boldsymbol{v}_i^k = \boldsymbol{v}_i^{k-1} - \mu \partial C / \partial \boldsymbol{v}_i^{k-1} \quad (9)$$

where $\boldsymbol{v}_i^{k-1}$ is the low-dimensional representation of $\boldsymbol{u}_i$ at iteration $k-1$ and $\mu \in \mathbb{R}^+$ is the step-size parameter. Relying on Eqs. (7a), (7b) and (8), the gradient update results in the following rule:

$$\boldsymbol{v}_i^k = \boldsymbol{v}_i^{k\text{-}1} - \mu' \mathbb{E}\left\{ \frac{(\boldsymbol{v}_i^{k\text{-}1} - \boldsymbol{v}_j^{k\text{-}1})\left(p_{ij} q_{ij}^{k\text{-}1} - (q_{ij}^{k\text{-}1})^2\right)}{p_{ij} |p_{ij} - q_{ij}^{k\text{-}1}|} : \forall j \in t\text{-}1, j \neq i \right\}$$

where $\mu' = \mu / (\sigma_V^2 ((t-1)-1))$.

Consequently, introducing the quantization-size $\varepsilon \in \mathbb{R}^+$ [9], the following *sparsification* strategy is proposed:

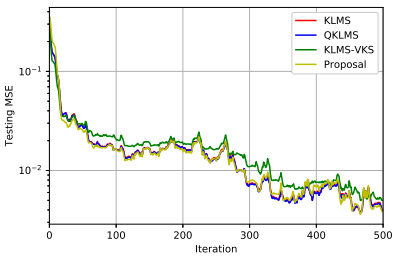
1. $\min_{1 \leq i \leq t-1} \|\boldsymbol{v}_t - \boldsymbol{v}_i\| \leq \varepsilon$: Update the closest sample weight to $\boldsymbol{u}_t$.

2. $\min_{1 \leq i \leq t-1} \|\boldsymbol{v}_t - \boldsymbol{v}_i\| > \varepsilon$: Add the input sample $\boldsymbol{u}_t$ to the dictionary.

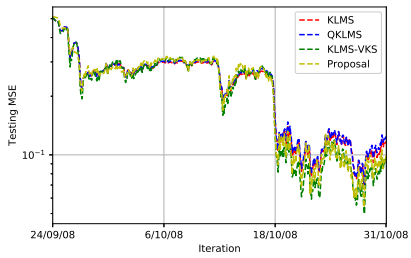
In terms of embedding preservation, the previous imposed restraints aim to select only the input data that encodes the global structures extracted from training samples. Thus, the main rationale behind the sparse dictionary building is to hold, as much as possible, those samples, which are more likely to appear.

2. RESULTS

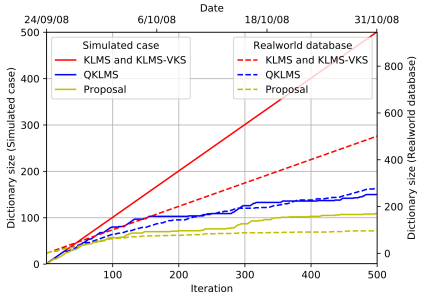
We validate the proposed kernel-based adaptive framework in the case of prediction tasks, using the mean-square-error (MSE) as a measure of performance. At each iteration of the training set, therefore, the learned filter is used to compute the MSE value on each test set as carried out in [1]. For a comparison purpose, the proposed variable bandwidth, adaptive learning-rate, and sparsification stretegy are contrasted with the following kernel-based adaptive filters: *i*) Kernel least-mean-square (noted as KLMS) as the simplest kernel-based adaptive strategy [10], *ii*) Quantized kernel least-mean-square (QKLMS) that introduces an online vector quantization method into KLMS [9]; *iii*) Kernel least-mean-square with variable kernel bandwidth (KLMS-VKS) described in [11].



(a) Learning curves of simulated case



(b) Learning curves of realworld database



(c) Dictionary size

Fig. 1. Performed results by each compared adaptive filter on tested datasets

The set-up of compared adaptive filters is as follows: *i*) the learning-rate is adjusted at η=0.2 for KLMS and QKLMS, while the initial learning-rate η_1 is set at 0 for KLMS-VKS and our proposal; *ii*) the Kernel bandwidth is set at $\sigma=\sqrt{1/2}$ for KLMS and QKLMS, which is also the initial bandwidth in our proposal and KLMS-VKS i.e., $\sigma_1=\sqrt{1/2}$; *iii*) the quantization size ε is set at 0.05 and 0.1 for QKLMS and our proposal, respectively; *iv*) the learning rate β is set at 0.1; *vii)* the correntropy bandwidth is set at λ=1; *v*) In the dimensionality reduction method, k=1000, n=2, μ=0.1 and σ_U, σ_V=0.2. All used kernel parameters had been adjusted heuristically.

Testing is carried out on the following two benchmarking datasets used in prediction tasks:

Mackey-Glass chaotic time-series:

Prediction performance is validated on a short-term signal set, which is generated by a chaotic system whose states are governed by a set of time-delayed differential equations. The task is to predict the current value using the previous ten consecutive samples. As experimented in [1], the data are normalized for the computation convenience, and for implementing the validation strategy, 500 samples are used as the training subset, while another 100 consecutive samples are the test subset.

Fig. 1(a) displays the learning curves, plotting the mean-square-error results performed by each compared solution versus the number of iterations. As seen, KLMS and KLMS-VKS methods show a relatively good performance since they achieve more stable MSE values through iterations. However, the evolution curves of network size in Fig. 1(c) make clear that their dictionary sizes linearly grows during training. This issue may be explained since both algorithms do not incorporate any sparsification technique, resulting in a significant drawback for implementation in online applications. By contrast, the number of samples of QKLMS algorithm grows very slowly, resulting in a final network that sizes only 150. Even that QKLMS and our proposal achieve similar MSE values, the former method demands a dictionary size significantly higher as seen in Fig. 1(c), and therefore, increasing the computational burden of online applications. As seen in Fig. 1(c), the proposed framework achieves a competitive performance, reaching the lowest network size through iterations and suggesting that its sparsification strategy (based on dimensionality reduction) helps to hold the most relevant samples to perform prediction tasks.

As regards the kernel bandwidth and learning-rate influence on the performed prediction, Table 1 displays the MSE evolution over the test set, showing that the proposed framework achieves the lowest MSE at iteration 100. Thus, there is an improvement in convergence time while competitive performance is maintained in future iterations, proving that our proposal converges to relatively low values of MSE, avoids overfitting, and provide stable solutions in real-world applications.

Wind speed data: This collection holds hourly wind speed records from the northern region of Colombia[1]. In this case, the performance is also evaluated in predicting the current value using the previous ten consecutive samples. The considered training set ranges from September-24-2008 to October-31-2008, and the test set ranges from May-28-2009 to June-02-2009.

Fig. 1(b) shows the learning curves estimated for the test set. The contrasted algorithms provide a robust performance through iterations. It is worth noting that the testing MSE decreases slower in all methods when compared with the learning curves of synthetic results (see Fig. 1(a)), clearly pointing out on the presence of highly non-stationary dynamics. This situation makes the kernel-based adaptive filters demand more time to encode the most relevant samples of this time-

[1]The dataset is publicly available at http://www.ideam.gov.co/solicitud-de-informacion

Table 1. Performed results on tested datasets at different iterations. The best overall method of each column are marked with bold notation. *MSE*-mean square error. *DS*-Dictionary Size.

Dataset	Method	Measure	Iteration				
			100	*200*	*300*	*400*	*500*
Mackey-Glass	KLMS	MSE	0.016	0.017	0.007	0.006	0.004
		DS	100	200	300	400	500
	QKLMS	MSE	0.017	0.017	0.007	0.006	0.004
		DS	80	103	126	136	150
	KLMS-VKS	MSE	0.021	0.019	0.011	0.008	0.005
		DS	100	200	300	400	500
	Proposal	MSE	0.016	0.007	0.007	0.006	0.004
		DS	**57**	**71**	**86**	**100**	**104**
			Iteration				
			28/09/08	*06/10/08*	*14/10/08*	*23/10/08*	*31/10/08*
Wind Speed	KLMS	MSE	0.253	0.299	0.249	0.084	0.115
		DS	100	300	500	700	900
	QKLMS	MSE	0.252	0.302	0.255	0.087	0.122
		DS	81	193	280	357	371
	KLMS-VKS	MSE	0.241	0.311	0.253	0.066	0.094
		DS	100	300	500	700	900
	Proposal	MSE	0.262	0.311	0.272	0.074	0.095
		DS	**62**	**88**	**96**	**108**	**121**

series correctly. The variable bandwidth and learning-rate, incorporated by our framework, promote the kernel-based adaptive filter to converge faster without significant loss of accuracy. As seen in Fig. 1(b), the evolution curves make clear also that our proposal reaches the lowest dictionary size during training while maintains a competitive MSE performance. However, if their initial values are inappropriately chosen at the beginning, the converging speed can be very slow. In this case, the suitable initial values of bandwidth and learning-rate can be selected using one of the methods developed on this account like the Silverman's rule of thumb.

Furthermore, the results presented in Table 1 suggest that the proposed framework is an adequate alternative to increase the convergence speed while maintains a high accuracy with the benefit of demanding a condensed dictionary size, and therefore, improving the performance of on-line prediction tasks.

3. CONCLUSION

In this study, a framework for kernel-based adaptive filters is introduced that addresses three main challenges of their online implementation: selection of appropriate bandwidth, learning-rate, and training samples. In particular, the first two stages are optimized based on nonlinear similarity cost function expressed over time. Namely, we propose to sequentially update the bandwidth and learning-rate parameters using a stochastic gradient algorithm that maximizes the correntropy function. Thus, the estimation error decreases along iterations, which means an improvement in convergence time while maintaining the robustness and simplicity of kernel-based adaptive filters. As the correntropy function is inherently insensitive to outliers, the proposed adaptive bandwidth and learning-rate provide an effective mechanism to eliminate the detrimental effect of outliers, and they are intrinsically different from the use of a threshold in conventional techniques.

To reduce the dictionary size, we also include a dimensionality reduction method that incorporates a sparsification strategy, employing a kernel-based cost function that quantifies the global structures of training samples. To this end, the proposed sparsification strategy is trained with the samples that are most likely to appear during the prediction task, starting with an empty dictionary and gradually adding new samples. As a result, the prediction task is performed by extracting the most relevant input data – concerning the embedding preservation – while maintaining a competitive performance. However, we must clarify that our sparsification strategy may be adversely affected with few training samples, due to it is more difficult to identify global structures under this scenario.

Validation on both datasets, synthetic and real-world, proves that the proposed framework converges to relatively low values of mean-square-error, avoiding overfitting while providing stable solutions in real-world applications.

We are in the process of expanding our research to other information theoretic measures and datasets. In the future, we plan to extend the results to the case where a more elaborate hyper-parameter tuning procedure is introduced into the compared kernel-based adaptive filters.

Acknowledgment

This work was supported by "*Programa Doctoral de Becas Colciencias – Convocatoria 617*" and "*Desarrollo de un sistema de monitoreo de condición y diagnóstico de fallas en línea de sistemas de generación de energía hidroeléctrica empleando una red de sensores inalámbricos de datos de alta resolución*".

4. REFERENCES

[1] W. Liu, J. Principe, and S. Haykin, *Kernel adaptive filtering: a comprehensive introduction*, vol. 57, John Wiley & Sons, 2011.

[2] Weihua Wang, Jihong Zhao, Hua Qu, Badong Chen, and Jose C Principe, "Convergence performance analysis of an adaptive kernel width mcc algorithm," *AEU-International Journal of Electronics and Communications*, vol. 76, pp. 71–76, 2017.

[3] Weifeng Liu, Puskal P Pokharel, and José C Príncipe, "Correntropy: Properties and applications in non-gaussian signal processing," *IEEE Transactions on Signal Processing*, vol. 55, no. 11, pp. 5286–5298, 2007.

[4] Songlin Zhao, Badong Chen, and Jose C Principe, "Kernel adaptive filtering with maximum correntropy criterion," in *The International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2011, pp. 2012–2017.

[5] Abhishek Singh and Jose C Principe, "A loss function for classification based on a robust similarity metric," in *Neural Networks (IJCNN), The 2010 International Joint Conference on*. IEEE, 2010, pp. 1–6.

[6] Ainara Garde, Leif Sörnmo, Raimon Jané, and Beatriz F Giraldo, "Correntropy-based spectral characterization of respiratory patterns in patients with chronic heart failure," *IEEE Transactions on Biomedical Engineering*, vol. 57, no. 8, pp. 1964–1972, 2010.

[7] Weifeng Liu, Puskal P Pokharel, and Jose C Principe, "Correntropy: A localized similarity measure," in *Neural Networks, 2006. IJCNN'06. International Joint Conference on*. IEEE, 2006, pp. 4919–4924.

[8] Chafic Saide, Régis Lengelle, Paul Honeine, Cédric Richard, and Roger Achkar, "Nonlinear adaptive filtering using kernel-based algorithms with dictionary adaptation," *International Journal of Adaptive Control and Signal Processing*, vol. 29, no. 11, pp. 1391–1410, 2015.

[9] Badong Chen, Songlin Zhao, Pingping Zhu, and José C Príncipe, "Quantized kernel least mean square algorithm," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 23, no. 1, pp. 22–32, 2012.

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

[11] B. Chen, J. Liang, N. Zheng, and J. Príncipe, "Kernel least mean square with adaptive kernel size," *Neurocomputing*, vol. 191, pp. 95–106, 2016.