entropy

Article

Recursive Minimum Complex Kernel Risk-Sensitive Loss Algorithm

Guobing Qian [1,2,*], **Dan Luo** [1] **and Shiyuan Wang** [1,*]

[1] College of Electronic and Information Engineering, Brain-inspired Computing & Intelligent Control of Chongqing Key Laboratory, Chongqing Key Laboratory of Nonlinear Circuits and Intelligent Information Processing, Southwest University, Chongqing 400715, China; luodan_2018@163.com

[2] School of Mathematics and Statistics, Southwest University, Chongqing 400715, China

[*] Correspondence: guobingqian@swu.edu.cn (G.Q.); wsy@swu.edu.cn (S.W.)

Received: 4 November 2018; Accepted: 23 November 2018; Published: 25 November 2018

Abstract: The maximum complex correntropy criterion (MCCC) has been extended to complex domain for dealing with complex-valued data in the presence of impulsive noise. Compared with the correntropy based loss, a kernel risk-sensitive loss (KRSL) defined in kernel space has demonstrated a superior performance surface in the complex domain. However, there is no report regarding the recursive KRSL algorithm in the complex domain. Therefore, in this paper we propose a recursive complex KRSL algorithm called the recursive minimum complex kernel risk-sensitive loss (RMCKRSL). In addition, we analyze its stability and obtain the theoretical value of the excess mean square error (EMSE), which are both supported by simulations. Simulation results verify that the proposed RMCKRSL out-performs the MCCC, generalized MCCC (GMCCC), and traditional recursive least squares (RLS).

Keywords: complex; kernel risk-sensitive loss; recursive; stability; EMSE

1. Introduction

As many noises are non-Gaussian distributed in practice, the performance of traditional second-order statistics-based similarity measures may deteriorate dramatically [1,2]. To efficiently handle the non-Gaussian noise, a higher order statistic called correntropy [3–6] was proposed. The correntropy is a nonlinear and local similarity measure widely used in adaptive filters [7–15], and usually employs a Gaussian function as the kernel function thanks to flexible and positive definiteness. However, the Gaussian kernel is not always the best choice [16]. Hence, Chen et al. proposed the generalized maximum correntropy criterion (GMCC) algorithm [16,17] using a generalized Gaussian density function as the kernel. Compared with traditional maximum correntropy criterion (MCC), the GMCC behaves better when the shape parameter is properly selected. In addition, the MCC can be regarded as a special case of GMCC. Considering that the error performance surface of correntropic loss is highly non-convex, Chen et al. proposed another algorithm named the minimum kernel risk-sensitive loss (MKRSL), which is defined in kernel space but also inherits the original form of risk-sensitive loss (RSL) [18,19]. The performance surface of the kernel risk-sensitive loss (KRSL) is more efficient than the MCC, resulting in a faster convergence speed and a higher accuracy. Furthermore, KRSL is also not sensitive to outliers.

Generally, adaptive filter has been mainly focused on the real domain and cannot be used to deal with complex-valued data directly. Recently, the complex domain adaptive filter has drawn more attention. Guimaraes et al. proposed the maximum complex correntropy criterion (MCCC) [20,21] and provided a probabilistic interpretation [20]. MCCC shows an obvious advantage over the least absolute deviation (LAD) [22], complex least mean square (CLMS) [23], and recursive least squares

(RLS) algorithms [24]. The stability analysis and the theoretical EMSE of the MCCC have been derived [25]. The MCCC has been extended to the generalized case [26]. The generalized MCCC (GMCCC) algorithm employs a complex generalized Gaussian density as a kernel and offers a desirable performance for handling the complex-valued data. In addition, a gradient-based complex kernel risk-sensitive loss (CKRSL) defined in kernel space has shown a superior performance [27]. Until now, there has been no report about the recursive complex KRSL (CKRSL) algorithm. Therefore, in this paper we first propose a recursive minimum CKRSL (RMCKRSL) algorithm. Then, we analyze the stability and calculate the theoretical value of the EMSE. Simulations show that the RMCKRSL is better than the MCCC, GMCCC, and traditional RLS. Simultaneously, the correctness of the theoretical analysis is also demonstrated by simulations.

The remaining parts of this paper are organized as follows: In Section 2, we provide the loss function of the CKRSL and propose the recursive MCKRSL algorithm. In Section 3 we analyze the stability and obtain the theoretical value of the EMSE for the proposed algorithm. In Section 4, simulations are performed to verify the superior convergence of the RMCKRSL algorithm and the correctness of the theoretical analysis. Finally, in Section 5 we draw a conclusion.

2. Fixed Point Algorithm under Minimizing Complex Kernel Risk-Sensitive Loss

2.1. Complex Kernel Risk-Sensitive Loss

Supposing there are two complex variables $C_1 = X_1 + jY_1$ and $C_2 = X_2 + jY_2$, the complex kernel risk-sensitive loss (CKRSL) is defined as [27]:

$$L_\lambda^c(C_1, C_2) = \frac{1}{\lambda} E[\exp[\lambda(1 - \kappa_\sigma^c(C_1 - C_2))]] \tag{1}$$

where X_1, X_2, Y_1 and Y_2 are real variables, λ is the risk-sensitive parameter, and $\kappa_\sigma^c(C_1 - C_2)$ is the kernel function.

This paper employs a Gaussian kernel which is expressed as:

$$\kappa_\sigma^c(C_1 - C_2) = \exp\left(-\frac{(C_1 - C_2)(C_1 - C_2)^*}{2\sigma^2}\right) \tag{2}$$

where σ is the kernel width.

2.2. Recursive Minimum Complex Kernel Risk-Sensitive Loss (RMCKRSL)

2.2.1. Cost Function

We define the cost function of MCKRSL as:

$$\begin{aligned} J_{MCKRSL} &= E\left[L_\lambda^c(e(k))\right] \\ &= \tfrac{1}{\lambda} E[\exp[\lambda(1 - \kappa_\sigma^c(e(k)))]] \end{aligned} \tag{3}$$

where

$$e(k) = d(k) - \mathbf{w}^H \mathbf{x}(k) \tag{4}$$

denotes the error at the k^{th} iteration, $d(k)$ represents the expected response at the k^{th} iteration, $\mathbf{w} = \left[\begin{array}{cccc} w_1 & w_2 & \cdots & w_m \end{array}\right]$ denotes the estimated weight vector, m is the length of adaptive filter, and $\mathbf{x}(k) = [x(k) \;\; x(k-1) \;\; \cdots \;\; x(k-m+1) \;\;]^T$ is the input vector, $(\cdot)^H$ and $(\cdot)^T$ denote the conjugate transpose and transpose, respectively.

2.2.2. Recursive Solution

Using the Wirtinger Calculus [28,29], the gradient of J_{MCKRSL} with respect to $\mathbf{w}^*$ is derived:

$$\frac{\partial J_{MCKRSL}}{\partial \mathbf{w}^*} = E\left[L_\lambda^c(e) \times \left(-d^*\mathbf{x} + \mathbf{x}\mathbf{x}^H\mathbf{w}\right)\right] \tag{5}$$

By making $\frac{\partial J_{MCKRSL}}{\partial \mathbf{w}^*} = 0$, we obtain the optimal solution

$$\mathbf{w} = \mathbf{R}^{-1}\mathbf{p} \tag{6}$$

where

$$\mathbf{R} = E\left\{h(e)\mathbf{x}\mathbf{x}^H\right\} \tag{7}$$

$$\mathbf{p} = E\{h(e)d^*\mathbf{x}\} \tag{8}$$

$$h[e] = \exp[\lambda(1 - \kappa_\sigma^c(e))]\kappa_\sigma^c(e) \tag{9}$$

It is noted that Equation (6) is actually a fixed point solution because $\mathbf{R}$ and $\mathbf{p}$ depend on $\mathbf{w}$. In practice, $\mathbf{R}$ and $\mathbf{p}$ are usually estimated as follows when the samples are finite:

$$\hat{\mathbf{R}} = \frac{1}{N}\sum_{k=1}^{N} h[e(k)]\mathbf{x}(k)\mathbf{x}^H(k) \tag{10}$$

$$\hat{\mathbf{p}} = \frac{1}{N}\sum_{k=1}^{N} h[e(k)]d^*(k)\mathbf{x}(k) \tag{11}$$

Hence, $\hat{\mathbf{R}}$, $\hat{\mathbf{p}}$ and $\mathbf{w}$ are updated as follows:

$$\begin{array}{c} \hat{\mathbf{R}}_k = \hat{\mathbf{R}}_{k-1} + h[e(k)]\mathbf{x}(k)\mathbf{x}^H(k) \\ \hat{\mathbf{p}}_k = \hat{\mathbf{p}}_{k-1} + h[e(k)]d^*(k)\mathbf{x}(k) \\ \mathbf{w}(k) = \hat{\mathbf{R}}_k{}^{-1}\,\hat{\mathbf{p}}_k \end{array} \tag{12}$$

Using the matrix inversion lemma [30], we may rewrite $\hat{\mathbf{R}}_k^{-1}$ in Equation (12) as:

$$\hat{\mathbf{R}}_k^{-1} = \hat{\mathbf{R}}_{k-1}^{-1} - \hat{\mathbf{R}}_{k-1}^{-1}\mathbf{x}(k)\left(h^{-1}(e(k)) + \mathbf{x}^H(k)\hat{\mathbf{R}}_{k-1}^{-1}\mathbf{x}(k)\right)^{-1} \times \mathbf{x}^H(k)\hat{\mathbf{R}}_{k-1}^{-1} \tag{13}$$

$$\mathbf{b}(k) = \hat{\mathbf{R}}_{k-1}^{-1}\mathbf{x}(k)\left(h^{-1}(e(k)) + \mathbf{x}^H(k)\hat{\mathbf{R}}_{k-1}^{-1}\mathbf{x}(k)\right)^{-1} \tag{14}$$

$$\left(h^{-1}(e(k)) + \mathbf{x}^H(k)\hat{\mathbf{R}}_{k-1}^{-1}\mathbf{x}(k)\right)\mathbf{b}(k) = \hat{\mathbf{R}}_{k-1}^{-1}\mathbf{x}(k) \tag{15}$$

and

$$\begin{aligned} \mathbf{b}(k) &= h(e(k))\hat{\mathbf{R}}_{k-1}^{-1}\mathbf{x}(k) \\ &\quad -\mathbf{b}(k)h(e(k))\mathbf{x}^H(k)\hat{\mathbf{R}}_{k-1}^{-1}\mathbf{x}(k) \\ &= h(e(k))\left(\hat{\mathbf{R}}_{k-1}^{-1} - \mathbf{b}(k)\mathbf{x}^H(k)\hat{\mathbf{R}}_{k-1}^{-1}\right)\mathbf{x}(k) \\ &= h(e(k))\hat{\mathbf{R}}_k^{-1}\mathbf{x}(k) \end{aligned} \tag{16}$$

After some algebraic manipulations, we may derive the recursive form of $\mathbf{w}(k)$ as follows:

$$\begin{aligned} \mathbf{w}(k) &= \hat{\mathbf{R}}_k{}^{-1}\left\{h(e(k))d^*(k)\mathbf{x}(k) + \hat{\mathbf{p}}_{k-1}\right\} \\ &= \mathbf{w}(k-1) + \mathbf{b}(k)e^*(k) \end{aligned} \tag{17}$$

Finally, Algorithm 1 summarizes the recursive MCKRSL (RMCKRSL) algorithm.

Algorithm 1: RMCKRSL.

```
Input: $\sigma$, $\lambda$, $d(k)$, $\mathbf{x}(k)$
1. Initializations: $\delta = 0.0001$, $\mathbf{p}_0 = 0$, $\mathbf{w}(0) = 0$, $\mathbf{R}_0 = \delta\mathbf{I}$, $\mathbf{R}_0^{-1} = \delta^{-1}\mathbf{I}$
2. While $\left\{ \mathbf{x}(k) \quad d(k) \right\}$ available, do
3.   $e(k) = d(k) - \mathbf{w}^H(k)\mathbf{x}(k)$
4.   $\kappa_\sigma^c(e(k)) = \exp\left(-|e(k)|^2/2\sigma^2\right)$
5.   $h[e(k)] = \exp[\lambda(1 - \kappa_\sigma^c(e(k)))]\kappa_\sigma^c(e(k))$
6.   $\mathbf{b}(k) = \hat{\mathbf{R}}_{k-1}^{-1}\mathbf{x}(k)\left(h^{-1}(e(k)) + \mathbf{x}^H(k)\hat{\mathbf{R}}_{k-1}^{-1}\mathbf{x}(k)\right)^{-1}$
7.   $\mathbf{w}(k) = \mathbf{w}(k-1) + \mathbf{b}(k)e^*(k)$
8.   $\hat{\mathbf{R}}_k^{-1} = \hat{\mathbf{R}}_{k-1}^{-1} - \hat{\mathbf{R}}_{k-1}^{-1}\mathbf{x}(k)\left(h^{-1}(e(k)) + \mathbf{x}^H(k)\hat{\mathbf{R}}_{k-1}^{-1}\mathbf{x}(k)\right)^{-1} \times \mathbf{x}^H(k)\hat{\mathbf{R}}_{k-1}^{-1}$
9. End while
10. $\hat{\mathbf{w}}_0 = \mathbf{w}(k)$
Output: Estimated filter weight $\hat{\mathbf{w}}_0$
```

3. Convergence Analysis

3.1. Stability Analysis

Supposing the desired signal is as follows:

$$d(k) = \mathbf{w}_0^H\mathbf{x}(k) + v(k) \tag{18}$$

we rewrite the error as:

$$\begin{aligned} e(k) &= \widetilde{\mathbf{w}}^H(k-1)\mathbf{x}(k) + v(k) \\ &= e_a(k) + v(k) \end{aligned} \tag{19}$$

where $\mathbf{w}_0$ is the system parameter to be estimated, $\widetilde{\mathbf{w}}(k-1) = \mathbf{w}_0 - \mathbf{w}(k-1)$, $v(k)$ represents the noise at discrete time k, and $e(k) = \widetilde{\mathbf{w}}^H(k-1)\mathbf{x}(k)$.

Furthermore, we rewrite $\mathbf{w}(k)$ as:

$$\begin{aligned} \mathbf{w}(k) &= \mathbf{w}(k-1) + h(e(k))e^*(k)\hat{\mathbf{R}}_k^{-1}\mathbf{x}(k) \\ &\approx \mathbf{w}(k-1) + \tfrac{a_0}{k}f(e(k))\bar{\mathbf{R}}^{-1}\mathbf{x}(k) \end{aligned} \tag{20}$$

where $f(e(k)) = h(e(k))e^*(k)$, $\bar{\mathbf{R}} = E\left\{\mathbf{x}\mathbf{x}^H\right\}$, a_0 =$1/E[\exp[\lambda(1-\kappa_\sigma^c(v))]\kappa_\sigma^c(v)]$, v represents the noise, and the second line is approximately obtained by using the following:

$$\begin{aligned} \hat{\mathbf{R}}_k^{-1} &= \left[\sum_{l=1}^{k} h[e(l)]x(l)x^H(l)\right]^{-1} \\ &= \left[k\left[\tfrac{1}{k}\sum_{l=1}^{k} h[e(l)]x(l)x^H(l)\right]\right]^{-1} \approx \tfrac{a_0}{k}\bar{\mathbf{R}}^{-1} \end{aligned} \tag{21}$$

Remark 1.

(1) *We can approximate the second line of Equation (20) when* $|e_a(l)|^2$ *is small enough, where* $e_a(l) = (\mathbf{w}_0 - \mathbf{w}(l-1))^H\mathbf{x}(l)$.

(2) *According to Equation (20), the RMCKRSL can be approximately viewed as a gradient descend method with variable steps a_0/k.*

(3) *We can estimate* $\bar{\mathbf{R}}$ *by* $\frac{1}{N}\sum\limits_{l=1}^{N}\mathbf{x}(l)\mathbf{x}^H(l)$, *where* $\mathbf{N}$ is the number of samples.

By multiplying $\bar{\mathbf{R}}^{1/2}$ on both sides of Equation (20), we obtain the following:

$$\bar{\mathbf{R}}^{1/2}\widetilde{\mathbf{w}}(k) = \bar{\mathbf{R}}^{1/2}\widetilde{\mathbf{w}}(k-1) - \frac{a_0}{k} f(e(k))\bar{\mathbf{R}}^{-1/2}\mathbf{x}(k) \tag{22}$$

where $f(e(k)) = \exp[\lambda(1-\kappa_\sigma^c(e(k)))]\kappa_\sigma^c(e(k))e^*(k)$.

Therefore,

$$\begin{aligned} &E\left\{\|\bar{\mathbf{R}}^{1/2}\widetilde{\mathbf{w}}(k)\|^2\right\} \\ &= E\left\{\|\bar{\mathbf{R}}^{1/2}\widetilde{\mathbf{w}}(k-1)\|^2\right\} - \tfrac{2a_0}{k}E\{\mathrm{Re}[e_a(k)f(e(k))]\} \\ &\quad +\tfrac{a_0{}^2}{k^2}E\left\{\|\mathbf{I}_m\|_F^2|f(e(k))|^2\right\} \end{aligned} \tag{23}$$

where $\|\cdot\|_F$ represents the Frobenius norm, $\mathrm{Re}(\cdot)$ is the real part, and $\mathbf{I}_m$ denotes the $m \times m$ identity matrix.

Then, we can determine that if

$$k \geq \frac{m\, a_0 E\left\{|f(e(k))|^2\right\}}{2E\{\mathrm{Re}[e_a(k)f(e(k))]\}} \tag{24}$$

the sequence $E\left\{\|\widetilde{\mathbf{w}}(k)\|^2\right\}$ is decreasing and the algorithm will converge.

3.2. Excess Mean Square Error

Let $S(k)$ be the excess mean square error (EMSE) and defined as:

$$S(k) = E\left[|e_a(k)|^2\right] \tag{25}$$

To derive the theoretical value of $S(k)$, we adopt some commonly used assumptions [8,27,31]:

(A1) $v(k)$ is zero-mean and independently identically distributed (IID); $e_a(k)$ is independent of $v(k)$ and also zero-mean;

(A2) $\mathbf{x}(k)$ is independent of $v(k)$, circular and stationary.

Thus, taking (23) and (25) into consideration, we obtain the following:

$$\begin{aligned} S(k+1) &= S(k) - \tfrac{2a_0}{k}E\{\mathrm{Re}[e_a(k)f(e(k))]\} \\ &\quad +\tfrac{ma_0{}^2}{k^2}E\left\{|f(e(k))|^2\right\} \end{aligned} \tag{26}$$

Similar to [27], we can obtain the following:

$$\begin{aligned} E\left\{|f(e(k))|^2\right\} &= E\left\{\exp[2\lambda(1-\kappa_\sigma^c(v))]\kappa_{\sigma/\sqrt{2}}^c(v)|v|^2\right\} \\ +S(k)\ &\times E\left\{\exp[-2\lambda(\kappa_\sigma^c(v)-1)]\kappa_{\sigma/\sqrt{2}}^c(v)R_1\right\} \end{aligned} \tag{27}$$

$$E\{\mathrm{Re}[e_a(k)f(e(k))]\} = S(k)E\{\exp[\lambda(1-\kappa_\sigma^c(v))]R_2\} \tag{28}$$

where

$$\begin{aligned} R_1 \quad &= \Big[1 - 3\tfrac{|v|^2}{\sigma^2} + \tfrac{|v|^4}{\sigma^4} + 3\lambda\tfrac{|v|^2}{\sigma^2}\kappa_\sigma^c(v) \\ &\quad -5\lambda\tfrac{|v|^4}{2\sigma^4}\kappa_\sigma^c(v) + \tfrac{|v|^4}{\sigma^4}\lambda^2\kappa_{\sigma/\sqrt{2}}^c(v)\Big] \end{aligned} \tag{29}$$

$$R_2 = \kappa_\sigma^c(v)\left(1 + \frac{1}{2\sigma^2}(\lambda\kappa_\sigma^c(v) - 1)|v|^2\right) \tag{30}$$

Thus,

$$S(k+1) = S(k) - \frac{2a_0a_1}{k}S(k) + \frac{a_0{}^2a_2}{k^2}S(k) + \frac{a_0{}^2a_3}{k^2} \tag{31}$$

where $a_1 = E\{\exp[\lambda(1-\kappa_\sigma^c(v))]R_2\}$, $a_2 = mE\left\{\exp[-2\lambda(\kappa_\sigma^c(v)-1)]\kappa_{\sigma/\sqrt{2}}^c(v)R_1\right\}$, $a_3 = mE\left\{\exp[2\lambda(1-\kappa_\sigma^c(v))]\kappa_{\sigma/\sqrt{2}}^c(v)|v|^2\right\}$.

It can be seen from Equation (31), that $S(k)$ is the solution to a first-order difference equation. Thus, we derive that:

$$S(k) = S_h(k) + S_p(k) \tag{32}$$

where $S_h(k) = c_1\lambda(k-1)\lambda(k-2)\cdots\lambda(1)$ is the homogeneous solution with $\lambda(k-1) = 1 - 2a_0a_1/(k-1) + {a_0}^2a_2/(k-1)^2$, and $S_p(k)$ is the particular solution where:

$$S_p(k) = \left(\frac{a_0a_3}{k}c_2\right)/\left(2a_1 - \frac{a_0a_2}{k}c_2\right) \tag{33}$$

and $c_2 \approx \frac{2a_0a_1}{2a_0a_1-[k/(k+1)]}$.

Remark 2. *The theoretical value of $S(k)$ in Equation (32) is reliable only when $|e_a|^2$ is small enough and k is large.* c_1 can be obtained by using the initial value of the EMSE. However, it is not necessary to calculate c_1 *in general, because $S_h(k) \ll S_p(k)$ when k is large. Thus, $S(k) \approx S_p(k)$.*

4. Simulation

In this section, two examples are used to illustrate the superior performance of the RMCKRSL i.e., system identification and nonlinear prediction. We obtained the simulation results by averaging 1000 Monte Carlo trials.

4.1. Example 1

We chose the length of the filter as five where the weight vector $\mathbf{w}_0 = \begin{bmatrix} w_1 & w_2 & \cdots & w_5 \end{bmatrix}$ is generated randomly, where $w_i = w_{Ri} + jw_{Ii}$, with $w_{Ri} \in N(0,\ 0.1)$ and $w_{Ii} \in N(0,\ 0.1)$ being the real and imaginary parts of w_i, and $N(\mu,\ \sigma^2)$ denoting the Gaussian distribution with μ and σ^2 being the mean and variance, respectively. The input signal $x = x_R + jx_I$ is also generated randomly, where $x_R,\ x_I \in N(0,\ 1)$. An additive complex noise, $v = v_R + jv_I$, with v_R and v_I being the real and imaginary parts, is considered in the simulations.

First, we verify the superiority of the RMCKRSL in the presence of contaminated Gaussian noise [17,19], i.e., $v(k) = (1-c(k))v_1(k) + c(k)v_2(k)$, where $v_1(k) = v_{1R}(k) + jv_{1I}(k)$, with $v_{1R},\ v_{1I} \in N(0,\ 0.1)$, $v_2(k) = v_{2R}(k) + jv_{2I}(k)$ with $v_{2R},\ v_{2I} \in N(0,\ 20)$ represents an outlier (or impulsive disturbances), $\mathrm{P}(c(k)=1) = 0.06$ is the occurrence probability of impulsive disturbances, $\mathrm{P}(c(k){=}\ 0) = 0.94$. To ensure a fair comparison, all the algorithms use the recursive iteration to search the optimal solution. The parameters for different algorithms are chosen experimentally to guarantee the desirable solution. The performances of different algorithms on the basis of weight error power $\|\mathbf{w}_0 - \mathbf{w}(k)\|^2$ are shown in Figure 1. It is clear that compared with the MCCC, GMCCC and traditional RLS, RMCKRSL has the best filtering performance.

Then, the validity of the theoretical EMSEs for MCKRSL is demonstrated. The noise model is also a contaminated Gaussian model, where $v_{1R},\ v_{1I} \in N(0,\ \sigma_v^2/2)$, $v_{2R},\ v_{2I} \in N(0,\ 20)$, $\mathrm{P}(c(k)=1) = 0.06$ and $\mathrm{P}(c(k){=}\ 0) = 0.94$. Figure 2 compares the values of the theoretical EMSEs and simulated ones under variations of σ_v^2. Obviously there is a good match between the theoretical EMSEs and the simulated ones. In addition, it has been shown that the value of EMSE becomes bigger with the increase of noise variance.

Next, we tested the influence of the outlier on the performance of the RMCKRSL algorithm. The noise model is also a contaminated Gaussian noise, where $v_{1R},\ v_{1I} \in N(0,\ 0.1)$, $v_{2R},\ v_{2I} \in N(0,\ \sigma_B^2/2)$, $\mathrm{P}(c(k)=1) = p$ and $\mathrm{P}(c(k){=}\ 0){=}\ 1-p$. Figure 3 compares the performances of different algorithms under different probability of outlier values (p), where the sample size is 5000 and the variance of the outlier is $\sigma_B^2 = 40$. One can observe that the proposed RMCKRSL algorithm is robust to the probability of an outlier and has better performance than the MCCC, GMCCC and RLS. Figure 4 depicts the performances of different algorithms under different variance of outlier (σ_B^2) values, where the sample size is also 5000 and the probability of an outlier is $p = 0.06$. It can be observed that the proposed RMCKRSL algorithm is also robust to the variance of an outlier and has better performance than other algorithms.

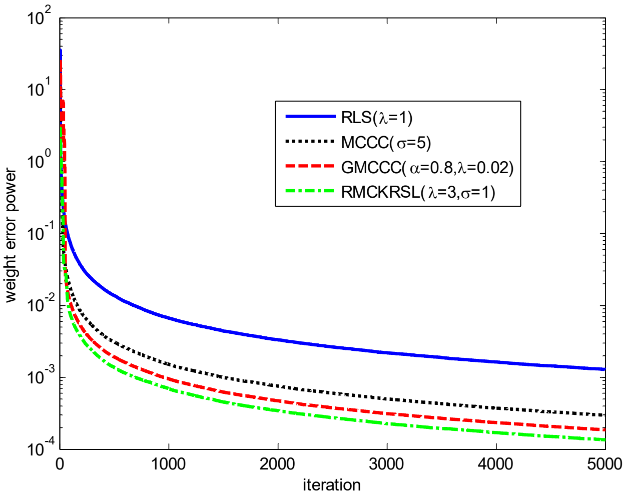

Figure 1. Learning curves of different algorithms.

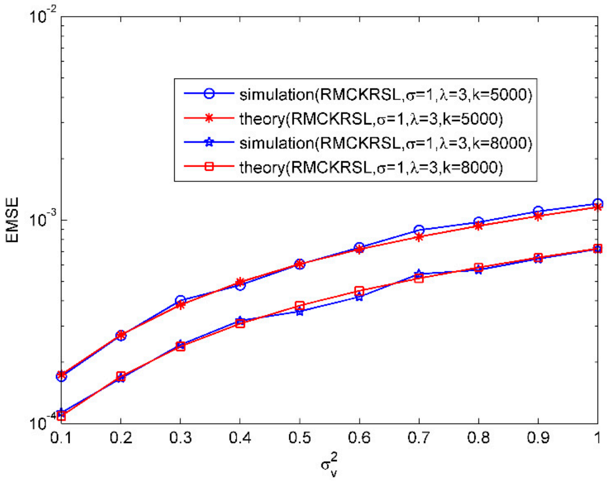

Figure 2. EMSE as a function of noise variances.

Finally, the influences of the kernel width σ and risk-sensitive parameter λ on the performance of the RMCKRSL are investigated. The noise model is also a contaminated Gaussian noise, where v_{1R}, $v_{1I} \in N(0,\ 0.1)$, v_{2R}, $v_{2I} \in N(0,\ 20)$, $\mathrm{P}(c(k) = 1) = 0.06$ and $\mathrm{P}(c(k) = 0) = 0.94$. Figures 5 and 6 present the performance of the RMCKRSL under a different kernel width σ and risk-sensitive parameter λ, respectively. One can see that both the kernel width σ and risk-sensitive parameter λ play an important role in the performance of the RMCKRSL. It is challenging to choose the optimal σ and λ because it is dependent on the statistical characteristic of the noise, which is unknown in the practical case. Thus, it is suggested that the parameters are chosen by experimentation.

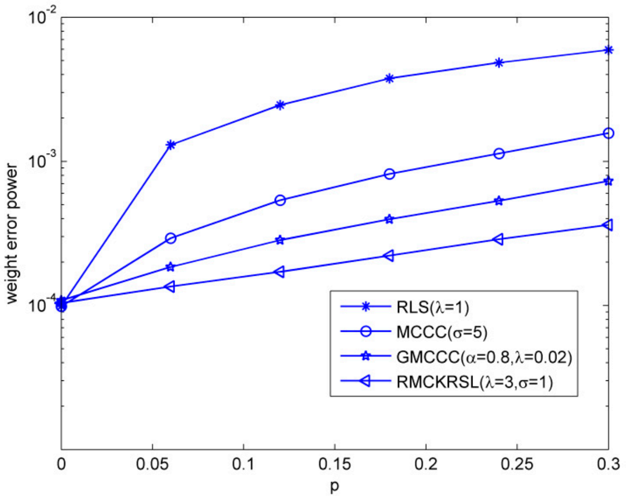

Figure 3. Influence of the probability of outliers.

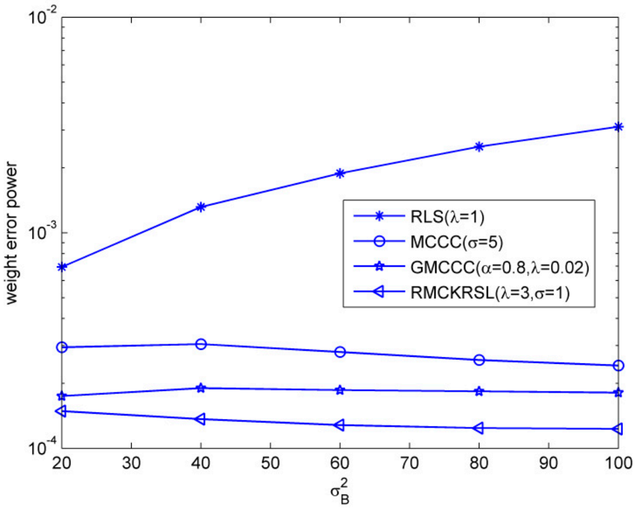

Figure 4. Influence of the variance of outliers.

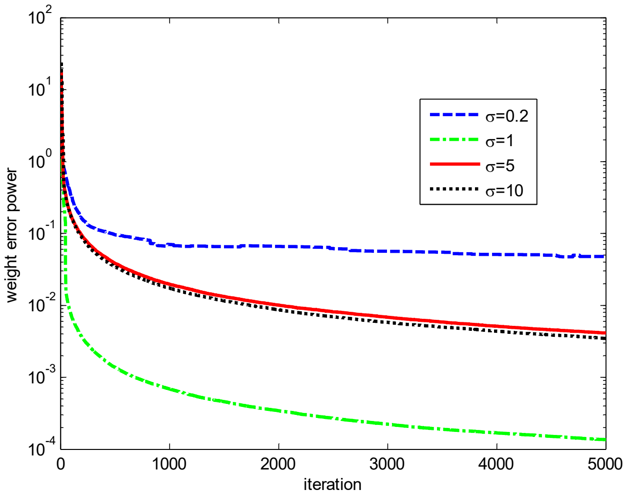

Figure 5. Influence of the σ ($\lambda = 3$).

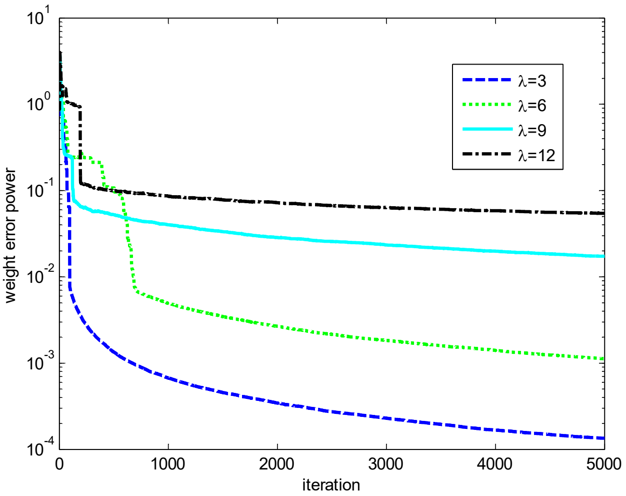

Figure 6. Influence of the λ ($\sigma = 1$).

4.2. Example 2

In this example, the superiority of the RMCKRSL is demonstrated by the prediction of a nonlinear system, where $s(t) = u_0[s_1(t) + js_2(t)]$, $s_1(t)$ is a Mackey-Glass chaotic time series described as follows [15]:

$$\frac{ds_1(t)}{dt} = -0.1s_1(t) + \frac{0.2s_1(t-30)}{1+s_1(t-30)^{10}} \tag{34}$$

$s_2(t)$ is the reverse of $s_1(t)$, and u_0 is a complex valued number whose real and imaginary parts are randomly generated and obey a uniform distribution over the interval [0, 1]. $s(t)$ is discretized by sampling with an interval of six seconds, and affected by the contaminated Gaussian noise $v(k) = (1-c(k))v_1(k) + c(k)v_2(k)$, where $v_1(k) = v_{1R}(k) + jv_{1I}(k)$, with v_{1R}, v_{1I} $\in N(0,\ 0.1)$, $v_2(k) = v_{2R}(k) + jv_{2I}(k)$ with v_{2R}, v_{2I} $\in N(0,\ 20)$, $\mathrm{P}(c(k)=1) = 0.06$, $\mathrm{P}(c(k)=0) = 0.94$. $s(k)$ is predicted by $\mathbf{x}(k) = [s(k-1) \quad s(k-2) \quad \cdots \quad s(k-6)]$ and the performance is measured by the mean square error (MSE) with $\mathrm{MSE}(k) = \frac{1}{N-6}\sum\limits_{l=7}^{N}\left(\left|s(l) - w^H(k)x(l)\right|^2\right)$. The convergence curves of different algorithms on the basis of MSE are compared in Figure 7. One may observe that the RMCKRSL has a faster convergence rate and better filter accuracy than other algorithms. In addition, the RLS behaves the worst since the minimum square error criterion is not robust to the impulse noise.

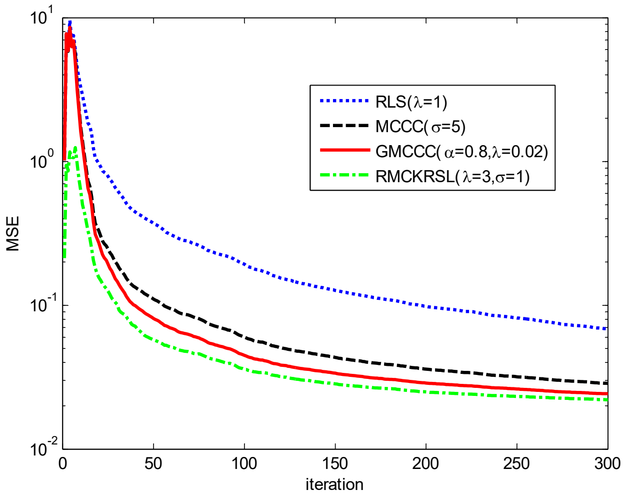

Figure 7. Convergence curves of different algorithms.

5. Conclusions

As a nonlinear similarity measure defined in kernel space, kernel risk-sensitive loss (KRSL) shows a superior performance in adaptive filter. However, there is no report about the recursive KRSL algorithm. Thus, in this paper we focused on the complex domain adaptive filter and proposed a recursive minimum complex KRSL (RMCKRSL) algorithm. Compared with the MCCC, GMCCC and traditional RLS algorithms, the proposed algorithm offers both a faster convergence rate and higher accuracy. Moreover, we derived the theoretical value of the EMSE, and demonstrated its correctness by simulations.

Author Contributions: Conceptualization, G.Q., D.L. and S.W.; methodology, S.W.; software, D.L.; validation, G.Q.; formal analysis, G.Q.; investigation, G.Q., D.L. and S.W.; resources, G.Q.; data curation, D.L.; writing—original draft preparation, G.Q. and D.L.; writing—review and editing, S.W.; visualization, D.L.; supervision, G.Q., D.L. and S.W.; project administration, G.Q.; funding acquisition, G.Q.

Funding: This research was funded by the China Postdoctoral Science Foundation Funded Project under grant 2017M610583, and Fundamental Research Funds for the Central Universities under grant SWU116013.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Principe, J.C. *Information Theoretic Learning: Renyi's Entropy and Kernel Perspectives*; Springer: New York, NY, USA, 2010.
2. Chen, B.; Zhu, Y.; Hu, J.; Principe, J.C. *System Parameter Identification: Information Criteria and Algorithms*; Newnes: Oxford, UK, 2013.
3. Liu, W.; Pokharel, P.P.; Pr´ıncipe, J. Correntropy: A localized similarity measure. In Proceedings of the 2006 IEEE International Joint Conference on Neural Network (IJCNN), Vancouver, BC, Canada, 16–21 July 2006; pp. 4919–4924.
4. Liu, W.; Pokharel, P.P.; Principe, J.C. Correntropy: Properties and applications in non-Gaussian signal processing. *IEEE Trans. Signal Process.* **2007**, *55*, 5286–5298. [CrossRef]
5. Singh, A.; Principe, J.C. Using correntropy as a cost function in linear adaptive filters. In Proceedings of the 2009 International Joint Conference on Neural Networks (IJCNN), Atlanta, GA, USA, 14–19 June 2009; pp. 2950–2955.
6. Singh, A.; Principe, J.C. A loss function for classification based on a robust similarity metric. In Proceedings of the 2010 International Joint Conference on Neural Networks (IJCNN), Barcelona, Spain, 18–23 July 2010; pp. 1–6.
7. Zhao, S.; Chen, B.; Principe, J.C. Kernel adaptive filtering with maximum correntropy criterion. In Proceedings of the 2011 International Joint Conference on Neural Networks (IJCNN), San Jose, CA, USA, 31 July–5 August 2011; pp. 2012–2017.
8. Chen, B.; Xing, L.; Liang, J.; Zheng, N.; Principe, J.C. Steady-state mean-square error analysis for adaptive filtering under the maximum correntropy criterion. *IEEE Signal Process. Lett.* **2014**, *21*, 880–884.
9. Wu, Z.; Peng, S.; Chen, B.; Zhao, H. Robust Hammerstein adaptive filtering under maximum correntropy criterion. *Entropy* **2015**, *17*, 7149–7166. [CrossRef]
10. Chen, B.; Wang, J.; Zhao, H.; Zheng, N.; Principe, J.C. Convergence of a Fixed-Point Algorithm under Maximum Correntropy Criterion. *IEEE Signal Process. Lett.* **2015**, *22*, 1723–1727. [CrossRef]
11. Wang, W.; Zhao, J.; Qu, H.; Chen, B.; Principe, J.C. Convergence performance analysis of an adaptive kernel width MCC algorithm. *AEU-Int. J. Electron. Commun.* **2017**, *76*, 71–76. [CrossRef]
12. Liu, X.; Chen, B.; Zhao, H.; Qin, J.; Cao, J. Maximum Correntropy Kalman Filter with State Constraints. *IEEE Access* **2017**, *5*, 25846–25853. [CrossRef]
13. Wang, F.; He, Y.; Wang, S.; Chen, B. Maximum total correntropy adaptive filtering against heavy-tailed noises. *Signal Process.* **2017**, *141*, 84–95. [CrossRef]
14. Chen, B.; Liu, X.; Zhao, H.; Principe, J.C. Maximum correntropy Kalman filter. *Automatica* **2017**, *76*, 70–77. [CrossRef]
15. Wang, S.; Dang, L.; Wang, W.; Qian, G.; Tse, C.K. Kernel Adaptive Filters with Feedback Based on Maximum Correntropy. *IEEE Access* **2018**, *6*, 10540–10552. [CrossRef]

16. He, Y.; Wang, F.; Yang, J.; Rong, H.; Chen, B. Kernel adaptive filtering under generalized Maximum Correntropy Criterion. In Proceedings of the 2016 International Joint Conference on Neural Networks (IJCNN), Vancouver, BC, Canada, 24–29 July 2016; pp. 1738–1745.
17. Chen, B.; Xing, L.; Zhao, H.; Zheng, N.; Príncipe, J.C. Generalized correntropy for robust adaptive filtering. *IEEE Trans. Signal Process.* **2016**, *64*, 3376–3387. [CrossRef]
18. Chen, B.; Wang, R. Risk-sensitive loss in kernel space for robust adaptive filtering. In Proceedings of the 2015 IEEE International Conference on Digital Signal Processing (DSP), Singapore, 21–24 July 2015; pp. 921–925.
19. Chen, B.; Xing, L.; Xu, B.; Zhao, H.; Zheng, N.; Príncipe, J.C. Kernel Risk-Sensitive Loss: Definition, Properties and Application to Robust Adaptive Filtering. *IEEE Trans. Signal Process.* **2017**, *65*, 2888–2901. [CrossRef]
20. Guimaraes, J.P.F.; Fontes, A.I.R.; Rego, J.B.A.; Martins, A.M.; Principe, J.C. Complex correntropy: Probabilistic interpretation and application to complex-valued data. *IEEE Signal Process. Lett.* **2017**, *24*, 42–45. [CrossRef]
21. Guimaraes, J.P.F.; Fontes, A.I.R.; Rego, J.B.A.; Martins, A.M.; Principe, J.C. Complex Correntropy Function: Properties, and application to a channel equalization problem. *Expert Syst. Appl.* **2018**, *107*, 173–181. [CrossRef]
22. Alliney, S.; Ruzinsky, S.A. An algorithm for the minimization of mixed l1 and l2 norms with application to Bayesian estimation. *IEEE Trans. Signal Process.* **1994**, *42*, 618–627. [CrossRef]
23. Mandic, D.; Goh, V. *Complex Valued Nonlinear Adaptive Filters: Noncircularity, Widely Linear and Neural Models (ser. Adaptive and Cognitive Dynamic Systems: Signal Processing, Learning, Communications and Control)*; John Wiley & Sons: New York, NY, USA, 2009.
24. Diniz, P.S.R. *Adaptive Filtering: Algorithms and Practical Implementation*, 4th ed.; Springer-Verlag: New York, NY, USA, 2013.
25. Qian, G.; Wang, S.; Wang, L.; Duan, S. Convergence Analysis of a Fixed Point Algorithm under Maximum Complex Correntropy Criterion. *IEEE Signal Process. Lett.* **2018**, *24*, 1830–1834. [CrossRef]
26. Qian, G.; Wang, S. Generalized Complex Correntropy: Application to Adaptive Filtering of Complex Data. *IEEE Access* **2018**, *6*, 19113–19120. [CrossRef]
27. Qian, G.; Wang, S. Complex Kernel Risk-Sensitive Loss: Application to Robust Adaptive Filtering in Complex Domain. *IEEE Access* **2018**, *6*, 2169–3536. [CrossRef]
28. Wirtinger, W. Zur formalen theorie der funktionen von mehr complexen veränderlichen. *Math. Ann.* **1927**, *97*, 357–375. [CrossRef]
29. Bouboulis, P.; Theodoridis, S. Extension of Wirtinger's calculus to reproducing Kernel Hilbert spaces and the complex kernel LMS. *IEEE Trans. Signal Process.* **2011**, *59*, 964–978.
30. Zhang, X. *Matrix Analysis and Application*, 2nd ed.; Tsinghua University Press: Beijing, China, 2013.
31. Picinbono, B. On circularity. *IEEE Trans. Signal Process.* **1994**, *42*, 3473–3482. [CrossRef]