Mitigating Outlier Effect in Online Regression: An Efficient Usage of Error Correntropy Criterion

Sajjad Bahrami and Ertem Tuncel
Department of Electrical and Computer Engineering
University of California, Riverside
Riverside, CA, USA
sbahr003@ucr.edu and ertem@ece.ucr.edu

***Abstract*—In this paper, a modified version of maximum correntropy criterion (MCC) with application in online regression (or adaptive filtering) is proposed. It is well known that information theoretic criteria such as error correntropy criterion (ECC) and error entropy criterion (EEC) have the advantage of better performance in supervised learning problems like regression and adaptive filtering when the error between system output and labels (sometimes called desired signals) contains outliers and/or does not follow a Gaussian distribution. Specifically, we improve the existing adaptive maximum correntropy criterion algorithm (known as AMCC) by simply eliminating major outliers during learning process. This elimination leads to better steady state performance than previously known algorithms.**

***Index Terms*—online regression, adaptive filtering, non-Gaussian noise, impulsive noise, outlier rejection, maximum correntropy criterion**

I. Introduction

In learning theory, it is widely established that mean square error (MSE) is not a reliable cost function when the error either is non-Gaussian or contains outliers (i.e., it has a heavy-tailed distribution). Although non-Gaussian error seems to be abnormal based on the central limit theorem, it exists in the real world due to ambient component noises or imperfect measurements, and should not be ignored. As an example, we can point to the existence of impulsive noise in underwater communications [1]–[5]. In this scenario, limited number of noise sources is involved because of limitation in frequency band of operation, and therefore central limit theorem does not hold anymore and we have to deal with non-Gaussian noise.

Two good and efficient alternatives for MSE criterion in aforementioned environments, proposed in an information theoretic sense, are EEC and ECC [6]–[8]. In fact, variance and correlation are substituted with entropy in EEC and correntropy in ECC, respectively. Assume we deal with an adaptive filter (which can be considered as an online regression problem) in which we try to learn the parameters of a system continuously as we are receiving new data samples such that error between system output and labels (or desired signals) is minimized. All we have are data samples and we do not know anything about data statistics. The idea behind using ECC in supervised learning is the fact that correntropy is a local similarity measure between system output and labels. We only have data samples to estimate the correntropy. Correntropy is defined based on the kernel used for Parzen error PDF estimation, and it can be shown that its estimation from samples reduces to estimation of error PDF at zero. This means that maximizing correntropy criterion is equivalent to maximizing estimated evaluation of error PDF at zero, and the resultant algorithm is called maximum correntropy criterion (MCC). Both learning algorithms based on EEC and ECC (called MEE and MCC, respectively) outperform MSE in the presence of non-Gaussian noise or outliers, and are known as robust algorithms for online regression (or adaptive filtering) in such environments. However, MCC has two significant advantages over MEE. First, MEE needs to put the error at the origin after minimizing the error entropy, e.g., by biasing the system output, while this is not always an easy task to do. MCC does not need such regularization [7]. Second, in each adaptation step there is a computational complexity of $O(N^2)$ for MEE while this complexity for MCC is $O(N)$.

MEE and MCC involve kernels in which kernel bandwidth can be viewed as a free parameter that can be also optimized to increase learning precision. Interestingly, this free parameter is much more flexible to be optimized in MCC compared to MEE. Although MEE and MCC have been compared to each other in some senses (for instance a new interesting information theoretic comparison of MEE and MCC criteria can be found in [9]), this superiority of MCC (i.e., more flexibly adaptable kernel bandwidth) has not been paid enough attention. This flexibility in kernel bandwidth adaptation for MCC emerges from smooth dependency of MCC to the value of the kernel bandwidth since error PDF estimation does not matter when we consider MCC (this will become more clear in the sequel) while in MEE, the kernel bandwidth can not be selected arbitrarily in the sense that very large or very small values for kernel bandwidth are prohibited. The reason for this prohibition is that MEE involves directly entropy estimation, and consequently Parzen error PDF estimation is involved, therefore the value of kernel bandwidth should be selected in a way that it compromises between bias and variance of Parzen error PDF estimation.

Adaptation of kernel bandwidth as an extra step in learning process has been already considered in the literature [10]–[17]. For instance, in [10] and [11] Kullback-Leibler (KL) divergence between the true and estimated error distribution is minimized as a second cost function in the overall adaptation problem. However, this approach will not be very efficient

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

(for instance when the initial weight vector is far from the optimal weight vector, it can not improve the convergence speed [18]). In another work [12], the author tried to adapt the kernel bandwidth for MCC based on the shape of error distribution which is measured by its kurtosis. However, a satisfying estimation of the shape of error distribution is not a straightforward task to do. Authors in another paper [13] tried to propose a simple algorithm for kernel bandwidth adaptation that involves no extra free parameters. In this algorithm, the kernel bandwidth in each iteration is updated based on instantaneous error and changed to a predetermined kernel bandwidth in order to avoid divergence when the updated kernel bandwidth is smaller than this predetermined value. Although this approach converges faster than previous algorithms, it almost keeps their steady-state behavior. The same authors modified their method in [14] in which they used another kernel bandwidth update rule. Although it is still based on instantaneous error and predetermined kernel bandwidth, the new update rule helps the method not only to converge faster but also to achieve a slightly lower steady state excess mean square error (EMSE) than that of MCC. Authors in [15] changed the Gaussian kernel used in the MCC definition and developed a new algorithm to update kernel bandwidth which converges faster than previous algorithms, and even can achieve a lower steady state misalignment especially in the environment where impulsive noise is considerably likely. Although the structure of their algorithm is very similar to previous gradient based algorithms, it is computationally more expensive compared to them. Recently, we proposed a hybrid method in [17] which achieves both fast convergence and low steady state misalignment, however many parameters are left to be predetermined. In addition, this approach is very sensitive to step size variation.

In this paper, we consider linear adaptive filtering (or online linear regression problem) used in various signal processing applications such as communication channel estimation, noise cancellation, system identification, etc. [19]. We modify AMCC algorithm in [14] by stopping the learning process in each step whenever we face a major outlier. This method results in significant decrease in steady state misalignment at the cost of some extra computations in order to check whether an error sample is a major outlier or not in each iteration.

The remainder of the paper is organized as follows. In Section II, we give a brief review on correntropy as an objective function. In Section III, our method is proposed. Section IV is devoted to presenting simulation results. Finally, we conclude the paper in Section V.

Throughout the paper we use the terms adaptive filtering and online regression interchangeably, and denote random variables by uppercase letters and their realization by lowercase ones. In addition, we use boldface letters to show vectors, and $\| \, . \, \|$ denotes 2 norm.

II. Correntropy as an Objective Function

Correntropy is defined as,

$$v\left(D,Y\right)=E_{D,Y}\left\{G_{\sigma}\left(D-Y\right)\right\}=E_{E}\left\{G_{\sigma}\left(E\right)\right\}, \quad (1)$$

in which D and Y are two random variables, E denotes the error $D-Y$, $G_{\sigma}(.)$ is a kernel function (usually Gaussian kernel) and σ is kernel bandwidth which affects the shape of the kernel function. As we only have data samples $\{d_n, y_n\}$, $n = 1 \cdots N$, (or equivalently $\{e_n\}, n = 1 \cdots N$) not joint PDF of D and Y (or equivalently not PDF of error E), we use a sample mean to approximate expectation in correntropy definition (1), i.e.

$$\hat{v}\left(D,Y\right)=\frac{1}{N}\sum_{n=1}^{N}G_{\sigma}\left(d_{n}-y_{n}\right)=\frac{1}{N}\sum_{n=1}^{N}G_{\sigma}\left(e_{n}\right), \quad (2)$$

in which we consider a Gaussian kernel, i.e.,

$$G_{\sigma}\left(D-Y\right)=\exp(-\frac{\parallel D-Y \parallel^{2}}{2\sigma^{2}}).$$

Equations (1) and (2) suggest that correntropy can be considered as a similarity measure between two random variables D and Y. In linear adaptive filtering, D is a random variable denoting labels (or sometimes called desired signal) and Y denotes the output of learned linear filter. However, why can we use correntropy as an objective function to minimize the error between labels and filter output? The answer is in the relation between approximations of correntropy in (2) and PDF of error. To be more clear, note that we can estimate a PDF based on the samples of that PDF using a non-parametric method called Parzen windows [20]. Using Parzen windows for error PDF estimation from N samples, we have:

$$\hat{p}_{E}\left(e\right)=\frac{1}{N}\sum_{n=1}^{N}G_{\sigma}\left(e-e_{n}\right). \quad (3)$$

By comparing (2) with (3) we can see $\hat{v}\left(D,Y\right)=\hat{p}_{E}\left(0\right)$. This means that maximizing the estimate of error PDF at 0 is equivalent to maximizing the estimate of correntropy. The correntropy maximization algorithm is called MCC in literature.

It is well known that for environments with Gaussian noise, MSE gives the optimum solution [21]. However, in non-Gaussian environments we need higher-order statistics [6] and this is why cornetropy outperforms MSE in such environments. Indeed, we can use Taylor expansion of Gaussian kernel in (1) and see that correntropy contains even-order moments of the error not only second-order moment as MSE [14], [22], i.e. (1) can be written as follows:

$$v\left(D,Y\right)=\frac{1}{\sqrt{2\pi}\sigma}\sum_{n=0}^{\infty}\frac{(-1)^{n}}{2^{n}\sigma^{2n}n!}E_{D,Y}\left\{\left(D-Y\right)^{2n}\right\}.$$

Moreover, to see how correntropy outperforms MSE in presence of outliers, it suffices to take a careful look at (2). Outliers are abnormally large error samples resulted from impulsive noise. As seen in (2), error samples are given a weight by the Gaussian kernel where error samples with smaller values have larger weights and consequently larger contribution in objective function utilized in adaptation. On the other hand, large error samples, or outliers, are given small weights and are filtered out by Gaussian kernel.

In filtering out the outliers, kernel bandwidth σ has a significant role. In fact, kernel bandwidth determines the magnitude of weight that should be assigned to a specific error sample. Generally, it affects the convergence rate and steady state misalignment, therefore a proper method for adaptation of σ will increase adaptive filtering efficiency.

Throughout the paper we try to learn the parameters of a linear adaptive filter. As mentioned earlier, d_n denotes label (or desired signal) at time instant n. Note that d_n is corrupted with measurement noise. Furthermore, $\mathbf{x}_n = \left[x_n, x_{n-1}, \cdots, x_{n-L+1}\right]^T$ is the input vector at time instant n with length L where L denotes the size of adaptive filter as well. We choose linear model for adaptive filter, therefore the output of this filter is $y_n = \mathbf{x}_n^T \mathbf{w}_{n-1}$ where $\mathbf{w}_{n-1}$ is filter parameter estimated at time instant $n-1$. Error sample at time n is obtained as $e_n = d_n - \mathbf{x}_n^T \mathbf{w}_{n-1}$.

Correntropy is a bounded function [22]. We use gradient ascent method to maximize correntropy. For simplicity, we consider stochastic gradient ascent in which we drop the expectation operator in (1) and only use the current error sample to approximate the correntropy. Therefore we have following online objective function:

$$J_{MCC}\left(\mathbf{w}_{n-1}\right) = G_\sigma\left(d_n - \mathbf{x}_n^T \mathbf{w}_{n-1}\right) = G_\sigma\left(e_n\right). \quad (4)$$

Then, gradient ascent algorithm is as follows:

$$\begin{aligned} \mathbf{w}_n &= \mathbf{w}_{n-1} + \mu \nabla J_{MCC}\left(\mathbf{w}_{n-1}\right) \\ &= \mathbf{w}_{n-1} + \frac{\mu}{\sigma^2} \exp\left(\frac{-e_n^2}{2\sigma^2}\right) e_n \mathbf{x}_n, \end{aligned} \quad (5)$$

where μ is step size and $\nabla J_{MCC}\left(\mathbf{w}_{n-1}\right)$ denotes the gradient of online objective function (4) with respect to $\mathbf{w}_{n-1}$.

In next section, we propose a method that achieves a lower steady state misalignment compared to previous algorithms.

III. Proposed Method

As mentioned earlier, we modify AMCC algorithm to achieve lower steady state misalignment. Therefore, we briefly review AMCC algorithm first, then move forward to explaining the modification.

A. AMCC Algorithm

Consider linear adaptive filtering in an environment with impulsive noise. Kernel bandwidth σ_n at each iteration is obtained such that (5) approaches to its optimum weight value faster. To this end, kernel bandwidth σ_n should maximize following term of (5) in each iteration:

$$h\left(\sigma^2\right) = \frac{1}{\sigma^2} \exp\left(-\frac{e_n^2}{2\sigma^2}\right). \quad (6)$$

Note that at each iteration n, error sample e_n has already been determined before updating kernel bandwidth, therefore we only need to take derivative of (6) with respect to σ^2. We have:

$$\begin{aligned} &\frac{\partial h\left(\sigma^2\right)}{\partial\left(\sigma^2\right)} = \\ &-\frac{1}{\sigma^4} \exp\left(-\frac{e_n^2}{2\sigma^2}\right) + \left(\frac{e_n^2}{2\left(\sigma^2\right)^3}\right) \exp\left(-\frac{e_n^2}{2\sigma^2}\right) = 0 \\ &\Longrightarrow \sigma_n^2 = \left\{\frac{{e_n}^2}{2}, \infty\right\}. \end{aligned}$$

Clearly, we have:

$$h\left(\sigma^2\right)\Big|_{\sigma^2 = \frac{e_n^2}{2}} = \frac{2}{e_n^2} \exp\left(-1\right) > \lim_{\sigma^2 \to \infty} h\left(\sigma^2\right) = 0,$$

therefore, $\sigma_n^2 = \frac{e_n^2}{2}$ maximizes the function $h\left(\sigma^2\right)$ for a fixed e_n. Figure 1 shows how $h\left(\sigma^2\right)$ in (6) varies with σ^2 for different values of e_n. As seen in Figure 1, the maximum of function $h\left(\sigma^2\right)$ which obtained at $\sigma_n^2 = \frac{e_n^2}{2}$ tends to infinity as e_n tends to zero, therefore in order to avoid divergence of the algorithm (5) when e_n is getting smaller, adaptive kernel bandwidth is modified as follows:

$$\sigma_n^2 = \frac{e_n^2}{2} + \sigma_0^2, \quad (7)$$

in which, as explained in [14], kernel bandwidth at iteration n switches to a predetermined kernel bandwidth σ_0 when error e_n is small.

Thus far we only talked about an adaptive kernel bandwidth for MCC that speeds up the convergence rate. From now on we focus on decreasing steady state misalignment.

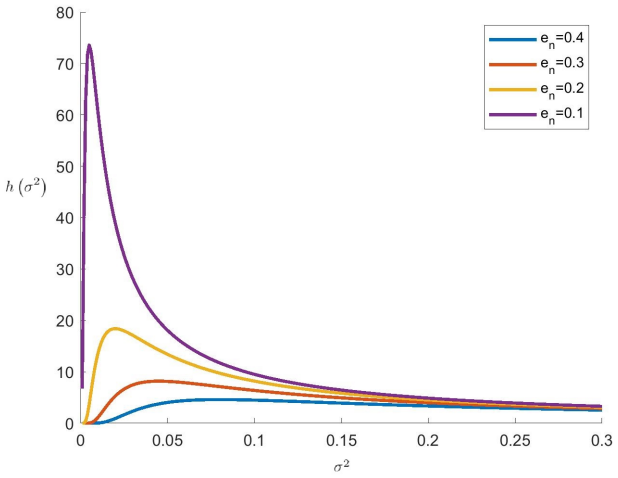

Fig. 1. Plot of the function $h\left(\sigma^2\right)$ for different values of e_n.

B. Modification to AMCC Algorithm

In order to reach a lower steady state misalignment we employ a filter with variable bandwidth in each iteration to reject major outliers. In fact, we modify the algorithm (5) as follows:

$$\mathbf{w}_n = \mathbf{w}_{n-1} + \frac{\mu}{\sigma_n^2} f\left(e_n\right) \exp\left(\frac{-e_n^2}{2\sigma_n^2}\right) e_n \mathbf{x}_n, \quad (8)$$

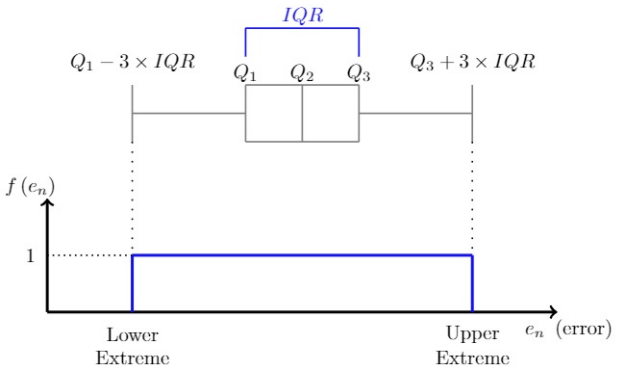

Fig. 2. Plot of the function $f(e_n)$ at time instant n.

in which $f(e_n)$ is the filter and σ_n^2 is substituted from (7). However, what are the boundaries that specify major outliers? In other words, how we can determine the variable bandwidth of the filter $f(e_n)$? We use running quartiles of the error samples. Generally speaking, median (or generally any quantile) of a data set is a robust quantity of data against outliers, therefore we can use the concept of outer fences to determine filter boundaries for major outlier rejection [23], [24]. Figure 2 denotes these boundaries. In this figure, Q_1, Q_2 and Q_3 are lower quartile (or 25th percentile), median, and upper quartile (or 75th percentile), respectively. In addition, $IQR = Q_3 - Q1$ stands for inter-quartile range, and outer fences are shown by:

$$lower\ extreme = Q_1 - 3 \times IQR,$$
$$upper\ extreme = Q_3 + 3 \times IQR.$$

Note that quartiles are functions of time n which means these filter extremes vary once new error sample is available, therefore we have to deal with running quartiles. Running quartile estimation from data samples has been widely studied in literature [25]–[29]. In this paper, we simply use order statistics in which we sort all observation samples at each time instant n. The complexity of this operation is $O(n)$ and we need to store all previous data samples, however we could choose a proper algorithm from aforementioned algorithms in order to decrease memory or computational requirements. Our algorithm is proposed in the following.

IV. Simulation Results

In this section, our simulation results show how our method outperforms other methods in term of steady state misalignment. We use the model in [30], [31] and [15] for impulsive noise environment. We assume that input samples x_i are drawn as $x_i \sim \mathcal{N}(0,1)$. Optimum weight vector of unknown filter is generated randomly and is a unit vector $\mathbf{w}_{opt} \in \mathcal{R}^L$ where $L = 5$ denotes the filter length. The desired signal at time instant n is modeled as,

Algorithm 1 Our Algorithm for Online Linear Regression

Inputs: $\{\mathbf{x}_n, d_n\}$
Output: $\mathbf{w}_n$
Initialisation : $\mathbf{w}_0 = \mathbf{0}$ and σ_0
for each iteration n **do**
  $e_n = d_n - \mathbf{x}_n^T \mathbf{w}_{n-1}$
  $\sigma_n^2 = \frac{e_n^2}{2} + \sigma_0^2$
  Sort error samples and find Q_1 and Q_3
  $IQR = Q_3 - Q_1$
  Lower Extreme $= Q_1 - 3 \times IQR$
  Upper Extreme $= Q_3 + 3 \times IQR$
  if (Lower Extreme $\leq e_n \leq$ Upper Extreme) **then**
    $f(e_n) = 1$
  else
    $f(e_n) = 0$
  end if

$$\mathbf{w}_n = \mathbf{w}_{n-1} + \frac{\mu}{\sigma_n^2} f(e_n) \exp\left(\frac{-e_n^2}{2\sigma_n^2}\right) e_n \mathbf{x}_n$$

end for
return $\mathbf{w}_n$

$$d_n = \mathbf{x}_n^T \mathbf{w}_{opt} + \nu_n + \eta_n,$$

where $\nu_n \sim \mathcal{N}(0, \sigma_{\nu,n}^2)$ and η_n are white Gaussian and impulsive measurement noises, respectively. We assume that there is 30dB signal to white Gaussian measurement noise ratio where this signal to noise ratio (SNR) is calculated as follows:

$$\mathrm{SNR} = 10 \log_{10}\left(\frac{E\left\{\left[\mathbf{x}_n^T \mathbf{w}_{opt}\right]^2\right\}}{\sigma_{\nu,n}^2}\right).$$

Impulsive measurement noise is created as $\eta_n = \beta_n \omega_n$ where $\beta_n \sim Bernoulli(p)$ in which p is probability of success (or equivalently the probability of existence of impulses in noise) and $\omega_n \sim \mathcal{N}\left(0, 1000E\left\{\left[\mathbf{x}_n^T \mathbf{w}_{opt}\right]^2\right\}\right)$. We assume $p = 0.2$ in our simulations. At each time instant n, we obtain $\mathbf{w}_n$, and accordingly we obtain misalignment based on the following normalized mean-square deviation (NMSD):

$$\mathrm{misalignment}_n = 10 \log_{10}\left(\frac{\| \mathbf{w}_n - \mathbf{w}_{opt} \|^2}{\| \mathbf{w}_{opt} \|^2}\right).$$

First, consider (5) in which there is no filter $f(e_n)$, and $\sigma_n^2 = \frac{e_n^2}{2} + \sigma_0^2$. Figure 3 shows how using (7) for σ_n^2 can increase convergence rate. Figure 4 shows how learning curves of (5) with σ_n from (7) vary with different values of predetermined kernel bandwidth σ_0. As seen in this figure, by increasing predetermined kernel bandwidth σ_0 convergence rate always decreases while steady state misalignment decreases first and then it increases.

Now, what happens when we employ filter $f(e_n)$ in (5), i.e. when we use (8). As illustrated in Figure 5, when we employ the filter $f(e_n)$, learning curve always converges slower to

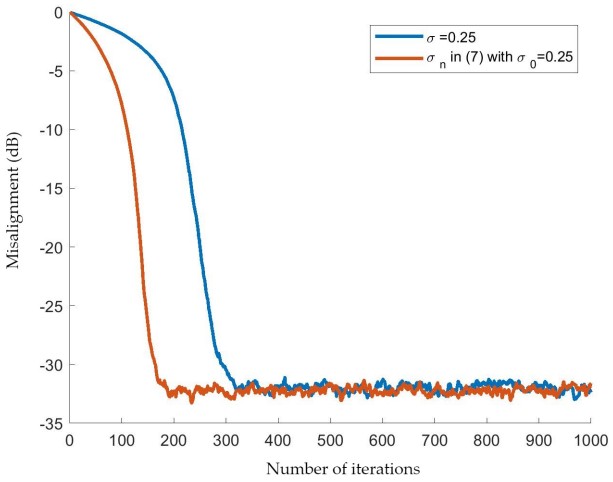

Fig. 3. Learning curves of MCC (5) with fixed $\sigma = 0.25$ and with variable σ_n from (7) with $\sigma_0 = 0.25$ ($\mu = 0.01$).

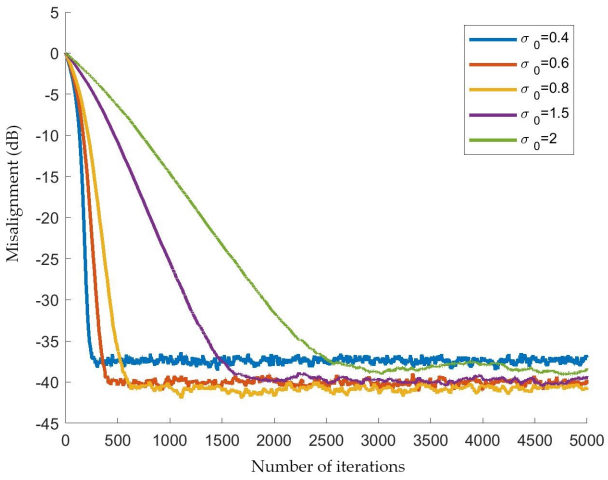

Fig. 4. Learning curves of MCC (5) with variable σ_n from (7) with different values of σ_0 ($\mu = 0.01$).

a lower steady state misalignment when we increase predetermined kernel bandwidth σ_0. Let us discuss the learning curve behaviour in Figures 4 and 5. As seen in these figures, by increasing predetermined kernel bandwidth σ_0 in (5) and (8) the gradient ascent whole step size $\frac{\mu}{\sigma_n^2}$ decreases and both algorithms converge slower. However, the steady state misalignment behaviour of our algorithm is different with that of (5). The reason is that when there is no filter $f(e_n)$, although we are decreasing whole step size value $\frac{\mu}{\sigma_n^2}$ with increasing σ_0 and we expect to achieve lower steady state misalignment with iterations, at the same time we are giving a big weight to outliers (i.e., large error samples) based on the correntropy definition which can result in higher steady state misalignment. Therefore there is a tradeoff between these two factors and once σ_0 is large enough the latter factor dominates the other one. We observe that this issue has been resolved in our algorithm as shown in Figure 5 in which learning curve always achieves lower steady state misalignment with increase in σ_0.

Figure 6 illustrates how learning curve of our algorithm changes with step size μ. As expected, larger step size results in faster convergence to a higher steady state misalignment.

Finally, Figure 7 shows how our proposed method in section III outperforms other algorithms from steady state misalignment point of view. These learning curves for 20000 iterations are obtained by averaging over 10 independent trials. Step size μ is set to 0.01. As seen, the LMS algorithm diverges to a high steady state misalignment when impulses occur in the noise (or equivalently when we have outlier error samples). AMCC algorithm in [14] outperform LMS when there is impulse in noise. VKW-MCC in [15] is both faster and achieves a lower misalignment compared to previous algorithms. Finally, our algorithm converges to the lowest steady state misalignment compared to other algorithms.

Note that we could combine our algorithm with a fast algorithm (e.g., recursive MCC) and propose a hybrid method

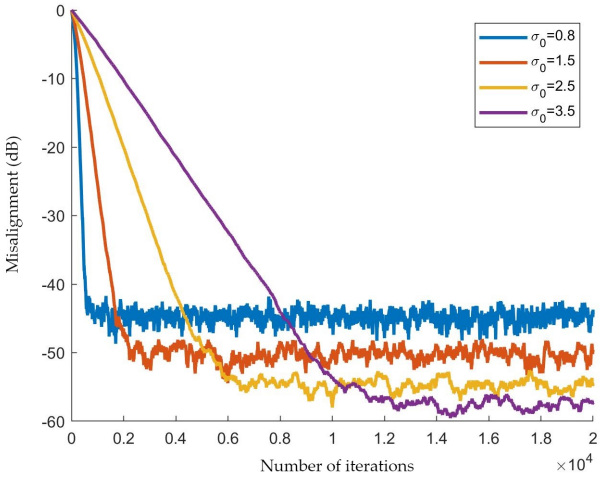

Fig. 5. Learning curves of our algorithm (8) with different values of σ_0 ($\mu = 0.01$).

like [17] in which the overall hybrid algorithm not only achieves a lower steady state misalignment but also converges faster.

V. CONCLUSIONS

This paper addresses the problem of online linear regression (or linear adaptive filtering) which has applications such as channel estimation. We consider the presence of outliers and impulsive noise in the environment. Correntropy is well known as a reliable cost function in such environments. In this paper, we use error samples running quartiles to find out whether a new error sample is a major outlier or not. If it is, we stop learning process (in which we use an existing algorithm called AMCC) based on that error sample and wait for next error sample to continue the learning process. Simulation results show that our algorithm achieves more accurate steady state performance compared to previous results.

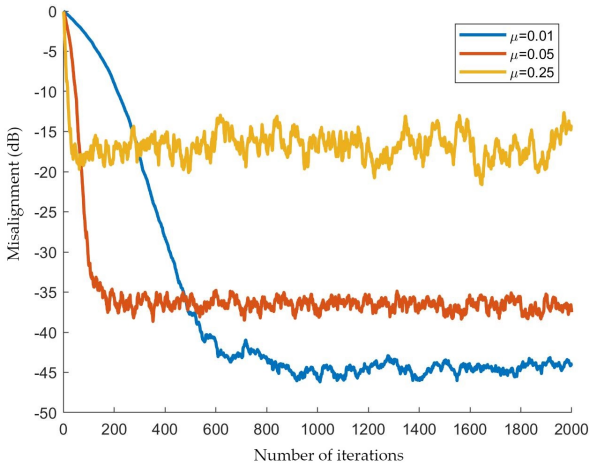

Fig. 6. Learning curves of our algorithm (8) for $\sigma_0 = 0.8$ and different values of step size μ.

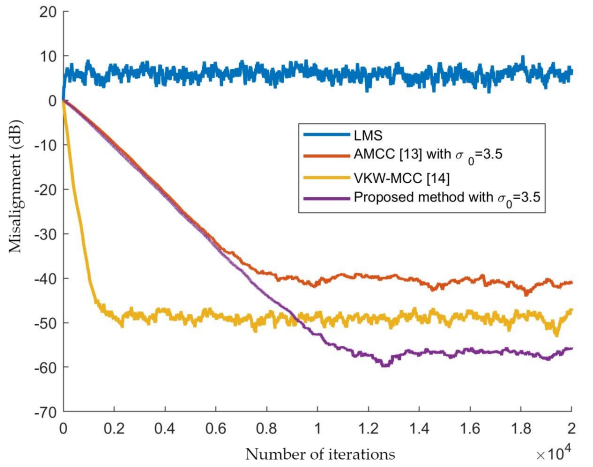

Fig. 7. Learning curves of different algorithms ($\mu = 0.01$).

REFERENCES

[1] M. Stojanovic and J. Preisig, "Underwater acoustic communication channels: Propagation models and statistical characterization," in IEEE Communications Magazine, vol. 47, no. 1, pp. 84-89, January 2009.

[2] X. Kuai, H. Sun, S. Zhou and E. Cheng, "Impulsive Noise Mitigation in Underwater Acoustic OFDM Systems," in IEEE Transactions on Vehicular Technology, vol. 65, no. 10, pp. 8190–8202, Oct. 2016.

[3] S. Banerjee and M. Agrawal, "On the performance of underwater communication system in noise with Gaussian Mixture statistics," 2014 Twentieth National Conference on Communications (NCC), Kanpur, 2014, pp. 1-6.

[4] P. Chen, Y. Rong, S. Nordholm, Z. He and A. J. Duncan, "Joint channel estimation and impulsive noise mitigation in underwater acoustic OFDM communication systems," in IEEE Transactions on Wireless Communications, vol. 16, no. 9, pp. 6165–6178, June 2017.

[5] P. Chen, Y. Rong, S. Nordholm and Z. He, "Joint channel and impulsive noise estimation in underwater acoustic OFDM systems," in IEEE Transactions on Vehicular Technology, vol. 66, no. 11, pp. 10567–10571, Aug. 2017.

[6] Principe, Jose C. Information theoretic learning: Renyi's entropy and kernel perspectives. Springer Science & Business Media, 2010.

[7] W. Liu, P. P. Pokharel and J. C. Principe, "Error Entropy, Correntropy and M-Estimation," 2006 16th IEEE Signal Processing Society Workshop on Machine Learning for Signal Processing, Arlington, VA, 2006, pp. 179-184.

[8] W. Liu, P. P. Pokharel and J. C. Principe, "Correntropy: Properties and Applications in Non-Gaussian Signal Processing," in IEEE Transactions on Signal Processing, vol. 55, no. 11, pp. 5286-5298, Nov. 2007.

[9] A. R. Heravi and G. A. Hodtani, "A New Information Theoretic Relation Between Minimum Error Entropy and Maximum Correntropy," in IEEE Signal Processing Letters, vol. 25, no. 7, pp. 921-925, July 2018.

[10] A. Singh and J. C. Príncipe, "Kernel width adaptation in information theoretic cost functions," 2010 IEEE International Conference on Acoustics, Speech and Signal Processing, Dallas, TX, 2010, pp. 2062-2065.

[11] H. Radmanesh and M. Hajiabadi, "Recursive Maximum Correntropy Learning Algorithm With Adaptive Kernel Size," in IEEE Transactions on Circuits and Systems II: Express Briefs, vol. 65, no. 7, pp. 958-962, July 2018.

[12] S. Zhao, B. Chen and J. C. Príncipe, "An adaptive kernel width update for correntropy," The 2012 International Joint Conference on Neural Networks (IJCNN), Brisbane, QLD, 2012, pp. 1-5.

[13] W. Wang, J. Zhao, H. Qu, B. Chen and J. C. Principe, "A switch kernel width method of correntropy for channel estimation," 2015 International Joint Conference on Neural Networks (IJCNN), Killarney, 2015, pp. 1-7.

[14] W. Wang, J. Zhao, H. Qu, B. Chen and J. C. Principe, "An adaptive kernel width update method of correntropy for channel estimation," 2015 IEEE International Conference on Digital Signal Processing (DSP), Singapore, 2015, pp. 916-920.

[15] F. Huang, J. Zhang and S. Zhang, "Adaptive Filtering Under a Variable Kernel Width Maximum Correntropy Criterion," in IEEE Transactions on Circuits and Systems II: Express Briefs, vol. 64, no. 10, pp. 1247-1251, Oct. 2017.

[16] L. Shi and H. Zhao and Y. Zakharov, "An Improved Variable Kernel Width for Maximum Correntropy," in IEEE Transactions on Circuits and Systems II: Express Briefs, Nov. 2018.

[17] S. Bahrami and E. Tuncel, "A new approach to online regression based on maximum correntropy criterion," 2019 IEEE International Workshop on Machine Learning for Signal Processing, 2019, Pittsburgh, PA, USA.

[18] W. Wang, J. Zhao, H. Qu, B. Chen, and J. C. Principe, "Convergence performance analysis of an adaptive kernel width MCC algorithm," AEU Int. J. Electron. Commun., vol. 76, pp. 71–76, Jun. 2017.

[19] Haykin, S. Adaptive Filter Theory. Prentice-Hall, 2002.

[20] Gramacki, A. Nonparametric Kernel Density Estimation and Its Computational Aspects. Springer International Publishing, 2018.

[21] Kay S. M. Fundamentals of Statistical Signal Processing. Prentice Hall PTR, 1993.

[22] S. Zhao, B. Chen and J. C. Principe. "Kernel adaptive filtering with maximum correntropy criterion," 2011 International Joint Conference on Neural Networks (IJCNN), San Jose, CA, 2011, pp. 2012-2017.

[23] Tukey, JW. Exploratory data analysis. Addison-Wesely, 1977.

[24] Suri, N., Murty, N. and Athithan, G. Outlier Detection: Techniques and Applications. Springer Nature, 2019.

[25] R. Jain and I. Chlamtac, "The P^2 algorithm for dynamic calculation of quantiles and histograms without storing observations," in Communications of the ACM, vol. 28, no. 10, pp. 1076–1085, 1985.

[26] H. L. Hammer and A. Yazidi, "Smooth estimates of multiple quantiles in dynamically varying data streams," Pattern Analysis and Applications, pp. 1–12, 2019.

[27] N. Tiwari and P. C. Pandey, "A technique with low memory and computational requirements for dynamic tracking of quantiles," Journal of Signal Processing Systems, vol. 91, no. 5, pp. 411–422, 2019.

[28] O. Arandjelović, "Targeted Adaptable Sample for Accurate and Efficient Quantile Estimation in Non-Stationary Data Streams," Machine Learning and Knowledge Extraction, vol. 1, no. 3, pp. 848–870, 2019.

[29] S. Bahrami and E. Tuncel, "An Efficient Running Quantile Estimation Technique alongside Correntropy for Outlier Rejection in Online Regression," to appear in 2020 IEEE International Symposium on Information Theory (ISIT), 2020, Los Angeles, CA, USA.

[30] I. Song, P. Park and R. W. Newcomb, "A Normalized Least Mean Squares Algorithm With a Step-Size Scaler Against Impulsive Measurement Noise," in IEEE Transactions on Circuits and Systems II: Express Briefs, vol. 60, no. 7, pp. 442-445, July 2013.

[31] F. Huang, J. Zhang and S. Zhang, "Combined-Step-Size Affine Projection Sign Algorithm for Robust Adaptive Filtering in Impulsive Interference Environments," in IEEE Transactions on Circuits and Systems II: Express Briefs, vol. 63, no. 5, pp. 493-497, May 2016.