Peer-reviewed version available at Sensors 2017, 17, , 824; doi:10.3390/s17040824

Article

A Robust Diffusion Estimation Algorithm with Self-adjusting Step-size in WSNs

Feng Chen [1,†,‡], Qing Ye [2,‡], Xiaodan Shao [2,‡] and Shukai Duan [2,*]

[1] Department of College of Electronic and Information Engineering, School of Mathematics and Statistics, Southwest University, and Key Laboratory of Nonlinear Circuits and Intelligent Information Processing, Chongqing 400715, China. e-mail: (fengchenmit@swu.edu.cn)

[2] Department of College of Electronic and Information Engineering, Southwest University, Chongqing 400715, China. e-mail: (qingye6@126.com, xiaodanshao3@126.com)

* Correspondence: duansk@swu.edu.cn; Tel: +023-68251252

† Current address: No.2, Tiansheng Road, Beibei District, Chongqing City, China

‡ These authors contributed equally to this work.

Abstract: In wireless sensor networks (WSNs), each sensor node can estimate the global parameter from the local data in distributed manner. This paper proposed a robust diffusion estimation algorithm based on minimum error entropy criterion with self-adjusting step-size, which are referred to as diffusion MEE-SAS (DMEE-SAS) algorithm. The DMEE-SAS algorithm has fast speed of convergence and is robust against non-Gaussian noise in the measurements. The detailed performance analysis of the DMEE-SAS algorithm is performed. By combining the DMEE-SAS with diffusion minimum error entropy (DMEE) algorithms, an Improving DMEE-SAS algorithm is proposed, in non-stationary environment where tracking is very important. The Improving DMEE-SAS algorithm can avoid insensitivity of the DMEE-SAS algorithm due to the small effective step-size near the optimal estimator, and obtain a fast convergence speed. Numerical simulations are given to verify the effectiveness and advantages of these proposed algorithms.

Keywords: robust diffusion estimation; self-adjusting step-size; non-Gaussian noise; wireless sensor networks

1. Introduction

Distributed estimation has become very popular for parameter estimation in wireless sensor networks. The objective is to enable the nodes to estimate a vector of parameters of interest from the observed data. Distributed estimation schemes over adaptive networks can be mainly classified into incremental strategies [15–17], consensus strategies [18,25], and diffusion strategies [3,4,7,12,19,27]. In the incremental strategies, data is processed in a cyclic fashion through the network. The consensus strategies rely on the fusion of intermediate estimates of multiple neighboring nodes. In the Diffusion strategies, information is processed at all nodes while the nodes communicate with all their neighbors to share their intermediate estimates. The diffusion strategies are particularly attractive because they are robust, flexible and fully-distributed, such as the diffusion least mean squares (DLMS) algorithm [7]. In this paper, we focus on the diffusion estimation strategies.

The performance of distributed estimation degrades severely when the signals are perturbed by non-Gaussian noise. Non-Gaussian noise may be natural, due to atmospheric phenomena, or man-made, due to either electric machinery present in the operation environment, or multipath telecommunications signals [20–22]. Recently, some researchers focus on improving robustness for non-Gaussian noise of distributed estimation methods. The efforts are mainly directed at searching for a more robust cost function to replace the MSE cost. To address this problem, the diffusion least mean p-power (DLMP) based on p-norm error criterion was proposed to estimate the parameters of the wireless sensor networks [5]. The diffusion minimum error entropy (DMEE) was proposed in [6]. By DMEE algorithm we refer the adapt-then-combine (ATC) DMEE algorithm in [6]. The DMEE algorithm achieved improved performance for non-Gaussian noise with the fixed step-size, but it still

Peer-reviewed version available at Sensors 2017, 17, , 824; doi:10.3390/s17040824

suffers from conflicting requirements between convergence rate and the steady-state mean square error. A large step-size leads to a fast convergence rate, but a large mean-square error at the steady state.

In this paper, we incorporate the minimum error entropy criterion with self-adjusting step-size (MEE-SAS) [8] into the cost function in diffusion distributed estimation. Then we figure out the diffusion-strategy solutions, which are referred to as the diffusion MEE-SAS (DMEE-SAS) algorithm. Numerical simulation results show that DMEE-SAS algorithm outperforms DLMS, DLMP and DMEE algorithms when the noise is modeled to be non-Gaussian noise. We also design an Improving DMEE-SAS algorithm by using a switching scheme between DMEE-SAS and DMEE algorithms for non-stationary environment, which tracks the changing estimator very effectively. The Improving DMEE-SAS algorithm can avoid the small effective step-size of DMEE-SAS algorithm when it close to the optimal estimator.

We organize the paper as follows. In section 2, we briefly revisit the MEE-SAS algorithm. In section 3, firstly, we propose the DMEE-SAS algorithm and analyze the mean and mean square performance for DMEE-SAS algorithm. Then we propose the Improving DMEE-SAS algorithm for non-stationary scenario. Simulation results are shown in section 4. Finally, we draw conclusions in section 5.

2. The Review of MEE-SAS Algorithm

A convenient evaluation of the integral operator in the formulation of quadratic Renyi's entropy using Gaussian kernel is obtained as follows:

$$\begin{aligned} H(e) &= -\log\left(\frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N} G_{\sigma\sqrt{2}}(e_j - e_i)\right) \\ &= -\log(V(e)). \end{aligned} \tag{1}$$

Where $e = [e_1, e_2, \cdots, e_N]$ and

$$G_{\sigma\sqrt{2}}(e_j - e_i) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{1}{2\sigma^2}(e_j - e_i)^2\right).$$

$$V(e) = \frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N} G_{\sigma\sqrt{2}}(e_j - e_i) \le V(0) = \frac{1}{\sigma\sqrt{2\pi}}.$$

The information potential $V(e)$ is defined as the argument of the log. The maximum value $V(0)$ of the information potential will be achieved when $e_1 = e_2 = \cdots = e_N$. The above results are obtained in the case of batch mode, where the N data points are fixed. For online training methods, we estimate the parameter using the stochastic information potential given below

$$V(e) \approx \frac{1}{L}\sum_{j=i-L+1}^{i} G_{\sigma\sqrt{2}}(e_i - e_j). \tag{2}$$

Where L is the latest L samples of e.

To minimize the entropy is equivalent to maximize the information potential since the log is a monotonic function. To maximize the information potential is equivalent to minimize the following cost function. Therefore, the cost function of MEE-SAS algorithm is

$$J_{MEE-SAS}(e) = \min_{w}[V(0) - V(e)]^2. \tag{3}$$

The MEE-SAS method has shown its ability to achieve faster speed than minimum error entropy (MEE) method and is robust to outliers. Based on these properties, we develop the diffusion MEE-SAS algorithms in the next section.

3. Proposed Algorithm

In this section, first, the diffusion MEE-SAS (DMEE-SAS) algorithm is proposed. Second, the detailed convergence and steady-state analyses of this algorithm are performed. Finally, an Improving scheme for diffusion DMEE-SAS algorithm is carried out to use in non-stationary scenario.

3.1. Diffusion MEE-SAS Algorithm

Consider a connected wireless sensor networks with K nodes. $k \in \{1, 2, \ldots, N\}$ is the node index and i is the time index. To proceed with the analysis, we assume a liner measurement model as follows:

$$\boldsymbol{d}_{k,i} = \boldsymbol{u}_{k,i}^T w^0 + v_{k,i}. \tag{4}$$

Where w^0 is a $M \times 1$ deterministic but unknown vector, $\boldsymbol{d}_{k,i}$ is a scalar measurement of some random process, $\boldsymbol{u}_{k,i}$ is the $M \times 1$ regression vector at time with zero mean and covariance σ_u^2, $v_{k,i}$ is the random noise signal at time i with zero mean and variance σ_v^2. For each node, we have

$$V(e_{k,i}) = \frac{1}{L} \sum_{j=i-L+1}^{i} G_{\sigma\sqrt{2}}(e_{k,i} - e_{k,j}) \leq V(0) = \frac{1}{\sigma\sqrt{2\pi}}. \tag{5}$$

Where $e_{k,i} = d_{k,i} - u_{k,i}^T w$. The maximum value $V(0)$ will be achieved when $e_{k,i} = e_{k,j}$, $j = i - L + 1, i - L + 2, \cdots, i$.

We seek an estimate of w^0 by minimizing a linear combination of local information. The individual local cost function for each node k is calculated as

$$J_k(w) = \sum_{l \in N_k} c_{l,k} E[V(0) - V(e_{l,i})]^2. \tag{6}$$

N_k denotes the one-hop neighbor set of node k, and $\{c_{lk}\}$ are some non-negative cooperative coefficients satisfying $c_{lk} = 0$ if $l \notin N_k$, $\mathbf{1}^T \bar{C} = \mathbf{1}^T$ and $\bar{C}\mathbf{1} = \mathbf{1}$. Here, $\bar{C}$ is a $N \times N$ matrix with individual entries $\{c_{lk}\}$ and $\mathbf{1}$ is a $N \times 1$ all-unity vector. The gradient of the individual local cost function is given by

$$\nabla J_k(w) = \sum_{l \in N_k} c_{lk} E[(\frac{2}{\sigma^2 L})(V(0) - V(e_{k,i})) \sum_{j=i-L+1}^{i} G_{\sigma\sqrt{2}}(e_{k,i} - e_{k,j})(e_{k,i} - e_{k,j})(u_{k,j} - u_{k,i})]. \tag{7}$$

We remove the expectation to generate stochastic gradient updates, then the (7) can be rewritten as

$$\nabla \hat{J}_k(w) = \sum_{l \in N_k} c_{lk} (\frac{2}{\sigma^2 L})(V(0) - V(e_{k,i})) \sum_{j=i-L+1}^{i} G_{\sigma\sqrt{2}}(e_{k,i} - e_{k,j})(e_{k,i} - e_{k,j})(u_{k,j} - u_{k,i}). \tag{8}$$

A gradient based algorithm for estimating w^0 at each node k can thus be derived as

$$\begin{aligned} w_{k,i} &= w_{k,i-1} - \mu_k \nabla \hat{J}_k(w) \\ &= w_{k,i-1} - \mu_k \frac{2}{\sigma^2 L} \sum_{l \in N_k} c_{lk} [V(0) - V(e_{k,i})] \sum_{j=i-L+1}^{i} G_{\sigma\sqrt{2}}(e_{k,i} - e_{k,j})(e_{k,i} - e_{k,j})(u_{k,j} - u_{k,i}). \end{aligned} \tag{9}$$

Peer-reviewed version available at Sensors 2017, 17, , 824; doi:10.3390/s17040824

Where μ_k is a positive step size. Using the general framework for diffusion-based distributed adaptive optimization [12], an adapt-then-combine (ATC) strategy for diffusion MEE-SAS algorithm can be formulated as

$$\begin{cases} \varphi_{k,i} = w_{k,i-1} - \mu_k \frac{2}{\sigma^2 L}[V(0) - V(e_{k,i})] \sum\limits_{j=i-L+1}^{i} G_{\sigma\sqrt{2}}(e_{k,i} - e_{k,j})(e_{k,i} - e_{k,j})(u_{k,j} - u_{k,i}), \\ w_{k,i} = \sum\limits_{l \in N_k} c_{lk} \varphi_{l,i}. \end{cases} \tag{10}$$

According to (10), the DMEE-SAS algorithm can be seen as a diffusion estimation algorithm with variable step size $\mu_k(i)$. Where

$$\mu_k(i) = 2\mu_k[V(0) - V(e_{k,i})]. \tag{11}$$

The DMEE-SAS algorithm is described formally in Algorithm 1.

Algorithm 1: DMEE-SAS Algorithm

Initialize: $w_{k,i} = 0$
for $i = 1 : T$
for each node k**:**
Adaptation
$\mu_k(i) = 2\mu_k[V(0) - V(e_{k,i})]$
$\varphi_{k,i} = w_{k,i-1} - \mu_k(i) \frac{1}{\sigma^2 L} \sum\limits_{j=i-L+1}^{i} G_{\sigma\sqrt{2}}(e_{k,i} - e_{k,j})(e_{k,i} - e_{k,j})(u_{k,j} - u_{k,i})$
Combination
$w_{k,i} = \sum\limits_{l \in N_k} c_{lk} \varphi_{l,i}$
end for

In the adaption step of DMEE-SAS algorithm, $V(0) - V(e_{k,i})$ is close to $V(0)$ when the algorithm starts, and it is close to 0 when the algorithm begins to converge. $V(0) - V(e_{k,i})$ is always a non-negative scalar quantity, which can accelerate the rate of convergence and achieve small steady-state estimation errors. The fast convergence rate and the small steady-state estimation errors of DMEE-SAS algorithm can be established against non-Gaussian noise in the measurements.

3.2. Performance Analysis

In this section, we analyze the mean and mean-square performance of the DMEE-SAS algorithm. For tractability of the analysis, here we fous on the case of batch mode. To briefly present the convergence property of the proposed algorithm in terms of global quantities, the following notations are introduced: $M_\tau = diag\{\mu_1(\tau)I_M, \ldots, \mu_K(\tau)I_M\}$, $W_\tau = col\{\boldsymbol{w}_{1,\tau}, \cdots \boldsymbol{w}_{K,\tau}\}$, $w^{(0)} = col\{w^0, \cdots, w^0\}$, $\tilde{W}_\tau = col\{\tilde{\boldsymbol{w}}_{1,\tau} \cdots \tilde{\boldsymbol{w}}_{K,\tau}\}$, $S = col\{s_1(w^0), \cdots, s_K(w^0)\}$, $C = \bar{C}^T \otimes I_M$. τ denotes the iteration index.

In order to make the analysis tractable, the followings are assumed:

Assumption 1: The regressor $\boldsymbol{u}_{k,i}$ is independent identically distributed (i.i.d) in time and spatially independent, and $E[\boldsymbol{u}_{k,i}] = 0$, $R_k = E[\boldsymbol{u}_{k,i}^T \boldsymbol{u}_{k,i}]$.

Assumption 2: The input noise $v_{k,i}$ is super-Gaussian noise. In addition, $v_{k,i}$ and the regressor $\boldsymbol{u}_{k,i}$ is independent from each other. we have $E[v_{k,i}] = 0$ and $E[v_{k,i}^2] = \xi_k^2$.

Assumption 3: The step-sizes, $\mu_k(i)$, $\forall k$, are small enough such that their squared values are negligible.

Peer-reviewed version available at Sensors 2017, 17, , 824; doi:10.3390/s17040824

3.2.1. Mean performance

When we consider the variable step size $\mu_k(i)$ of DMEE-SAS algorithm as a new step size factor, we can also seek the optimal estimator w^0 by minimizing the following individual local cost function

$$G_k(w) = \sum_{l \in N_k} c_{lk} E(V(0) - V(e_{l,i})). \tag{12}$$

Where

$$V(e_{l,i}) = \frac{1}{N^2} \sum_{i=1}^{N} \sum_{j=1}^{N} G_{\sigma\sqrt{2}}(e_{l,i} - e_{l,j})$$

We obtain the first gradient of $E(V(0) - V(e_{k,i}))$ as follows

$$g_k(w) = E(\frac{1}{\sigma^2 N^2} G_{\sigma\sqrt{2}}(e_{k,i} - e_{k,j})(e_{k,i} - e_{k,j})(u_{k,j} - u_{k,i})]. \tag{13}$$

The instantaneous implementation for (13) is as follows

$$\hat{g}_k(w) = \frac{1}{\sigma^2 N^2} \sum_{i=1}^{N} \sum_{j=1}^{N} G_{\sigma\sqrt{2}}(e_{k,i} - e_{k,j})(e_{k,i} - e_{k,j})(u_{k,j} - u_{k,i}). \tag{14}$$

We consider the gradient error caused by approximating the expectations with their instantaneous values [9]. The gradient error at iteration τ and each node k is defined as follows:

$$s_k(w_{k,\tau-1}) = \hat{g}_k(w_{k,\tau-1}) - g_k(w_{k,\tau-1}). \tag{15}$$

Using (10), the update equation of the intermediate estimate can be rewritten as

$$\varphi_{k,\tau} = w_{k,\tau-1} - \mu_k(g_k(w_{k,\tau-1}) + s_k(w_{k,\tau-1})). \tag{16}$$

According to [29], $((e_{k,i} - e_{k,j})/\sigma = 0$ when $w = w^0$. And the Hessian matrix function $H_k(w^0)$ of $J_k(w)$ is calculated as

$$\begin{aligned} H_k(w^0) &= \frac{\partial g_k(w)}{\partial w}|_{w^0} \\ &= \frac{\partial}{\partial w} \frac{1}{\sigma^2} E[G_{\sigma\sqrt{2}}(e_{k,i} - e_{k,j})(e_{k,i} - e_{k,j})(u_{k,j} - u_{k,i})] \\ &= \frac{1}{\sigma^2} E[G_{\sigma\sqrt{2}}(e_{k,i} - e_{k,j})(u_{k,j} - u_{k,i})^T (u_{k,j} - u_{k,i}) - \\ &\quad \frac{1}{\sigma^2} G_{\sigma\sqrt{2}}(e_{k,i} - e_{k,j})(e_{k,i} - e_{k,j})^2 (u_{k,j} - u_{k,i})^T (u_{k,j} - u_{k,i})] \\ &= \frac{1}{\sigma^2} E[u_{k,i}^T u_{k,i} + u_{k,j}^T u_{k,j}] - \frac{1}{\sigma^4} E[v_{k,i}^2 + v_{k,j}^2] E[u_{k,i}^T u_{k,i} + u_{k,j}^T u_{k,j}] \\ &= \frac{2R_k}{\sigma^2} - \frac{4\xi_k^2 R_k}{\sigma^4}. \end{aligned} \tag{17}$$

Based on the Theorem 1.2.1 of [10], we obtain

$$\begin{aligned} g_k(w_{k,\tau-1}) &= g_k(w^0) - (\int_0^1 H_k(w^0 - x\tilde{w}_{k,\tau-1})dx)\tilde{w}_{k,\tau-1} \\ &= -(\int_0^1 H_k(w^0 - x\tilde{w}_{k,\tau-1})dx)\tilde{w}_{k,\tau-1}. \end{aligned} \tag{18}$$

Peer-reviewed version available at Sensors 2017, 17, , 824; doi:10.3390/s17040824

Where $\tilde{w}_{k,\tau-1} = w^0 - w_{k,\tau-1}$ is the weight error vector for node k. We assume that the estimate of each node converges to the vicinity of the unknown vector w^0. Therefore $\tilde{w}_{k,\tau-1}$ is small enough such that it is negligible, yielding

$$\begin{aligned} g_k(w_{k,\tau-1}) &\approx -(\int_0^1 H_k(w^0)dx)\tilde{w}_{k,\tau-1} \\ &= -H_k(w^0)\tilde{w}_{k,\tau-1}. \end{aligned} \tag{19}$$

We can also obtain the approximation of the gradient error at the vicinity of w^0, which is given by

$$\begin{aligned} s_k(w_{k,\text{ø}-1}) &\approx s_k(w^0) \\ &= \hat{g}_k(w^0) - g_k(w^0) \\ &= \frac{1}{\sigma^2}\sum_{i=1}^{N}\sum_{j=1}^{N} G_{\sigma\sqrt{2}}(v_{k,i} - v_{k,j})(v_{k,i} - v_{k,j})(u_{k,j} - u_{k,i}). \end{aligned} \tag{20}$$

Substituting (19) and (20) into (16), an approximation of intermediate estimate can be obtained at the vicinity of

$$\varphi_{k,\tau} = w_{k,\tau-1} + \mu_k(\tau)(H_k(w^0)\tilde{w}_{k,\tau-1} - s_k(w^0)). \tag{21}$$

By substituting (21) into the second equation of (10), we get the estimate of unknown parameter as follows

$$w_{k,\tau} = \sum_{l\in N_k} c_{lk}[w_{l,\tau-1} + \mu_k(i)(H_k(w^0)\tilde{w}_{l,\tau-1} - s_l(w^0))]. \tag{22}$$

Using global quantities defined above gives the update equation for the network estimate vector as

$$W_\tau = C(W_{\tau-1} + M_\tau H\tilde{W}_{\tau-1} - M_\tau S). \tag{23}$$

Where H collects the Hessian matrix across the network into the global vector $H = diag(H_1(w^0), \cdots, H_N(w^0))$. Noting that $Cw^{(0)} = w^{(0)}$, subtraction of both sides of (23) from w^0 gives

$$\tilde{W}_\tau = C(I_{MN} - M_\tau H)\tilde{W}_{\tau-1} + CM_\tau S. \tag{24}$$

In view of assumptions A1 and A2, $\tilde{W}_\tau$, H and C are independent of each other. Hence taking expectation of both sides of (24) leads to

$$E[\tilde{W}_\tau] = E[C](I_{MN} - E[M_\tau]H)E[\tilde{W}_{\tau-1}] + CM_\tau E[S]. \tag{25}$$

We can easily find that $E[S] = col\{E[s_1(w^0), \cdots, s_N(w^0)]\} = 0$, the equation (25) has therefore been reduced to this form

$$E[\tilde{W}_\tau] = E[C](I_{MN} - E[M_\tau]H)E[\tilde{W}_{\tau-1}]. \tag{26}$$

From (26), we observe that in order to be stable for the Algorithm 1 in the mean sense, the matrix $E[C](I_{MN} - E[M_\tau]H)$ should be stable. All the entries of $E(C)$ are non-negative and all the rows of it add up to unity. Therefore, to ensure the stability in the mean, it should hold that

$$|\lambda_{\max}\{I_{MN} - E[M_\tau]H\}| < 1. \tag{27}$$

We use the notion $\lambda_{\max}(A)$ to denote the maximum eigenvalue of a Hermitian matrix A. The algorithm will therefore be stable in the mean if

$$\prod_{\tau=0}^{\infty}[I_{MN} - E[\mu_k(\tau)]H_k(w^0)] \to 0.$$

Peer-reviewed version available at Sensors 2017, 17, , 824; doi:10.3390/s17040824

Thus, we note that a sufficient condition for unbiasedness is

$$0 < E[\mu_k(\tau)] < \frac{2}{\lambda_{\max}\{H_k(w^0)\}} \quad \Leftrightarrow 0 < \mu_k < \frac{1}{\lambda_{\max}\{H_k(w^0)\}E[V(0) - V(e_{k,\tau})]}. \tag{28}$$

3.2.2. Mean-square Performance

In order to make the presentation clearer, we shall introduce the following notation

$$\mathbf{\Gamma} = (\mathbf{I}_{MN} - M_\tau H)C^T\mathbf{\Sigma}C(\mathbf{I}_{MN} - M_\tau H).$$

Performing weighted energy balance on both sides of (24), and taking expectations gives

$$E[\|\tilde{W}_\tau\|_\Sigma^2] = E[\|\tilde{W}_{\tau-1}\|_\Gamma^2] + E[S^T M_\tau C^T\mathbf{\Sigma}CM_\tau S]. \tag{29}$$

Where $\mathbf{\Sigma}$ is an arbitrary symmetric nonnegative-definite matrix, and the notion $\|a\|_\Sigma^2 = a^T\mathbf{\Sigma}a$ represents a weighted vector norm for any Hermitian $\mathbf{\Sigma} > 0$. By defining

$$r = vec\{E[\mathbf{\Gamma}]\}, \grave{} = vec\{\mathbf{\Sigma}\}.$$

Where the vec(.) notation stacks the columns of its matrix argument on top of each other. We can modify (29) to

$$E[\|\tilde{W}_\tau\|_{\grave{}}^2] = E[\|\tilde{W}_{\tau-1}\|_r^2] + E[S^T M_\tau C^T\mathbf{\Sigma}CM_\tau S]. \tag{30}$$

Using the following relationship of the vectorization operator and the Kronecker product [28]:

$$vec(ABC) = (C^T \otimes A)vec\{B\}.$$

We can obtain that

$$r = \phi\grave{}. \tag{31}$$

Where

$$\phi = E[(\mathbf{I}_{MN} - E[M_\tau]H) \otimes (\mathbf{I}_{MN} - E[M_\tau]H)]\beta. \tag{32}$$
$$\beta = E[C^T \otimes C^T].$$

Considering Assumption 3, we can approximate (32) as

$$\begin{aligned}\phi &\approx (\mathbf{I}_{M^2N^2} - \mathbf{I}_{MN} \otimes E[M_\tau]H - E[M_\tau]H \otimes \mathbf{I}_{MN})\beta \\ &= (\mathbf{I}_{MN} - E[M_\tau]H) \otimes (\mathbf{I}_{MN} - E[M_\tau]H)\beta.\end{aligned} \tag{33}$$

Using the following relationship of the vectorization operator and the matrix trace [28]:

$$Tr\{\mathbf{A}^T\mathbf{B}\} = vec^T(\mathbf{B})vec(\mathbf{A}).$$

We find that

$$E[S^T M_\tau C^T\mathbf{\Sigma}CM_\tau S] = vec^T Q\beta\theta. \tag{34}$$

Where

$$Q = E[M_\tau SS^T M_\tau].$$

Substituting (31) and (34) into (30), we can then reformulate recursion as follows

$$E[\|\tilde{W}_\tau\|_{\grave{}}^2] = E[\|\tilde{W}_{\tau-1}\|_{\phi\theta}^2] + vec^T Q\beta\theta. \tag{35}$$

Peer-reviewed version available at Sensors 2017, 17, , 824; doi:10.3390/s17040824

It is known that (35) is stable and convergent if the matrix ϕ is stable [11]. Form the equation

$$\beta^T 1_{M^2N^2} = E[(C \otimes \mathbf{I}_M)1_{MN} \otimes (C \otimes \mathbf{I}_M)1_{MN}] = 1_{M^2N^2}.$$

We know that all the entries of β in (34) are non-negative and all its columns sum up to unity. Using the property $\lambda(\mathbf{A} \otimes \mathbf{A}) = \lambda^2(\mathbf{A})$, the stability of ϕ has the same conditions as the stability of $\mathbf{I_{MN}} - E[M_\tau]H$. Therefore, we choose the step size in accordance with (28) which can keep the DMEE-SAS stable in the mean-square sense.

3.3. An Improving Scheme for DMEE-SAS Algorithm

The too small effective step size near the optimal estimator will hinder the tracking ability of DMEE-SAS algorithm in non-stationary environment. In non-stationary environment, the optimal estimator has small changes. A random-walk model is commonly used in the literature to describe the non-stationarity of the weight vector [11].

Therefore, we try to combine the DMEE-SAS algorithm with DMEE algorithm [6] in non-stationary environment where tracking is important. DMEE-SAS algorithm should be used due to the faster convergence when the algorithm start, and DMEE algorithm should be used when algorithm begins to converge. We use Lyapunov stability theory [1] to analyze the switching time for each node.

Lyapunov energy function is a method for analyzing the convergence characteristics of dynamic systems. The cost function can be viewed as a Lyapunov energy function. For DMEE-SAS algorithm, the continuous-time learning rule is

$$\dot{w} = -\mu_{DMEE-SAS} \frac{\partial J_k(w)_{DMEE-SAS}}{\partial w}. \tag{36}$$

The temporal dynamics for the Lyapunov energy that describes the DMEE-SAS algorithm can be obtained as follows

$$\begin{aligned} \dot{J}_k(w)_{DMEE-SAS} &= \sum_{l \in N_k} c_{lk}(-2)[V(0) - V(e_{l,i})] \frac{\partial V(e_{l,i})^T}{\partial w} \dot{w} \\ &= \sum_{l \in N_k} c_{lk}(-4)\mu_{k,DMEE-SAS}[V(0) - V(e_{l,i})]^2 \left\| \frac{\partial V(e_{l,i})}{\partial w} \right\|^2. \end{aligned} \tag{37}$$

The individual local energy function for DMEE algorithm can be written as

$$J_k(w)_{DMEE} = -\sum_{l \in N_k} c_{lk} V(e_{l,i}). \tag{38}$$

For DMEE algorithm, the continuous-time learning rule is

$$\dot{w} = -\mu_{DMEE} \frac{\partial J_k(w)_{DMEE}}{\partial w}. \tag{39}$$

In a similar way, the temporal dynamics for the Lyapunov energy that describes the DMEE algorithm can be obtained as follows

$$\begin{aligned} \dot{J}_k(w)_{DMEE} &= \sum_{l \in N_k} c_{lk} \frac{\partial V(e_{l,i})^T}{\partial w} \dot{w} \\ &= \sum_{l \in N_k} c_{lk}(-\mu_{l,DMEE}) \left\| \frac{\partial V(e_{l,i})}{\partial w} \right\|^2. \end{aligned} \tag{40}$$

Peer-reviewed version available at Sensors 2017, 17, , 824; doi:10.3390/s17040824

The switching time is determined as

$$\left|\dot{J}_k(w)_{DMEE-SAS}\right| > \left|\dot{J}_k(w)_{DMEE}\right| \Leftrightarrow V(e_{l,i}) < V(0) - \frac{1}{2}\sqrt{\frac{\mu_{l,DMEE}}{\mu_{l,DMEE-SAS}}}(l \in N_k). \tag{41}$$

When the condition of (41) is met, we should switch from DMEE-SAS algorithm to DMEE-SAS algorithm. We introduce the following auxiliary variable

$$s_{k,i} = \begin{cases} 1, V(e_{k,i}) < V(0) - \frac{1}{2}\sqrt{\frac{\mu_{k,DMEE}}{\mu_{k,DMEE-SAS}}} \\ 0, \text{otherwise} \end{cases}$$

This yields the following algorithm, which refer to as the improving DMEE-SAS algorithm:

$$\begin{cases} \varphi_{k,i} = w_{k,i-1} - s_{k,i}\mu_{k,DMEE-SAS}\frac{2}{\sigma^2 L}[V(0) - V(e_{k,i})]\sum\limits_{j=i-L+1}^{i} G_{\sigma\sqrt{2}}(e_{k,i} - e_{k,j})(e_{k,i} - e_{k,j})(u_{k,j} - u_{k,i}), \\ -(1 - s_{k,i})\mu_{k,DMEE}\frac{1}{\sigma^2 L}\sum\limits_{j=i-L+1}^{i} G_{\sigma\sqrt{2}}(e_{k,i} - e_{k,j})(e_{k,i} - e_{k,j})(u_{k,j} - u_{k,i}) \\ w_{k,i} = \sum\limits_{l\in N_k} c_{lk}\varphi_{l,i}. \end{cases} \tag{42}$$

For the purpose of clarity, we summarize the procedure of the Improving DMEE-SAS algorithm in Algorithm 2.

Algorithm 2: Improving DMEE-SAS Algorithm

Initialize: $w_{k,i} = 0$
for $i = 1 : T$
for each node k:
Adaptation
each node calculates the switching time using (41).
each node updates intermediate estimate $\varphi_{k,i}$ according to the first equation of (42).
Combination
$w_{k,i} = \sum\limits_{l\in N_k} c_{lk}\varphi_{l,i}$
end for

4. Simulation results

20 sensors are randomly placed in a square 100×100 shown in Fig. 1. The communication distance is set as 50. In this paper, the performance of the steady-state network MSD [7] is adopted for performance comparison. All of the performance measures are averaged over 100 trials.

We employ the super-Gaussian distribution as the noise model in our simulations. We generate the noise from the zero-mean generalized Gaussian distribution (GGD)of pdf $q_V(v) = \propto \exp(-|v|^p)$,where p is a positive shape parameter of GGD [26]. We set $p = 0.6$ to make the noise distribution be super-Gaussian.

a) In stationary environment

Here, the proposed DMEE-SAS algorithm performance is compared with that of some existing algorithms in the literature. We assume the communication link is ideal link. The unknown parameter vector w^0 is set to $\frac{1}{\sqrt{6}}, \frac{1}{\sqrt{6}}, \frac{1}{\sqrt{6}}, \frac{1}{\sqrt{6}}, \frac{1}{\sqrt{6}}, \frac{1}{\sqrt{6}}]^T$.

We set the window length $L = 8$ and kernel size σ=1.5 for both DMEE and DMEE-SAS algorithms. Further, the p is 1.2 for DLMP algorithm. The steady state MSD curves are plotted in Fig. 2. It is found that DMEE-SAS algorithm is robust to the non-Gaussian noises and performs better than DLMP algorithm [5] and DLMS [7]. DMEE-SAS algorithm achieves better convergence

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 28 February 2017 doi:10.20944/preprints201702.0106.v1

Peer-reviewed version available at Sensors 2017, 17, , 824; doi:10.3390/s17040824

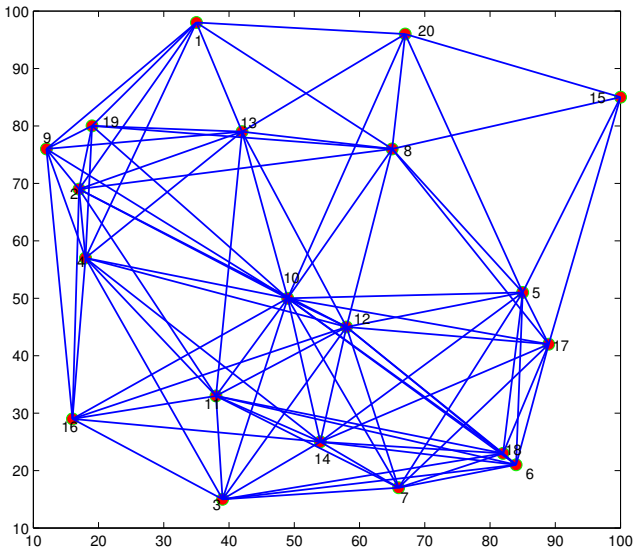

Figure 1. Network topology.

performance than the DMEE [6] algorithm when the DMEE-SAS and DMEE algorithms achieve comparable performance.

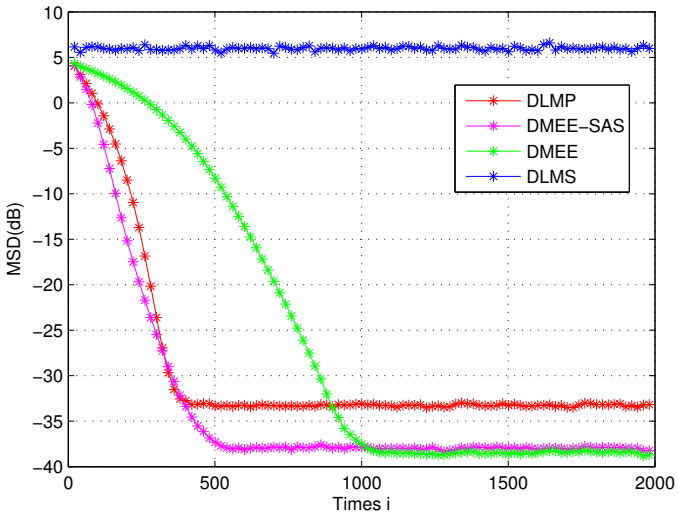

Figure 2. Transient MSD curve.

b) In non-stationary environment

Here, the simulations are carried out in the same environments as those shown in 5.1 subsection, except for the optimal estimator w^0. We compare the proposed Improving DMEE-SAS algorithm with other algorithms.

Motivated by [14], we assume a time-varying w^0 of length 6 as follows:

$$w_i^0 = \frac{1}{2}[a_{1,i}, a_{2,i}, a_{3,i}, a_{4,i}, a_{5,i}, a_{6,i}]^T.$$

Where $a_{k,i} = [\cos(wi + \frac{(k-1)}{2}\pi)]$ for $k = 1, 2, 3, 4, 5, 6$ and $w = \frac{\pi}{3000}$.

The unknown link is assume to change at time 6000. In Fig. 3, the Improving DMEE-SAS algorithm can detect the weight vector change and the performance of it is better than DLMS algorithm. We observe that Improving DMEE-SAS and DMEE algorithms achieve comparable performance and Improving DMEE-SAS achieves better convergence performance than the DMEE algorithm. When compared with DMEE-SAS algorithm, the Improving DMEE-SAS algorithm exhibits a significant improvement in performance when the estimate near to optimal estimator. Improving DMEE-SAS algorithm achieves a low MSD and fast rate of convergence in the non-stationary environment.

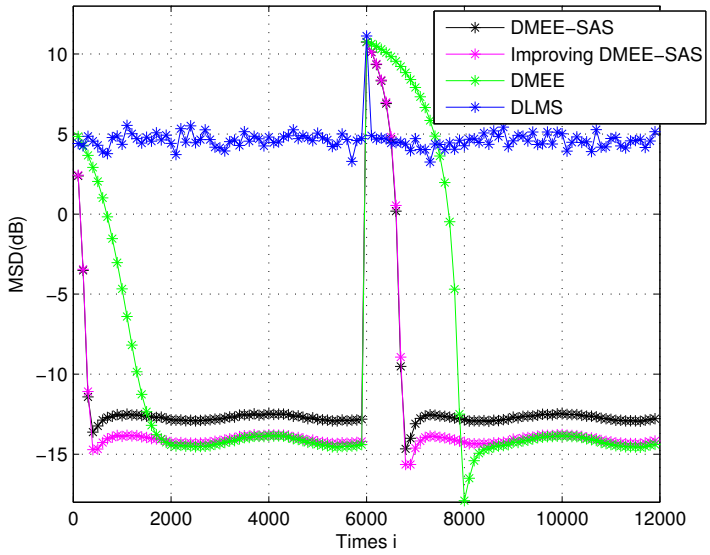

Figure 3. MSD learning curves in a non-stationary environment.

5. Conclusions

In this paper, a robust diffusion estimation algorithm with self-adjusting step-size is developed which called DMEE-SAS algorithm. The mean and mean square convergence analysis of this new algorithm are carried out, and a sufficient condition for ensuring the stability is obtained. Simulation results illustrate that DMEE-SAS algorithm can achieve better performance than the DLMS, robust DLMP, and DMEE algorithms in non-Gaussian noise scenario. Besides, we propose the Improving DMEE-SAS algorithm using in the non-stationary scenario where the unknown parameter is changing over time. The Improving DMEE-SAS algorithm combined the DMEE-SAS WITH DMEE algorithms and it can avoid the small effective step-size of DMEE-SAS algorithm when close to the optimal estimator.

Acknowledgment

This work was supported in part by the Doctoral Fund of Southwest University (No.SWU113067), the National Natural Science Foundation of China (Grant Nos.61672436, 61571372, 61372139, 61101233, 60972155), Fundamental Research Funds for the Central Universities (Grant Nos.XDJK2014A009, XDJK2016A001, XDJK2014C018) and Program for New Century Excellent Talents in University(Grant Nos.[2013]47).

References

1. Khalil, Hassan K, Nonlinear Systems, Macmillan, New York, 1992.
2. Kong, Jun-Taek and Lee, Jae-Woo and Song, Woo-Jin, Diffusion LMS algorithm with multi-combination for distributed estimation over networks. IEEE, 2013 Asilomar Conference on Signals, Systems and Computers, 2013: 438-441.
3. Xiao, Lin and Boyd, Stephen, Fast linear iterations for distributed averaging. Systems & Control Letters, 2004, 53(1): 65-78.
4. Chouvardas, Symeon and Slavakis, Konstantinos and Theodoridis, Sergios, Adaptive robust distributed learning in diffusion sensor networks. IEEE Transactions on Signal Processing, 2011, 59(10): 4692-4707.
5. Wen, F., Diffusion least-mean P-power algorithms for distributed estimation in alpha-stable noise environments. Electronics Letters, 2013, 49(21): 1355-1356.
6. Li, Chunguang and Shen, Pengcheng and Liu, Ying and Zhang, Zhaoyang, Diffusion information theoretic learning for distributed estimation over network. IEEE Transactions on Signal Processing, 2013, 61(61): 4011-4024.
7. Cattivelli, Federico S and Sayed, Ali H, Diffusion LMS strategies for distributed estimation. IEEE Transactions on Signal Processing, 2010, 58(3): 1035-1048.
8. Han, Seungju and Rao, Sudhir and Erdogmus, Deniz and Jeong, Kyu-Hwa and Principe, Jose, A minimum-error entropy criterion with self-adjusting step-size (MEE-SAS). Signal Processing, 2007, 87(11): 2733-2745.
9. Arablouei, Reza and Werner, Stefan and Doğançay, Kutluyıl, Analysis of the gradient-descent total least-squares adaptive filtering algorithm. IEEE Transactions on Signal Processing, 2014, 62(5): 1256-1264.
10. Kelley, C. T., Iterative methods for optimization. SIAM, Philadelphia. J.appl.probab, 1999, 9: 878-878.
11. Sayed, AH, Adaptive Filters. Hoboken. NJ: John Wiley & Sons, 2008.
12. Chen, Jianshu and Sayed, Ali H, Diffusion adaptation strategies for distributed optimization and learning over networks. IEEE Transactions on Signal Processing, 2012, 60(8): 4289-4305.
13. Sayin, Muhammed O and Vanli, N Denizcan and Kozat, Suleyman Serdar, A novel family of adaptive filtering algorithms based on the logarithmic cost. IEEE Transactions on Signal Processing, 2014, 62(17): 4411-4424.
14. Zhao, Xiaochuan and Tu, Sheng-Yuan and Sayed, Ali H, Diffusion adaptation over networks under imperfect information exchange and non-stationary data. IEEE Transactions on Signal Processing, 2012, 60(7): 3460-3475.
15. Lopes, Cassio G and Sayed, Ali H, Incremental adaptive strategies over distributed networks. IEEE Transactions on Signal Processing, 2007, 55(8): 4064-4077.
16. Rabbat, Michael G and Nowak, Robert D, Quantized incremental algorithms for distributed optimization. IEEE Journal on Selected Areas in Communications, 2005, 23(4): 798-808.
17. Nedic, Angelia and Bertsekas, Dimitri P, Incremental subgradient methods for nondifferentiable optimization. SIAM Journal on Optimization, 2001, 12(1): 109-138.
18. Kar, Soummya and Moura, José MF, Distributed consensus algorithms in sensor networks with imperfect communication: Link failures and channel noise. 2009, 57(1): 355-369.
19. Cattivelli, Federico S and Sayed, Ali H, Diffusion strategies for distributed Kalman filtering and smoothing. 2010, 55(9): 2069-2084.
20. Middleton, David, Non-Gaussian noise models in signal processing for telecommunications: new methods an results for class A and class B noise models. IEEE Transactions on Information Theory, 1999, 45(4): 1129-1149.
21. Zoubir, Abdelhak M and Koivunen, Visa and Chakhchoukh, Yacine and Muma, Michael, Robust estimation in signal processing: A tutorial-style treatment of fundamental concepts. IEEE Signal Processing Magazine, 2012, 29(4): 61-80.
22. Zoubir, Abdelhak M and Brcich, Ramon F, Multiuser detection in heavy tailed noise. Digital Signal Processing, 2002, 12(2): 262-273.
23. Khalili, Azam and Tinati, Mohammad Ali and Rastegarnia, Amir and Chambers, Jonathon A, Transient analysis of diffusion least-mean squares adaptive networks with noisy channels. nternational Journal of Adaptive Control and Signal Processing, 2012, 26(2): 171-180.

24. Zhao, Xiaochuan and Tu, Sheng-Yuan and Sayed, Ali H, Diffusion adaptation over networks under imperfect information exchange and non-stationary data. IEEE Transactions on Signal Processing, 2012, 60(7): 3460-3475.
25. Carli, Ruggero and Chiuso, Alessandro and Schenato, Luca and Zampieri, Sandro, Distributed Kalman filtering based on consensus strategies. IEEE Journal on Selected Areas in Communications, 2008, 26(4): 622-633.
26. Li, Hualiang and Li, Xi-Lin and Anderson, Matthew and Adali, Tülay, A class of adaptive algorithms based on entropy estimation achieving CRLB for linear non-Gaussian filtering. IEEE Transactions on Signal Processing, 2012, 60(4): 2049-2055.
27. Cattivelli, Federico S and Lopes, Cassio G and Sayed, Ali H, A diffusion RLS scheme for distributed estimation over adaptive networks. 2007 IEEE 8th Workshop on Signal Processing Advances in Wireless Communications, 2007, 1-5.
28. Abadir, Karim M and Magnus, Jan R, Matrix algebra. Cambridge University Press, 2005, 1.
29. Shen, Pengcheng and Li, Chunguang, Minimum Total Error Entropy Method for Parameter Estimation. IEEE Transactions on Signal Processing, 2015, 63(15): 4079-4090.