Generalized Student's *t* Sparse Kernel Learning for Robust Signal Processing

Long Pan, Libiao Peng, Xifeng Li, Dongjie Bi and Yongle Xie
University of Electronic Science and Technology of China, Chengdu, Sichuan, China
E-mail: 202111060909@std.uestc.edu.cn

***Abstract*—The growing need for robust signal processing in non-Gaussian environments has become critical across multiple application domains, particularly in communication channel equalization, signal enhancement, and state estimation systems. This work proposes a novel sparse kernel learning method that integrates maximum generalized Student's *t*-kernel mixture correntropy regression with vector projection mechanism. More specifically, a kernel adaptive filtering algorithm, named GS-tKMCR, is proposed. This algorithm employs the generalized Student's *t* kernel mixture correntropy as the optimality criterion to effectively extract signal features from noise-contaminated data. Furthermore, to regulate the expansion of network size and mitigate the average computational complexity, a vector projection-based sparsification strategy is employed which dynamically prunes redundant bases during the recursive update process. Experimental results demonstrate that the proposed algorithm exhibits excellent performance in handling signals contaminated with non-Gaussian noise, outperforming four state-of-the-art kernel adaptive filtering algorithms.**

I. Introduction

Kernel Adaptive Filtering (KAF) is a technique that combines kernel methods with adaptive signal processing, widely used in complex signal processing scenarios under nonlinear and non-stationary environments[1]. With the rapid advancement of artificial intelligence and Internet of Things technologies, its advantages in high-dimensional data modeling have become increasingly prominent. This technique maps original data to a high-dimensional reproducing kernel Hilbert space using kernel functions, transforming nonlinear separable problems into linearly solvable ones. Typical applications include channel equalization and noise suppression in communication systems [2–4], speech enhancement [5], biomedical signal processing [6], financial time series prediction [7], and real-time signal optimization in intelligent sensor networks[8]. Compared to traditional linear filtering methods, KAF significantly enhances robustness and generalization in complex environments, making it a core tool for analyzing nonlinear dynamic systems in the era of big data.

In adaptive filtering, the second-order similarity measure in the data space is a viable measurement method in Gaussian noise environment, such as the mean square error (MSE)[9], affine projection[10] and least squares method[11]. However, the performance of kernel adaptive filtering algorithm using second-order metric is significantly degraded in non-Gaussian noise denoising[12]. In order to solve these problems, the non-second-order similarity measure method is proposed, such as mean p-power[13]. However, in general, the MPE criterion is still not able to provide the desired filtering performance under the impulsive noise. In recent years, nonlinear metric methods based on reproducing kernel Hilbert space (RKHS) have become a hot research topic[14]. As representatives of information theory, correntropic loss (C-loss)[15] and generalized correntropic loss[16] utilize the higher-order statistics of the data to derive some robust adaptive filtering methods under impulsive noise. On the downside, the non-convexity of the correntropic loss leads to its poor convergence performance. So as to enhance the robustness and accuracy, maximum mixture correntropy (MMC) is developed because of its better surface performance in the RKHS[17].

In general, the Gaussian function is choosing as the kernel function in MMC. Nevertheless, the Gaussian kernel function cannot exhibit the optimal performance sometimes, especially when it comes to the situation where large outliers and noise with multimodality distribution exist. To handle this problem, we utilize the generalized Student's t kernel [18] to release the power of non-quadratic cost function. And a new KAF algorithm is derived under maximum generalized Student's *t*-kernel mixture correntropy regression in this paper. It's worth noting that the size of the algorithm network will grow linearly as the input data continues to increase, eventually resulting in huge computation and storage overhead.

Various sparsification criteria have been proposed to control the growth of network size, including the quantization method [19–22], approximate linear dependency [23], and surprise criterion [24]. However, these approaches suffer from certain limitations, such as high computational cost or difficulty in determining appropriate thresholds. To overcome these drawbacks, this paper introduces an innovative online sparsification method called Vector Projection [25]. By leveraging discarded data to correct the dictionary and fully utilizing the information from data outside the dictionary, this method enables effective data sparsification.

To summarize the key contributions of this paper, we propose the maximum generalized Student's *t* kernel mixture correntropy regression and derive an optimal solution. Moreover, by incorporating the vector projection method, we devise a novel sparse KAF algorithm that enables efficient updates of solutions. The effectiveness of the proposed algorithm is validated through simulations on nonlinear channel equalization, which underscore its superior performance compared to

979-8-3315-7206-8/25/$31.00

existing methods.

II. Main Results

A. Generalized Student's t Kernel Function

The generalized Student's t-distribution encompasses both the exponential power distribution and the Student's t-distribution as specific instances, with mathematical expressions that can be represented by:

$$f(x) = \boldsymbol{K}(v,\beta)\left(1+\frac{|x|^{\beta}}{v}\right)^{-\frac{v+1}{\beta}} \quad -\infty < x < +\infty \tag{1}$$

with

$$\boldsymbol{K}(v,\beta) = \frac{\beta}{2v^{\frac{1}{\beta}}\boldsymbol{B}(\frac{v}{\beta},\frac{1}{\beta})} \tag{2}$$

$$\boldsymbol{B}(a,b) = \int_0^1 t^{a-1}(1-t)^{b-1}\,dt \tag{3}$$

in which $v, \beta > 0$.

Thanks to the additional flexibility of shape parameter v and β, the generalized Student's t-distribution function is able to capture various shapes at the peak and tail of the distribution. This property of generalized Student's t-distribution makes it more flexible to express more information in comparison with the Gaussian distribution. When $\beta = 2$, the standard Student's t-distribution is achieved. As the degrees of freedom v increase, the distribution curve of (1) becomes progressively flatter. In the limit as $v \to +\infty$, the function (1) converges to a Gaussian distribution. Inversely, as $v \to 0$, the function (1) approaches the Dirac delta function.

Based on Mercer's kernel theorem, the generalized Student's t kernel can be derived as:

$$\kappa_{GSt}(\boldsymbol{X},\boldsymbol{Y}) = \left(\frac{1}{v}\left(\frac{||\boldsymbol{X}-\boldsymbol{Y}||^2}{\sigma^2}\right)^{\frac{\beta}{2}}+1\right)^{-\frac{v+1}{\beta}} \tag{4}$$

in which $\sigma > 0$ means the kernel bandwidth. v and β are the shape parameters.

B. Generalized Student's t Kernel Mixture Correntropy

As a non-quadratic statistical measure, the correntropy is powerful to dig up more information contained in the original datasets. Given two random variables V and M, the mathematical form of correntropy can be expressed as:

$$\begin{aligned} S(V,W) &= \mathbf{E}\left[\kappa\left(V-W\right)\right] \\ &= \mathbf{E}\left[\langle\boldsymbol{\varphi}(V),\boldsymbol{\varphi}(W)\rangle_{\mathbb{F}}\right]. \end{aligned} \tag{5}$$

The Mercer kernel, denoted as $\kappa(\cdot)$, plays a crucial role in the formulation of the proposed approach. Additionally, the expectation operator $\mathbf{E}[\cdot]$ is employed to derive the empirical correntropy. To approximate this correntropy, a sample estimator is utilized, which can be expressed as:

$$\hat{S}_N = \frac{1}{N}\sum_{i=1}^{N}\kappa(x_i - y_i). \tag{6}$$

Based on the weighted mixture of two kernels, the mixture correntropy can be build as:

$$M(\boldsymbol{X},\boldsymbol{Y}) = \mathbf{E}\left[\alpha\kappa_1(e) + (1-\alpha)\kappa_2(e)\right] \tag{7}$$

in which κ_1, κ_2 are two different kernels. $0 \le \alpha \le 1$ is the weight of kernel. The mixture correntropy is the extension of typical correntropy (5). With appropriate coefficients, the mixture correntropy has better learning performance compared to the original correntropy [17].

Based on (4) , the generalized Student's t kernel mixture correntropy is defined as follows:

$$M_{GStKAF}(\boldsymbol{X},\boldsymbol{Y}) = \mathbf{E}\left[\alpha\kappa_{GSt_1}(e) + (1-\alpha)\kappa_{GSt_2}(e)\right] \tag{8}$$

Notice that $M_{GStKAF}(\boldsymbol{X},\boldsymbol{Y})$ is positive and bounded: $0 \le M_{GStKAF}(\boldsymbol{X},\boldsymbol{Y}) \le 1$. When $\boldsymbol{X} = \boldsymbol{Y}$, $M_{GStKAF}(\boldsymbol{X},\boldsymbol{Y}) = 1$.

Let $\boldsymbol{X} = [m,n]^T, \boldsymbol{Y} = [0,0]^T, \sigma_1 = 2, v_1 = 1, \beta_1 = 1.8, \sigma_2 = 5, v_2 = 5, \beta_2 = 1.8, \alpha = 0.5$. The surface of $M_{GStKAF}(\boldsymbol{X},\boldsymbol{Y})$ is shown in Fig. 1. The gradient of $M_{GStKAF}(\boldsymbol{X},\boldsymbol{Y})$ increases with the distance of the two random variables initially increasing, which is conducive to the convergence of learning process. As the distance continues to increase, the gradient decreases. By utilizing this property, the fluctuation of learning performance caused by large outliers can be effectively suppressed.

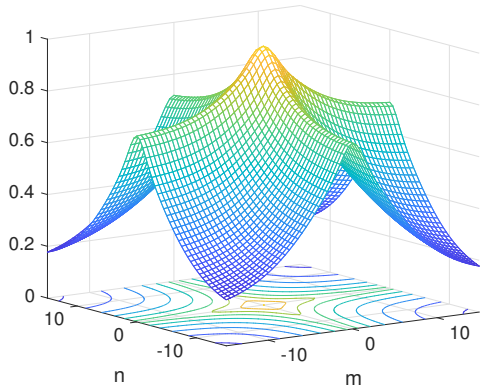

Fig. 1. Surface of the M_{GStKAF}

C. Maximum Generalized Student's t Kernel Mixture Correntropy Regression

Consider the problem of maximum generalized Student's t kernel mixture correntropy regression, which involves finding a continuous mapping function $f : \mathbb{U} \to \mathbb{R}$ based on a sequence of training-target data pairs $\{\boldsymbol{u}(j), d(j)\}_{j=1}^{n}$. Here, $\boldsymbol{u}(j) \in \mathbb{U} \subset \mathbb{R}^{k\times 1}$ represents a k-dimensional input vector, while $d(j) \in \mathbb{R}$ denotes the corresponding desired output. To facilitate this process, we utilize a high-dimensional feature space $\mathbb{H}$ that serves as a RKHS. By leveraging the kernel method, we can efficiently search for the optimal function f within this feature space $\mathbb{H}$.

$$f\left(\boldsymbol{u}\left(j\right)\right)=\boldsymbol{w}_i^T\boldsymbol{\varphi}_j \tag{9}$$

Here, $\boldsymbol{w}_i \in \mathbb{H}$ represents the output weight vector at the i-th iteration, and $\boldsymbol{\varphi}_j = \kappa(u(j),\cdot)$ denotes the kernel mapping in the space $\mathbb{H}$.

The optimal solution for $\boldsymbol{w}_i$ can be obtained by maximizing the generalized Student's *t* kernel mixture correntropy:

$$\begin{aligned}\max_{\boldsymbol{w}_i} M\left(\boldsymbol{w}_i\right)=&\sum_{j=1}^{i}\left[\alpha\kappa_{GSt_1}(e_{i,j})+(1-\alpha)\kappa_{GSt_2}(e_{i,j})\right]\\&-\frac{\lambda}{2}\left\|\boldsymbol{w}_i\right\|_{\mathbb{H}}^2\end{aligned} \tag{10}$$

in which $\kappa_{GSt}\left(\cdot\right)$ denotes the kernel function, $e_{i,j}=d(j)-f\left(\boldsymbol{u}\left(j\right)\right)=d(j)-\boldsymbol{w}_i^T\boldsymbol{\varphi}_j$,$\lambda>0$ and $\frac{\lambda}{2}\left\|\boldsymbol{w}_i\right\|^2$ is the regularization term.

By taking the derivative of the cost function (10) with respect to the output weight vector $\boldsymbol{w}_i$ and setting it to zero, we can get the following solution:

$$\begin{aligned}&\frac{\partial M\left(\boldsymbol{w}_i\right)}{\partial \boldsymbol{w}_i}=0\Rightarrow\\&\qquad\boldsymbol{w}_i=[\lambda\boldsymbol{I}_i+\boldsymbol{\Phi}_i\boldsymbol{S}_i\boldsymbol{\Phi}_i^T]^{-1}\boldsymbol{\Phi}_i\boldsymbol{D}_i\end{aligned} \tag{11}$$

with

$$\begin{cases}\boldsymbol{\Phi}_i=(\boldsymbol{\varphi}(\mathcal{D}_1),\boldsymbol{\varphi}(\mathcal{D}_2),...,\boldsymbol{\varphi}(\mathcal{D}_L))\\ \boldsymbol{S}_i=diag[s_1,s_2,...,s_i]\\ \boldsymbol{D}_i=\boldsymbol{S}_i(d_1,d_2,...,d_i)^T\\ s_j=\alpha\frac{2(v_1+1)}{\beta_1 v_1\sigma_1^{\beta_1}}\kappa_{GSt_1}(e_{ij})^{\frac{\beta_1}{v_1+1}+1}\\ \qquad+(1-\alpha)\frac{2(v_2+1)}{\beta_2 v_2\sigma_2^{\beta_2}}\kappa_{GSt_2}(e_{ij})^{\frac{\beta_2}{v_2+1}+1}\end{cases} \tag{12}$$

where $\boldsymbol{I}_i$ is an $i\times i$ identity matrix, $\boldsymbol{\Phi}_i$ means the mapping vector matrix in $\mathbb{H}$. $(\mathcal{D}_1,\mathcal{D}_2,...,\mathcal{D}_L)$ denotes the data stored in the dictionary. $(d_1,d_2,...,d_i)^T$ is a column vector of the desired output.

It is possible to re-express (11) as:

$$\boldsymbol{w}_i=\boldsymbol{\Phi}_i\lambda^{-1}\left(\boldsymbol{D}_i-\boldsymbol{S}_i\boldsymbol{\Phi}_i^T\boldsymbol{w}_i\right)=\boldsymbol{\Phi}_i\boldsymbol{\Omega}_i \tag{13}$$

In (13), $\boldsymbol{w}_i$ is expressed as a linear combination of the training matrix $\boldsymbol{\Phi}_i$ and the weight vector $\boldsymbol{\Omega}_i$. Therefore, one can obtain

$$\begin{aligned}\boldsymbol{\Omega}_i&=\lambda^{-1}\left(\boldsymbol{D}_i-\boldsymbol{S}_i\boldsymbol{\Phi}_i^T\boldsymbol{w}_i\right)\\&=\lambda^{-1}\left(\boldsymbol{D}_i-\boldsymbol{S}_i\boldsymbol{\Phi}_i^T\boldsymbol{\Phi}_i\boldsymbol{\Omega}_i\right)\\&=(\lambda\boldsymbol{I}_i+\boldsymbol{S}_i\boldsymbol{\Phi}_i^T\boldsymbol{\Phi}_i)^{-1}\boldsymbol{D}_i\\&=(\lambda\boldsymbol{I}_i+\boldsymbol{S}_i\boldsymbol{K}_i)^{-1}\boldsymbol{D}_i\end{aligned} \tag{14}$$

where $\boldsymbol{K}_i=<\boldsymbol{\varphi}_j,\boldsymbol{\varphi}_k>_{\mathbb{F}},j,k\in[\mathcal{D}_1,\mathcal{D}_2,...,\mathcal{D}_L]$.

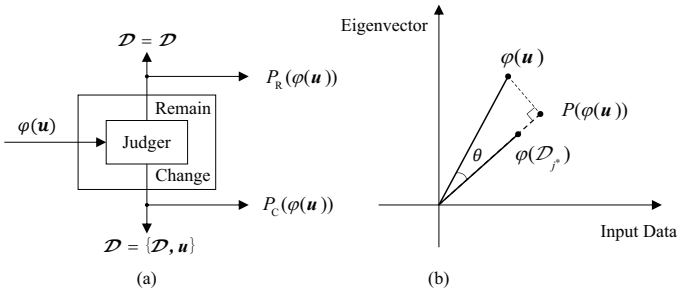

Fig. 2. (a) The judger of VP criterion (b) The principle of vector projection

D. Vector Projection Method

In order to reduce the computational and storage volume of calculating $\boldsymbol{\Phi}_i$ and $\boldsymbol{\Omega}_i$, the vector projection method is utilized to derive the recursive process. The main idea of vector projection is shown in Fig. 2. Let $\varphi(\boldsymbol{u})$ be a mapping vector of the input $\boldsymbol{u}$. $\varphi(\boldsymbol{u})$ will enter into one of two different stage (e.g. Remain-Stage or Change-Stage) under the supervision of a judger. The Remain-Stage is to keep the dictionary size constant and change the corresponding existing elements in the dictionary by the vector projection operation, while the Change-Stage is to expand the dictionary size and absorb the new elements into the dictionary. Specifically, we define an evaluation variable to measure the value of the input data:

$$\cos\left(\varphi\left(\boldsymbol{u}\right),\varphi\left(\boldsymbol{\mathcal{D}}\right)\right)=\frac{\left\langle\varphi\left(\boldsymbol{u}\right),\varphi\left(\mathcal{D}_{j^*}\right)\right\rangle_{\mathbb{F}}}{\left\|\varphi\left(\boldsymbol{u}\right)\right\|_{\mathbb{F}}\left\|\varphi\left(\mathcal{D}_{j^*}\right)\right\|_{\mathbb{F}}} \tag{15}$$

where $\boldsymbol{\mathcal{D}}$ is the dictionary with L elements,

$$j^*=arg\max_{1\le j\le L}\cos\left(\varphi\left(\boldsymbol{u}\right),\varphi\left(\mathcal{D}_j\right)\right)$$

and $\mathcal{D}_{j^*}$ is the j^*th member of $\boldsymbol{\mathcal{D}}$.

When $\cos(\varphi(\boldsymbol{u}),\varphi(\boldsymbol{\mathcal{D}}))\ge\varepsilon_c$, the dictionary is updated by:

$$P_R(\varphi(\boldsymbol{u}))=\frac{\kappa(\boldsymbol{u},\mathcal{D}_{j^*})}{\|\varphi(\mathcal{D}_{j^*})\|_{\mathbb{F}}^2}\varphi(\mathcal{D}_{j^*}) \tag{16}$$

while the dictionary size remains the same. Otherwise, the dictionary is extended by absorbing new element:

$$P_C(\varphi(\boldsymbol{u}))=\varphi(\boldsymbol{u}) \tag{17}$$

Herein, ε_c is a judge threshold that determines whether an element should enter the dictionary or not.

E. The Proposed Algorithm

Let $\boldsymbol{Q}_i=(\lambda\boldsymbol{I}_i+\boldsymbol{S}_i\boldsymbol{K}_i)^{-1}$. Utilizing the vector projection technique, we formulate the data dictionary for the iterative algorithm in two distinct scenarios:

1) Scenario 1: $\cos(\varphi(\boldsymbol{u}),\varphi(\boldsymbol{\mathcal{D}}))\ge\varepsilon_c$.

In this case, we have $\boldsymbol{\mathcal{D}}_i=\boldsymbol{\mathcal{D}}_{i-1}$ and $\boldsymbol{Q}_i=\boldsymbol{Q}_{i-1}$ a. $\varphi(\boldsymbol{u}(i))$ is incorporated into the j^*th group by vector projection, where $j^*=arg\max_{1\le j\le L}\cos(\varphi(\boldsymbol{u}_i),\varphi(\mathcal{D}_{ij}))$. Then, the we have

$$\begin{cases}\boldsymbol{S}_i=\boldsymbol{S}_{i-1}+a_{j^*i}^2S_{j^*i}\boldsymbol{J}_{j^*}\boldsymbol{J}_{j^*}^T\\ \boldsymbol{D}_i=\boldsymbol{D}_{i-1}+a_{j^*i}S_{j^*i}d_i\boldsymbol{J}_{j^*}\end{cases} \tag{18}$$

where $S_{j^*i} = \alpha\kappa_{GSt_1}(d_i - a_{j^*i}\boldsymbol{w}^T\varphi(\mathcal{D}_{ij^*})) + (1-\alpha)\kappa_{GSt_2}(d_i - a_{j^*i}\boldsymbol{w}^T\varphi(\mathcal{D}_{ij^*}))$; $a_{j^*i} = \alpha\frac{\kappa_{GSt_1}(\boldsymbol{u}_i,\mathcal{D}_{ij^*})}{\|\varphi(\mathcal{D}_{ij^*})\|_{\mathbb{F}}^2} + (1-\alpha)\frac{\kappa_{GSt_2}(\boldsymbol{u}_i,\mathcal{D}_{ij^*})}{\|\varphi(\mathcal{D}_{ij^*})\|_{\mathbb{F}}^2}$; and $\boldsymbol{J}_{j^*}$ is a column vector of L indices, with the j^*th element being 1 and all other elements being zero. Utilizing (18), the matrix $\boldsymbol{Q}$ can be obtained through the following derivation:

$$\begin{aligned}\boldsymbol{Q}_i &= [\boldsymbol{Q}_{i-1}^{-1} + a_{j^*i}^2 S_{\sigma,v}^{j^*i}\boldsymbol{J}_{j^*}\boldsymbol{J}_{j^*}^T\boldsymbol{K}_{i-1}]^{-1} \\ &= \boldsymbol{Q}_{i-1} - \frac{\boldsymbol{Q}_{i-1}(j^*)\boldsymbol{K}_{i-1}(j^*)^T\boldsymbol{Q}_{i-1}}{(a_{j^*i}^2 S_{j^*i})^{-1} + \boldsymbol{K}_{i-1}(j^*)^T\boldsymbol{Q}_{i-1}(j^*)} \end{aligned} \tag{19}$$

The j^*th columns of $\boldsymbol{Q}_{i-1}$ and $\boldsymbol{K}_{i-1}$ are represented by $\boldsymbol{Q}_{i-1}(j^*)$ and $\boldsymbol{K}_{i-1}(j^*)$, respectively. Furthermore, one can update the vector $\boldsymbol{\Omega}_i$ in (13) as

$$\boldsymbol{\Omega}_i = \boldsymbol{\Omega}_{i-1} + \frac{(a_{j^*i}^{-1}d_i - \boldsymbol{K}_{i-1}(j^*)^T\boldsymbol{\Omega}_{i-1})\boldsymbol{Q}_{i-1}(j^*)}{(a_{j^*i}^2 S_{j^*i})^{-1} + \boldsymbol{K}_{i-1}(j^*)^T\boldsymbol{Q}_{i-1}(j^*)} \tag{20}$$

2) Scenario 2: $\cos(\varphi(\boldsymbol{u}), \varphi(\boldsymbol{\mathcal{D}})) < \varepsilon_c$.

The dictionary is expanded in this case where $\boldsymbol{\mathcal{D}}_i = \{\boldsymbol{\mathcal{D}}_{i-1}, \boldsymbol{u}_i\}$, $\boldsymbol{\Phi}_i = [\boldsymbol{\Phi}_{i-1}, \varphi(\boldsymbol{u}_i)]$. Meanwhile,

$$\boldsymbol{S}_i = \begin{bmatrix}\boldsymbol{S}_{i-1} & \boldsymbol{0} \\ \boldsymbol{0}^T & s_i\end{bmatrix}, \boldsymbol{K}_i = \begin{bmatrix}\boldsymbol{K}_{i-1} & \boldsymbol{h}_i \\ \boldsymbol{h}_i^T & \kappa_{ii}\end{bmatrix}, \tag{21}$$

where $\boldsymbol{h}_i = \boldsymbol{\Phi}_{i-1}^T\varphi_i$. Furthermore, we have

$$\boldsymbol{Q}_i = \begin{bmatrix}\boldsymbol{Q}_{i-1} + \hat{r}_i^{-1}s_i\hat{\boldsymbol{z}}_{\hat{B}i}\hat{\boldsymbol{k}}_i^T & -\hat{r}_i^{-1}\hat{\boldsymbol{z}}_{\hat{B}i} \\ -\hat{r}_i^{-1}s_i\hat{\boldsymbol{k}}_i^T & \hat{r}_i^{-1}\end{bmatrix}, \tag{22}$$

with

$$\begin{cases}\hat{\boldsymbol{z}}_{\hat{B}i} = \boldsymbol{Q}_{i-1}\boldsymbol{S}_{i-1}\boldsymbol{h}_i, \\ \hat{\boldsymbol{k}}_i = \boldsymbol{Q}_{i-1}^T\boldsymbol{h}_i, \\ \hat{r}_i^{-1} = \kappa_{ii}s_i + \lambda - s_i\boldsymbol{h}_i^T\hat{\boldsymbol{z}}_{\hat{B}i}.\end{cases} \tag{23}$$

Moreover, Substituting (21), (22) and $\boldsymbol{D}_i = [(\boldsymbol{D}_{i-1})^T, s_i d_i]^T$ to (14), we have

$$\boldsymbol{\Omega}_i = \begin{bmatrix}\boldsymbol{\Omega}_{i-1} - \hat{r}_i^{-1}s_i\hat{\boldsymbol{z}}_{\hat{B}i}(d_i - \boldsymbol{h}_i^T\boldsymbol{\Omega}_{i-1}) \\ \hat{r}_i^{-1}s_i(d_i - \boldsymbol{h}_i^T\boldsymbol{\Omega}_{i-1})\end{bmatrix}, \tag{24}$$

In conclusion, the kernel adaptive filtering algorithm named GS-tKMCR has been derived and summarized in Table ***Algorithm*1**. It has been observed that the overall computational complexity of the proposed algorithm is $\mathcal{O}(L^2)$, where L represents the size of the dictionary. In comparison to the non-sparse recursive algorithm, the proposed algorithm exhibits significantly improved computational efficiency when $L \ll i$.

Algorithm 1: GS-tKMCR

Initialization:

covariance $\sigma_1, \sigma_2 > 0$, $\beta_1, \beta_2 > 0$, $v_1, v_2 > 0$
regularization factor $\gamma > 0$,
threshold $0 < \varepsilon_c < 1$,
mixture coefficient $0 \le \alpha \le 1$

Estimate:

$L = 1$, $\boldsymbol{\mathcal{D}}_1 = \{\boldsymbol{u}_1\}$,
$s_1 = \alpha\frac{2(v_1+1)}{\beta_1 v_1\sigma_1^{\beta_1}}\kappa_{GSt_1}(e_1)^{\frac{\beta_1}{v_1+1}+1}$
$\quad +(1-\alpha)\frac{2(v_2+1)}{\beta_2 v_2\sigma_2^{\beta_2}}\kappa_{GSt_2}(e_1)^{\frac{\beta_2}{v_2+1}+1}$,
$\boldsymbol{S}_1 = [s_1]$, $\quad \boldsymbol{Q}_1 = (\lambda + s_1\kappa_{11})^{-1}, \boldsymbol{\Omega}_1 = \boldsymbol{Q}_1 s_1 d_1$

Computation:

for *i=2:n* **do**
  if $\cos(\varphi(\boldsymbol{u}(i)), \varphi(\boldsymbol{\mathcal{D}})) \ge \varepsilon_c$ **then**
    Dictionary: $L \Leftarrow L, \boldsymbol{\mathcal{D}}_i = \boldsymbol{\mathcal{D}}_{i-1}$
    Update:
    $\boldsymbol{S}_i = \boldsymbol{S}_{i-1} + a_{j^*i}^2 S_{j^*i}\boldsymbol{J}_{j^*}\boldsymbol{J}_{j^*}^T$
    $\boldsymbol{Q}_i = \boldsymbol{Q}_{i-1} - \frac{\boldsymbol{Q}_{i-1}(j^*)\boldsymbol{K}_{i-1}(j^*)^T\boldsymbol{Q}_{i-1}}{(a_{j^*i}^2 S_{j^*i})^{-1} + \boldsymbol{K}_{i-1}(j^*)^T\boldsymbol{Q}_{i-1}(j^*)}$
    $\boldsymbol{\Omega}_i = \boldsymbol{\Omega}_{i-1} + \frac{(a_{j^*i}^{-1}d_i - \boldsymbol{K}_{i-1}(j^*)^T\boldsymbol{\Omega}_{i-1})\boldsymbol{Q}_{i-1}(j^*)}{(a_{j^*i}^2 S_{j^*i})^{-1} + \boldsymbol{K}_{i-1}(j^*)^T\boldsymbol{Q}_{i-1}(j^*)}$
  end
  else
    Dictionary: $L \Leftarrow L+1, \boldsymbol{\mathcal{D}}_i = \{\boldsymbol{\mathcal{D}}_{i-1}, \boldsymbol{u}_i\}$
    Update:
    $\boldsymbol{h}_i = \boldsymbol{\Phi}_{i-1}^T\varphi_i$
    $e_i = d_i - \boldsymbol{h}_i^T\boldsymbol{\Omega}_{i-1}$
    $s_i = \alpha\frac{2(v_1+1)}{\beta_1 v_1\sigma_1^{\beta_1}}\kappa_{GSt_1}(e_i)^{\frac{\beta_1}{v_1+1}+1}$
    $\quad +(1-\alpha)\frac{2(v_2+1)}{\beta_2 v_2\sigma_2^{\beta_2}}\kappa_{GSt_2}(e_i)^{\frac{\beta_2}{v_2+1}+1}$
    $\boldsymbol{S}_i = \begin{bmatrix}\boldsymbol{S}_{i-1} & \boldsymbol{0} \\ \boldsymbol{0}^T & s_i\end{bmatrix}$, $\boldsymbol{K}_i = \begin{bmatrix}\boldsymbol{K}_{i-1} & \boldsymbol{h}_i \\ \boldsymbol{h}_i^T & \kappa_{ii}\end{bmatrix}$
    $\hat{\boldsymbol{z}}_{\hat{B}i} = \boldsymbol{Q}_{i-1}\boldsymbol{S}_{i-1}\boldsymbol{h}_i$
    $\hat{\boldsymbol{k}}_i = \boldsymbol{Q}_{i-1}^T\boldsymbol{h}_i$
    $\hat{r}_i^{-1} = \kappa_{ii}s_i + \lambda - s_i\boldsymbol{h}_i^T\hat{\boldsymbol{z}}_{\hat{B}i}$
    $\boldsymbol{Q}_i = \begin{bmatrix}\boldsymbol{Q}_{i-1} + \hat{r}_i^{-1}s_i\hat{\boldsymbol{z}}_{\hat{B}i}\hat{\boldsymbol{k}}_i^T & -\hat{r}_i^{-1}\hat{\boldsymbol{z}}_{\hat{B}i} \\ -\hat{r}_i^{-1}s_i\hat{\boldsymbol{k}}_i^T & \hat{r}_i^{-1}\end{bmatrix}$
    $\boldsymbol{\Omega}_i = \begin{bmatrix}\boldsymbol{\Omega}_{i-1} - \hat{r}_i^{-1}s_i\hat{\boldsymbol{z}}_{\hat{B}i}(d_i - \boldsymbol{h}_i^T\boldsymbol{\Omega}_{i-1}) \\ \hat{r}_i^{-1}s_i(d_i - \boldsymbol{h}_i^T\boldsymbol{\Omega}_{i-1})\end{bmatrix}$.
  end
end

III. Experimental Results

A. Experimental Settings

In this section, the proposed algorithm is utilized to address a nonlinear channel equalization problem. This problem is characterized by a nonlinear channel model that consists of a linear filter combined with a memoryless nonlinear component. More specifically, the input to the nonlinear channel is a binary signal represented as $\{s(1), s(2), ..., s(N)\}$, which is distorted by α-stable noise. The resulting received signal is indicated by $\{r(1), r(2), ..., r(N)\}$. In order to reconstruct the original signal with maximum accuracy, an inverse filter is designed to counteract the distortions introduced by the nonlinear channel.

This study employs a sample set $\{([r(i), r(i+1), ..., r(i+l)], s(i-D))\}$, where l signifies the length of time embedding and and D corresponds to the equalization lag time. Specifically, we set $l = 5$ and $D = 2$. Within the nonlinear channel model, the input signal follows $x(i) = s(i) + 0.6s(i-1)$, whereas the output is characterized by $r(i) = x(i) - 0.8x(i)^2 + n(i)$, incorporating α-stable noise $n(i)$. The experimental setup

involves 1000 training data points and 200 testing data points. To bolster reliability, 20 independent Monte Carlo simulations are executed per test. Additionally, the noise variance is fixed at 0.04, with all algorithm parameters detailed in Table I.

TABLE I
VALUES OF PARAMETERS

Parameters	v_1	σ_1	β_1	v_2	σ_2	β_2	α	γ
Values	1	10	1.8	12	0.004	1.9	0.5	0.001

B. Comparison of Performance Between the Proposed Algorithm and Other KAF Algorithms

In this study, we compare the performance of our proposed algorithm with four state-of-the-art KAF algorithms, namely QKRLS[26], QKRLP[19], QKRMC[20], and SP-KRMC-W[21], using the same comparison methodology as in MG time series prediction. The results are presented in Table II and Fig.3-Fig.4. Our experimental observations indicate that:

1) From Fig. 3, it is evident that our proposed algorithm outperforms the other algorithms in terms of accuracy performance when applied to nonlinear channel equalization problems that are contaminated by non-Gaussian noise. Specifically, our proposed algorithm exhibits significantly lower mean squared error values, indicating a higher degree of accuracy in recovering the original signal.

2) As indicated by Table II, the proposed algorithm exhibits the smallest variance, which is a strong indicator of the algorithm's robustness in nonlinear channel equalization problems. This observation is further supported by the results presented in Fig. 3.

3) Employing the vector projection method, the network size of the proposed algorithm is reduced from 100% to 67%, as revealed by Fig.4. This represents a significant reduction in the number of parameters required for the algorithm to operate effectively. Compared to the other algorithms, the proposed algorithm exhibits a much sparser network size, which is an important feature for practical applications where computational efficiency and memory usage are crucial considerations. Overall, these results demonstrate the effectiveness and efficiency of the proposed algorithm in nonlinear channel equalization problems.

TABLE II
TESTING MSES AT FINAL ITERATION

Algorithm	Testing MSE	Network Size	Variance
QKRLS	$1.2\text{x}10^{-1}$	992	$1.4\text{x}10^{-1}$
QKRLP	$1.1\text{x}10^{-1}$	957	$1.5\text{x}10^{-1}$
QKRMC	$9.9\text{x}10^{-2}$	823	$9.0\text{x}10^{-2}$
SP-KRMC-W	$1.0\text{x}10^{-1}$	856	$1.2\text{x}10^{-1}$
Proposed	$7.0\text{x}10^{-2}$	675	$5.5\text{x}10^{-2}$

IV. CONCLUSION

This paper proposes a novel approach to improving estimation accuracy performance in non-Gaussian noise environments, through the use of a generalized Student's t kernel adaptive filtering algorithm under the maximum mixture correntropy criterion. Additionally, a sparsity criterion, known as vector projection, is employed to reduce the network size during algorithm implementation. The experimental results highlight the superior accuracy and sparsity of the proposed algorithm in comparison to other state-of-the-art methods. Further research can be conducted to design more refined cost functions, which could potentially improve the performance of the algorithm even further. Additionally, other sparsity criteria may be investigated to further reduce the network size, which is crucial for practical applications where computational efficiency and memory usage are important considerations. These efforts could lead to the development of even more effective and efficient algorithms for nonlinear system identification tasks in non-Gaussian noise environments.

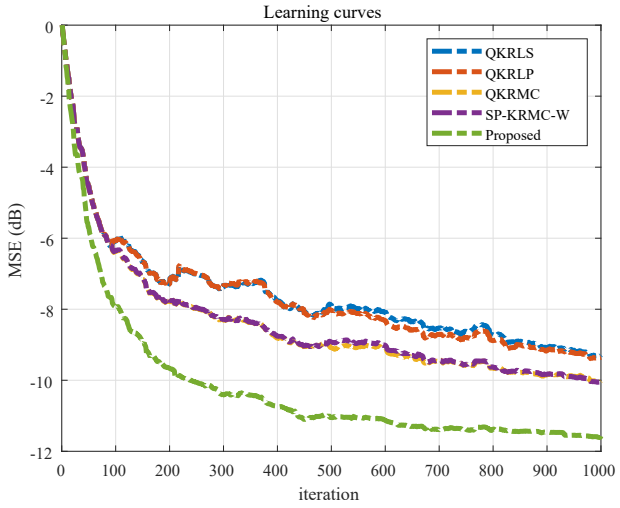

Fig. 3. Learning curves of different algorithms.

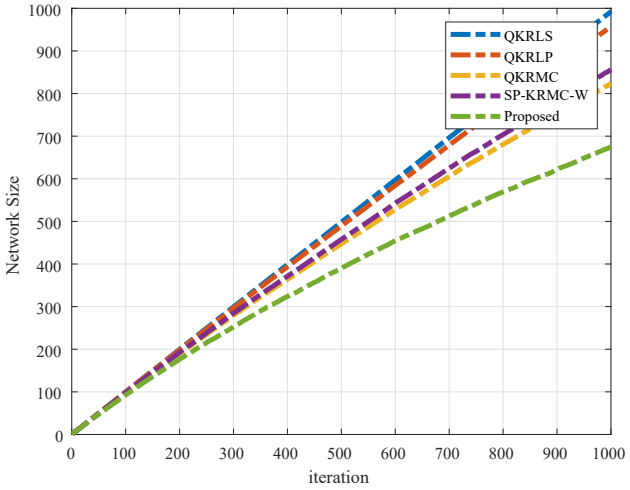

Fig. 4. Network size of different algorithms in the training process

ACKNOWLEDGMENT

This work was supported by the National Natural Science Foundation of China under Grant 62027803.

REFERENCES

[1] Weifeng Liu, Jose C Principe, and Simon Haykin. *Kernel adaptive filtering: a comprehensive introduction*. John Wiley & Sons, 2011.

[2] Renato Candido, Marcio Eisencraft, and Magno TM Silva. Equalization in chaos-based communication systems using kernel adaptive filtering. *Journal of Communication and Information Systems*, 34(1):55–63, 2019.

[3] Sonia Boscolo, Tu T Nguyen, Abdallah AI Ali, Stylianos Sygletos, and Andrew D Ellis. Kernel adaptive filtering-based phase noise compensation for pilot-free optical phase conjugated coherent systems. *Optics Express*, 30(11):19479–19493, 2022.

[4] Yijie Tang, Guobing Qian, Wenqi Wu, and Ying-Ren Chien. An efficient filtering algorithm against impulse noise in communication systems. *Journal of Internet Technology*, 24(2):357–362, 2023.

[5] Haiquan Zhao, Yuan Gao, and Yingying Zhu. Robust subband adaptive filter algorithms-based mixture correntropy and application to acoustic echo cancellation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 31:1223–1233, 2023.

[6] Ahmed S Eltrass and Noha H Ghanem. A new automated multi-stage system of non-local means and multi-kernel adaptive filtering techniques for eeg noise and artifacts suppression. *Journal of Neural Engineering*, 18(3):036023, 2021.

[7] Long Shi, Jinghua Tan, Jun Wang, Qing Li, Lu Lu, and Badong Chen. Robust kernel adaptive filtering for nonlinear time series prediction. *Signal Processing*, 210:109090, 2023.

[8] Wenling Li, Zidong Wang, Jun Hu, Junping Du, and Weiguo Sheng. Kernel adaptive filtering over complex networks. *IEEE Transactions on Neural Networks and Learning Systems*, 35(3):4339–4346, 2022.

[9] Weifeng Liu, Puskal P. Pokharel, and Jose C. Principe. The kernel least-mean-square algorithm. *IEEE Transactions on Signal Processing*, 56(2):543–554, 2008.

[10] Tao Yu, Wenqi Li, Rodrigo C de Lamare, and Yi Yu. M-estimate affine projection spline adaptive filtering algorithm: Analysis and implementation. *Digital Signal Processing*, 123:103452, 2022.

[11] Jiacheng He, Gang Wang, Bei Peng, Qi Sun, Zhenyu Feng, and Kun Zhang. Mixture quantized error entropy for recursive least squares adaptive filtering. *Journal of the Franklin Institute*, 359(3):1362–1381, 2022.

[12] Steven Van Vaerenbergh and Ignacio Santamaría. A comparative study of kernel adaptive filtering algorithms. In *2013 IEEE Digital Signal Processing and Signal Processing Education Meeting (DSP/SPE)*, pages 181–186. IEEE, 2013.

[13] Lina Peng, Tao Zhang, Shiyuan Wang, Gangyi Huang, and Shanmou Chen. Diffusion adagrad minimum kernel risk sensitive mean p-power loss algorithm. *Signal Processing*, 202:108773, 2023.

[14] Nikhil Cherian Kurian, Kashyap Patel, and Nithin V George. Robust active noise control: An information theoretic learning approach. *Applied Acoustics*, 117:180–184, 2017.

[15] Mujahid N Syed, Panos M Pardalos, and Jose C Principe. On the optimization properties of the correntropic loss function in data analysis. *Optimization Letters*, 8:823–839, 2014.

[16] Tao Zhang and Shiyuan Wang. Nyström kernel algorithm under generalized maximum correntropy criterion. *IEEE Signal Processing Letters*, 27:1535–1539, 2020.

[17] Badong Chen, Xin Wang, Na Lu, Shiyuan Wang, Jiuwen Cao, and Jing Qin. Mixture correntropy for robust learning. *Pattern Recognition*, 79:318–327, 2018.

[18] Andrew Harvey and Rutger-Jan Lange. Volatility modeling with a generalized t distribution. *Journal of Time Series Analysis*, 38(2):175–190, 2017.

[19] Badong Chen, Songlin Zhao, Pingping Zhu, and Jose C Principe. Quantized kernel recursive least squares algorithm. *IEEE transactions on neural networks and learning systems*, 24(9):1484–1491, 2013.

[20] Tianyu Shen, Weijie Ren, and Min Han. Quantized generalized maximum correntropy criterion based kernel recursive least squares for online time series prediction. *Engineering Applications of Artificial Intelligence*, 95:103797, 2020.

[21] Ji Zhao, Hongbin Zhang, and Gang Wang. Projected kernel recursive maximum correntropy. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 65(7):963–967, 2017.

[22] Yuanlian Huo, Danfeng Wang, Yongfeng Qi, and Peijun Lian. A new gaussian kernel filtering algorithm involving the sparse criterion. *Circuits, Systems, and Signal Processing*, 42(1):522–539, 2023.

[23] Xinyan Hou, Haiquan Zhao, Xiaoqiang Long, and Weidong Jin. The kernel recursive maximum total correntropy algorithm. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 69(12):5139–5143, 2022.

[24] Chang Liu, Xifeng Li, Dongjie Bi, Libiao Peng, and Yongle Xie. Sparse q-laplace kernel online prediction for indoor localization in the internet of things. *Engineering Applications of Artificial Intelligence*, 126:107144, 2023.

[25] Yu Xia and Yeming Wang. Improved hybrid response surface method based on double weighted regression and vector projection. *Mathematical Problems in Engineering*, 2022(1):5104027, 2022.

[26] Wentao Ma, Jiandong Duan, Weishi Man, Haiquan Zhao, and Badong Chen. Robust kernel adaptive filters based on mean p-power error for noisy chaotic time series prediction. *Engineering Applications of Artificial Intelligence*, 58:101–110, 2017.