ICIC Express Letters
Volume 11, Number 3, March 2017
ICIC International ©2017 ISSN 1881-803X
pp. 697–703

A KERNEL-BASED ALGORITHM FOR NONLINEAR ADAPTIVE FILTERING PROBLEMS

Xiangli Zhang[1], Hong Peng[1,*] and Jun Wang[2]

[1]Center of Radio Administrator and Technology Development
School of Computer and Software Engineering
[2]School of Electrical Engineering and Information Engineering
Xihua University
No. 999, Jinzhou Road, Jinniu District, Chengdu 610039, P. R. China
*Corresponding author: ph.xhu@hotmail.com

Received August 2016; accepted November 2016

Abstract. *The linear adaptive filtering algorithms fail to deal with nonlinear adaptive filtering problems. In this paper a kernel-based adaptive filtering algorithm is proposed to deal with nonlinear adaptive filtering problems, called kernel normalized least mean square algorithm. By means of the principle of kernel methods, the proposed kernel-based adaptive filtering algorithm is derived from classical normalized least mean square algorithm. The kernel normalized least mean square algorithm is evaluated on two nonlinear adaptive filtering problems: time series prediction and nonlinear channel equalization. Simulation results demonstrate the availability and performance of the kernel normalized least mean square algorithm.*

Keywords: Adaptive filtering, Kernel methods, Kernel adaptive filtering, Time series prediction, Nonlinear channel equalization

1. **Introduction.** Adaptive filtering is an important part in statistical signal processing, which can adaptively adjust its characteristics by parameter optimization [1, 2, 3, 4]. Therefore, it has a stronger ability to solve digital signal processing problems in comparison to the traditional filters with the fixed parameters. Now, adaptive filtering techniques have been widely used in a lot of signal processing areas, such as channel equalization, system identification and noise cancellation. Generally, adaptive filtering techniques can be classified into two types: linear filtering and nonlinear filtering. If the input-output of a filter is described in a linear relationship way, it is a linear adaptive filter; otherwise, it is a nonlinear adaptive filter. Least mean square (LMS) algorithm, as a member of a family of stochastic gradient algorithms, is one of the most famous adaptive filtering algorithms. LMS algorithm has been widely used in many signal processing areas because it has a small computation coat and is easy to implement. In recent years, nonlinear adaptive filtering problems in image and signal processing have received a lot of attention. However, the performances of LMS and other linear filtering algorithms are not ideal when they are used to deal with the nonlinear filtering problems; for example, they are not able to satisfy the requirement of these problems in terms of convergence, tracking speed and accuracy.

Kernel methods are a novel kind of machine learning, which provide a technique that can induce a nonlinear algorithm from the corresponding linear algorithm. Kernel methods have been widely used in pattern recognition, clustering analysis and signal processing problems [5, 6, 7, 8, 9]. In recent years, it has received a lot of attention on how to apply kernel methods to processing the nonlinear filtering problems. Liu et al. [10] proposed a nonlinear adaptive filtering algorithm using kernel methods, KLMS, which was a nonlinear version of classical least mean square algorithm. Subsequently, Pokharel et al. [11] developed a kernel least mean square algorithm with constrained growth (KLMSC). Usually, the nonlinear filtering algorithms induced by kernel methods are called kernel adaptive filtering algorithms. Our motivation stays on how to induce a nonlinear filtering algorithm from classical normalized least mean square algorithm (NLMS) via kernel methods and discuss its application in two nonlinear adaptive filtering problems. The contribution of the work is proposing a nonlinear filtering algorithm, called kernel normalized least mean square algorithm (KNLMS).

The rest of this paper is organized as follows. In Section 2, the classical normalized least mean square algorithm is reviewed, and then the proposed kernel adaptive filtering algorithm is described in detail. Two nonlinear adaptive filtering problems are used to demonstrate simulation results in Section 3. Finally, conclusions are drawn in Section 4.

2. Kernel Normalized Least Mean Square Algorithm.

In this section, classical normalized least mean square (NLMS) algorithm is reviewed, and then the proposed kernel normalized least mean square (KNLMS) algorithm is described in detail.

2.1. NLMS.

Similar to LMS algorithm, NLMS is also a linear filtering algorithm. For a linear filter, its input-output is described as a linear relation. Assume that $\{u_1, u_2, \ldots, u_N\}$ is an input signal sequence, and the corresponding filter output is $\{y_1, y_2, \ldots, y_N\}$, where N is the number of samples. NLMS algorithm minimizes the following empirical risk to determine the optimal weights w:

$$\min_{w} R_{emp}(w) = \sum_{i=1}^{N} (y_i - w(u_i)) \tag{1}$$

where w is the weights of the linear filter. According to stochastic gradient approach, the weight updating for NLMS algorithm can be described as follows:

$$\begin{cases} w_0 = 0 \\ e_n = y_n - w_{n-1}(u_n) \\ w_n = w_{n-1} + \eta_n e_n u_n \end{cases} \tag{2}$$

where e_n is the priori error, and η_n is step size and can be computed by

$$\eta_n = \frac{a}{< u_n, u_n >}, \quad 0 < a < 2 \tag{3}$$

Based on Equation (2), we have after n iterations

$$w_n = \sum_{i=1}^{n} \eta_i e_i u_i \tag{4}$$

Thus, for a new input $\tilde{u}$, the corresponding output of the filter is

$$\begin{cases} \tilde{y} = w_n(\tilde{u}) = \sum\limits_{n=1}^{N} \eta_i e_i < u_i, \tilde{u} > \\ e_n = y_n - \sum\limits_{n=1}^{N} \eta_i e_i < u_i, \tilde{u} >, \quad n = 1, 2, \ldots, N \end{cases} \tag{5}$$

It can be found that the input-output relation can be expressed in an inner product form. This is a premise that NLMS algorithm can be used to induce a nonlinear algorithm by kernel methods.

2.2. KNLMS.

Kernel methods are the known machine learning methods that provide an efficient technique to induce the nonlinear algorithm from a linear algorithm. The kernel technique uses a nonlinear mapping to transform input samples from input space into a high-dimensional feature space, and then in the feature space a linear algorithm is used to deal with the mapped data. When a linear algorithm is used, it is required that the computations on the mapped data can be expressed in an inner product form. Thus, inner product operation in the high-dimensional feature space can be characterized by a kernel function in the original input space. Because it does not need to explicitly compute inner products of feature vectors in the high dimensional feature space and all the calculations are carried out only in the original input space via a kernel function, the kernel technique can avoid the computation complexity in high-dimensional feature space. Kernel methods have been widely used in pattern recognition, such as support vector machines, kernel clustering and kernel discriminant analysis. In this work kernel methods are used to induce a kernel-based adaptive filtering algorithm from classical normalized least mean square algorithm (NLMS), called kernel normalized least mean square algorithm (KNLMS), which is a nonlinear version of classical normalized least mean square algorithm.

Assume that $\{u_1, u_2, \ldots, u_N\}$ is an input signal sequence, and the corresponding filter output is $\{y_1, y_2, \ldots, y_N\}$, where N is the number of samples. Let ϕ be a nonlinear mapping, which maps samples $\{u_1, u_2, \ldots, u_N\}$ in input space into a high-dimensional feature space F, i.e., $\{z_1, z_2, \ldots, z_N\}$, where $z_i = \phi(u_i)$, $i = 1, 2, \ldots, N$. In the high-dimensional feature space F, classical normalized least mean square algorithm is considered to tackle adaptive filtering problem considered. In the context of adaptive filtering, normalized least mean square algorithm searches for the optimal weights by minimizing the following empirical risk.

$$\min_w R_{emp}(w) = \sum_{i=1}^{N}(y_i - w(z_i)) \tag{6}$$

where w is the weights of the linear filter. In the high-dimensional feature space, classical normalized least mean square (NLMS) algorithm is used to update the weights as follows:

$$\begin{cases} w_0 = 0 \\ e_n = y_n - w_{n-1}(z_n) = y_n - w_{n-1}(\phi(u_n)) \\ w_n = w_{n-1} + \eta_n e_n z_n = w_{n-1} + \eta_n e_n \phi(u_n) \end{cases} \tag{7}$$

where e_n is the priori error, and η_n is step size and can be computed by

$$\eta_n = \frac{a}{< z_n, z_n >} = \frac{a}{< \phi(u_n), \phi(u_n) >}, \quad 0 < a < 2 \tag{8}$$

where $< \cdot, \cdot >$ denotes the inner product operation. Based on Equation (8), weights can be iterated as follows

$$\begin{aligned} w_n &= w_{n-1} + \eta_n e_n \phi(u_n) \\ &= w_{n-2} + \eta_{n-1} e_{n-1} \phi(u_{n-1}) + \eta_n e_n \phi(u_n) \\ &= w_{n-3} + \eta_{n-2} e_{n-2} \phi(u_{n-2}) + \eta_{n-1} e_{n-1} \phi(u_{n-1}) + \eta_n e_n \phi(u_n) \\ &\quad \cdots\cdots \qquad \cdots\cdots \\ &= w_1 + \eta_2 e_2 \phi(u_2) + \cdots + \eta_{n-1} e_{n-1} \phi(u_{n-1}) + \eta_n e_n \phi(u_n) \\ &= w_0 + \eta_1 e_1 \phi(u_1) + \eta_2 e_2 \phi(u_2) + \cdots + \eta_{n-1} e_{n-1} \phi(u_{n-1}) + \eta_n e_n \phi(u_n) \\ &= \sum_{i=1}^{n} \eta_i e_i \phi(u_i) \end{aligned}$$

Thus, we have

$$\begin{aligned} e_n &= y_n - w_{n-1}(\phi(u_n)) = y_n - \sum_{i=1}^{n-1} \eta_i e_i < \phi(u_i), \phi(u_n) > \\ &= y_n - \sum_{i=1}^{n-1} \eta_i e_i k(u_i, u_n) \end{aligned} \tag{9}$$

where $k(u_i, u_n) =< \phi(u_i), \phi(u_n) >$ denotes the inner product in feature space F, known as kernel function. Therefore, after N step training, the input-output relation of kernel normalized least mean square (KNLMS) algorithm can be described by

$$\begin{cases} \tilde{y} = \sum_{i=1}^{N} \eta_i e_i k(u_i, \tilde{u}) \\ e_n = y_n - \sum_{i=1}^{n-1} \eta_i e_i k(u_i, u_n), \quad n = 1, 2, \ldots, N \\ \eta_n = \frac{a}{k(u_n, u_n)}, \quad n = 1, 2, \ldots, N \end{cases} \tag{10}$$

It can be observed from Equation (10) that all computations are carried out in original input space via kernel function. This illustrates that the computations require only inner products (kernel function) rather than all information of a mapping ϕ. In kernel methods, Gaussian kernel and polynomial kernel are two commonly used kernel functions, and are given, respectively, by

$$\begin{cases} k(u, u') = \exp\big(-\delta ||u - u'||^2\big) \\ k(u, u') = \big(< u, u' > +1\big)^p \end{cases} \tag{11}$$

where δ is the width parameter of Gaussian kernel, and p is the order of polynomial kernel. However, note that kernel normalized least mean square algorithm will be degenerated into original normalized least mean square algorithm if the Gaussian kernel is used. Therefore, the polynomial kernel is used in kernel normalized least mean square algorithm. From Equation (10), KNLMS is a variable step-size nonlinear adaptive filtering algorithm, so it can be viewed as variable step-size version of KLMS algorithm.

Based on Equation (10), kernel normalized least mean square algorithm can be summarized as follows.

Table 1. Kernel normalized least mean square algorithm

Input: $D = \{(u_i, y_i) | i = 1, 2, \ldots, N\}$, //input data
Begin
/*Initialization*/
$y^0 = 0$;
$a = 0.2$; //learning rate
$p = 5$; //the order of polynomial kernel used
$n = 0$;
foreach $(u_n, y_n) \in D$ do
$\eta_n = a/k(u_n, u_n)$; //Compute step size
$y^{n-1} = \sum_{i=1}^{n-1} \eta_i e_i k(u_i, u_n)$; //Compute output of network
$e_n = y_n - y^{n-1}$; //compute error
$y^n = y^{n-1} + \eta_n e_n k(u_n, u_n)$
endfor
End.

3. **Simulation Results.** In this section two examples will be used to demonstrate the performance of the proposed KNLMS algorithm: time series prediction and nonlinear channel equalization. The simulation experiments mainly focus on the comparison of KNLMS algorithm with kernel least mean square (KLMS) algorithm and two classical adaptive filtering algorithms (LMS and NLMS).

3.1. **Time series prediction.** The first simulation example is the short-term prediction of the Mackey Glass chaotic time series with parameter $\tau = 30$, and the sampling period 6s. The time embedding is 10 and a segment of 500 samples is used as the training data and another 100 as the test data. All the data is corrupted by Gaussian noise with zero mean and 0.04 variance. In the experiment, KLMS, LMS and NLMS are used to compare the performance of the proposed KNLMS. Gaussian kernel with width parameter $\delta = 1$ is used in KLMS. Polynomial kernel with the order $p = 5$ is used in KNLMS since the KNLMS cannot use Gaussian kernel as stated before. For all the four algorithms, learning rate $\alpha = 0.2$ is chosen.

In the experiment, mean square error (MSE) is used as a measure to evaluate the performance of the four algorithms. Table 2 provides the experimental results of the four algorithms in terms of MSE, which are the average and the standard deviation of MSE obtained by the algorithms for 50 runs. LMS and NLMS are two classical linear algorithms, while KLMS and KNLMS are their nonlinear versions respectively. As one can observe in Table 2, KLMS and KNLMS have lower average values and standard deviations compared with LMS and NLMS. This illustrates that the performances of the two nonlinear filtering algorithms are significantly better than that of the linear filtering algorithms, LMS and NLMS. It can be seen that KNLMS has lower average value compared with KLMS; however, its standard deviation is slightly higher than that of KLMS.

Figure 1 shows the learning curves of the four filtering algorithms. Compared with LMS and NLMS, KNLMS has an obvious advantage since it has a faster convergence and can converge to lower MSE value. Moreover, KNLMS is better than KLMS in terms of convergence.

In order to further validate the applicability of the four algorithms, different noise variances are used in the data: $\sigma = 0.005, 0.02, 0.04, 0.1$. Table 3 provides the comparison results of the four algorithms under different noise variances. As presented in Table 3,

Table 2. The comparison results of the four algorithms in terms of MSE

Algorithms	Training MSE		Testing MSE	
	average	std.	average	std.
LMS	0.021	0.002	0.026	0.007
NLMS	0.018	0.0005	0.020	0.0012
KLMS	0.0074	0.0003	0.0069	0.0008
KNLMS	0.0056	0.00047	0.0062	0.0099

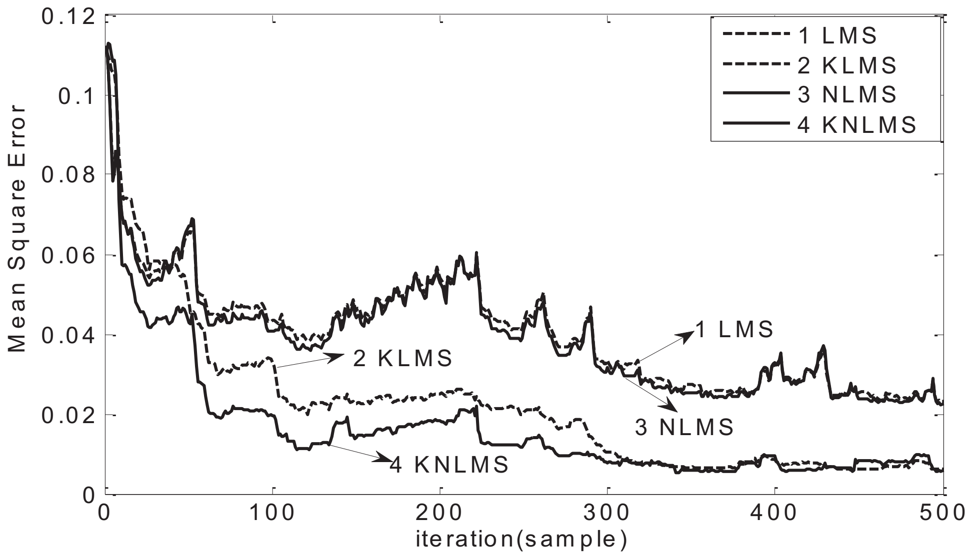

Figure 1. Learning curves of the four algorithms

Table 3. The MSE results of the four algorithms under different noise variances

Cases	LMS		KLMS		NLMS		KNLMS	
	average	std.	average	std.	average	std.	average	std.
Training ($\sigma = 0.005$)	0.017	5e-5	0.0050	2e-5	0.015	6e-5	0.0027	3e-5
Testing ($\sigma = 0.005$)	0.018	0.0002	0.0041	0.0001	0.016	0.0001	0.0030	0.0104
Training ($\sigma = 0.02$)	0.018	0.0002	0.0055	0.0001	0.016	0.0003	0.0035	0.0002
Testing ($\sigma = 0.02$)	0.018	0.0007	0.0046	0.0004	0.017	0.0006	0.0038	0.0103
Training ($\sigma = 0.04$)	0.021	0.002	0.0074	0.0003	0.018	0.001	0.0056	0.00047
Testing ($\sigma = 0.04$)	0.026	0.007	0.0069	0.0008	0.020	0.001	0.0062	0.0099
Training ($\sigma = 0.1$)	0.033	0.001	0.019	0.001	0.032	0.002	0.022	0.004
Testing ($\sigma = 0.1$)	0.031	0.005	0.018	0.003	0.037	0.004	0.029	0.012

KNLMS and KLMS are better than NLMS and LMS, and KNLMS has lower average value but slightly higher standard deviation compared with KLMS.

3.2. **Nonlinear channel equalization.** The channel equalization problem is described as follows. A binary sequence $(s_1, s_2, \ldots, s_N)$ is fed into a generally nonlinear channel and is further corrupted by additive Gaussian noise at the receiver end of the channel, and then the signal sequence is observed as $(r_1, r_2, \ldots, r_N)$. The aim of channel equalization is to construct an "inverse" filter that reproduces the original signal with as low an error rate as possible. It is easy to formulate it as a regression problem, with samples $\{(r_{t+D}, r_{t+D-1}, \ldots, r_{t+D-l}), s_t\}$, where l is the embedding length, and D is the equalization time lag.

The following channel model is used in the experiment: $z_t = s_t + 0.5s_{t-1}$, $\tau_t = z_t - 0.9z_t^2 + n_\sigma$, where n_σ is the white Gaussian noise with a variance of σ^2. Testing is finished on a 5000-sample random test sequence. The proposed KNLMS algorithm is compared with KLMS and two conventional algorithms, LMS and NLMS. Gaussian kernel with width parameter $\delta = 1$ is used in KLMS and polynomial kernel with the order $p = 5$ is used in KNLMS, and $l = 5$ and $D = 2$. The experimental results are listed in Table 4, where each entry consists of the average and the standard deviation for 100 repeated independent tests. The results in Table 4 show that KNLMS slightly outperforms KLMS in terms of the bit error rate (BER). It can be clearly observed from Table 4 that KNLMS outperforms the conventional LMS and NLMS substantially as can be expected because the channel is nonlinear.

Table 4. The performance comparison in NCE with different noise levels σ

Cases	LMS		KLMS		NLMS		KNLMS	
	average	std.	average	std.	average	std.	average	std.
BER ($\sigma = 0.1$)	0.162	0.014	0.020	0.012	0.155	0.013	0.018	0.001
BER ($\sigma = 0.4$)	0.177	0.012	0.058	0.008	0.152	0.018	0.056	0.190
BER ($\sigma = 0.8$)	0.218	0.012	0.130	0.010	0.203	0.111	0.128	0.007

4. **Conclusions.** This paper discussed a kernel-based adaptive filtering algorithm, KNLMS, which is a nonlinear version of classical normalized least mean square (NLMS) algorithm, called kernel normalized least mean square algorithm. The KNLMS was tested on time series prediction and nonlinear channel equalization problems to demonstrate the availability of the proposed KNLMS algorithm by comparing it with the existing kernel-based adaptive filtering algorithm, KLMS, and two classical normalized least mean square

algorithms, LMS and NLMS. Experimental results indicate that the proposed KNLMS is slightly better than KLMS and outperforms the conventional LMS and NLMS for channel equalization. Our further work is that the KNLMS is considered to solve more signal processing problems, for example, noise cancelation.

Acknowledgements. This work was partially supported by the National Natural Science Foundation of China (No. 61472328), Research Fund of Sichuan Science and Technology Project (No. 2015HH0057), China.

REFERENCES

[1] S. Haykin, *Adaptive Filter Thoery*, 4th Edition, Prentice-Hall, Upper Saddle River, NJ 07458, USA, 2002.
[2] A. Sayed, *Fundamentals of Adaptive Filtering*, Wiley, New York, 2003.
[3] J. Wang, P. Shi and H. Peng, Membrane computing model for IIR filter design, *Information Sciences*, vol.329, pp.164-176, 2016.
[4] H. Peng and J. Wang, A hybrid approach based on tissue P systems and artificial bee colony for IIR system identification, *Neural Computing and Applications*, DOI: 10.1007/s00521-016-2201-3, 2016.
[5] V. Vapnik, *The Nature of Statistical Learning Theory*, Springer-Verlag, New York, 1995.
[6] J. Shawe-Taylor and N. Cristilanini, *Kernel Methods for Pattern Analysis*, Cambridge University Press, Cambridge, England, 2004.
[7] H. Peng, J. Wang and W. Wang, Image watermarking method in multiwavelet domain based on support vector machines, *Journal of Systems and Software*, vol.83, pp.1470-1477, 2010.
[8] H. Peng, B. Li, X. Luo, J. Wang and Z. Zhang, A learning-based audio watermarking scheme using kernel Fisher discriminant analysis, *Digital Signal Processing*, vol.23, no.1, pp.382-389, 2013.
[9] H. Peng, J. Wang and Z. Zhang, Audio watermarking scheme robust against desynchronization attacks based on kernel clustering, *Multimedia Tools and Applications*, vol.62, no.3, pp.681-699, 2013.
[10] W. Liu, P. P. Pokharel and J. C. Principe, The kernel least mean square algorithm, *IEEE Trans. Signal Processing*, vol.56, no.2, pp.543-554, 2008.
[11] P. P. Pokharel, W. Liu and J. C. Principe, Kernel least mean square algorithm with constrained growth, *Signal Processing*, vol.89, no.3, pp.257-265, 2009.