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

Trimmed Affine Projection Algorithms

Badong Chen, Xiaohan Yang, Hong Ji, Hua Qu, Nanning Zheng, Jose C. Principe

***Abstract*—The least trimmed squares (LTS) estimator is a robust estimator as it can avoid undue influence from outliers. The exact solution of the LTS estimation is however hard to find and if the number of data is large then the method is unfeasible. In this work, we apply the LTS criterion to adaptive filtering and develop the trimmed affine projection algorithm (TAPA) and kernel trimmed affine projection algorithm (KTAPA). The proposed adaptive algorithms are very robust to outliers and have low computational complexity. Simulation results confirm their excellent and robust performance.**

***Index Terms*—**Least trimmed squares (LTS) estimator; affine projection algorithm (APA); kernel affine projection algorithm (KAPA).

I. Introduction

THE classical least squares (LS) estimator usually assumes that errors are Gaussian and i.i.d., and it may perform poorly when the data are non-Gaussian, particularly in the case of large outliers (observations that significantly deviates from the bulk of data). Thus, many robust estimators are proposed to deal with the problem of outliers. The least trimmed squares (LTS) estimator, which minimizes the sum of the M smallest squared residuals, is a well-known high break-down point robust estimator [1-5]. Fig. 1 shows a simple example demonstrating how the performance of the LS deteriorates, whereas that of the LTS is little affected by the outliers. Although the LTS estimator has desirable robustness properties and asymptotic efficiency, the exact computation of LTS is in general computationally very demanding [1-3]. There is a clear academic interest in making the procedure more computationally feasible. In this work, we will apply the LTS criterion to develop some robust adaptive filtering algorithms with low computational complexity.

Adaptive filtering algorithms [6] like the least mean square (LMS), affine projection algorithms (APA) and recursive least squares (RLS) are simple online estimators, which solve the LS problem in an iterative manner. In recent years, kernel adaptive filtering algorithms are also developed, which are a class of nonlinear adaptive filtering algorithms derived in reproducing kernel Hilbert space (RKHS) by using the linear structure (i.e. inner products) of this space to implement the well-established linear adaptive algorithms [7]. The kernel least mean square (KLMS) [8], kernel affine projection algorithms (KAPAs) [9], and kernel recursive least squares (KRLS) [10] are typical examples of the kernel adaptive filtering algorithms.

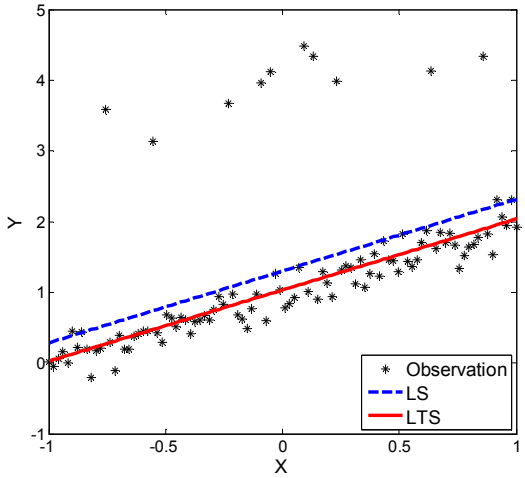

Fig. 1. An example demonstrating the robustness of LTS regression.

The APA algorithms appear as intermediate complexity algorithms between LMS and RLS, which inherit the simplicity of LMS while reducing the gradient noise by using multiple samples [6,7]. They provides a unifying framework for adaptive filtering including the sliding-window RLS. Therefore, in the present paper, our focus is mainly on the APA algorithms and their kernelized versions. The rest of the paper is organized as follows. The LTS estimator is briefly described in section II. The trimmed APA (TAPA) and kernel trimmed APA (KTAPA) are then developed in section III. The new algorithms are robust to outliers and have low computational cost. Encouraging simulation results are provided in section IV and finally, conclusion is given in section V.

II. LTS Estimator

Let us consider the linear regression model:

$$y_i = W^T x_i + \varepsilon_i, \quad i = 1, \cdots, N \tag{1}$$

where $y_i \in \mathbb{R}$ represents the dependent variable (the observed output), $x_i = \left[x_{i,1}, x_{i,2}, \cdots, x_{i,p} \right] \in \mathbb{R}^p$ denotes the input vector, and $W \in \mathbb{R}^p$ stands for the underlying parameter vector (or weigh vector) that needs to be estimated. The term ε_i denotes random noises. In ordinary least squares, the parameter vector W is estimated by minimizing the sum of the squared

Badong Chen, Xiaohan Yang, Hong Ji, Hua Qu and Nanning Zheng are with the School of Electronic and Information Engineering, Xi'an Jiaotong University, China (e-mail: chenbd@mail.xjtu.edu.cn).

Jose C. Principe is with the Department of Electrical and Computer Engineering, University of Florida, USA (e-mail: principe@cnel.ufl.edu).

errors:

$$\hat{W}_{LS} = \arg\min_{W\in\mathbb{R}^p} \sum_{i=1}^{N} e_i^2 = \arg\min_{W\in\mathbb{R}^p} \sum_{i=1}^{N} \left(y_i - W^T x_i\right)^2 \qquad (2)$$

where $e_i = y_i - W^T x_i$ denotes the error sample (residual). It is well-known that the LS estimator is very sensitive to outliers. To address this issue, the trimmed LS estimator was introduced [1]. Let M be an integer such that $0 < M \le N$. The LTS estimator of W is defined as

$$\hat{W}_{LTS} = \arg\min_{W\in\mathbb{R}^p} \sum_{i=1}^{M} \left(e^2\right)_{i,N} \qquad (3)$$

where $\left(e^2\right)_{1,N} \le \cdots \le \left(e^2\right)_{N,N}$ are the ordered squared errors. The LTS method is very robust to outliers and simultaneously possesses desirable asymptotic properties.

Let z_i be the indicator for whether observation i is a good observation or not. The LTS estimator can be obtained by solving the mixed integer programming problem [2, 3]:

$$\begin{aligned} &\min_{W,Z} \sum_{i=1}^{N} z_i \left(y_i - W^T x_i\right)^2 \\ &s.t. \quad \sum_{i=1}^{N} z_i = M \\ &\quad z_i \in \{0,1\} \end{aligned} \qquad (4)$$

Although some efforts have been made to reduce the computational complexity, the exact calculation of the LTS estimator is still very expensive.

III. TRIMMED APA ALGORITHMS

Assume that x_i, y_i and ε_i are i.i.d. samples from random variables x, y and ε. The optimal weight vector W can be obtained by solving

$$\min_W E\left[(y - W^T x)^2\right] + \lambda \|W\|^2 \qquad (5)$$

where E denotes the expectation operator, and $\lambda \ge 0$ is the regularization factor. The closed-form solution of (5) is

$$W = (\lambda I + R_x)^{-1} r_{yx} \qquad (6)$$

where $R_x = E\left[xx^T\right]$, $r_{yx} = E\left[yx\right]$, and I is a $p \times p$ identity matrix. The solution (6) can be recursively solved using a gradient based or Newton's recursion method. For example, we consider

$$\begin{aligned} W_i &= W_{i-1} + \eta\left[r_{yx} - (\lambda I + R_x) W_{i-1}\right] \\ &= (1 - \eta\lambda) W_{i-1} + \eta\left[r_{yx} - R_x W_{i-1}\right] \end{aligned} \qquad (7)$$

where η denotes step-size. The APA algorithm can then be easily derived by approximating R_x and r_{yx} in (7) using the L most recent observations (let i be the current instant):

$$\begin{cases} \hat{R}_x = \dfrac{1}{L} X_i X_i^T \\ \hat{r}_{yx} = \dfrac{1}{L} X_i Y_i \end{cases} \qquad (8)$$

where $X_i = \left[x_{i-L+1}, \cdots, x_i\right]_{p\times L}$, $Y_i = \left[y_{i-L+1}, \cdots, y_i\right]^T$ (x_i and y_i are set zero if the subscripts are less than 1). Combining (7) and (8) yields:

$$W_i = (1 - \eta\lambda) W_{i-1} + \eta X_i \left[Y_i - X_i^T W_{i-1}\right] \qquad (9)$$

Unlike the above ordinary APA algorithm, the trimmed APA (TAPA) algorithm approximates R_x and r_{yx} using M ($0<M\le L$) observations selected from the L most recent observations based on the idea of LTS. Specifically, the matrix X_i and the vector Y_i in (9) are replaced by the matrix $(X)_i$ and vector $(Y)_i$, which are

$$\begin{cases} (X)_i = \left[(x)_{1,i}, \cdots, (x)_{M,i}\right]_{p\times M} \\ (Y)_i = \left[(y)_{1,i}, \cdots, (y)_{M,i}\right]^T \end{cases} \qquad (10)$$

where $\left\{(x)_{j,i}, (y)_{j,i}\right\}$ ($j = 1, \cdots, L$) are ordered L most recent observations such that

$$\left((y)_{1,i} - W_{i-1}^T (x)_{1,i}\right)^2 \le \cdots \le \left((y)_{L,i} - W_{i-1}^T (x)_{L,i}\right)^2 \qquad (11)$$

The pseudocode for TAPA is listed in Table 1.

TAPA Algorithm

Input: $\{x_i, y_i\}, i = 1, 2, \cdots$

Parameter Setting: η, λ, L, M ($0<M\le L$)

Initialization: W_0

Computation:

while $\{x_i, y_i\}$ ($i \ge 1$) **do**

1) compute the $L\times 1$ error vector $\xi_i = \left[e_{i,1}, \cdots, e_{i,L}\right]^T$:

$$\xi_i = Y_i - X_i^T W_{i-1}$$

2) arrange the errors in ascending order of magnitude:

$$(e_{i,j_1})^2 \le \cdots \le (e_{i,j_L})^2$$

3) update the weight vector :

$$W_i = (1 - \eta\lambda) W_{i-1} + \eta (X)_i \zeta_i$$

where

$$\begin{cases} (X)_i = \left[x_{i-L+j_1}, \cdots, x_{i-L+j_M}\right]_{p\times M} \\ \zeta_i = \left[e_{i,j_1}, \cdots, e_{i,j_M}\right]^T \end{cases}$$

end while

Table 1. The pseudocode for TAPA

Remark: The proposed TAPA algorithm is computationally simple. Compared with the ordinary APA algorithm, the only extra computational cost is the ordering of the L errors, which is not significant especially when L is small. If $M = L$, the TAPA algorithm will reduce to the APA algorithm.

One can extend the TAPA algorithm into a high (possibly infinite) dimensional RKHS and derive the kernel TAPA

(KTAPA) algorithm. Let us consider the general nonlinear regression model:

$$y_i = f(x_i) + \varepsilon_i, \quad i = 1, \cdots, N \tag{12}$$

where $f(.)$ is the underlying nonlinear mapping from the input space to output space. Let f_i be the learned mapping in RKHS H_κ induced by a Mercer kernel $\kappa(x_i, x_j)$ [7]. Then the KTAPA algorithm can be derived as

$$f_i = (1 - \eta\lambda) f_{i-1} + \eta \left(K\right)_i \zeta_i \tag{13}$$

where

$$\begin{cases} \left(K\right)_i = \left[\kappa(x_{i-L+j_1}, .), \cdots, \kappa(x_{i-L+j_M}, .)\right] \\ \zeta_i = \left[e_{i,j_1}, \cdots, e_{i,j_M}\right]^T \end{cases} \tag{14}$$

in which the indices $j_1, \cdots, j_M$ are obtained by arranging the error vector $\xi_i = \left[e_{i,1}, \cdots, e_{i,L}\right]^T = Y_i - f_{i-1}(X_i)$ in ascending order of magnitude, where

$$\begin{cases} Y_i = \left[y_{i-L+1}, \cdots, y_i\right]^T \\ f_{i-1}(X_i) = \left[f_{i-1}(x_{i-L+1}), \cdots, f_{i-1}(x_i)\right]^T \end{cases} \tag{15}$$

If the Mercer kernel κ is a radial kernel (e.g. the Gaussian kernel), the KTAPA algorithm will produce naturally a radial basis function (RBF) network with growing size. Denote $C(i)$ the set of RBF centers (the dictionary) at iteration i, and $\alpha(i)$ the corresponding coefficient vector. The learned mapping f_i can be expressed as

$$f_i = \sum_{j=1}^{size(C(i))} \alpha_j(i) \kappa\left(C_j(i), .\right) \tag{16}$$

where $\alpha_j(i)$ and $C_j(i)$ denote, respectively, the j-th element of the coefficient vector $\alpha(i)$ and dictionary $C(i)$. The pseudocode for KTAPA is summarized in Table 2.

KTAPA Algorithm

Input: $\{x_i, y_i\}, i = 1, 2, \cdots$

Parameter Setting: $\kappa, \eta, \lambda, L, M$ ($0 < M \le L$)

Initialization: $f_0 = 0$, $\alpha(0) = \phi$, $C(0) = \phi$

Computation:

while $\{x_i, y_i\}$ ($i \ge 1$) **do**

1) compute the $L \times 1$ error vector $\xi_i = \left[e_{i,1}, \cdots, e_{i,L}\right]^T$:

$$\xi_i = Y_i - f_{i-1}(X_i)$$

2) arrange the errors in ascending order of magnitude:

$$(e_{i,j_1})^2 \le \cdots \le (e_{i,j_L})^2$$

3) update $C(i)$ and $\alpha(i)$ based on (13)

end while

Table 2. The pseudocode for KTAPA

Remark: The computational complexity of KTAPA is similar to the KAPA algorithm and can be significantly reduced if one constrains the network growth using some sparsification or quantization methods [7,11-14].

IV. Simulation Results

We now present simulation results to demonstrate the performance of the proposed algorithms.

A. FIR System Identification

Consider the FIR system identification where the underlying system has transfer function

$$G(z) = 0.5 - 0.8z^{-1} + 1.2z^{-2} + 0.1z^{-3} + 2.4z^{-4} - 1.9z^{-5} + 0.8z^{-6} + 1.7z^{-7} \tag{17}$$

The common input to the unknown system and the adaptive filter (which has the same structure as the unknown system) is a white Gaussian process with unit power. The output of the unknown system is disturbed by an impulsive (long-tailed) noise with symmetric alpha-stable ($S\alpha S$) distribution whose characteristic function is ($\gamma > 0$, $0 < \alpha \le 2$) [15]

$$\psi_{\gamma,\alpha}(\omega) = \exp\left(-\gamma |\omega|^\alpha\right) \tag{18}$$

The alpha-stable noise is very useful in modeling outliers. The performance measure adopted is the mean square deviation (MSD) defined as

$$\text{MSD} = \frac{\|\mathrm{W}^* - \mathrm{W}_i\|^2}{\|\mathrm{W}^*\|^2} \tag{19}$$

where $W^* = [0.5, -0.8, 1.2, 0.1, 2.4, -1.9, 0.8, 1.7]^T$.

First, we show how the values of the trimming constant M affect the performance of TAPA. Let the parameters L, η and λ be $L = 10$, $\eta = 0.01$, $\lambda = 0$. The noise parameters are $\gamma = 0.1$, $\alpha = 1.2$. The average convergence curves (over 100 independent Monte Carlo runs) of TAPA with different trimming constants are shown in Fig. 2. As expected, when the trimming constant is larger, the algorithm will converge faster but will be more sensitive to the outliers.

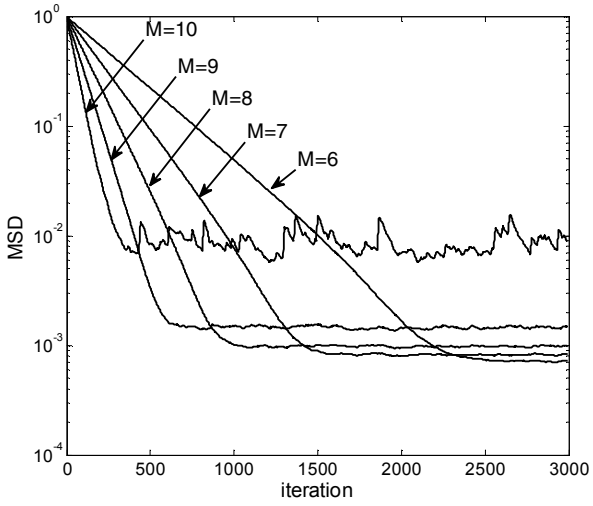

Fig. 2. Average convergence curves with different trimming constants.

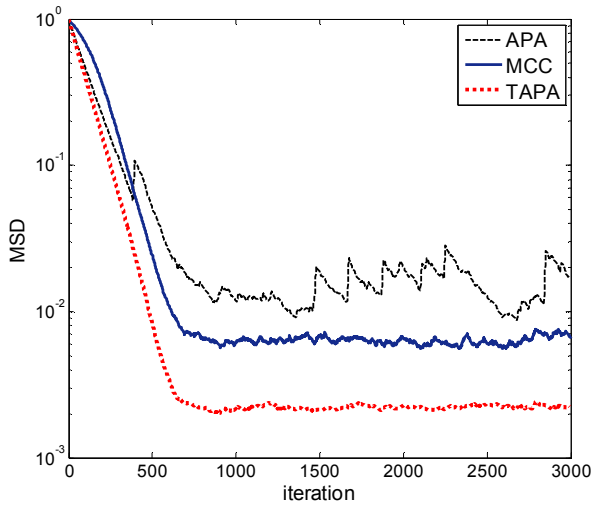

Fig. 3. Average convergence curves (alpha-stable noise).

Algorithms	MSD
APA	0.0181±0.0059
MCC	0.0071±0.0002
TAPA	0.0020±0.0005

Table 3. MSD at final iteration (alpha-stable noise).

Next, we compare the performance of three algorithms: APA, TAPA, and the maximum correntropy criterion (MCC) based adaptation algorithm [16]. The MCC algorithm is also very robust to outliers [16]. The sliding window lengths of APA and TAPA are both set at $L = 10$, and the regularization factor and trimming constant of TAPA are $\lambda = 0$, $M = 8$. The kernel width of MCC is set at 3.5 so as to achieve desirable performance. The step-sizes of the three algorithms are adjusted such that they have similar initial convergence rate. The average convergence curves are illustrated in Fig. 3, and the MSDs at final iteration are listed in Table 3. Simulation results indicate that the TAPA algorithm may perform much better than APA, and is even more robust to outliers than MCC algorithm.

It is worth noting that the performance of TAPA is not always better than APA and MCC. In fact, if the noise distribution is not long-tailed (hence there are few outliers), the APA may perform better than TAPA. This is confirmed by simulation results shown in Fig. 4, where noise is Gaussian distributed with variance 0.04. In this simulation, the kernel width of MCC is set at 2.5, and the step-sizes of the three algorithms are chosen such that the steady-state MSDs are visually identical. One can see clearly from Fig. 4 that the APA achieves a faster convergence speed than both TAPA and MCC. The TAPA achieves a faster initial convergence speed but a slower overall convergence rate than MCC.

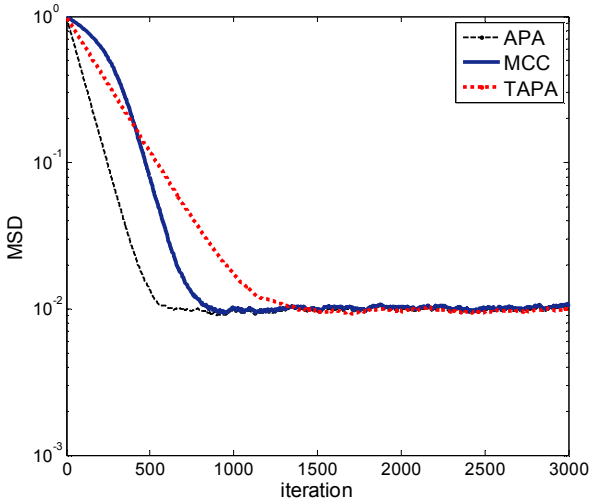

Fig. 4. Average convergence curves (Gaussian noise).

B. Mackey-Glass Chaotic Time Series Prediction

The second example is the Mackey-Glass (MG) time series prediction. The time series is generated from the following time-delay ordinary differential equation: [7]

$$\frac{dx(t)}{dt} = -bx(t) + \frac{ax(t-\tau)}{1+x(t-\tau)^{10}} \tag{20}$$

with $a = 0.2, b = 0.1, \tau = 30$, and is discretized at a sampling period of 6 seconds. Our goal is to predict the present value using the previous seven points. A segment of 1000 samples is used as the training data and another 100 as the test data. The training data are corrupted by additive $S\alpha S$ noise with parameters $\gamma = 0.1$, $\alpha = 1.4$.

In this example, we compare the performance of several adaptive algorithms including APA, TAPA, KLMS, KAPA, and KTAPA. For KLMS, KAPA and KTAPA, the Gaussian kernel with width 1.0 is chosen as the Mercer kernel. For TAPA and KTAPA, the sliding window length is $L = 10$, and the trimming constant is $M = 8$. The step-sizes of these algorithms are experimentally selected so as to achieve a good trade-off between the convergence speed and steady-state accuracy. Fig. 5 illustrates the learning curves averaged over 50 Monte Carlo runs with different segments of data. The testing MSE is calculated based on the 100 test data (during testing the filter is fixed). From Fig. 5 we observe: 1) the kernel adaptive algorithms (KLMS, KAPA, KTAPA) perform much better than linear adaptive algorithms (APA, TAPA) due to their universal approximation property; 2) the trimmed adaptive algorithms (TAPA, KTAPA) perform better than their non-trimmed counterparts (APA, KAPA) due to their robustness to outliers. The testing MSEs at final iteration are summarized in Table 4.

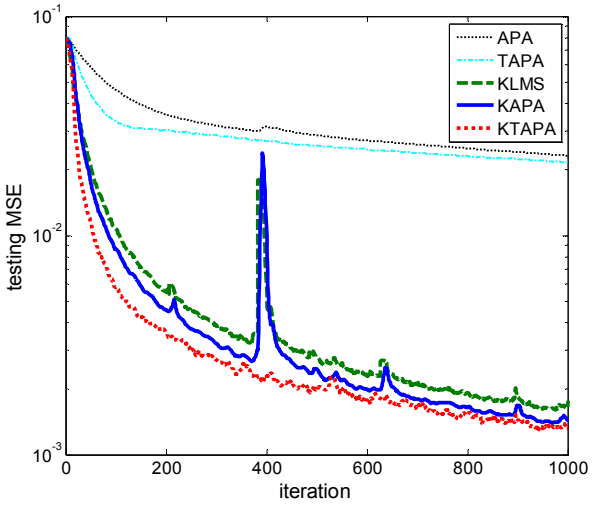

Fig. 5. Average learning curves in MG time series prediction.

Algorithms	Testing MSE
APA	0.023593±0.00025223
TAPA	0.02193±0.00023026
KLMS	0.0016714±5.5721e-05
KAPA	0.0014656±6.7807e-05
KTAPA	0.0013607±3.6702e-05

Table 4. Testing MSE at final iteration.

In [13], a quantization approach is proposed to constrain the network growth of KLMS. This quantization method can also be applied to KTAPA algorithm, and the new algorithm is call the quantized KTAPA (QKTAPA) algorithm. In the following, we compare the performance of KTAPA and QKTAPA. The experimental setting is the same as the previous experiment. For QKTAPA, the quantization size is set at $\varepsilon = 0.2$ (see [13] for the description of online vector quantization) . The average learning curves and the network growth curves are shown in Fig. 6 and Fig. 7, respectively. The testing MSEs at final iteration are given in Table 5. Simulation results suggest that, by properly selecting a quantization size, the network size (number of the RBF centers) of the QKTAPA will decrease significantly (in this example, the network size is reduced to less than 200), while with little loss in performance.

Algorithms	Testing MSE
KTAPA	0.0035484±6.1328e-05
QKTAPA	0.0046321±8.9293e-05

Table 5. Testing MSE of KTAPA and QKTAPA .

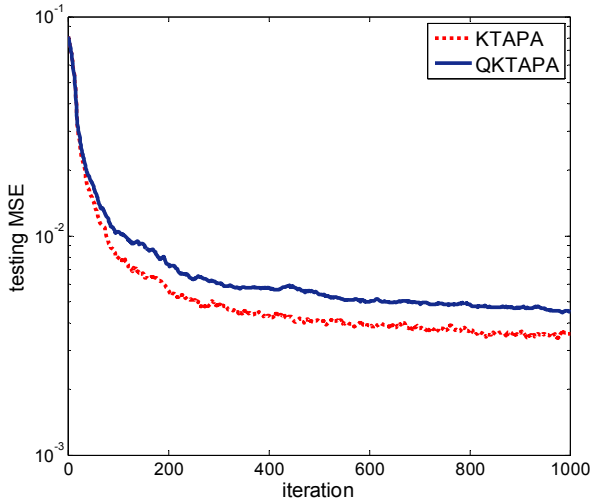

Fig. 6. Average learning curves of KTAPA and QKTAPA.

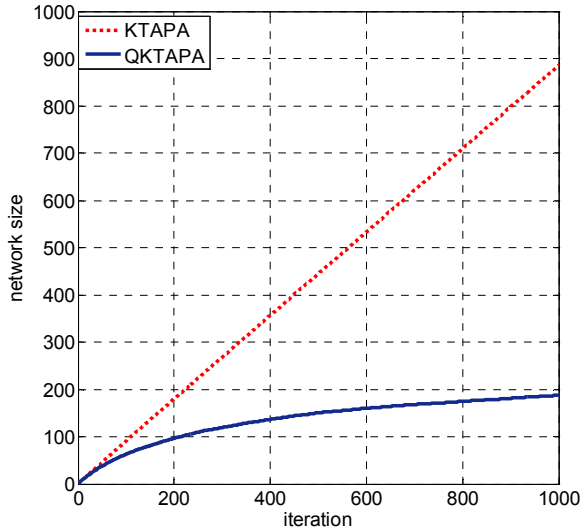

Fig. 7. Network growth curves of KTAPA and QKTAPA.

V. CONCLUSION

In most practical situations, the real-world data obtained from the environment are often contaminated by outliers. This is a crucial problem for adaptive signal processing and machine learning algorithms. Great efforts have been devoted to design robust statistical methods to reduce or even remove the affects of the outliers. The trimmed least squares (LTS) estimation is one of such robust methods, which has very desirable properties and forms the basis for many other robust methods. In this work, we apply the idea of LTS to develop some new adaptive filtering algorithms, namely the trimmed affine projection algorithm (TAPA) and the kernel trimmed affine projection algorithm (KTAPA). The proposed algorithms are very robust to outliers, and their computational complexity is very low compared with the exact calculation of the LTS solution.

ACKNOWLEDGMENT

This work was supported by National Natural Science Foundation of China (no. 61372152), and 973 Program (no. 2012CB316400, 2012CB316402).

REFERENCES

[1] P. Rousseeuw, A. Leroy, *Robust Regression and Outlier Detection*, New York: Wiley, 1987.

[2] E. W. Bai, A random least-trimmed-squares identification algorithm. *Automatica*, 39(9), 1651-1659, 2003.

[3] T. D. Nguyen, R. Welsch, Outlier detection and least trimmed squares approximation using semi-definite programming. *Computational Statistics & Data Analysis*, 54(12), 3212-3226, 2010.

[4] A. Rusiecki, Robust LTS backpropagation learning algorithm. In *Computational and Ambient Intelligence* (p. 102-109). Springer Berlin Heidelberg, 2007.

[5] A. Rusiecki, Robust learning algorithm based on LTA Estimator. *Neurocomputing*,vol. 120 (23), 624-632, 2013.

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

[7] W. Liu, J. Principe, S. Haykin, *Kernel Adaptive Filtering: A Comprehensive Introduction*, Wiley, 2010.

[8] W. Liu, P. Pokharel, J. C. Principe, The kernel least mean square algorithm, *IEEE Transactions on Signal Processing*, vol. 56, pp. 543-554, 2008.

[9] W. Liu, J. Principe, Kernel affine projection algorithm, *EURASIP J. Adv. Signal Process.*, vol. 2008, Article ID 784292, 12 pages, doi: 10.1155/2008/784292.

[10] Y. Engel, S. Mannor, R. Meir, The kernel recursive least-squares algorithm, *IEEE Transactions on Signal Processing*, vol. 52, pp. 2275-2285, 2004.

[11] C. Richard, J. C. M. Bermudez, P. Honeine, Online prediction of time series data with kernels, *IEEE Transactions on Signal Processing*, vol. 57, pp. 1058-1066, 2009.

[12] W. Liu, Il Park, J. C. Principe, An information theoretic approach of designing sparse kernel adaptive filters, *IEEE Transactions on Neural Networks*, vol. 20, pp. 1950-1961, 2009

[13] 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, 2012, pp. 22-32.

[14] B. Chen, S. Zhao, P. Zhu, J. C. Principe, Quantized Kernel Recursive Least Squares Algorithm, *IEEE Transactions on Neural Networks and Learning Systems*, vol. 24, 2013, pp. 1484-1491.

[15] C. L. Nikias, M. Shao, *Signal Processing with Alpha-Stable Distributions and Applications.* Wiley, 1995.

[16] A. Singh, J. C. Principe, Using correntropy as a cost function in linear adaptive filters. In *Neural Networks, 2009. IJCNN 2009. International Joint Conference on* (pp. 2950-2955). IEEE.