NOVEL ADAPTIVE EQUALIZERS FOR THE NONLINEAR CHANNEL USING THE KERNEL LEAST MEAN SQUARES ALGORITHM

BỘ CÂN BẰNG THÍCH NGHI MỚI CHO KÊNH VỆ TINH PHI TUYẾN SỬ DỤNG GIẢI THUẬT BÌNH PHƯƠNG TRUNG BÌNH TỐI THIỂU KERNEL

Nguyen Viet Minh

ABSTRACT

The combination of the kernel trick and the least-mean-square (LMS) algorithm provides an interesting sample by sample update for an adaptive equalizer in reproducing Kernel Hilbert Spaces (RKHS), which is named here the KLMS. This paper shows that in the finite training data case, the KLMS algorithm is well-posed in RKHS without the addition of an extra regularization term to penalize solution norms. In this paper, we propose an algorithm for Kernel equalizers based on LMS algorithm with more simple computation, while the convergence rate will be adjusted based on the algorithm's control step size. The solution can be applied to the equalizers in OFDM satellite systems in order to reduce output errors and capacity of computation.

Keywords: *Kernel method; LMS algorithm; satellite channel; channel equalizers.*

TÓM TẮT

Sự kết hợp của phương pháp kernel với giải thuật bình phương trung bình tối thiểu (LMS) cho phép nâng cấp từng mẫu đối với bộ cân bằng thích nghi trong không gian tái tạo Hilbert Kernel (RKHS), được gọi là KLMS. Bài báo chứng tỏ rằng trong trường hợp số liệu hướng dẫn hữu hạn, giải thuật KLMS thích hợp trong không gian RKHS mà không cần thêm một giới hạn ổn định mở rộng. Trong bài báo này, một giải thuật được đề xuất cho bộ cân bằng kernel dựa trên LMS với việc tính toán đơn giản hơn trong khi tốc độ hội tụ có thể được điều chỉnh dựa trên kích thước bước điều khiển của thuật toán. Giải pháp này có thể được áp dụng cho bộ cân bằng trong hệ thống thông tin vệ tinh OFDM giúp giảm lỗi đầu ra và khối lượng tính toán.

Từ khóa: *Phương pháp kernel; giải thuật LMS; kênh vệ tinh; cân bằng kênh.*

Posts and Telecommunications Institute of Technology
Email: minhnv@ptit.edu.vn
Received:10 October 2019
Revised: 13 November 2019
Accepted: 20 December 2019

1. INTRODUCTION

Nowadays, the OFDM satellite information systems are considered to be strong nonlinear systems. Under the influence of radio transmission medium, the nonlinearity of the channel causes the signal to be intercepted between the symbols, ISI, and the interference between the subcarriers, ICI. Signal predistortion techniques at the transmitters [11] or equalizers at the receivers can be used to eliminate these interferences. The proposed control algorithms usually use the Volterra series. These algorithms are respresented in high order series [8] therefore they are extremely complex. Over the past ten years, adaptive nonlinear equalizers are being used in satellite channels [8]. These equalizers mainly use artificial neural networks [8, 11] and RBF networks are the most commonly used method. RBF equalizers, with simple structures, have the advantage of being adequate for nonlinear channels. However, their most basic disadvantage is that only the optimal local root can be found. Therefore, the output errors will be very large when these equalizers are used in OFDM satellite information systems. To overcome this disadvantage, kernel equalizers have been proposed with the application of kernel method to traditional equalization algorithms for the purpose of simplifying computation and thus improving the equalization efficiency [6, 7] [9, 10].

In this paper, we propose a new equalization method using multikernel technique which operates based on adaptive KLMS algorithm. Because this method uses the gradient principle therefore the computation is simple and effective [11]. This equalization algorithm is mainly based on least mean squares (LMS) algorithm and is kernel standardized accepts consistent criteria for directory design [12].

Basically, the LMS multikernel algorithm is still based on gradient princile. However, due to the specificity of the multikernel, there are different application hypotheses. In [1], to restrain imposing optimal weight, the authors used a port fuction softmax $\psi_k(n)$, therefore limits the application areas of the equalizer. In [2], the authors developed a multikernel learning algorithm based on the results of Bach et al. 2004 [3] and the extension of Zien and Ong 2007 [13]. The optimization tool is based on Shalev-Shwarts and Singer 2007 [14]. This is a generic framework for designing and analyzing the most statistic gradient descent algorithm. However, they are not commonly used for the fuctions with strong convexity. Do et al. 2009 [15] proposed the Pegasos algorithm, which has relatively good convergence with small λ. The disadvantage of this algorithm is that it requires knowing the upper limit of the optimal root.

In this paper, we propose an algorithm for kernel equalizers based on LMS algorithm that does not require the above factors to make the computation more simple, while the convergence rate will be adjusted based on the algorithm's control step size. The LMS kernel algorithm makes the output error of the equalizer smaller than the conventional LMS algorithm, therefore it is consistent with the equalizers in OFDM satellite systems.

The structure of this paper is presented as follow: Section 2: Kernel method; Section 3: KLMS equalizer; Section 4: Simulation and Section 5: Conclusion.

2. KERNEL METHOD

Kernel trick gives an algorithm which uses inner products in it's calculations. We can construct an alternative algorithm, by replacing each of the inner products with a positive definite kernel function.

Kernel Function: Given a set X, a 2-variable function $K : X \times X \rightarrow C$ is called **positive definite kernel function** ($K \geq 0$) provided that for each $n \in N$ and for every choice of n distinct points $\{x_1,......,x_n\} \subseteq X$ the Gram matrix of K regarding $\{x_1,......,x_n\}$ is positive definite.

The elements of the Gram Matrix (or kernel Matrix) of K regarding $\{x_1,......,x_n\}$ are given by the relation:

$$(K(x_i;x_j))_{i,j} = K(x_i,x_j) \text{ for } i;j = 1,...,n \quad (1)$$

The Gram Matrix is a Hermitian Matrix i.e. a matrix equal to it's Conjugate Transpose. Such a matrix being Positive Definite means that $\lambda \geq 0$ for each and every one of it's eigenvalues λ.

Kernel Trick:

Consider a set X and a positive definite (kernel) function

$K : X \times X \rightarrow R$. The RKHS theory ensures:

• the existence of a corresponding (Reproducing Kernel) Hilbert Space H, which is a vector subspace of F (X;R) (Moore's Theorem).

• the existence of a representation $\Phi : X \rightarrow H : \Phi(x) = k_x$ (feature representation) which maps each element of X to an element of H ($k_x \in H$ is called the reproducing kernel function for the point x).

so that:

$$\langle \Phi(x);\Phi(y)\rangle_H = \langle k_x;k_y\rangle_H = k_y(x) = K(x,y)$$

Thus:

• Through the feature map, the kernel trick succeeds in transforming a non-linear problem within the set X into a linear problem inside the "better" space H .

• We may, then, solve the linear problem in H, which usually is a relatively easy task, while by returning the result in space X. We obtain the final, non-linear, solution to our original problem.

Some Kernel functions:

The most widely used kernel functions include the Gaussian kernel:

$$K(\mathbf{x}_i,\mathbf{x}_j) = e^{-a\|\mathbf{x}_i - \mathbf{x}_j\|^2} \quad (2)$$

as well as the polynomial kernel:

$$K(\mathbf{x}_i,\mathbf{x}_j) = (\mathbf{x}_i^T \mathbf{x}_j + 1)^p \quad (3)$$

But there are plenty of other choices (e.g. linear kernel, exponential kernel, Laplacian kernel etc.)

Lots of algorithms capable of operating with kernels including adaptive filters (Least Mean Squares Algorithm) etc.

3. KLMS EQUALIZERS

The Channel Equalization Task aims at designing an inverse filter which acts upon the filter's output, x_n, thus producing the original input signal as close as possible.

We execute the algorithm NKLMS for the set of examples

$$((x_n,x_{n-1},\ldots,x_{n-k+1}),y_{n-D})$$

where k > 0 is the "equalizer's length" and D the "equalizer's time delay" (present at almost any equalization set up).

In other words, the equalizer's result at each time instance n corresponds to the estimation of y_{n-D}.

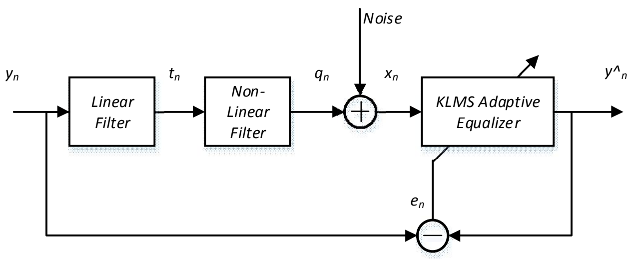

Figure 1. Equalization Task

Motivation:

Suppose we wish to discover the mechanism of a function

$$F : X \subset R^M \rightarrow R \text{ (true equalizer)}$$

having at our disposal just a sequence of example inputs-outputs

$$\{(\mathbf{x}_1,d_1),(\mathbf{x}_2,d_2),\ldots,(\mathbf{x}_n,dn),\ldots\}$$

(where $\mathbf{x}_n \in X \subset R^M$ and $d_n \in R$ for every $n \in N$).

Objective of a typical Adaptive Learning algorithm: to determine, based on the given "training" data, the proper input-output relation, $f_{\mathbf{w}}$, member of a parametric class of functions $H = \{f_{\mathbf{w}} : X \rightarrow R, \mathbf{w} \in R^v\}$, so as to minimize the value of a predefined loss function $L(\mathbf{w})$.

$L(\mathbf{w})$ calculates the error between the actual result dn and the estimation $f_{\mathbf{w}}(\mathbf{x}_n)$, at every step n.

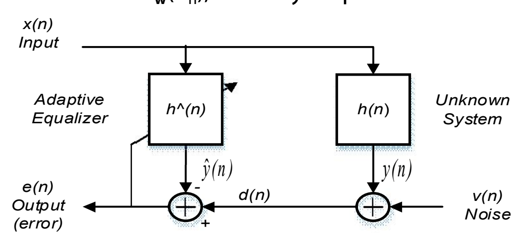

Figure 2. Adaptive Equalizer

Stochastic Gradient Descent method: at each instance time n = 1;2,…,N the gradient of the mean square error

$$-\nabla L(w) = 2E[(d_n - \mathbf{w}_{n-1}{}^T\mathbf{x}_n)(\mathbf{x}_n)] = 2E[e_n\mathbf{x}_n] \quad (4)$$

approximated by it's value at every time instance n

$$E[e_n\mathbf{x}_n] \approx e_n\mathbf{x}_n \quad (5)$$

leads to the step update (or weight-update) equation, which, towards the direction of reduction, takes the form:

$$\mathbf{w}_n = \mathbf{w}_{n-1} + \mu e_n\mathbf{x}_n \quad (6)$$

Note: parameter μ expresses the size of the "learning step" towards the direction of the descent.

The Least-Mean Square Code:

- **w = 0**
- for i = 1 to N (e.g. N = 5000)

  $f \equiv \mathbf{w}^T \mathbf{x}_i$

  $e = d_i - f$ (a priori error)

  $\mathbf{w} = \mathbf{w} + \mu e\mathbf{x}_i$
- end for

Variation: generated by replacing the last equation of the aforementioned iterative process with

$$\mathbf{w} = \mathbf{w} + \frac{\mu e}{\|x_i\|^2} x_i \quad (7)$$

called Normalized LMS. It's optimal learning rate has been proved to be obtained when $\mu = 1$.

Settings for the Kernel LMS algorithm :

- new hypothesis space: the space of linear functionals

  $\mathbf{H}_2 = \{T_\mathbf{w} : H \to R, T_\mathbf{w}(\phi(\mathbf{x})) = \langle \mathbf{w};\phi(\mathbf{x})\rangle_H, \mathbf{w} \in H\}$
- new sequence of examples: $\{(\phi(\mathbf{x}_1),d_1),\dots,(\phi(\mathbf{x}_n),d_n)\}$
- determine a function

  $f(\mathbf{x}_n) \equiv T_\mathbf{w}(\phi(\mathbf{x}_n)) = <\mathbf{w}, f(\mathbf{x}n)>_H, \quad \mathbf{w} \in H$

so as to minimize the loss function:

$$L(w) \equiv E[|d_n - f(\mathbf{x}_n)|^2] = E[|d_n - \langle \mathbf{w},\phi(\mathbf{x}_n)\rangle_H|^2]$$

- once more:

$$e_n = d_n - f(\mathbf{x}_n)$$

We calculate the Frechet derivative:

$$\nabla L(\mathbf{w}) = -2E[e_n\phi(\mathbf{x}_n)]$$

which again (according to LMS rational...) we approximate by it's value for each time instance *n*

$$\nabla L(\mathbf{w}) = -2e_n\phi(\mathbf{x}_n)$$

eventually getting, towards the direction of minimization

$$\mathbf{w}_n = \mathbf{w}_{n-1} + \mu e_n\phi(\mathbf{x}_n) \quad (8)$$

The Kernel Least-Mean Square Code:

- **Inputs**: the data $(\mathbf{x}_n,y_n)$ and their number N
- **Output**: the expansion $\mathbf{w} = \sum_{k=1}^{N} \alpha_k K(\cdot;\mathbf{u}_k)$, where $\alpha_k = \mu e_k$
- **Initialization**:

$f^0 = 0$, n: the learning step, μ: the parameter μ of the learning step

Define: vector $\alpha = 0$, array $D = \{.\}$ and the parameters of the *kernel* function.

- **for** $n = 1\dots N$ **do**

  if $n == 1$ **then**

  $f_n = 0$

  else

  Calculate the equalizer output

  $f_n = \sum_{k=1}^{M} \alpha_k K(\mathbf{u}_k, x_n)$

  end if

  Calculate the error: $e_n = d_n - f_n$

  $\alpha_n = \mu e_n$

  Register the new center $\mathbf{u}_n = \mathbf{x}_n$ at the center's list, i.e.

  $D = \{D,\mathbf{u}_n\}$, $\alpha^T = \{\alpha^T;\alpha_n\}$
- **end for**

Notes on Kernel LMS algorithm: After *N* steps of the algorithm, the input-output relation is

$$\boldsymbol{w}_n = \boldsymbol{\mu} \sum_{k=1}^{n} e_k \phi(\boldsymbol{x}_k)$$
$$f(\boldsymbol{x}_n) = \boldsymbol{\mu} \sum_{k=1}^{n-1} e_k K(\boldsymbol{x}_k, \boldsymbol{x}_n) \quad (9)$$

We can, again, use a normalised version:

$$\mathbf{w}_n = \mathbf{w}_{n-1} + \frac{\mu e_n}{K(x_n, x_n)} \boldsymbol{\phi}(\boldsymbol{x}_n) \quad (10)$$

getting the normalized KLMS (NKLMS).(replacing the step $a_n = \mu e_n$ with $a_n = \mu e_n/k$, where $k = K(\mathbf{x}_n,\mathbf{x}_n)$ would have already been calculated at some earlier step).

4. SIMULATIONS

In order to test the performance of KLMS algorithm we consider a typical non-linear channel equalization task. The non-linear channel consists of a linear filter

$$t_n = 0.8y_n + 0.7y_{n-1}$$

and a memoryless non-linearity

$$q_n = t_n + 0.8t_n^2 + 0.7t_n^3$$

Then, the signal gets effected by additive white Gaussian noise being finally observed as x_n. Noise level has been set equal to 15dB.

We used 50 sets of 5000 input signal samples each (Gaussian random variable with zero mean and unit variance) comparing the performance of standard LMS with that of KLMS.

We consider all algorithms in their normalized version. The step update parameter was set for optimum results (in terms of the steady-state error rate). Time delay was also configured for optimum results.

The learning curve is plotted in Figure 3. We compare the performance of the conventional LMS and the KLMS. The Gaussian kernel with $a = 0.1$ is used in the KLMS for best results, and $l = 5$ and $D = 2$. The results are presented in Table II; each entry consists of the average and the standard deviation for 100 repeated independent tests. The results in Table 1 show that, the KLMS outperforms the

P-ISSN 1859-3585 | E-ISSN 2615-9619

conventional LMS in terms of the bit error rate (BER) as can be expected because the channel is nonlinear. The regularization parameter for the LMS and the learning rate of KLMS were set for optimal results.

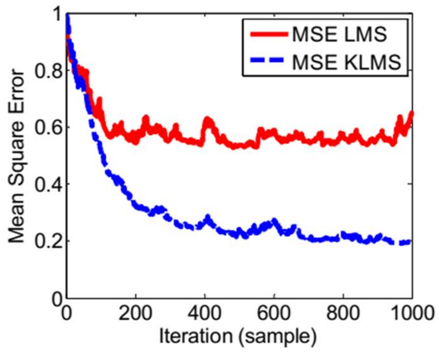

Figure 3. The learning curves of the LMS (η = 0.005) and kernel LMS (η = 0.1) in the nonlinear channel equalization (σ = 0.4)

Table 1. Performance comparison in nce with different noise levels σ

Algorithms	Linear LMS (η = 0.005)	KLMS (η=0.1)
BER (σ = 0.1)	0.162±0.014	0.020±0.012
BER (σ = 0.4)	0.177±0.012	0.058±0.008
BER (σ = 0.8)	0.218±0.012	0.130±0.010

5. CONCLUSIONS

This paper proposes the KLMS algorithm used in Nonlinear Satellite Channel Equalization. Since the update equation of the KLMS can be written as inner products, KLMS can be efficiently computed in the input space. This capability includes modeling of nonlinear systems, which is the main reason why the kernel LMS can achieve good performance in the nonlinear channel equalization.

Demonstrated by the experiments, the KLMS has general applicability due to its simplicity since it is impractical to work with batch mode kernel methods in large data sets. The KLMS is very useful in problems like nonlinear channel equalization The superiority of KLMS is obvious, which was of no surprise as LMS is incapable of handling non-linearities.

REFERRENCES

[1]. Rosha Pokharel, Sohan Seth, Jose C. Principe. *Mixture Kernel Least Mean Square*. NSF IIS 0964197.

[2]. Francesco Orabona, Luo Jie, Barbara Caputo, 2012. *MultiKernel Learning With Online-Batch Optimization*. Journal of Machine Learning Research 13, 227-253.

[3]. F. R. Bach, G. R. G. Lanckriet, and M. I. Jordan, 2004. *Multiple kernel learning, conic duality, and the SMO, algorithm*. In Proc. of the International Conference on Machine Learning.

[4]. P. Bartlett, E. Hazan, and A. Rakhlin, 2008."*Adaptive online gradient descent*. In Advances in Neural Information Processing Systems 20, pages 65–72. MIT Press, Cambridge, MA.

[5]. F. Orabona, L. Jie, and B. Caputo, 2010. *Online-batch strongly convex multi kernel learning.* In Proc. Of the 23rd IEEE Conference on Computer Vision and Pattern Recognition..

[6]. Yukawa Masahiro, 2012. *Multi-Kernel Adaptive Filtering*. IEEE transactions on signal processing, vol. 60, no. 9, pp. 4672–4682.

[7]. M. Yukawa, 2011. *Nonlinear adaptive filtering techniques with multiple kernels*. 2011 19th European Signal Processing Conference, Barcelona, pp. 136-140.

[8]. W. Liu, J. Principe, and S. Haykin, 2010. *Kernel Adaptive Filtering*. New Jersey, Wiley.

[9]. B. Scholkopf and A. Smola, 2001. *Learning with kernels: Support vector machines, regularization, optimization, and beyond.* MIT press.

[10]. Y. Nakajima and M. Yukawa, 2012. *Nonlinear channel equalization by multikernel adaptive filter*. in Proc. IEEE SPAWC.

[11]. J. Principe, W. Liu and S. Haykin, 2011. *Kernel Adaptive Filtering: A Comprehensive Introduction*. Wiley, Vol. 57.

[12]. C. Richard, J Bermudez and P. Honeine, 2009. *Online Prediction of Time Series Data With Kernel*. IEEE Trans. Signal Processing, Vol. 57, No 3.

[13]. A. Zien and C. S. Ong, 2007. *Multiclass multiple kernel learning*. In Proc. of the International Conference on Machine Learning.

[14]. S. Shalev-Shwartz and Y. Singer, 2007. *Logarithmic regret algorithms for strongly convex repeated games*. Technical Report 2007-42, The Hebrew University.

[15]. C. B. Do, Q. V. Le, and Chuan-Sheng Foo, 2009. *Proximal regularization for online and batch learning*. In Proc. of the International Conference on Machine Learning.

THÔNG TIN TÁC GIẢ

Nguyễn Viết Minh

Học viện Công nghệ Bưu chính Viễn thông