2014 International Joint Conference on Neural Networks (IJCNN)
July 6-11, 2014, Beijing, China

Kernel Robust Mixed-Norm Adaptive Filtering

Jin Liu
School of Software Engineering
Xi'an Jiaotong University
Xi'an, China
xjtuliujn@gmail.com

Hua Qu, Badong Chen, Wentao Ma
School of Electronic and Information Engineering
Xi'an Jiaotong University
Xi'an, China
chenbd@mail.xjtu.edu.cn

***Abstract*—Kernel methods are powerful for developing a nonlinear learning algorithm in a high-dimensional linear space. The least mean square (LMS) and the least absolute deviation (LAD) are two well-known linear adaptive filtering algorithms. The former performs very well when the noise is Gaussian, while the later possesses desirable performance when the noise has a long-tailed distribution (e.g. alpha-stable distribution). The combination of the LMS and LAD yields a robust mixed-norm (RMN) algorithm. In this paper, we combine the popular kernel methods and the RMN algorithm to develop a new kernel adaptive filtering algorithm, namely the kernel RMN (KRMN) algorithm, which is a robust adaptive algorithm in reproducing kernel Hilbert space (RKHS). The mean square convergence is analyzed, and the excellent and robust performance of the new algorithm is demonstrated by the simulation results of nonlinear time series prediction.**

Keywords—kernel; mixed-norm; robust adaptive filtering.

I. INTRODUCTION

The kernel method is a powerful nonparametric modeling tool. It transforms the input data into a high-dimensional feature space via a reproducing kernel such that the inner products operation in the feature space can be computed efficiently through the kernel evaluations (so called the "kernel trick") [1]. Then, appropriate linear methods can subsequently be applied to the transformed data. Successful examples of kernel learning methods include support vector machines (SVM)[2, 5], kernel principal component analysis (KCPA), kernel least-mean-square (KLMS), etc. Efficient experiments of kernel methods are usually conducted in the environments corrupted by Gaussian noises. In practice, however, the data are in general non-Gaussian, and the noises with long-tailed distributions (e.g. alpha-stable noise) are common [3]. In the literature of linear adaptive filtering, the least absolute deviation (LAD) algorithm was proposed to deal with this problem. It has been shown that the LAD algorithm is much more robust to outliers than the conventional least mean square (LMS) algorithm, especially when the measurement noises have a long-tailed distribution. The robust mixed-norm (RMN) algorithm, with a cost function as a convex mixture of the mean-square error (MSE) and mean absolute error [4], is a linear combination of the LMS and LAD, which is often superior to either LMS or LAD [4]. The goal of this work is to extend the RMN algorithm to RKHS. The new algorithm is called the kernel robust mixed-norm (KRMN) algorithm, which performs well in the presence of impulse noises due to the robustness of the mixed-norm criterion [4,7,8].

This work was supported by the National Natural Science Foundation of China (61371807, 61372152), and the Key Project of major national science and technology on new generation of broadband wireless mobile communication network (2012ZX03001023-003, 2012ZX03001008-003, 2013ZX03002010 -003).

The rest of the paper is organized as follows. In section II, the kernel method and RMN algorithm are introduced. In Section III, the KRMN algorithm is derived. Then, the mean square convergence analysis is presented in section IV, and simulation results are given in section V. Finally, in section VI, we give the conclusion.

II. PRELIMINARY KNOWLEDGE

A. Kernel Method

A Mercer kernel [1,6] is a continuous, symmetric, positive definite function k: $X \times X \to R$, where X is the input domain, a compact subset of R^{L}. By Mercer theorem, any Mercer kernel can be expanded as follows:

$$k\left(x,x^{'}\right)=\sum_{i=1}^{\infty}\zeta_i\varphi_i(x)\varphi_i(x^{'}) \tag{1}$$

where ζ_i and φ_i are, respectively, the non-negative eigenvalue and eigenfunction. Therefore, a nonlinear mapping φ can be constructed as:

$$\varphi: X \to F$$

$$\varphi(x)=\left[\sqrt{\zeta_1}\varphi_1(x),\sqrt{\zeta_2}\varphi_2(x),\cdots\right] \tag{2}$$

which transforms the input data into a high-dimensional feature space F (essentially the same as the RKHS). By the construction of the mapping, the following equality holds (so called the "kernel trick"):

$$\varphi(x)^T\varphi(x')=k(x,x') \tag{3}$$

B. Robust Least Mean Mixed-Norm (RMN) Algorithm

The RMN algorithm is based on the minimization of the following convex combination of the error norms[4]:

$$J_{RMN}(i)=\lambda E(e^2(i))+(1-\lambda)E(|e(i)|) \tag{4}$$

where λ is the mixing parameter, $\lambda\in[0,1]$, and $e(i)$ is the error between the output of the linear adaptive filter and the desired response $d(i)$:

$$e(i)=d(i)-W^T(i-1)u(i) \tag{5}$$

where $W(i-1)$ is the estimated weight vector at iteration $i-1$, and $u(i)$ is the input vector. When $\lambda=1$, (4) becomes the mean

square error cost for the LMS algorithm, whereas when $\lambda = 0$, (4) becomes the mean absolute error for the LAD algorithm. With an appropriate λ, the mixed-norm provides a mechanism to mitigate the disturbance caused by the outliers . The gradient of the cost function J_{RMN} with respect to W is:

$$\nabla J_{RMN}(i) = -E[2\lambda e(i) + (1-\lambda)\operatorname{sign}(e(i))]u(i) \tag{6}$$

The RMN algorithm uses an instantaneous estimation of the gradient vector, which leads to the following update rule:

$$\begin{aligned} W(i+1) &= W(i) - \mu\nabla J_{RMN}(i) \\ &= W(i) + \mu[2\lambda e(i) + (1-\lambda)\operatorname{sign}(e(i))]u(i) \end{aligned} \tag{7}$$

where μ is the step size controlling the stability and convergence rate. Note that if $\lambda = 1$, the algorithm (7) reduces to the LMS algorithm; while if $\lambda = 0$, it becomes the LAD update rule.

III. KERNEL RMN ALGORITHM

It is assumed that the adaptive filter has a linear FIR structure in the RMN algorithm. Thus, poor performance can be expected if the underlying mapping between the desired output $d(i)$ and the input $u(i)$ is highly nonlinear. In order to overcome this limitation, the kernel robust mixed-norm (KRMN) adaptive filter algorithm is proposed in the following. This method uses the Mercer theorem to transform the data $u(i)$ into RKHS as $\varphi(u(i))$, and $\Omega^T\varphi(i)$ is a much more powerful model than $W^T u(i)$ because of the difference in the dimensions of $u(i)$ and $\varphi(i) = \varphi(u(i))$, where Ω denotes the weight vector in RKHS. We use the RMN algorithm on the new example sequence $\{\varphi(i), d(i)\}$:

$$\begin{cases} \Omega(0) = 0 \\ e(i) = d(i) - \Omega^T(i-1)\varphi(i) \\ \Omega(i) = \Omega(i-1) + \mu[2\lambda e(i) + (1-\lambda)\operatorname{sign}(e(i))]\varphi(i) \end{cases} \tag{8}$$

where $\Omega(i)$ denotes the estimate (at iteration i) of the weight vector in RKHS.

However, the dimensionality of $\varphi(\cdot)$ is high and it is only implicitly known, so we need an alternative way of carrying out the computation. The repeated application of the weight-update equation through iterations yields:

$$\begin{aligned} \Omega(i) &= \Omega(i-1) + \mu[2\lambda e(i) + (1-\lambda)\operatorname{sign}(e(i))]\varphi(i) \\ &= [\Omega(i-2) + \mu[2\lambda e(i-1) + (1-\lambda)\operatorname{sign}(e(i-1))]\varphi(i-1)] \\ &\quad + \mu[2\lambda e(i) + (1-\lambda)sign(e(i))]\varphi(i) \\ &= \Omega(i-2) + \mu[[2\lambda e(i-1) + (1-\lambda)\operatorname{sign}(e(i-1))] \\ &\quad + [2\lambda e(i) + (1-\lambda)\operatorname{sign}(e(i))]\varphi(i)] \\ &= \Omega(0) + \mu\sum_{j=1}^{i}(2\lambda e(j) + (1-\lambda)\operatorname{sign}(e(j))]\varphi(j) \end{aligned} \tag{9}$$

Setting $\Omega(0) = 0$, we have

$$\Omega(i) = \mu\sum_{j=1}^{i}(2\lambda e(j) + (1-\lambda)\operatorname{sign}(e(j))]\varphi(j) \tag{10}$$

Thus, the output of the system to a new input $u(i)$ can be expressed in terms of inner products between transformed inputs:

$$\begin{aligned} y(i) &= \Omega(i)^T\varphi(i) \\ &= \mu\sum_{j=1}^{i}(2\lambda e(j) + (1-\lambda)\operatorname{sign}(e(j))]\varphi(j)^T\varphi(i) \\ &= \mu\sum_{j=1}^{i}(2\lambda e(j) + (1-\lambda)\operatorname{sign}(e(j))\kappa(u(j), u(i)) \end{aligned} \tag{11}$$

According to the description above, the KRMN algorithm can be summarized as follows:

Algorithm. The kernel RMN algorithm

Initialization

Choose step-size, kernel function κ

$$\alpha_1 = \mu[2\lambda d(1) + (1-\lambda)\operatorname{sign}(d(1))], C(1) = \{u(1)\}$$

Computation

while $\{u(i), d(i)\}$ available **do**

%Compute the output

$$y(i) = \sum_{j=1}^{i}\alpha_j\kappa(u(j), u(i))$$

%Compute the error

$$e(i) = d(i) - y(i)$$

%Compute the coefficient

$$\alpha_i = \mu[2\lambda e(i) + (1-\lambda)\operatorname{sign}(e(i))]$$

%Update the dictionary

$$C(i) = \{C(i-1), u(i)\}$$

end while

KRMN algorithm is simple, but we need to pay attention to two aspects that are still unspecified. The first one is how to select a proper kernel function k ; and the second one is how to select a good mixing parameter λ. For the first one, it is well-known that Gaussian kernel induces a RKHS with universal approximating capability [1], which is usually a default choice in kernel adaptive filtering. The Gaussian kernel is defined by

$$k(u, v) = \exp(-l \mid u - v \mid^2) \tag{12}$$

where $l > 0$ is the kernel parameter. After choosing the kernel function, the next thing is to select a suitable mixing parameter λ. This parameter can be chosen in two different ways. The first way is to set λ at a constant value, which is fixed during the learning. The second way is to update λ with an adaptive method. In this work, we use a sliding window approach to get a threshold that determines when λ will change [9]. The threshold is set at the mean squared error within a window:

$$\tau = \frac{2\sum_{i=1}^{n} e_i^2}{n} \tag{13}$$

where n denotes the window length, $\{e_i\}$ are the errors in this window. If the new error is larger than the threshold, we reduce the λ value such that the KLAD algorithm occupies a main position. The robustness of the KLAD makes the KRMN stable. If the new error is smaller than the threshold, we make the λ close to 1, and hence the KLMS [6] algorithm occupies a main position.

IV. Mean Square Convergence Analysis

The mean square convergence behavior is another key aspect of the kernel adaptive filters. For classical linear adaptive filters, the convergence analysis has been extensively studied [10]. In this direction, we mention here the works by Al-Naffouri and Sayed [11, 13], whose approach is based on the fundamental energy conservation relation (ECR). Recently, this relation has been extended into RKHS for analyzing the mean-square convergence performance of the kernel adaptive filters [12, 14]. In this section, we study the mean-square convergence performance of the proposed KRMN algorithm, based on the ECR given in *Lemma 1*.

Lemma 1 [12, 14]. Consider the nonlinear system identification case in which $d(i) = \Omega^* \varphi_i + v(i)$, where Ω^* denotes the unknown weight vector (in RKHS) that needs to be estimated, and $v(i)$ stands for the disturbance noise. Define the weight error vector $\tilde{\Omega}(i) = \Omega^*(i) - \Omega(i)$, *a priori* error $e_a(i) = \tilde{\Omega}(i-1)^T \varphi(i)$, and *a posterior* error $e_p(i) = \tilde{\Omega}(i)^T \varphi(i)$. Then the following equality holds:

$$\left\|\tilde{\Omega}(i)\right\|_F^2 + e_a^2(i) = \left\|\tilde{\Omega}(i-1)\right\|_F^2 + e_p^2(i) \tag{14}$$

Based on *Lemma 1*, we can easily derive:

$$e_p(i) = e_a(i) - \mu[2\lambda e(i) + (1-\lambda)\operatorname{sign}(e(i))]\kappa(u(i), u(i)) \tag{15}$$

Substituting (15) into (14) and after some straightforward manipulations, we have:

$$\begin{aligned} &E[\left\|\tilde{\Omega}(i)\right\|_F^2] = E[\left\|\tilde{\Omega}(i-1)\right\|_F^2] \\ &-2\mu E[e_a(i)(2\lambda e(i) + (1-\lambda)\operatorname{sign}(e(i)))] \\ &+\mu^2 E[(2\lambda e(i) + (1-\lambda)\operatorname{sign}(e(i)))^2] \end{aligned} \tag{16}$$

Here, we assume that the noise $v(i)$ is zero-mean, independent, identically distributed, and independent of the *a priori* estimation error $e_a(i)$. This assumption is commonly used in the convergence analysis of most adaptive filtering algorithms. Under this assumption, we can derive

$$\begin{aligned} &E[\left\|\tilde{\Omega}(i)\right\|_F^2] = E[\left\|\tilde{\Omega}(i-1)\right\|_F^2] - 2\mu E[e_a(i)(2\lambda e(i) + (1-\lambda)\operatorname{sign}(e(i)))] \\ &+\mu^2 E[(2\lambda e(i) + (1-\lambda)\operatorname{sign}(e(i)))^2] \\ &\quad = E[\left\|\tilde{\Omega}(i-1)\right\|_F^2] - 2\mu E[2\lambda e_a^2(i) + e_a(i)(1-\lambda)\operatorname{sign}(e(i)))] \\ &+\mu^2 E[(4\lambda^2 e^2(i) + 2\lambda e(i)(1-\lambda)\operatorname{sign}(e(i)) + ((1-\lambda)\operatorname{sign}(e(i)))^2] \\ &\quad = E[\left\|\tilde{\Omega}(i-1)\right\|_F^2] - 2\mu E[2\lambda e_a^2(i) + e_a(i)(1-\lambda)\operatorname{sign}(e(i)))] \\ &+\mu^2 E[(4\lambda^2 e^2(i) + 2\lambda e(i)(1-\lambda)\operatorname{sign}(e(i)) + ((1-\lambda)\operatorname{sign}(e(i)))^2] \\ &\quad = E[\left\|\tilde{\Omega}(i-1)\right\|_F^2] - 2\mu E[2\lambda e_a^2(i) + \beta e_a(i)] \\ &+\mu^2 E[(4\lambda^2 e^2(i) + 2\lambda\beta e(i) + \beta^2] \\ &\quad = E[\left\|\tilde{\Omega}(i-1)\right\|_F^2] - 2\mu E[2\lambda e_a^2(i) + \beta e_a(i)] \\ &+\mu^2 (E[(4\lambda^2 e_a^2(i) + 2\lambda\beta e_a(i) + \beta^2] + 4\lambda^2\xi^2) \end{aligned} \tag{17}$$

where $\beta = (1-\lambda)\operatorname{sign}(e(i)))$, and ξ^2 denotes the noise variance. From the (17), we observe that:

$$\begin{aligned} &E\left[\left\|\tilde{\Omega}(i)\right\|_F^2\right] \le E\left[\left\|\tilde{\Omega}(i-1)\right\|_F^2\right] \Leftrightarrow -2\mu E[2\lambda e_a^2(i) + \beta e_a(i)] \\ &+\mu^2 (E[(4\lambda^2 e_a^2(i) + 2\lambda\beta e_a(i) + \beta^2] + 4\lambda^2\xi^2) \le 0 \\ &\Leftrightarrow \mu \le \frac{2E[2\lambda e_a^2(i) + \beta e_a(i)]}{(E[(4\lambda^2 e_a^2(i) + 2\lambda\beta e_a(i) + \beta^2] + 4\lambda^2\xi^2)} \end{aligned} \tag{18}$$

Therefore, if the step-size satisfies (18), the sequence of the weight error power in F will be monotonically decreased (and hence convergent). From (18), an upper bound for the step-size can be derived as follow:

$$\begin{aligned} \mu &\le \frac{2E[2\lambda e_a^2(i) + \beta e_a(i)]}{(E[(4\lambda^2 e_a^2(i) + 2\lambda\beta e_a(i) + \beta^2] + 4\lambda^2\xi^2)} \\ &\le \frac{E[2\lambda e_a^2(i)]}{(E[(4\lambda^2 e_a^2(i) + \beta^2] + 4\lambda^2\xi^2)} \\ &\overset{a}{\le} \sup_{E(e_a^2)\in\varepsilon} \frac{2\lambda E[\| \tilde{\Omega} \|_F^2]}{(E[(4\lambda^2 \| \tilde{\Omega} \|_F^2 + \beta^2] + 4\lambda^2\xi^2)} \\ &\overset{b}{=} \sup_{E[\|\Omega^*\|_F^2]\in\psi} \frac{2\lambda E[\| \Omega^* \|_F^2]}{(E[(4\lambda^2 \| \Omega^* \|_F^2 + \beta^2] + 4\lambda^2\xi^2)} \end{aligned} \tag{19}$$

where ε, ψ denote, respectively, the feasible set of $E[e_a^2(i)]$ and $E[\| \tilde{\Omega} \|_F^2]$.

V. Simulation Results

For evaluating the performance of the proposed KRMN algorithm, we conduct Monte Carlo simulations to demonstrate its performance in Mackey-Glass chaotic time series prediction. A segment of 1000 samples is used for training, and another 100 data are used for testing. The data are corrupted by additive noises with a distribution modeled as a mixture of the Gaussian (with zero mean) and a long-tailed alpha-stable distribution. In the simulations, we compare the performance of KLMS, KLAD, and KRMN algorithms. According to the selection methods of the mixing parameter λ, we define two kinds of algorithm as KRMN with fixed λ and KRMN with adaptive λ.

We show the convergence behaviors of the algorithms. In the simulation, the step sizes for the KLMS, KLAD, KRMN with fixed λ, and KRMN with adaptive λ are set at 0.1, 0.01, 0.05, and 0.07, respectively. The fixed λ for KRMN is set at 0.6, and the window length for the adaptive λ is set at 10. The convergence curves for all the algorithms are shown in Fig.1. One can see clearly that although the steady-state testing MSEs of the proposed algorithms are almost the same as the KLMS, their convergence speeds are much faster. In addition, the convergence curves of the two KRMN algorithms are relatively smooth, while the learning curve of the KLMS has large fluctuation especially when the outliers happen. The KLAD algorithm also converges smoothly, but its convergence speed is very low.

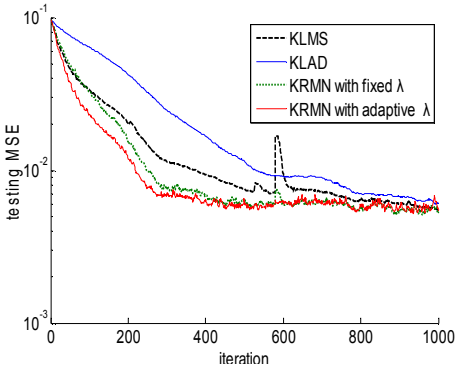

Fig.1. Convergence curves of different algorithms

VI. CONCLUSION

In this work, a new kernel adaptive filtering algorithm, namely the kernel robust mixed-norm (KRMN) algorithm, is developed, which is a linear combination of the KLMS and KLAD algorithms. The new algorithm is computationally simple, and robust to outliers. Simulation results show that the KRMN algorithm can outperform both KLMS and KLAD if properly choosing the mixing parameter.

REFERENCES

[1] T. Hofmann, B. Schölkopf, A.J. Smola. "Kernel methods in machine learning," The annals of statistics, vol.36, no.3, pp.1171-1220, Jul. 2008.

[2] C.J.C. Burges, "A tutorial on support vector machines for pattern recognition", Data Min. Knowl. Discov, vol.2, no. 2, pp. 121-167, 1998.

[3] P.G. Georgiou, P. Tsakalides, C. Kyriakakis. "Alpha-stable modeling of noise and robust time-delay estimation in the presence of impulsive noise". Multimedia, IEEE Transactions on, 1(3): 291-301, 1999.

[4] J. Chambers, A. Avlonitis. "A robust mixed-norm adaptive filter algorithm". Signal Processing Letters, IEEE, 4 (2): 46-48, 1997.

[5] S.S. Keerthi, C J Lin. "Asymptotic behaviors of support vector machines with Gaussian kernel". Neural computation, 15(7): 1667-1689, 2003.

[6] W. Liu, P. Pokharel, and J. C. Principe, "The kernel least mean square algorithm," IEEE Trans. on Signal Process., vol. 56, no. 2, pp. 543-554, 2008.

[7] J.A. Chambers, O. Tanrikulu, A.G. Constantinides. "Least mean mixed-norm adaptive filtering". Electronics letters, 30 (19): 1574-1575, 1994.

[8] V.J. Mathews and S.H. Cho, "Improved convergence analysis of stochastic gradient adaptive filters using the sign algorithm," IEEE Trans. Acoust, Speech, Signal Processing, vol. ASSP-35, no. 4, pp. 450–454, 1987.

[9] Y. Chi, H. Wang, P. S. Yu, R. R. Muntz "Moment: Maintaining closed frequent item setsover a stream sliding window," in 2004 Proc. Fourth IEEE International Conference on Data Mining, pp.59-66.

[10] A.H. Sayed, "Fundamentals of adaptive filtering."John Wiley & Sons, 2003.

[11] T.Y. Al-Naffouri, A.H. Sayed, "Transient analysis of data-normalized adaptive filters,"IEEE Transactions on Signal Processing 51 (2003) 639–652.

[12] B. Chen, S. Zhao, P. Zhu, J.C. Principe, "Mean square convergence analysis for kernel least mean square algorithm". Signal Processing, 92 (2012) 2624–2632.

[13] T.Y. Al-Naffouri, A.H. Sayed, "Adaptive filters with error nonlinearities: mean-square analysis and optimum design," EURASIP Journal on Applied Signal Processing 4 (2001) ,pp.192–205. .

[14] B. Chen, S. Zhao, P. Zhu, J.C. Principe."Quantized kernel least mean square algorithm". IEEE Transactions on Neural Networks and Learning Systems, vol. 23, no.1, 22-32, 2012.