Contents lists available at ScienceDirect

Digital Signal Processing

www.elsevier.com/locate/dsp

Variable learning rates kernel adaptive filter with single feedback

Ji Zhao [a], Hongbin Zhang [a,*], Xiaofeng Liao [b]

[a] School of Electronic Engineering, University of Electronic Science and Technology of China, Chengdu, Sichuan 611731, PR China
[b] School of Electronic and Information Engineering, Southwest University, Chongqing, 400715, PR China

ARTICLE INFO

Article history:
Available online xxxx

Keywords:
Kernel adaptive filter
Single feedback
Variable learning rate
Mean square convergence

ABSTRACT

In this paper, we propose a novel kernel adaptive filtering algorithm, which called variable learning rates kernel adaptive filter with single feedback (SF-VLRKAF). Based on a feedback structure, the past information can be used to estimate current output to improve the filtering performance, because of a momentum term existing in the weight update equation. A switch ON–OFF normalized variable learning rate is developed to obtain a tradeoff between convergence rate and filtering accuracy of the proposed algorithm. The weights, in the SF-VLRKAF, are updated at each iteration to avoid local minimum, and the analysis of mean square convergence is performed. Furthermore, a sufficient condition ensuring mean square convergence is obtained by applying the energy conservation relation. Moreover, we derive the lower and upper bounds on a theoretical result of the steady-state excess mean square error. Simulations for nonlinear regression, chaotic time-series predictions and real-world applications are presented to illustrate the effectiveness of the new algorithm.

1. Introduction

During the past decade, there has been a growing interest in kernel methods, e.g., support vector machines, Gaussian processes, regularization networks [1], and these kernel methods have been extensively applied in many areas including time series prediction, channel equalization, pattern classification, dimensionality reduction, image processing and nonlinear regression [2,3]. However, since most kernel methods are batch algorithms, they are not suitable for real-time applications. Therefore, the extension of kernel methods to online algorithms is quite necessary.

Recently, the kernel adaptive filter (KAF) as a class of efficient online learning algorithms has been widely studied. KAF is a natural generalization of linear adaptive filters in the reproducing kernel Hilbert spaces (RKHS), which allows nonlinear learning problems to be solved in the input space as convex optimization problems in the transformed space. The inner product operation in the RKHS can be computed efficiently by the kernel trick, which will be explained in the next section. From the data space perspective, the KAFs can be developed in two different spaces, i.e., the original input space and RKHS. Classical adaptive algorithms can be directly kernelized to obtain some elegant KAFs, e.g., the kernel least mean square (KLMS) [4], the kernel affine projection algorithm (KAPA) [5,6], the kernel recursive least squares (KRLS) [7] and the extended kernel recursive least squares (EKRLS) [8]. The KRLS and EKRLS can also be explained from the viewpoint of Gaussian processes [9,10]. Moreover, the KAFs can be simply derived in the RKHS, for instance, one can use the statistical steepest descent method to obtain the updated forms [2,3,11]. The inner relation of these two spaces can be illustrated from the viewpoint of isomorphism [12]. In addition, according to the types of kernel functions used in the KAFs, the nonlinear filters can also be classified into the single-kernel KAFs (SK-KAFs) and the multiple-kernel KAFs (MK-KAFs). In the SK-KAFs, a reasonable kernel has been assumed available prior to verify the efficiency of these algorithms, such as the KLMS, KAPA and KRLS, etc., [1–8,11–21]. This assumption, however, is not always realistic, particularly when the non-stationary data are considered. And finding a reasonable kernel is costly even though possible. Hence, MK-KAFs are proposed by finding more than one kernel in adapting to non-stationary environments [22–24].

Furthermore, as a kind of learning algorithms, the KAFs bear different performances on the basis of different updated forms. There are mainly three kinds of updated forms in KAFs: 1) there is only one component of weight updated at each learning process, others remain unchanged [4,16]; 2) some components of weights, e.g., the latest k components are updated as new data become available, typically a least-squares cost function is minimized [5,6]; and 3) all components of weights are updated at each training process to avoid local minimum, which is similar to the treatment

* Corresponding author.
E-mail address: zhanghb@uestc.edu.cn (H. Zhang).

https://doi.org/10.1016/j.dsp.2018.06.007
1051-2004/

in neural networks [1,3,7,11]. In addition, there is much interest in developing complex kernel algorithms to effectively deal with complex valued signals [25]. But it is beyond the scope of this paper.

It is worth noting that the main bottlenecks of KAFs are their growing structure and computational complexity with the increase of training samples. Online sparsification is therefore required to maintain a reasonable dictionary size for computational efficiency as well as memory efficiency in the implementation of KAFs. The most widely investigated criteria of online sparsification are the distance, the coherence criterion (CC), as well as the Babel criterion [26]. Resource-allocating networks have introduced the distance criterion to control their complexity in radial-basis-function networks, and retain the most mutually distant samples [16,17,27]. In recent applications of compressed sensing [15,28], samples with mutually least coherence are retained. As an extension of CC, the Babel criterion uses the cumulative coherence as a measure of diversity [29]. It is worth noting that these criteria are based on an approximation process [26]. In addition, there are other classical criteria, e.g., the approximate linear dependency (ALD) [7], the novelty criterion [27], the surprise criterion [14], the sparsity-promoting regularizations [18,19], the sliding window methods [30], and the fixed budget models [31–33]. Based on these sparsification methods, KAFs can be efficiently applied in real-time applications.

Regarding the filter structure, the KAFs have feedforward structure and feedback structure. Most KAFs have feedforward structures. The structure is independent of the previous output of the system [1,2,4–8,12–19]. It is common to consider, nevertheless, the past information provided to KAFs to construct feedback structure. This approach has been widely used in signal processing and neural networks [3,11,20,21,34]. Recently, we have proposed a KAF based on feedback, namely, the kernel least mean square with single feedback (SF-KLMS) [11]. In the SF-KLMS, there is only one delayed output applied to update the weights in a recurrent fashion, and the SF-KLMS can effectively improve the convergence rate of a KAF.

In this paper, we propose a new KAF algorithm, namely, the variable learning rates kernel adaptive filter with single feedback (SF-VLRKAF), which is based on an efficient structure [11]. Compared with the SF-KLMS, we conduct some minor changes of the original algorithm to match the theoretical proof. In the SF-VLRKAF, only a single delayed output in a linear form is applied to update all components of weights (feedforward weight and feedback weight) to minimize the cost function, e.g., least mean square. A switch ON–OFF normalized variable learning rate strategy is proposed to further improve the filtering performance. We treat the feedback term as a special input based on the idea of the energy conservation relation (ECR) and real-time recurrent learning (RTRL) structure [34]. And then, the stability of SF-VLRKAF is analyzed by the ECR, and a sufficient condition for mean square convergence is also obtained. In addition, we derive effective lower and upper bounds on the steady-state excess mean square error of the proposed algorithm. In this work, the coherence-based criterion for sparsification is utilized to deal with the increase of the structure size.

The rest of this paper is organized as follows. In Section 2, before presenting the SF-VLRKAF algorithm, we briefly introduce the concept of the Mercer kernel. Section 3 provides an analysis of the mean square performance, and presents the main theoretical results. Simulation results are presented in Section 4 to verify the effectiveness of SF-VLRKAF. Finally, discussions are given in Section 5.

2. The SF-VLRKAF algorithm

Nonlinear signal processing problems are not readily solved by using linear models in their original low dimensional spaces. However, the nonlinear models can be established via applying a nonlinear mapping function, which transforms low dimensional spaces into high dimensional feature spaces. And, it is commonly known that Mercer kernels can be used to implement the nonlinear operations.

2.1. Mercer kernels

The Mercer kernel is a continuous, symmetric and positive-definite function $k(\cdot,\cdot): \mathbb{U} \times \mathbb{U} \rightarrow \mathbb{R}$, where $\mathbb{U} \subset \mathbb{R}^{l\times 1}$ is the input space. The most important property of a Mercer kernel is the kernel trick, which allows inner-product based algorithms to be performed implicitly in a feature space $\mathbb{F}$ by replacing all inner-products by kernels [13], i.e.,

$$k(\boldsymbol{u}, \boldsymbol{u}') = \langle \varphi(\boldsymbol{u}), \varphi(\boldsymbol{u}') \rangle, \tag{1}$$

where $\boldsymbol{u} \in \mathbb{U}$; $\varphi(\cdot)$ is a nonlinear mapping function in the RKHS; and $\varphi(\boldsymbol{u}) \in \mathbb{F}$ with high (or infinity) dimensionality. A commonly used kernel is the Gaussian kernel, i.e.,

$$k(\boldsymbol{u}, \boldsymbol{u}') = \exp(-\frac{\|\boldsymbol{u} - \boldsymbol{u}'\|^2}{2h^2}), \tag{2}$$

where $h > 0$ is a kernel bandwidth, and the notation $\|\cdot\|$ is the Euclidean norm in $\mathbb{U}$. The Gaussian kernel is unit-norm, i.e., $k(\boldsymbol{u}, \boldsymbol{u}) = 1$ for any sample $\boldsymbol{u} \in \mathbb{U}$.

It has been proved that, when the Gaussian kernel used, for any continuous input–output mapping $f: \mathbb{U} \rightarrow \mathbb{R}$ and $\forall \epsilon > 0$, there exist some useful sampled data $\{\boldsymbol{u}(\mathcal{C}_1), \boldsymbol{u}(\mathcal{C}_2, \ldots, \boldsymbol{u}(\mathcal{C}_m))\} \subseteq \mathbb{U}$ and kernel weights $\{\omega_1, \omega_2, \ldots, \omega_m\} \subseteq \mathbb{R}$ such that [35]

$$\|f(\cdot) - \boldsymbol{a}^T\varphi(\cdot)\|_{\boldsymbol{u}} = \|f(\cdot) - \sum_{l=1}^{m} \omega_l \langle \varphi(\boldsymbol{u}(\mathcal{C}_l)), \varphi(\cdot)\rangle\|_{\boldsymbol{u}} < \epsilon, \tag{3}$$

where $\boldsymbol{a} = \sum_{l=1}^{m} \omega_l \varphi(\boldsymbol{u}(\mathcal{C}_l))$ is a vector in $\mathbb{F}$; $\boldsymbol{u}(\mathcal{C}_l)$ denotes the l-th element of a dictionary $\boldsymbol{\mathcal{C}}(i)$ at discrete time i with m members, i.e., $\boldsymbol{\mathcal{C}}(i) = \{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_m\}$; and $\|\cdot\|_{\boldsymbol{u}}$ is a nonnegative functional in $\mathbb{U}$. Equation (3) implies that the linear model $f(\cdot) = \boldsymbol{a}^T\varphi(\cdot)$ has the universal approximation property [13]. Based on the property, a well-known representer theorem claims that

$$f(\cdot) = \sum_{l=1}^{m} \omega_l \langle \varphi(\boldsymbol{u}(\mathcal{C}_l)), \varphi(\cdot)\rangle = \sum_{l=1}^{m} \omega_l k(\boldsymbol{u}(\mathcal{C}_l), \cdot), \tag{4}$$

which is a theoretical basis of this paper.

2.2. SF-VLRKAF

As mentioned above, most of KAFs are feedforward networks, therefore, a KAF with feedback structure is proposed in this work. The linear recurrent kernel online learning algorithm with sparse updates (LRKOL) algorithm contains multiple same recurrent signals [3], however, our algorithm considers only single feedback information to get better filtering performance.

Fig. 1 shows the structure of the proposed kernel filter, where $\boldsymbol{u}(i) \in \mathbb{U} \subseteq R^{l\times 1}$ is the input column vector at discrete time i; $k_{li} = k(\boldsymbol{u}(\mathcal{C}_l), \boldsymbol{u}(i))$ is the Gaussian kernel; $\boldsymbol{\omega}(i) = [\omega_1(i), \omega_2(i), \ldots, \omega_m(i)]^T \in \mathbb{R}^{m\times 1}$ is the feedforward weight vector (FFW), $\gamma(i) \in \mathbb{R}$ is the feedback weight scalar (FBW); and $y(i) \in \mathbb{R}$ is the estimated output. Therefore, the relationship between the input and the output can be expressed as