A Parallel Adaptive Filtering Algorithm Based on the Mean-Square Deviation Analysis for Large-Scale Data

Sang Mok Jung *
* Agency for Defense Development, Daejoen, South Korea
E-mail: illus2@postech.ac.kr

***Abstract*—This paper proposes a parallel adaptive filtering algorithm via analysis of the mean-square deviation for large-scale data. In this algorithm, large-scale data is divided into several sub-blocks to reduce the computational cost. Based on each data sub-block, a normalized least-mean-square algorithm estimates the parameters of interest at each sub-filter. Furthermore, the mean-square deviation analysis of the estimation result at each sub-filter leads to a variable step-size method and an intermittent-update method. These methods provide not only fast convergence rate and small steady-state error but also high computational efficiency. Finally, the estimation results of each sub-filter are combined through a combination method by determining the weights for each estimation result based on their error variance. The proposed combination method provides robustness to abnormalities of the data. Simulation results show that the proposed algorithm performs well for estimation with large-scale data.**

I. Introduction

Large-scale data has received wide attention from researchers in various fields because volumes of data are rapidly grown by the Internet, social media, and mobile devices. In this era of data deluge, handling large-scale data has been one of the most noteworthy research in signal processing areas [1-6].

There are important challenges to deal with large-scale data because it often degrades performance of traditional signal processing methods. Using large-scale data sets without considering the characteristic of the data may be infeasible or limit the applicability of traditional methods. Many signal processing areas are currently facing this challenge of coping with massive volume and dimensionality of data [7]. As part of response to these challenges, Large-scale data optimization problems have been widely studied such as compressed sensing methods [8-10], high-order tensors [11], [12], coordinate descent methods [13-15].

Furthermore, adaptive stochastic gradient algorithms such as least-mean-square (LMS) and recursive least-squares (RLS) algorithms for big data have also received a great deal of attention [16]. Such adaptive filtering algorithms can be widely used in online learning for large-scale data analytics because of its computational simplicity and ease of implementation [16-18]. Analyzing and processing of large-scale data poses significant challenges in the adaptive filtering algorithms because of the data dimensionality. In general, sub-samples of given large-scale data are used for solving optimization problems [19-22]. They choose and use only a few specific sub-sample data to alleviate the computational burden. However, each sub-sample of data can have valuable and important information even if it is not chosen. Therefore, new paradigms and techniques are required for large-scale data on the adaptive filters, which can consider not only the computational efficiency but also importance of all the given data.

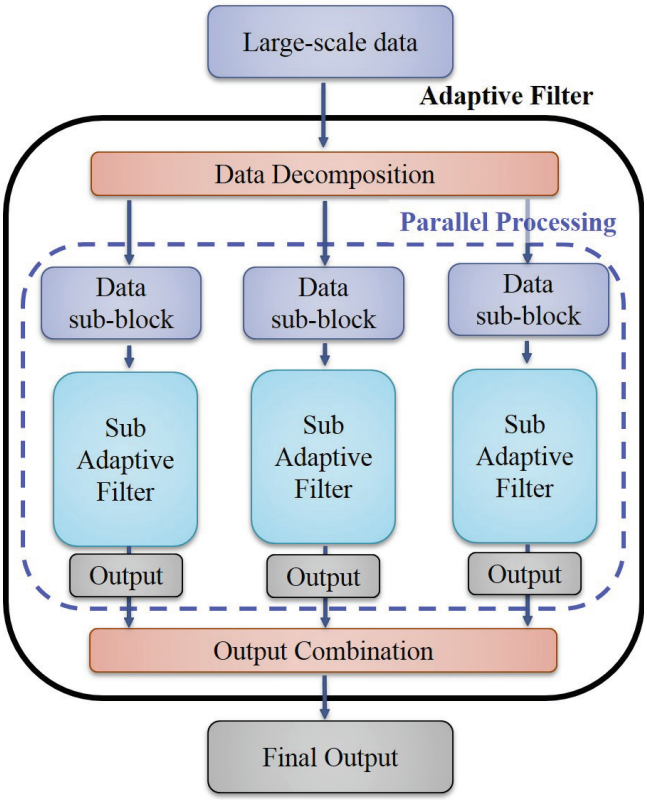

Fig. 1. Parallel-processing framework for adaptive filtering of large-scale data.

This paper proposes a parallel-processing framework (Fig. 1) to solve the parameter estimation problem for adaptive filtering of large-scale data. Given large-scale data is decomposed into several data sub-blocks. This decomposition reduces the computational complexity and makes the adaptive

filters implementable and scalable for large-scale data. Based on each data sub-block with adequate size, the normalized least-mean-square (NLMS) algorithm is derived to estimate the target parameter at each sub-filter. Furthermore, the variable step-size method and the intermittent-update method are derived by analyzing the mean-square deviation (MSD) performance of the NLMS algorithm at each sub-filter. The variable step-size method controls the step size to improve the performance in terms of the convergence rate and the steady-state error. The intermittent-update method performs occasional updates to improve the computational efficiency by avoiding the redundant updates. As can be seen in Fig. 1, the output of each sub-filter is combined with the other outputs through a combination method. By adopting the combination concept in diffusion adaptive filtering algorithms [23], the combination method combines all outputs of the sub-filters through the weighted sum based on their inverse of the error variance. Because the preciseness of each output is assumed to be proportional to its inverse of the error variance, the proposed combination method provides effective and robust weights for combining each output. The performance of the proposed algorithmic framework for large-scale data is verified by simulations. Because the adaptive filters contribute an important part of statistical signal processing, the proposed framework provides new capabilities and opportunities for signal processing of big data.

In this paper, bold symbols are used for column vectors (lower-case) and matrices (upper-case). For arbitrary nonnegative integers a and b, the following notations are used:

$(\cdot)^T$	transpose of a vector or matrix;
$\mathrm{Tr}(\cdot)$	trace of a matrix;
$E(\cdot)$	expectation of a random variable;
$\mathcal{R}^{a\times b}$	set of real matrices of dimension $a \times b$;
$\|\cdot\|$	Euclidean norm of a vector;
$\mathbf{I}_a$	identity matrix of dimension $a \times a$.

II. Parallel adaptive filtering for large-scale data

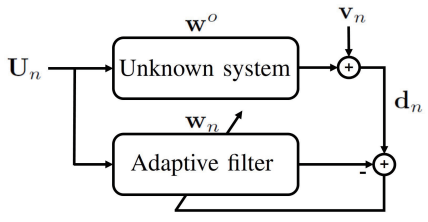

Fig. 2. Block diagram of parameter estimation using adaptive filter

Consider a parameter estimation model (Fig. 2). The unknown weight vector that is to be estimated is represented as $\mathbf{w}^o \in \mathcal{R}^{M\times 1}$. The weight vector of the adaptive filter at any particular iteration n is represented as $\mathbf{w}_n \in \mathcal{R}^{M\times 1}$. The input data is considered as a large-scale matrix, which is represented at iteration n for arbitrary positive integers $0 < L < M$ as

$$\mathbf{U}_n = [\mathbf{u}_{1,n}, \mathbf{u}_{2,n}, \cdots, \mathbf{u}_{L,n}] \in \mathcal{R}^{M\times L}, \tag{1}$$

where $\mathbf{u}_{i,n} = [u_{i,n}, u_{i,n-1}, \cdots, u_{i,n-M+1}]^T \in \mathcal{R}^{M\times 1}$ and $\mathbf{u}_{i,n}$ for all $i \in [1, \cdots, L]$ is comprised of a zero-mean white Gaussian signal. The measurement noise vector at iteration n is defined as

$$\mathbf{v}_n = [v_{1,n}, v_{2,n}, \cdots, v_{L,n}]^T \in \mathcal{R}^{L\times 1}, \tag{2}$$

which is comprised of a zero-mean white Gaussian signal with variance σ_v^2 and it is assumed to be independent from the input matrix $\mathbf{U}_n$. Then, the desired vector measured at iteration n is represented via a linear regression model as

$$\begin{aligned}\mathbf{d}_n &= \mathbf{U}_n^T\mathbf{w}^o + \mathbf{v}_n \in \mathcal{R}^{L\times 1},\\ &= [d_{1,n}, d_{2,n}, \cdots, d_{L,n}]^T,\end{aligned} \tag{3}$$

where $d_{i,n} = \mathbf{u}_{i,n}^T\mathbf{w}^o + v_{i,n}$. The structure of large-scale data is similar to that of the affine projection algorithm (APA). However, statistical properties of data are different because data matrix of the APA are made by stacking several past input vectors. In general, the estimate of the unknown weight vector can be obtained via the gradient-based recursion such as iterative solver including interior point methods and centralized online schemes. For large-scale data, however, estimation problem reinvents itself because their sizes are too large to process locally. The implementation of such estimators requires heavy computational resources.

A. Decomposition of large-scale data

As mentioned earlier, the size of the data can significantly degrade the performance of the adaptive filtering algorithms in the aspect of computational complexity. Therefore, the large-scale data that is arising in (3) might cause problems of computational burden. To overcome this drawback, given large-scale data $\mathbf{U}_n$ is decomposed into N sub-blocks as

$$\mathbf{U}_n = \begin{bmatrix} \bar{\mathbf{U}}_{1,n} & \bar{\mathbf{U}}_{2,n} & \cdots & \bar{\mathbf{U}}_{N,n} \end{bmatrix}, \tag{4}$$

where $\bar{\mathbf{U}}_{k,n}$ denotes the k-th data sub-block for $k \in [1, \cdots, N]$. From the decomposed data, the unknown system is estimated in parallel way at the sub-filter based on each data sub-block. Because each sub-block has reasonable size, the problem of computational complexity can be efficiently solved. The input data, the measurement noise vector, and the desired vector for k-th sub-filter are expressed as

$$\bar{\mathbf{U}}_{k,n} = [\mathbf{u}_{\mathcal{S}_k(1),n}, \mathbf{u}_{\mathcal{S}_k(2),n}, \cdots, \mathbf{u}_{\mathcal{S}_k(\bar{L}),n}] \in \mathcal{R}^{M\times \bar{L}}, \tag{5}$$

$$\mathbf{v}_{k,n} = [v_{\mathcal{S}_k(1),n}, v_{\mathcal{S}_k(2),n}, \cdots, v_{\mathcal{S}_k(\bar{L}),n}]^T \in \mathcal{R}^{\bar{L}\times 1}, \tag{6}$$

$$\mathbf{d}_{k,n} = [d_{\mathcal{S}_k(1),n}, d_{\mathcal{S}_k(2),n}, \cdots, d_{\mathcal{S}_k(\bar{L}),n}]^T \in \mathcal{R}^{\bar{L}\times 1}, \tag{7}$$

where $\bar{L} \triangleq L/N$, $\mathcal{S}_k(i) \triangleq (k-1)\bar{L}+i$. Then, the desired signal of block k can be represented as

$$\mathbf{d}_{k,n} = \bar{\mathbf{U}}_{k,n}^T \mathbf{w}^o + \mathbf{v}_{k,n}. \tag{8}$$

Based on the above measurements of k-th data sub-block, the output of k-th sub-filter is defined as an estimated weight vector using k-th data sub-block such that $\mathbf{w}_{k,n}$. Then, an *a priori* error vector of k-th sub-filter can be defined as

$$\mathbf{e}_{k,n} \triangleq \mathbf{d}_{k,n} - \bar{\mathbf{U}}_{k,n}^T \mathbf{w}_{k,n}. \tag{9}$$

B. NLMS algorithm at each sub-filter

Based on the each data sub-block, the NLMS algorithm is adopted to estimate the unknown system $\mathbf{w}^o$ at each sub-filter. From [17], the weight update equation of the NLMS algorithm of k-th sub-filter is represented as

$$\mathbf{w}_{k,n+1} = \mathbf{w}_{k,n} + \mu_{k,n}\bar{\mathbf{U}}_{k,n}(\bar{\mathbf{U}}_{k,n}^T\bar{\mathbf{U}}_{k,n})^{-1}\mathbf{e}_{k,n}, \tag{10}$$

where $\mu_{k,n}$ is a step size.

C. Mean-square deviation analysis of each sub-filter

The weight error vector of k-th sub-filter is defined as

$$\widetilde{\mathbf{w}}_{k,n} \triangleq \mathbf{w}^o - \mathbf{w}_{k,n}. \tag{11}$$

Then, the *a priori* error (9) and the NLMS algorithm (10) can be described in terms of $\widetilde{\mathbf{w}}_{k,n}$ as

$$\widetilde{\mathbf{w}}_{k,n+1} = \widetilde{\mathbf{w}}_{k,n} - \mu_{k,n}\bar{\mathbf{U}}_{k,n}\mathbf{G}_{k,n}\mathbf{e}_{k,n} \tag{12}$$

$$\mathbf{e}_{k,n} = \bar{\mathbf{U}}_{k,n}^T\widetilde{\mathbf{w}}_{k,n} + \mathbf{v}_{k,n}, \tag{13}$$

where $\mathbf{G}_{k,n} \triangleq (\bar{\mathbf{U}}_{k,n}^T\bar{\mathbf{U}}_{k,n})^{-1}$. The transition matrix is defined as

$$\mathbf{\Phi}_{k,(n+1,n)} \triangleq \mathbf{I}_M - \mu_{k,n}\bar{\mathbf{U}}_{k,n}\mathbf{G}_{k,n}\bar{\mathbf{U}}_{k,n}^T, \tag{14}$$

$$\mathbf{\Phi}_{k,(n,n)} \triangleq \mathbf{I}_M, \tag{15}$$

$$\mathbf{\Phi}_{k,(n,m)} \triangleq \mathbf{\Phi}_{k,(n,l)}\mathbf{\Phi}_{k,(l,m)}, \tag{16}$$

where $\mathbf{I}_M \in \mathcal{R}^{M\times M}$ is the identity matrix. Because there is no dependency between $\mathbf{v}_{k,n+1}$ and $\mathbf{v}_{k,n}$, an augmented model can be derived from (12), (13), and (14) as

$$\widetilde{\mathbf{w}}_{k,n+1} = \begin{bmatrix}-\mu_{k,n}\bar{\mathbf{U}}_{k,n}\mathbf{G}_{k,n} & \mathbf{\Phi}_{k,(n+1,n)}\end{bmatrix}\begin{bmatrix}\mathbf{v}_{k,n}\\ \widetilde{\mathbf{w}}_{k,n}\end{bmatrix}. \tag{17}$$

The conditional covariance matrix of $\widetilde{\mathbf{w}}_{k,n}$ is defined as

$$\mathbf{P}_{k,n} \triangleq E\left(\widetilde{\mathbf{w}}_{k,n}\widetilde{\mathbf{w}}_{k,n}^T|\bar{\mathcal{U}}_{k,n}\right) \tag{18}$$

for a given set $\bar{\mathcal{U}}_{k,n} = \{\bar{\mathbf{U}}_{k,j}|1 \le j < n\}$. Then, the MSD of k-th sub-filter is defined as

$$p_{k,n} \triangleq E(\widetilde{\mathbf{w}}_{k,n}^T\widetilde{\mathbf{w}}_{k,n}|\bar{\mathcal{U}}_{k,n}) = \mathrm{Tr}(\mathbf{P}_{k,n}). \tag{19}$$

Because $\mathbf{v}_{k,n}$ consists the white Gaussian signal and the dependencies of $\widetilde{\mathbf{w}}_{k,n}$ and $\mathbf{v}_{k,n}$ can be neglected, post-multiplying the transpose of (17) to itself and taking conditional expectation leads to

$$\begin{aligned}\mathbf{P}_{k,n+1} &= E\left(\widetilde{\mathbf{w}}_{k,n+1}\widetilde{\mathbf{w}}_{k,n+1}^T|\bar{\mathcal{U}}_{k,n}\right)\\ &= \begin{bmatrix}-\mu_{k,n}\bar{\mathbf{U}}_{k,n}\mathbf{G}_{k,n} & \mathbf{\Phi}_{k,(n+1,n)}\end{bmatrix}\begin{bmatrix}\sigma_v^2\mathbf{I}_{\bar{L}} & \mathbf{0}\\ \mathbf{0} & \mathbf{P}_{k,n}\end{bmatrix}\\ &\quad\times\begin{bmatrix}-\mu_{k,n}\mathbf{G}_{k,n}\bar{\mathbf{U}}_{k,n}^T\\ \mathbf{\Phi}_{k,(n+1,n)}^T\end{bmatrix},\end{aligned} \tag{20}$$

where $\mathbf{0}$ is a zero matrix with appropriate dimensions. From (20), the recursion of the covariance matrix $\mathbf{P}_{k,n}$ is derived as

$$\begin{aligned}\mathbf{P}_{k,n+1} &= \mathbf{\Phi}_{k,(n+1,n)}\mathbf{P}_{k,n}\mathbf{\Phi}_{k,(n+1,n)}^T\\ &\quad+\mu_{k,n}^2\sigma_v^2\bar{\mathbf{U}}_{k,n}\mathbf{G}_{k,n}^2\bar{\mathbf{U}}_{k,n}^T\\ &= \mathbf{P}_{k,n} - \mu_{k,n}\bar{\mathbf{U}}_{k,n}\mathbf{G}_{k,n}\bar{\mathbf{U}}_{k,n}^T\mathbf{P}_{k,n}\\ &\quad-\mu_{k,n}\mathbf{P}_{k,n}\bar{\mathbf{U}}_{k,n}\mathbf{G}_{k,n}\bar{\mathbf{U}}_{k,n}^T\\ &\quad+\mu_{k,n}^2\bar{\mathbf{U}}_{k,n}\mathbf{G}_{k,n}\bar{\mathbf{U}}_{k,n}^T\mathbf{P}_{k,n}\bar{\mathbf{U}}_{k,n}\mathbf{G}_{k,n}\bar{\mathbf{U}}_{k,n}^T\\ &\quad+\mu_{k,n}^2\sigma_v^2\bar{\mathbf{U}}_{k,n}\mathbf{G}_{k,n}^2\bar{\mathbf{U}}_{k,n}^T.\end{aligned} \tag{21}$$

By taking the trace on both sides of (21), the MSD recursion of k-th sub-filter is derived as

$$\begin{aligned}p_{k,n+1} = p_{k,n} &- (2\mu_{k,n}-\mu_{k,n}^2)\mathrm{Tr}\left(\bar{\mathbf{U}}_{k,n}\mathbf{G}_{k,n}\bar{\mathbf{U}}_{k,n}^T\mathbf{P}_{k,n}\right)\\ &+\mu_{k,n}^2\sigma_v^2\mathrm{Tr}\left(\mathbf{G}_{k,n}\right).\end{aligned} \tag{22}$$

Since $\bar{\mathbf{U}}_{k,n}\mathbf{G}_{k,n}\bar{\mathbf{U}}_{k,n}^T$ and $\mathbf{P}_{k,n}$ are positive semi-definite matrix, from [24], the values of $\mathrm{Tr}\left(\bar{\mathbf{U}}_{k,n}\mathbf{G}_{k,n}\bar{\mathbf{U}}_{k,n}^T\mathbf{P}_{k,n}\right)$ can be approximated as

$$\mathrm{Tr}\left(\bar{\mathbf{U}}_{k,n}\mathbf{G}_{k,n}\bar{\mathbf{U}}_{k,n}^T\mathbf{P}_{k,n}\right) \approx \frac{\bar{L}}{M}\mathrm{Tr}\left(\mathbf{P}_{k,n}\right). \tag{23}$$

Substituting (23) into (22) leads the MSD recursion to

$$\begin{aligned}p_{k,n+1} \approx &\left(1-(2\mu_{k,n}-\mu_{k,n}^2)\frac{\bar{L}}{M}\right)p_{k,n}\\ &+\mu_{k,n}^2\sigma_v^2\mathrm{Tr}\left(\mathbf{G}_{k,n}\right).\end{aligned} \tag{24}$$

In the steady-state, the MSD of k-th sub-filter converges to its steady-state value as $p_{k,n} \approx p_{k,ss}$, where $p_{k,ss}$ is the steady-state MSD of k-th sub-filter and defined as

$$p_{k,ss} \triangleq \lim_{n\to\infty} p_{k,n}. \tag{25}$$

Because $p_{k,n}$ is equal to $p_{k,n+1}$ in the steady-state, the steady-state MSD is obtained from (24) as

$$p_{k,ss} = \frac{M}{\bar{L}}\frac{\mu_{k,n}\sigma_v^2\mathrm{Tr}\left(\mathbf{G}_{k,n}\right)}{2-\mu_{k,n}}. \tag{26}$$

D. Performance improvements

To improve the estimation performance and reduce the computational complexity, the variable step-size and the intermittent-update methods are proposed from the mean-square deviation analysis.

1) Variable step-size method: To achieve improvements of the performance, the step size is controlled through the variable step-size method. By minimizing the MSD recursion in (24) with respect to $\mu_{k,n}$, the variable step size that leads to the largest decrease the MSD of k-th sub-filter can be chosen. According to

$$\begin{aligned}\frac{\partial p_{k,n+1}}{\partial \mu_{k,n}} &= (-2+2\mu_{k,n})\frac{\bar{L}}{M}p_{k,n} + 2\mu_{k,n}\sigma_v^2 \mathrm{Tr}\left(\mathbf{G}_{k,n}\right) \\ &= 0, \end{aligned} \tag{27}$$

the variable step size is obtained as follows:

$$\mu_{k,n} = \frac{\frac{\bar{L}}{M}p_{k,n}}{\frac{\bar{L}}{M}p_{k,n} + \sigma_v^2 \mathrm{Tr}\left(\mathbf{G}_{k,n}\right)}. \tag{28}$$

By using the variable step size, the NLMS algorithm can achieve fast convergence rate and small steady-state estimation error. For the proposed algorithm with the variable step size, the convergence of the MSD recursion of k-th sub-filter is verified. Because $p_{k,n}$ and $\sigma_v^2 \mathrm{Tr}\left(\mathbf{G}_{k,n}\right)$ are always positive, it is obvious that the proposed variable step size has boundary as $0 < \mu_n < 1$. Substituting (28) into (24), the difference equation of the MSD recursion of k-th sub-filter is derived as

$$p_{k,n+1} - p_{k,n} = -\mu_{k,n}^2 \left(\frac{\bar{L}}{M}p_{k,n} + \sigma_v^2 \mathrm{Tr}\left(\mathbf{G}_{k,n}\right) \right) < 0. \tag{29}$$

Consequently, the variable step size guarantees the monotonic decrease of the MSD of k-th sub-filter. Furthermore, the convergence of overall MSD for $\mathbf{w}_n$ is also guaranteed because the combination weight a_k in (32) is a convex parameter.

2) Intermittent-update method: To increase the computational efficiency, the intermittent-update method is proposed. The proposed method adjusts the update interval to perform the intermittent update. That is, the NLMS adaptation is intermittently performed from the previous adaptation. The update interval of k-th sub-filter at iteration n is defined as $t_{k,n} \in [1, \cdots, M]$. The update interval $t_{k,n}$ is dynamically adjusted through the MSD information on whether the adaptive filter reaches the steady state or not. From (24) and (26), the update interval $t_{k,n}$ is adjusted according to the difference between the current MSD value and the steady-state MSD value as follows:

$$t_{k,n+1} \triangleq \begin{cases} \min(M, t_{k,n}+1) \text{ if } |p_{k,n} - p_{k,ss}| \le \zeta_1, \\ \max(1, t_{k,n}-1) \text{ if } |p_{k,n} - p_{k,ss}| > \zeta_2, \\ t_{k,n} \qquad\qquad \text{if } \zeta_1 < |p_{k,n} - p_{k,ss}| \le \zeta_2, \end{cases} \tag{30}$$

where ζ_1 and ζ_2 are threshold parameters used to check whether the filter reaches the steady state. It is satisfied that $\zeta_2 \ge \zeta_1 > 0$. Since the intermittent adaptation avoids the redundant updates, the proposed algorithm requires fewer updates and computations.

E. Combination of output at each sub-filter

At each sub-filter, the unknown system $\mathbf{w}^o$ is estimated by the improved NLMS algorithm with the variable step-size (28) and the intermittent-update method (30). The estimation output at each sub-filter should be combined to yield the estimate for the given large-scale data. Therefore, it is needed to design a combination strategy. To compute $\mathbf{w}_n$ from $\mathbf{w}_{k,n}$ for all k, a combination method is proposed as follows:

$$\mathbf{w}_n = \sum_{k=1}^{N} a_k \mathbf{w}_{k,n}, \tag{31}$$

where a_k is a combination weight parameter. The estimate $\mathbf{w}_n$ is computed from weighted sum of $\mathbf{w}_{k,n}$ for all k. To satisfy $\sum_{k=1}^{N} a_k = 1$ as a convex parameter, the weight a_k is determined as follows:

$$a_k \triangleq \frac{\sigma_{e,k}^{-2}(n)}{\sum_{m=1}^{N} \sigma_{e,m}^{-2}(n)}, \tag{32}$$

where $\sigma_{e,k}^2(n)$ is the variance of the error vector at k-th sub-filter, which can be considered as the reliability indicator because the inverse of the error variance provides information that how much the estimation output of each sub-filter is precise. Therefore, the combination method can effectively combine the outputs of each sub-filter. The variance of the error vector $\sigma_{e,k}^2(n)$ can be estimated by following time-averaging method:

$$\sigma_{e,k}^2(n+1) = \delta \sigma_{e,k}^2(n) + (1-\delta)\frac{1}{\bar{L}}\mathbf{e}_{k,n}^T \mathbf{e}_{k,n}, \tag{33}$$

where δ is a forgetting factor. From the proposed combination method, the estimate $\mathbf{w}_n$ for given large-scale data is computed as follows:

$$\mathbf{w}_n = \sum_{k=1}^{N} \left(\sum_{m=1}^{N} \sigma_{e,m}^{-2}(n) \right)^{-1} \sigma_{e,k}^{-2}(n) \mathbf{w}_{k,n}. \tag{34}$$

After combining all outputs, the estimate $\mathbf{w}_{k,n}$ of k-th sub-filter is replaced by the combined result $\mathbf{w}_n$ for all k as

$$\mathbf{w}_{k,n} = \mathbf{w}_n \text{ for all } k. \tag{35}$$

This replacement extends the advantages of combining to each sub-filter. Consequentially, the proposed parallel-processing framework can provide not only high computational efficiency but also enhancements of preciseness and robustness to abnormalities of the data.

Table I summarizes the proposed algorithm.

III. Simulation results

The computer simulations are performed to estimate an unknown system $\mathbf{w}^o$, which is randomly generated with 3000 taps ($M = 3000$). The size of input data is set to $\mathbf{U}_n \in \mathcal{R}^{3000\times 2000}$. The input data is generated with zero-mean white Gaussian distribution. The signal-to-noise ratio (SNR) for the measurement is defined as follows:

$$\text{SNR} \triangleq 10\log_{10}\frac{E(\mathbf{y}_n^T \mathbf{y}_n)}{E(\mathbf{v}_n^T \mathbf{v}_n)}, \tag{36}$$

where $\mathbf{y}_n = \mathbf{U}_n^T \mathbf{w}^o$. For all simulations, SNR is set to 20 dB. The noise variance σ_v^2 is assumed to be available. In practice,

TABLE I
THE PROPOSED ALGORITHM SUMMARY

Initialization: $p_{k,0} = 1$, $t_{k,0} = 1$, and $\text{flag}_k = 0$ for all k

N, δ, ζ_1, ζ_2 : user defined values

M, $\bar{L}$, σ_v^2 : known values

for each iteration n

for each block k

$$\mathbf{e}_{k,n} = \mathbf{d}_{k,n} - \bar{\mathbf{U}}_{k,n}^T \mathbf{w}_{k,n}$$

$$\sigma_{e,k}^2(n) = \delta\sigma_{e,k}^2(n-1) + (1-\delta)\tfrac{1}{\bar{L}}\mathbf{e}_{k,n}^T\mathbf{e}_{k,n}$$

if $n == \text{flag}_k + t_{k,n}$

$$\mathbf{G}_{k,n} = (\bar{\mathbf{U}}_{k,n}^T\bar{\mathbf{U}}_{k,n})^{-1}$$

$$\mu_{k,n} = \frac{\frac{\bar{L}}{M}p_{k,n}}{\frac{\bar{L}}{M}p_{k,n} + \sigma_v^2 \text{Tr}\left(\mathbf{G}_{k,n}\right)}$$

$$p_{k,n+1} = \left(1 - (2\mu_{k,n} - \mu_{k,n}^2)\tfrac{\bar{L}}{M}\right)p_{k,n} + \mu_{k,n}^2\sigma_v^2\text{Tr}\left(\mathbf{G}_{k,n}\right)$$

$$p_{k,ss} = \frac{M}{\bar{L}}\frac{\mu_{k,n}\sigma_v^2\text{Tr}\left(\mathbf{G}_{k,n}\right)}{2-\mu_{k,n}}$$

$$t_{k,n+1} = \begin{cases} \min(M, t_{k,n}+1) \text{ if } |p_{k,n} - p_{k,ss}| \le \zeta_1, \\ \max(1, t_{k,n}-1) \text{ if } |p_{k,n} - p_{k,ss}| > \zeta_2, \\ t_{k,n} \qquad \text{if } \zeta_1 < |p_{k,n} - p_{k,ss}| \le \zeta_2, \end{cases}$$

$$\mathbf{w}_{k,n+1} = \mathbf{w}_{k,n} + \mu_{k,n}\bar{\mathbf{U}}_{k,n}\mathbf{G}_{k,n}\mathbf{e}_{k,n}$$

$$\text{flag}_k = n$$

else

$$\mu_{k,n} = \mu_{k,n-1},\ p_{k,n+1} = p_{k,n}$$

$$\mathbf{w}_{k,n+1} = \mathbf{w}_{k,n}$$

end

end for

$$\mathbf{w}_{n+1} = \sum_{k=1}^{N}\left(\sum_{m=1}^{N}\sigma_{e,m}^{-2}(n)\right)^{-1}\sigma_{e,k}^{-2}(n)\mathbf{w}_{k,n+1}$$

$$\mathbf{w}_{k,n+1} = \mathbf{w}_{n+1} \text{ for all } k$$

end for

it can be easily estimated during silences [24]. The simulation value of the MSD at iteration n can be computed as follows:

$$\text{MSD}_n \triangleq E\left(\|\mathbf{w}^o - \mathbf{w}_n\|^2\right), \tag{37}$$

where $\mathbf{e}_n = \mathbf{d}_n - \bar{\mathbf{U}}_n^T\mathbf{w}_n$. The normalized MSD (NMSD) value is used to a performance index, which is defined as $\text{NMSD}_n \triangleq E\left(\|\mathbf{w}^o - \mathbf{w}_n\|^2/\|\mathbf{w}^o\|^2\right)$. For the proposed algorithm, the user defined parameters are set to as $\lambda = 0.99$ (the forgetting factor), $\zeta_1 = 8 \times 10^{-5}$, and $\zeta_2 = 1 \times 10^{-4}$.

A. Verification of the mean-square deviation analysis

A Monte-Carlo simulations are carried out in Fig. 3 to discuss regarding the accuracy of the proposed theoretical analysis (24) about the MSD recursions to predict the practical MSD. It demonstrates how the proposed analysis precisely predicts the MSD learning behavior of the proposed framework. The simulation results show that the proposed analysis excellently predicts the learning behavior of the practical MSD. Therefore, proposed theoretical analysis of the mean-square deviation has enough accuracy to derive the variable step-size method and the intermittent-update method.

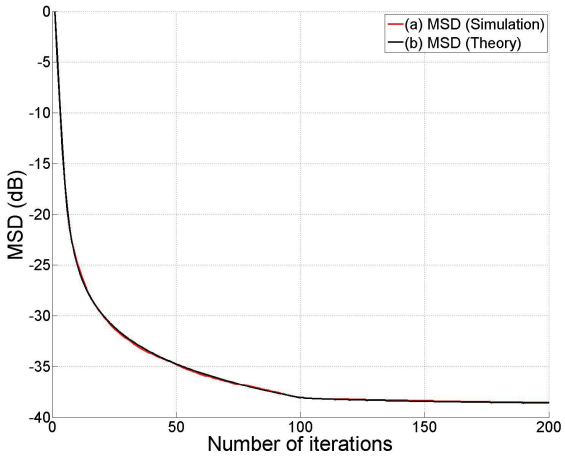

Fig. 3. MSD learning curves of the simulation result (red line) and its prediction (black line) using the proposed theoretical analysis for $N = 1$.

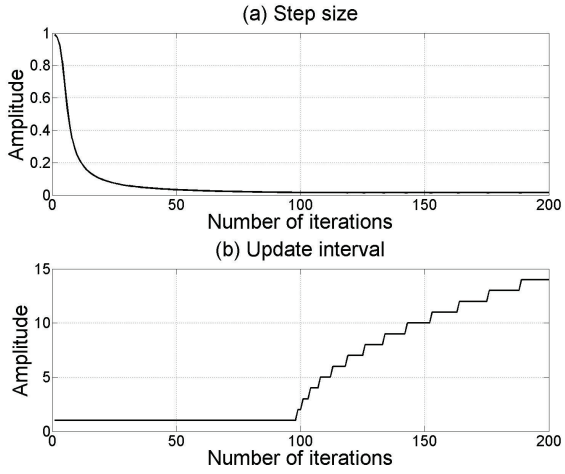

Fig. 4. Validation of the variable step-size method and the intermittent-update method for $N = 1$ (a) variation of step sizes (b) variation of update intervals.

Fig. 4 (a) shows that proposed variable step-size method is works well. The step size automatically decreases to yield a low steady-state error and a fast convergence rate. Fig. 4 (b) show that the update interval is dynamically adjusted by the proposed intermittent-update method. The intermittent-update method improves the efficiency of the proposed algorithm by reducing redundant adaptation processes.

B. Comparison of the learning behavior

Because there is no adaptive filtering algorithm to deal with large-scale input matrix, the proposed algorithmic framework is compared with the conventional NLMS algorithm which

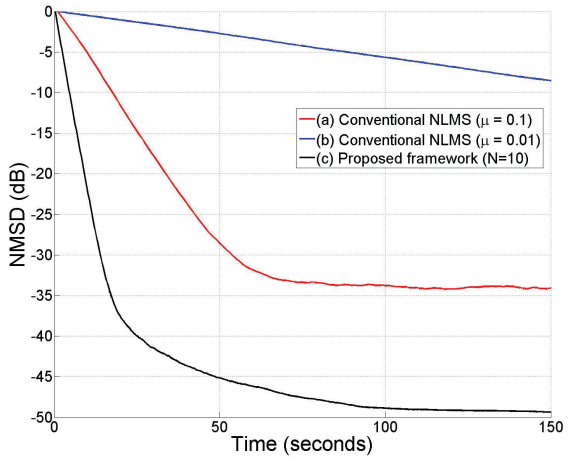

Fig. 5. NMSD learning curves of the conventional NLMS and the proposed framework for $N = 10$.

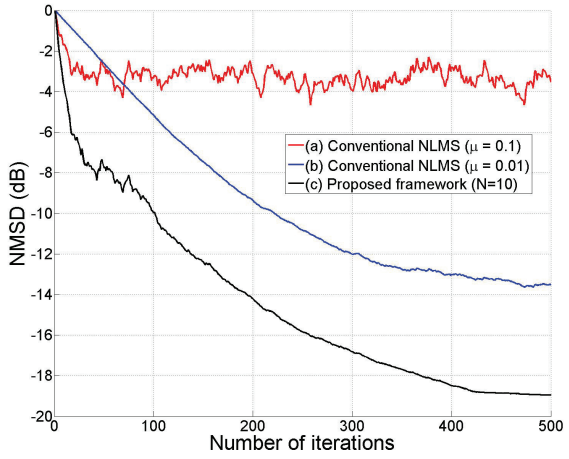

Fig. 6. NMSD learning curves for the impulsive measurement noise. $p_r = 0.01$ and $N = 10$ for the proposed algorithm.

is modified to fit the matrix input data to verify the learning performance. The update equation of the NLMS algorithm for large-scale data is defined as

$$\mathbf{w}_n = \mathbf{w}_{n-1} + \mu \mathbf{U}_n (\mathbf{U}_n^T \mathbf{U}_n)^{-1} \mathbf{e}_n. \tag{38}$$

Fig. 5 represents the NMSD learning curves versus those computation times of the NLMS and the proposed algorithm for large-scale data. Using simulations based on computation times instead of iterations provides fair results, which is similar to [13-15]. The computation time is obtained by recording process time to perform the algorithms. As can be seen, the proposed algorithm has the smaller steady-state error and the faster convergence rate than the conventional NLMS algorithm. When the larger data is applied to the system, efficiencies of the proposed framework will be more improved than the conventional algorithm.

An additional simulation is carried out in Fig. 6 when the system measurement is disturbed by an impulsive noise In this simulation, the measurement noise vector $\mathbf{v}_n$ is considered as a contaminated-Gaussian impulsive noise defined as $\boldsymbol{\eta}_n = \mathbf{v}_n + \mathbf{B}_n \mathbf{g}_n$, where

$$\mathbf{B}_n \triangleq \begin{bmatrix} b_{1,n} & & & \\ & b_{2,n} & & \\ & & \ddots & \\ & & & b_{L,n} \end{bmatrix} \in \mathcal{R}^{L \times L} \tag{39}$$

is a square matrix with all entries outside the main diagonal equal to zero and the diagonal elements $b_{i,n}$ is a switch sequence of ones and zeros, which is modeled as a Bernoulli random process with occurrence probability $P_r(b_{i,n} = 1) = p_r$; $\mathbf{g}_n = [g_{1,n}, g_{2,n}, \cdots, g_{L,n}]^T \in \mathcal{R}^{L \times 1}$ is zero-mean white Gaussian sequences with variance $\sigma_g^2 = 1000\sigma_y^2$. As can be seen in Fig. 6, the conventional NLMS algorithm is very sensitive to disturbance on the system measurement. Therefore, it suffers from performance degradation in the presence of impulsive noise. However, the proposed framework shows the good performance because the parallel-processing effectively suppress the effect of bad information from the combination method in (31) and (32).

IV. Conclusion

This paper proposed a parallel adaptive filtering algorithm for large-scale data. In the proposed algorithm, large-scale data was divided into several data sub-blocks to reduce the size of data. Based on each data sub-block, the NLMS algorithm estimated the unknown parameters and outputs of each sub-filter were effectively and robustly combined by the combination method. Furthermore, the mean-square deviation analysis of each sub-filter led to the variable step-size method and the intermittent-update method, which improved the performance of the proposed algorithm in terms of the convergence rate, the steady-state error, and the computational complexity. Simulations showed that the proposed algorithm works well for large-scale data.

References

[1] J. Fan, F. Han, and H. Liu, "Challenges of big data analysis," *National Science Review*, vol. 1, no. 2 pp. 293-314, Feb. 2014.
[2] C. L. P. Chen and C.-Y. Zhang, "Data-intensive applications, challenges, techniques and technologies: A survey on Big Data," *Information Sciences*, vol. 275, pp. 314-347, Aug. 2014.
[3] K. Slavakis, G. B. Giannakis, and G. Mateos, "Modeling and optimization for big data analytics," *IEEE Signal Process. Mag.*, vol. 31, no. 5, pp. 18-31, Sep. 2014.
[4] V. Cevher, S. Becker, and M. Schmidt, "Convex optimization for big data: Scalable, randomized, and parallel algorithms for big data analytics," *IEEE Signal Process. Mag.*, vol. 31, no. 5, pp. 32-43, Sep. 2014.
[5] A. Sandryhaila and J. M. F. Moura, "Big Data Analysis with Signal Processing on Graphs: Representation and processing of massive data sets with irregular structure," *IEEE Signal Process. Mag.*, vol. 31, no. 5, pp. 80-90, Sep. 2014.

[6] V. Krishnamurthy, O. Namvar, and M. Hamdi, "Signal processing of social networks:Interactive sensing and learning," *Found. Trends Signal Process.*, vol. 7, no. 1-2. pp. 1-19, 2014.
[7] J. Ahrens, B. Hendrickson, G Long, .S Miller, R. Ross, and D. Williams, "Data-intensive science in the us doe: case studies and future challenges," *Comput. Sci. Eng.*, vol. 13, no. 6, pp. 14-24. 2011.
[8] D. L. Donoho, "Compressed sensing," *IEEE Trans. Inform. Theory*, vol. 52, no. 4, pp. 1289-1306, Apr. 2006.
[9] E. J. Candes and M. B. Wakin, "An introduction to compressive sampling," *IEEE Signal Process. Mag.*, vol. 25, no. 2, pp. 21-30, Mar. 2008.
[10] C. Yang, X. Zhang, C. Zhong, C. Liu, and J. Pei, "A spatiotemporal compression based approach for efficient big data processing on Cloud," *J. Comput. Syst. Sci.*, vol. 80, no. 8, pp. 1563-1583, Dec. 2014.
[11] N. Vervliet, O. Debals, L. Sorber, L. De Lathauwer, "Breaking the curse of dimensionality using decompositions of incomplete tensors: Tensor-based scientific computing in big data analysis," *IEEE Signal Process. Mag.*, vol. 31, no. 5, pp. 71-79, Sep. 2014.
[12] M. Mardani, G. Mateos, G. B. Giannakisi, "Subspace learning and imputation for streaming big data matrices and tensors," *IEEE Trans. Signal Process.*, vol. 63, no. 10, pp. 2663-2677, May. 2015.
[13] F. Facchinei and G. Scutari, "Parallel selective algorithms for nonconvex big data optimization," *IEEE Trans. Signal Process.*, vol. 63, no. 7, pp. 1874-1889, Apr. 2015.
[14] A. Daneshmand, F. Facchinei, V. Kungurtsev, and G. Scutari, "Hybrid random/deterministic parallel algorithms for convex and nonconvex big data optimization," *IEEE Trans. Signal Process.*, vol. 63, no. 15, pp. 3914-3929 , Aug. 2015.
[15] M. Hong, M. Razaviyayn, Z.-Q. Luo, and J.-S. Pang, "A unified algorithmic framework for block-structured optimization involving big data: with applications in machine learning and signal process," *IEEE Signal Process. Mag.*, vol. 33, pp. 57-77, Jan. 2016.
[16] K. Slavakis, S.-J. Kim, G. Mateos, and G. B. Giannakis, "Stochastic approximation vis-a-vis online learning for big data analytics," *IEEE Signal Process. Mag.*, vol. 31, no. 6, pp. 124-129, Sep. 2014.
[17] S. Haykin, *Adaptive Filter Theory*, New Jersey, NJ, USA: Prentice-Hall, 2002.
[18] A. H. Sayed, *Fundamentals of Adaptive Filtering*, New York, NY, USA: Wiley, 2003.
[19] M. Mahoney, "Randomized algorithms for matrices and data," *Found. Trends Mach. Learn.*, vol. 3, no. 2, pp. 123-224, Feb. 2011.
[20] M. Spiliopoulou, M. Hatzopoulos, and Y. Cotronis, "Parallel optimization of large join queries with set operators and aggregates in a parallel environment supporting pipeline," *IEEE Trans. Knowl. Data Eng.*, vol. 8, no. 3, pp. 429-445, Jun. 1996.
[21] J. Yan, N. Liu, S. Yan, Q. Yang, W. Fan, W. Wei, and Z. Chen, "Trace-oriented feature analysis for large-scale text data dimension reduction," *IEEE Trans. Knowl. Data Eng.*, vol. 23, no. 7, pp. 1103-1117, Jul. 2011.
[22] B. Recht and C. Rè, "Parallel stochastic gradient algorithms for large-scale matrix completion," *Mathematical Programming Computation*, vol 5, no. 2, pp 201-226, Jun. 2013.
[23] S. M. Jung, J.-H. Seo, and P. Park, "A variable step-size diffusion normalized least-mean-square algorithm with a combination method based on mean-square deviation," *Circuits, Sys., and Signal Process.*, vol. 34, no. 10, pp. 3291-3304, Oct. 2015.
[24] S. M. Jung and P. Park, "Stabilization of a bias-compensated normalized least-mean-square algorithm for noisy inputs," *IEEE Trans. Signal Process.*, vol. 65, no. 11, pp. 2949-2961, 2017.