SCIENCE CHINA
Information Sciences

• RESEARCH PAPER •

September 2014, Vol. 57 092103:1–092103:10
doi: 10.1007/s11432-013-4944-0

The incremental subgradient methods on distributed estimations in-network

FENG Hui, JIANG ZiDong, HU Bo* & ZHANG JianQiu

Department of Electronic Engineering, Fudan University, Shanghai 200433*, China*

Received June 25, 2013; accepted November 8, 2013; published online January 5, 2014

Abstract A unified framework on distributed estimations in-network using incremental subgradient (IS) methods is introduced. The IS methods meet the requirement of the asynchronously processing, which are efficient solvers developed recently focusing on separable non-differentiable convex optimization problems. The main contribution of this paper is to formalize distributed estimators as equivalent separable convex optimization problems, where general skills and several specific cases on signal estimations are presented. Analytical and simulation results show that the IS framework can solve general nonlinear estimation problems in-network, and achieve comparable performances as the centralized estimators.

Keywords distributed signal processing, signal processing in-network, distributed estimation, incremental subgradient, optimization methods

Citation Feng H, Jiang Z D, Hu B, et al. The incremental subgradient methods on distributed estimations in-network. Sci China Inf Sci, 2014, 57: 092103(10), doi: 10.1007/s11432-013-4944-0

1 Introduction

Over the past decades, signal processing in-network has attracted lots of interest from various research fields. The fundamental motivation is to design a reasonable scheme which can distribute the computational task over the whole network. In many practical networks, such as wireless ad-hoc networks, the concurrent transmission rate is bounded [1]. It implies that any distributed algorithm on such networks has to consider the communication cost besides the computational complexity on single node.

Traditional in-network signal processing methods are fulfilled on a fusion center, where centralized algorithms handle all the data or measurements from nodes and then export the results to the network. Such methods suffer from some difficulties: 1) heavy communication costs and long delay between the nodes and the fusion center; 2) possible single point failure on the whole system. Considering the deficiencies of centralized methods, it is desirable to develop real distributed methods which do not need any fusion centers. In this way, several benefits are present: 1) all nodes do computation with local (nearby) information following simple rules; 2) all nodes only exchange information with their neighbor nodes; 3) the network's dynamic behavior will converge to the expected result.

Among varieties of distributed processing ways, consensus average (CA) and distributed optimization approaches are two of the mainstream methods used in recent publications. In consensus context, each

*Corresponding author (email: bohu@fudan.edu.cn)

node only exchanges information with its neighbors within communication range. By proper designed distributed communication schemes, the whole network will converge to a consensus state, usually the average of all nodes' initial states, which is faster and more efficient than all-to-all broadcast schemes [2]. Under reasonable assumptions, distributed estimation expressions on a static signal on a network, e.g. best linear unbiased estimator (BLUE), linear minimum mean squares error estimator (LMMSE), least mean squares estimator (LMS) and even Kalman filter (KF), could be decomposed into a summation of many local calculations, and can then be solved in CA framework [3]. CA based methods require synchronized transmission over the network, which restricts their applications in some practical scenarios. CA methods also perform not very well in convergence rate and communication efficiency, compared with other distributed methods as introduced as follows.

Distributed optimization is another approach to solving the signal processing problems in-network. For instance, many estimators in statistical processing field with analytical solutions are derived from some optimization problems. One usually used distributed optimization method is the alternating direction method of multipliers (AD-MoM) [4], which has been applied to solve distributed estimation problems [5–8]. AD-MoM is a general optimization method based on the Lagrangian dual framework, so it can be applied to the general distributed estimation on deterministic and/or random signals [5,6], such as maximum likelihood (ML), BLUE, maximum a posteriori (MAP), and MMSE. Furthermore, recursive estimation problems such as recursive least squares filter (RLS), LMS and KF can also be integrated into the AD-MoM framework [7,8]. AD-MoM methods need so-called bridge nodes to accomplish the information exchange over the network, whereas the selection of bridge nodes is an intractable problem [9]. Also, AD-MoM requires synchronous information exchange among neighboring nodes, which comes with high communication cost and computation complexity in some cases as discussed in Section 5.

Incremental subgradient (IS) method is an asynchronous approach developed to solve separable convex optimization problems [10–12]. Incremental approach only handles single objection in one iteration and relay the intermittent result to the successor. Compared with the traditional gradient methods, IS does not require object functions be differentiable, which broadens the field of its applications. IS has been used to solve the least square regression problem [10] and SVM classification case [13], while it can potentially be used to solve any separable optimization problems as indicated in [14]. Besides determined objective functions, the IS methods can also be used to solve stochastic optimization problem [15,16], which actually is some incremental stochastic gradient algorithms.

There are several incremental ways of the IS methods, mainly classified by determined or random relay paths. The seminal paper [10] proposed cyclic and uniform randomized relay modes, and analyzed their convergence properties of several stepsize rules. Later, Markovian relay mode is also proposed [16,17], where relay among neighboring nodes follow a Markovian transition probability matrix.

Our main contribution of this work is to formalize the classical distributed estimation problems as optimization problems, so that distributed estimations can be accomplished with the IS framework. The outline of this paper is as follows. Some necessary definitions and related work on the IS methods are given and simply reviewed in Section 2. The key equivalent relationships between some typical distributed estimators with closed-form expressions and the separated convex optimization problems are established in Section 3, where the general form of the distributed incremental subgradient (DIS) estimator in-network is formulized. Specific forms of the DIS estimator on the classical distributed estimation problems are presented in Sections 4 and 5. The simulation results of our DIS estimator are reported in Section 6. Finally, some concluding remarks are given in Section 7.

2 Incremental subgradient methods

Consider a distributed estimation problem on a network with N independent observers, where only single-hop communications between neighbors are allowed. The network is modeled as an undirected graph with symmetric links. We consider the problem as

$$\min \ f(\boldsymbol{x}) = \sum_{i=1}^{N} f_i(\boldsymbol{x}) \ \text{ s.t. } \ \boldsymbol{x} \in \mathcal{X}, \tag{1}$$

where f_i are convex functions (but not necessarily differentiable), and $\mathcal{X}$ is a nonempty, closed, and convex subset of $\mathbb{R}^n$. The signal $\boldsymbol{x} \in \mathbb{R}^p$ to be estimated may be a deterministic or random signal. The observation of the node i to signal $\boldsymbol{x}(t)$ is described as $\boldsymbol{z}_i(t) = \boldsymbol{f}_i(\boldsymbol{x}(t),\ \boldsymbol{w}_i(t))$, where $\boldsymbol{w}_i(t)$ denotes the measurement noise. We focus on random relay modes of the IS methods, which is more suitable for in-network processing.

The IS iterative process can be written as

$$\boldsymbol{x}(k+1) = P_{\mathcal{X}}(\boldsymbol{x}(k) - \alpha_k g_{\omega_k}(\boldsymbol{x}(k))), \tag{2}$$

where α_k is the positive stepsize, $P_{\mathcal{X}}$ is the projection operator on the set $\mathcal{X}$. $g_{\omega_k}(\cdot)$ is a subgradient of $f_{\omega_k}(\cdot)$, where ω_k is a sequence random variables over the index set $1, \ldots, N$. The procedure of the iteration is that a sub-function $f_{\omega_k}(\cdot)$ is selected each instance k, and then steps forward the negative subgradient direction of $f_{\omega_k}(\cdot)$ at $\boldsymbol{x}_k$. Compared with traditional gradient or subgradient based methods, the main advantage of the IS methods is that each node only needs to carry out local optimization without global information.

2.1 Uniform random relay mode

In uniform random relay mode [10], ω_k is a sequence of random variables uniformly selected over the index set. In order to ensure that the iteration converges to f^* as $k \to \infty$, we offer the following assumption:

Assumption 1. The set of subgradients $\{g_{\omega_k}(\cdot)\}$ for all $k \leqslant 0$ is bounded, i.e., there exists a positive constant C such that $\|g_{\omega_k}(\cdot)\| \leqslant C$ with probability 1.

Several stepsize rules which the constant and diminishing stepsize rules are two fundamental ones.

Theorem 1 (Constant stepsize rule [10])**.** Assume Assumption 1 holds. Then, for the sequence $\boldsymbol{x}_k$ generated by the incremental method (2) in uniform random relay mode, with the constant positive stepsize α, we have

(a) if $f^* = -\infty$, then $\inf_{k \geqslant 0} f(x_k) = f^*$ almost surely (a.s.),

(b) if $f^* > -\infty$, then $\inf_{k \geqslant 0} f(x_k) \leqslant f^* + \frac{\alpha N C^2}{2}$ a.s.

Theorem 2 (Diminishing stepsize rule [10])**.** Assume Assumption 1 holds and let the optimal set $\mathcal{X}^*$ be nonempty. If the stepsize diminishes such that $\alpha_k > 0$, $\sum_{k=0}^{\infty} \alpha_k = \infty$, $\sum_{k=0}^{\infty} \alpha_k^2 < \infty$, then, for the sequence $\boldsymbol{x}_k$ generated by the incremental method (2) in uniform random relay mode converges to some optimal solution $\boldsymbol{x}^*$ almost surely.

2.2 Markovian relay mode

Although owing to fast convergence rate of the uniform random relay mode, the random relay between any pair across the network is not applicable. If only neighboring transfer is allowed, Markovian relay mode is more feasible. One kind of transition matries in [16], also named the Perron matrix in [2], is given for instance

$$[\boldsymbol{P}]_{i,j} = \begin{cases} \varepsilon, & \text{if } j \neq i \text{ and } j \in \mathcal{N}_i, \\ 1 - \varepsilon|\mathcal{N}_i|, & \text{if } j = i, \\ 0, & \text{otherwise}, \end{cases} \tag{3}$$

where $[\boldsymbol{P}]_{i,j}$ is the transfer probability from node i to j at time k, $\mathcal{N}_i$ is the neighbor set of node i, and $|\mathcal{N}_i|$ is its cardinality, i.e., the degree of node i. The constant ε is chosen to guarantee that the Perron matrix is primitive such that we have the following lemma.

Lemma 1 ([2])**.** Let $\varepsilon < 1/\Delta$, where Δ is the maximum degree of all nodes. The matrix $\boldsymbol{P}$ owns the following properties, if the underlying topology is a connected undirected graph, i.e., there is an edge from node i to j if and only if $[\boldsymbol{P}]_{i,j} > 0$.

(a) The Perron matrix $\boldsymbol{P}$ is primitive and hence has one strictly maximum eigenvalue.

(b) All eigenvalues of $\boldsymbol{P}$ lie within the interval $[-1, 1]$, which can be ordered as $-1 \leqslant \mu_1 \leqslant \mu_2 \leqslant \cdots \leqslant \mu_{N-1} \leqslant \mu_N = 1$.

(c) $\boldsymbol{P}$ is a doubly stochastic matrix, which means $\mathbf{1}^{\mathrm{T}}\boldsymbol{P} = \mathbf{1}^{\mathrm{T}}$ and $\boldsymbol{P}^{\mathrm{T}}\mathbf{1} = \mathbf{1}$.

Theorem 3 (Constant stepsize rule [17])**.** Assume Assumption 1 holds. Then, for the sequence $\boldsymbol{x}_k$ generated by (2) in Markovian relay mode, with the constant positive stepsize α, we have (a) If $f^* = -\infty$, then $\liminf_{k\to\infty} f(x_k) = f^*$ a.s. (b) If $f^* > -\infty$, then $\liminf_{k\to\infty} f(x_k) \leqslant f^* + \frac{\alpha K C_0^2}{2}$ a.s., where $K = \max_i E[(R_k^i)^2]$, and R_k^i is the recurrent time of node i.

Theorem 4 (Diminishing stepsize rule [16])**.** Assume Assumption 1 holds and let the set $\mathcal{X}$ be bounded. Assume a stepsize rule $\alpha^k = \alpha/k^p$, where $2/3 < p \leqslant 1$. Then, for the sequence $\boldsymbol{x}_k$ generated by (2) in Markovian relay mode, we have $\liminf_{k\to\infty} f(x_k) = f^*$ a.s.

It is obvious the result of the Markovian relay mode is weaker than the uniform relay mode. However, it does not degrade the performance a lot in practice. A trick in Markovian relay mode is that each node preservers two intermittent results, one is the current value and the other is the minimal value in history.

3 IS estimation framework

3.1 Equivalence on distributed estimation problems

The central issue of this paper is how to transform an analytical global optimization solution into a separable optimization problem and solve it using the IS methods. Many distributed estimators have the closed-form expressions as given in the following lemmas, which have equivalent relationships with constructed distributed optimization problems.

Lemma 2. Assume $\sum_{i=1}^N {\boldsymbol{A}_i}^{\mathrm{T}}\boldsymbol{A}_i$ is invertible. The solution $\hat{\boldsymbol{x}} = (\sum_{i=1}^N {\boldsymbol{A}_i}^{\mathrm{T}}\boldsymbol{A}_i)^{-1}\sum_{i=1}^N {\boldsymbol{A}_i}^{\mathrm{T}}\boldsymbol{b}_i$ is equivalent to a distributed optimization problem $\hat{\boldsymbol{x}} = \arg\min_{\boldsymbol{x}} \sum_{i=1}^N \|\boldsymbol{A}_i\boldsymbol{x} - \boldsymbol{b}_i\|_2^2$.

Proof. Note $\|\boldsymbol{A}_i\boldsymbol{x} - \boldsymbol{b}_i\|_2^2$ is a convex function of x, take the derivative of $\sum_{i=1}^N \|\boldsymbol{A}_i\boldsymbol{x} - \boldsymbol{b}_i\|_2^2$ with respect to x and let it be zero, $\sum_{i=1}^N \left[\boldsymbol{A}_i^{\mathrm{T}}(\boldsymbol{A}_i\boldsymbol{x} - \boldsymbol{b}_i)\right] = \mathbf{0}$, then we have the unique result.

Lemma 3. If all $\boldsymbol{A}_i$ are positive definite matrices, the solution $\hat{\boldsymbol{x}} = (\sum_{i=1}^N \boldsymbol{A}_i)^{-1}\sum_{i=1}^N \boldsymbol{b}_i$ is equivalent to a distributed optimization problem $\hat{\boldsymbol{x}} = \arg\min_{\boldsymbol{x}} \sum_{i=1}^N \|\boldsymbol{A}_i^{1/2}\boldsymbol{x} - \boldsymbol{A}_i^{-1/2}\boldsymbol{b}_i\|_2^2$.

Proof. By the eigen-decomposition of $\boldsymbol{A}_i$, we know $\boldsymbol{A}_i^{1/2}$ and $\boldsymbol{A}_i^{-1/2}$ exist and are also positive definite. Note that $\|\boldsymbol{A}_i^{1/2}\boldsymbol{x} - \boldsymbol{A}_i^{-1/2}\boldsymbol{b}_i\|_2^2$ is a convex function of x, take the derivative of $\sum_{i=1}^N \|\boldsymbol{A}_i^{1/2}\boldsymbol{x} - \boldsymbol{A}_i^{-1/2}\boldsymbol{b}_i\|_2^2$ with respect to x and let it be zero: $\sum_{i=1}^N [(\boldsymbol{A}_i^{1/2})^{\mathrm{T}}(\boldsymbol{A}_i^{1/2}\boldsymbol{x} - \boldsymbol{A}_i^{-1/2}\boldsymbol{b}_i)] = \sum_{i=1}^N (\boldsymbol{A}_i\boldsymbol{x} - \boldsymbol{b}_i) = \mathbf{0}$. Then we have the unique result, where we use the fact that $\sum_{i=1}^N \boldsymbol{A}_i$ is invertible if all $\boldsymbol{A}_i$ are positive definite.

Special equivalent cases on BLUE and LMMSE were proposed in [5,6]. Here we give the general forms which can be applied in more cases.

3.2 General randomized DIS estimator

As we have built a bridge from analytical estimators to incremental subgradient optimization problems, we can present the general algorithms of our DIS estimator as Algorithm 1.

4 Distributed estimators on deterministic signals

4.1 DIS-MLE

For a deterministic signal $\boldsymbol{x}$, each sensor node observes $\mathbf{z}_i \in \mathbb{R}^{L_i}$ independently with the probability density function (pdf) $p_i(\boldsymbol{z}_i; \boldsymbol{x})$. If $p_i(\boldsymbol{z}_i; \boldsymbol{x})$ is known, the maximum likelihood estimator (MLE) is given by $\hat{\boldsymbol{x}}_{ML} = \arg\min_{\boldsymbol{x}} -\sum_{i=1}^N \ln[p_i(\boldsymbol{z}_i; \boldsymbol{x})]$. Here we assume

Algorithm 1 General randomized DIS estimator

1. Initialization

Select an initial value $\boldsymbol{x}(0)$ and do other initializations.

2. Incremental iteration

For $k = 0, 1, 2, \ldots$

Calculate the subgradient $\boldsymbol{g}_i(\boldsymbol{x}(k))$ of f_i,

Select a stepsize α_k,

$\boldsymbol{x}(k+1) = P_{\mathcal{X}}(\boldsymbol{x}(k) - \alpha_k \boldsymbol{g}_i(\boldsymbol{x}(k)))$,

Transmit $\boldsymbol{x}(k+1)$ to node $j \in \mathcal{N}_i$ according to (3).

3. Stopping

When $\boldsymbol{x}(k)$ reaches to a steady value, stop the iteration.

Assumption 2. The underlying graph of the network is fully-connected, where the transmissions between nodes are error-free.

Assumption 3. The pdfs $p_i(\boldsymbol{z}_i; \boldsymbol{x})$ are log-concave with respect to $\boldsymbol{x}$.

The network connectivity in Assumption 2 ensures the utilization of all observations. The log-concave in Assumption 3 is the requirements of global uniqueness of the centralized ML estimator. Algorithm 1 can be applied to DIS-MLE directly, where $f_i = -\ln[p_i(\boldsymbol{z}_i; \boldsymbol{x})]$ is a convex function.

4.2 DIS-BLUE

For a popular linear measurement model $\boldsymbol{z}_i = \boldsymbol{H}_i\boldsymbol{x} + \boldsymbol{w}_i$. Let $\boldsymbol{H}^{\mathrm{T}} = [\boldsymbol{H}_1^{\mathrm{T}}, \boldsymbol{H}_2^{\mathrm{T}}, \ldots, \boldsymbol{H}_N^{\mathrm{T}}]^{\mathrm{T}}$, $\boldsymbol{w}^{\mathrm{T}} = [\boldsymbol{w}_1^{\mathrm{T}}, \boldsymbol{w}_2^{\mathrm{T}}, \ldots, \boldsymbol{w}_N^{\mathrm{T}}]^{\mathrm{T}}$, and the total observations vector $\boldsymbol{z}^{\mathrm{T}} = [\boldsymbol{z}_1^{\mathrm{T}}, \boldsymbol{z}_2^{\mathrm{T}}, \ldots, \boldsymbol{z}_N^{\mathrm{T}}]^{\mathrm{T}}$. The augmented measurement formula is $\boldsymbol{z} = \boldsymbol{H}\boldsymbol{x} + \boldsymbol{w}$, for which the Best Linear Unbiased Estimator (BLUE) is

$$\hat{\boldsymbol{x}}_{BL} = \left(\boldsymbol{H}^{\mathrm{T}}\boldsymbol{C}^{-1}\boldsymbol{H}\right)^{-1}\boldsymbol{H}^{\mathrm{T}}\boldsymbol{C}^{-1}\boldsymbol{z}, \tag{4}$$

where $\boldsymbol{C}$ is the blocked variance matrix of all noise $\boldsymbol{w}_i$. However, we cannot solve (4) in a centralized way. If all the noises are uncorrelated, (4) can be decomposed into

$$\hat{\boldsymbol{x}}_{BL} = \left(\sum_{i=1}^{N}\boldsymbol{H}_i^{\mathrm{T}}\boldsymbol{C}_{ii}^{-1}\boldsymbol{H}_i\right)^{-1}\sum_{i=1}^{N}\boldsymbol{H}_i^{\mathrm{T}}\boldsymbol{C}_{ii}^{-1}\boldsymbol{z}_i, \tag{5}$$

where $\boldsymbol{C}_{ii}$ is the variance matrix of noise $\boldsymbol{w}_i$. It is well known that the original derivation of BLUE is a solution of a constrained optimization problem [18]. However, we construct an unconstrained optimization to get the equivalent result as indicated in Proposition 1,

Proposition 1. The analytical estimator (5) is equivalent to the following optimization problem: $\hat{\boldsymbol{x}}_{BL} = \arg\min_{\boldsymbol{x}}\sum_{i=1}^{N} f_i(\boldsymbol{x})$, where $f_i(\boldsymbol{x}) = \|\boldsymbol{C}_{ii}^{-1/2}\boldsymbol{H}_i\boldsymbol{x} - \boldsymbol{C}_{ii}^{-1/2}\boldsymbol{z}_i\|_2^2$.

Proof. Note the covariance matrix $\boldsymbol{C}_{ii}$ is positive definite, so $\boldsymbol{C}_{ii}^{-1/2}$ exists and is also positive definite. Letting $\boldsymbol{A}_i = \boldsymbol{C}_{ii}^{-1/2}\boldsymbol{H}_i$ and $\boldsymbol{b}_i = \boldsymbol{C}_{ii}^{-1/2}\boldsymbol{z}_i$, we complete the proof by Lemma 2, where we use the fact that $\boldsymbol{H}_i^{\mathrm{T}}\boldsymbol{C}_{ii}^{-1}\boldsymbol{H}_i$ is positive definite such that $\sum_{i=1}^{N}\boldsymbol{H}_i^{\mathrm{T}}\boldsymbol{C}_{ii}^{-1}\boldsymbol{H}_i$ is invertible.

Algorithm 1 can be applied here, where the subgradient of $f_i(\boldsymbol{x})$ is equal to its gradient $\boldsymbol{g}_i(\boldsymbol{x}) = 2(\boldsymbol{H}_i^{\mathrm{T}}\boldsymbol{C}_{ii}^{-1}\boldsymbol{H}_i\boldsymbol{x} - \boldsymbol{H}_i^{\mathrm{T}}\boldsymbol{C}_{ii}^{-1}\boldsymbol{z}_i)$. Compared with the conventional derivation of BLUE, the optimization problem in Proposition 1 focuses on the signal $\boldsymbol{x}$ directly, rather than the linear gain on the observations.

4.3 DIS-LSE

For the estimation performance in the observation space, there is a well-known least square estimator (LSE) to minimize the square error $\hat{\boldsymbol{x}}_{LS} = \arg\min_{\boldsymbol{x}}\|\boldsymbol{z} - \boldsymbol{H}\boldsymbol{x}\|_2^2 = \arg\min_{\boldsymbol{x}}\sum_{i=1}^{N}\|\boldsymbol{z}_i - \boldsymbol{H}_i\boldsymbol{x}\|_2^2$, which leads to a closed-form expression $\hat{\boldsymbol{x}}_{LS} = (\boldsymbol{H}^{\mathrm{T}}\boldsymbol{H})^{-1}\boldsymbol{H}^{\mathrm{T}}\boldsymbol{z} = (\sum_{i=1}^{N}\boldsymbol{H}_i^{\mathrm{T}}\boldsymbol{H}_i)^{-1}\sum_{i=1}^{N}\boldsymbol{H}_i^{\mathrm{T}}\boldsymbol{z}_i$.

In fact, LSE can be considered as a special case of BLUE when all the variance matrices are identical matrices. So Algorithm 1 can be applied here, where $f_i(\boldsymbol{x}) = \|\boldsymbol{z}_i - \boldsymbol{H}_i\boldsymbol{x}\|_2^2$ and its subgradient is equal to the gradient $\boldsymbol{g}_i(\boldsymbol{x}) = 2(\boldsymbol{H}_i^{\mathrm{T}}\boldsymbol{H}_i\boldsymbol{x} - \boldsymbol{H}_i^{\mathrm{T}}\boldsymbol{z}_i)$.

5 Distributed estimators on random signals

5.1 DIS-MAP

If the signal $\boldsymbol{x}$ is a random variable with known a priori pdf $p(\boldsymbol{x})$, the MAP estimator under independently conditional observations can be written as $\hat{\boldsymbol{x}}_{MAP} = \arg\min_{\boldsymbol{x}} -\sum_{i=1}^{N} (\ln[p_i(\boldsymbol{z}_i|\boldsymbol{x})] + N^{-1}\ln[p(\boldsymbol{x})])$.

As in Assumption 3, we assume

Assumption 4. $f_i(\boldsymbol{x}) = -\ln[p_i(\boldsymbol{z}_i|\boldsymbol{x})] - N^{-1}\ln[p(\boldsymbol{x})]$ is convex with respect to $\boldsymbol{x}$.

Then Algorithm 1 can be applied here, where $f_i(\boldsymbol{x})$ is not necessarily differentiable.

5.2 DIS-LMMSE

For the convenience of computation, $\boldsymbol{x}$ and $\boldsymbol{z}_i$ are assumed to be zero-mean without loss of generality, as we can compensate for the mean value after computation. Further assuming that $\boldsymbol{x}$ is the linear combination of observations $\boldsymbol{x}_i$, the LMMSE estimator is formulized as $\hat{\boldsymbol{x}}_{\text{LMMSE}} = \boldsymbol{C}_{\boldsymbol{xz}}\boldsymbol{C}_{\boldsymbol{zz}}^{-1}\boldsymbol{z}$, where $\boldsymbol{C}_{xz}$ is the covariance matrix of $\boldsymbol{x}$ and $\boldsymbol{z}$, and $\boldsymbol{C}_{zz}$ is the variance matrix of $\boldsymbol{z}$ [18]. For the linear model $\boldsymbol{z}_i = \boldsymbol{H}_i\boldsymbol{x} + \boldsymbol{w}_i$ when the noises $\boldsymbol{w}_i$ are uncorrelated, the LMMSE result can be decomposed into $\hat{\boldsymbol{x}}_{\text{LMMSE}} = (\boldsymbol{C}_{\boldsymbol{xx}}^{-1} + \sum_{i=1}^{N}\boldsymbol{H}_i^{\mathrm{T}}\boldsymbol{C}_{\boldsymbol{w}_i}^{-1}\boldsymbol{H}_i)^{-1}\sum_{i=1}^{N}\boldsymbol{H}_i^{\mathrm{T}}\boldsymbol{C}_{\boldsymbol{w}_i}^{-1}\boldsymbol{z}_i$.

Proposition 2. The analytical LMMSE estimator has the equivalent optimization form as $\hat{\boldsymbol{x}}_{\text{LMMSE}} = \arg\min_{\boldsymbol{x}}\sum_{i=1}^{N}\|\boldsymbol{A}_i^{1/2}\boldsymbol{x} - \boldsymbol{A}_i^{-1/2}\boldsymbol{b}_i\|_2^2$, where $\boldsymbol{A}_i = N^{-1}\boldsymbol{C}_{\boldsymbol{xx}}^{-1} + \boldsymbol{H}_i^{\mathrm{T}}\boldsymbol{C}_{\boldsymbol{w}_i}^{-1}\boldsymbol{H}_i$ and $\boldsymbol{b}_i = \boldsymbol{H}_i^{\mathrm{T}}\boldsymbol{C}_{\boldsymbol{w}_i}^{-1}\boldsymbol{z}_i$.

Proof. Note $\boldsymbol{C}_{\boldsymbol{xx}}$ and $\boldsymbol{C}_{\boldsymbol{w}_i}$ are positive definite matrices, so $\boldsymbol{C}_{\boldsymbol{xx}}^{-1}$ and $\boldsymbol{C}_{\boldsymbol{w}_i}^{-1}$ exist and are also positive definite. Let $\boldsymbol{A}_i = N^{-1}\boldsymbol{C}_{\boldsymbol{xx}}^{-1} + \boldsymbol{H}_i^{\mathrm{T}}\boldsymbol{C}_{\boldsymbol{w}_i}^{-1}\boldsymbol{H}_i$ and $\boldsymbol{b}_i = \boldsymbol{H}_i^{\mathrm{T}}\boldsymbol{C}_{\boldsymbol{w}_i}^{-1}\boldsymbol{z}_i$. The proof is completed by Lemma 3, where we use the fact that $\boldsymbol{A}_i = N^{-1}\boldsymbol{C}_{\boldsymbol{xx}}^{-1} + \boldsymbol{H}_i^{\mathrm{T}}\boldsymbol{C}_{\boldsymbol{w}_i}^{-1}\boldsymbol{H}_i$ is positive definite.

Then Algorithm 1 can be applied here, where $f_i(\boldsymbol{x}) = \|\boldsymbol{A}_i^{1/2}\boldsymbol{x} - \boldsymbol{A}_i^{-1/2}\boldsymbol{b}_i\|_2^2$, and its subgradient is equal to the gradient $\boldsymbol{g}_i(\boldsymbol{x}) = 2[(N^{-1}\boldsymbol{C}_{\boldsymbol{xx}}^{-1} + \boldsymbol{H}_i^{\mathrm{T}}\boldsymbol{C}_{\boldsymbol{w}_i}^{-1}\boldsymbol{H}_i)\boldsymbol{x} - \boldsymbol{H}_i^{\mathrm{T}}\boldsymbol{C}_{\boldsymbol{w}_i}^{-1}\boldsymbol{z}_i]$.

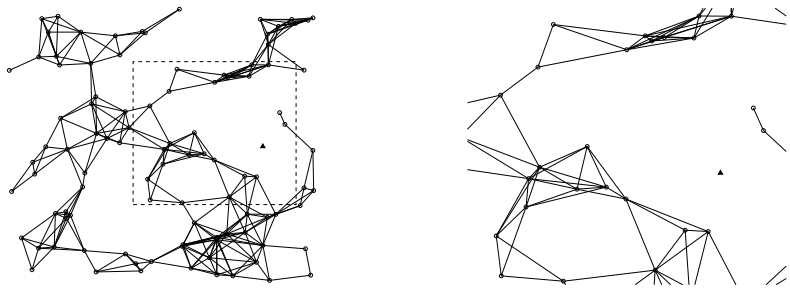
6 Simulation

In this section, we will show the simulation results for the IS methods and compare them with other distributed estimation methods. As illustrated in the introduction, we choose the Markovian relay mode for all the simulations.

We choose a network topology as shown in Figure 1, where the circles represent the sensors, the triangle represents the signal source, and the edges are the links among sensors. Although the topology is arbitrarily chosen, all the algorithms discussed above can be applied to any other topology. We use this sensor network to estimate the amplitude of the signal. The gains in the communication channels are inversely proportional to the square of the distances between the transmitter and sensors. We assume that all the links are error-free, and the observation noises are zero-mean and independent and identically distributed (i.i.d.). To compare our algorithm with others, we define the normalized error as $e_{\text{norm}} = \|\hat{\boldsymbol{x}}_{\text{dis}} - \boldsymbol{x}_{\text{true}}\|_2/\|\boldsymbol{x}_{\text{true}}\|_2$, where $\hat{\boldsymbol{x}}_{\text{dis}}$ is the distributed estimate, and $\boldsymbol{x}_{\text{true}}$ is the actual signal.

In a typical sensor network, the energy consumption mainly comes from computation and communication. We define the computational cost as the total number of calculation in the network, where we take addition and multiplication as one calculation for simplicity. Also we define the communicational cost as the total number of the communication data from sensors to their neighbors, where we take each value transmission as one communication for simplicity. For example, if a node sends three parameters to its four neighbors and another node does the same, the total communicational cost is $2 \times 3 \times 4 = 24$. The computation and communication costs in one iteration of different algorithms used in simulation are listed in Table 1.

In previous sections we have shown that the IS methods can be applied to many distributed estimators such as MLE, BLUE, MAP, and LMMSE. We will discuss these cases one by one. To all the IS methods



(a) (b)

Figure 1 The network topology used in simulations. (a) The network topology, $N = 100$, average degree is 6, and the size is 100×100; (b) the partial enlargement of the dotted rectangle in (a).

Table 1 The computational and communicational costs in one iteration. The computation cost of algorithms marked with "$*$" may be imprecise. Because they do not have analytical solutions, we use the matlab function fminbnd to find the optimal points iteratively. We suppose the computation cost in one iteration of fminbnd is 5, which may be the best situation

	Computation cost	Communication cost
DIS-MLE	5	2
D-MLE*	8000	840
DIS-BLUE	4	2
D-BLUE	4100	840
CA-BLUE	3600	1200
DIS-MAP	6	2
D-MAP*	8000	840
DIS-LMMSE	4	2
D-LMMSE	4100	840

in the simulation, we choose the diminishing stepsize rule $\alpha_k = C/(k+1)$, where k is the iteration index, and C is a constant.

6.1 Distributed MLE

Here we test the convergence rate of DIS-MLE in non-Gaussian noise case, and compare it with the D-MLE in [5]. In the simulation model, $z_i = H_i x + w_i$, where the actual signal amplitude $x = 1$. The channel gain $H_i = 10^4/\mathrm{dist}(i)^2$, where the $\mathrm{dist}(i)$ represents the distance between sensor i and the transmitter in Fig.1. The observation noise w_i is chosen to follow Laplace distribution, whose probability density function is $p(x) = \frac{1}{2b}\exp(-\frac{|x-\mu|}{b})$, where μ is a location parameter and $b > 0$ is a scale parameter. Here we set $\mu = 0$ and $b = \sqrt{5}/2$.

In order to eliminate the effects of the randomization in the IS methods, we achieve the final normalized error by running the IS 100 times and averaging 100 sets of normalized error in one observation. All the definitions and assumptions are same in the following simulation cases, except for explicit specification.

In each iteration of DIS-MLE, one addition and two multiplications are needed to calculate the sub-gradient, and one addition and two multiplications are needed to update the local estimation value. It only needs to transmit two values, the local estimation value and iteration index, in one link of the whole network. Figures 2 and 3 display the relationships between the normalized error and the computational or communicational cost for DIS-MLE and D-MLE.

These results indicate that the convergence rate of DIS-MLE versus computational or communicational cost is faster than that of D-MLE at first. This is because in every iteration only one sensor processes the data and transmits local results to one neighbor in the IS methods, while in D-MLE, every sensor should process data and transmit to all neighbors during an iteration. Both methods own similar estimation accuracy if increasing the simulation steps further.

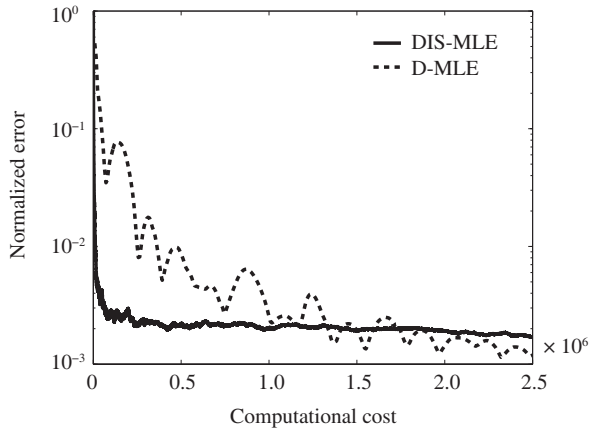


Figure 2 Normalized error vs. computational cost for DIS-MLE and D-MLE.

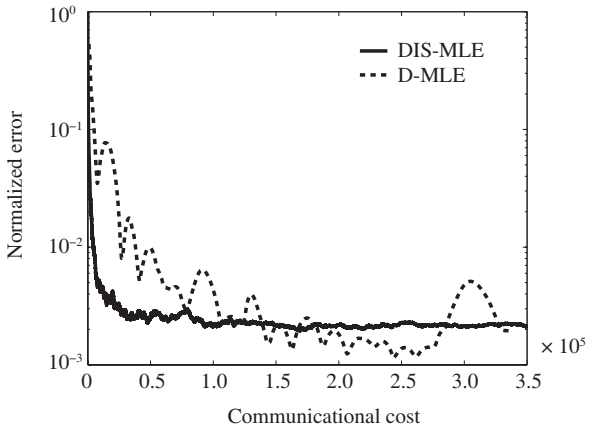


Figure 3 Normalized error vs. communicational cost for DIS-MLE and D-MLE.

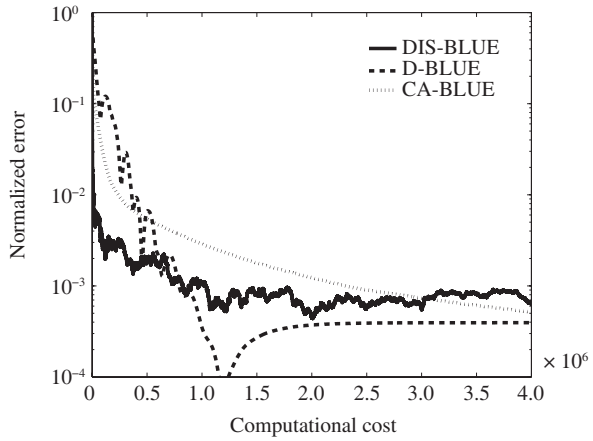


Figure 4 Normalized error vs. computational cost for DIS-BLUE, D-BLUE and CA-BLUE.

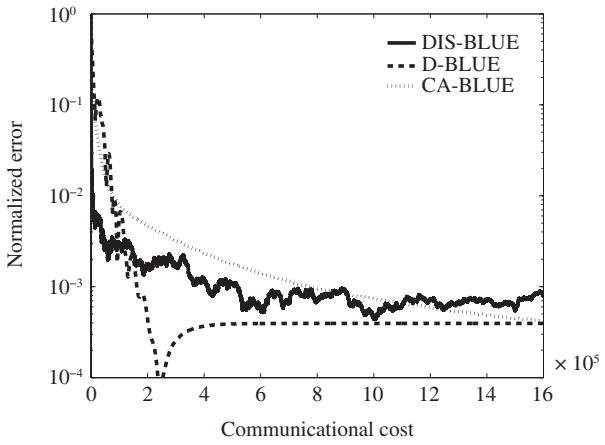


Figure 5 Normalized error vs. communicational cost for DIS-BLUE, D-BLUE and CA-BLUE.

6.2 Distributed BLUE

Here we test the convergence of DIS-BLUE, and compare it with D-BLUE [5] and CA-BLUE in [3]. In the model, the actual signal amplitude $x = 1$, and the observation noise $w_i \sim N(0, 2.5)$. In each iteration of DIS-BLUE, only one addition and one multiplication are needed to calculate the subgradient, and one addition and one multiplication are needed to update the local estimation value. As it is in DIS-MLE, we only need to transmit two values in the whole network.

Figures 4 and 5 display the relationships between the normalized error and the computational cost or communicational cost for DIS-BLUE, D-BLUE and CA-BLUE.

First, the convergence rate of DIS-BLUE is much higher than other algorithms. And the normalized errors will be less than 0.2% at small calculation and communicational cost. D-BLUE and CA-BLUE have advantages with cost up to 10^6 computation and 10^5 communication respectively. However, it is not feasible for sensors with limited computation and communication resource. We think that DIS-BLUE algorithm is a better choice among other distributed BLUE algorithms for practical application.

6.3 Distributed MAP

In the model for MAP, we assume that the actual signal amplitude x is a random variable with Gaussian distribution $\mathcal{N}(1, 0.01)$, and observation noise w_i obeys Laplace distribution with $\mu = 0$ and $b = \sqrt{5}/2$. In every iteration of DIS-MAP, two additions and one multiplication are needed to calculate the subgradient,

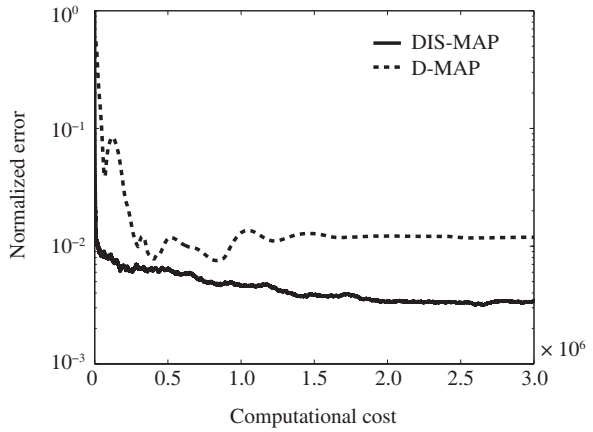


Figure 6 Normalized error vs. computational cost for DIS-MAP and D-MAP.

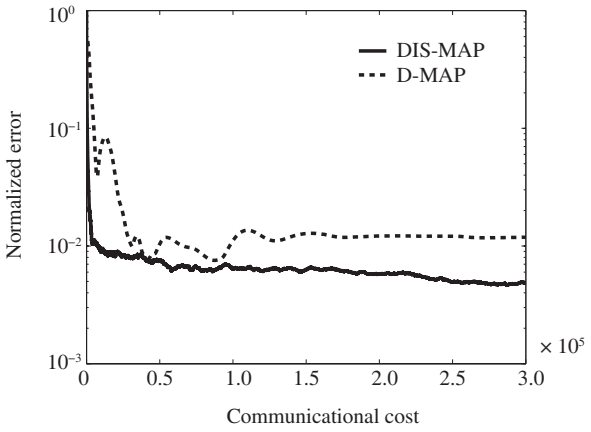


Figure 7 Normalized error vs. communicational cost for DIS-MAP and D-MAP.

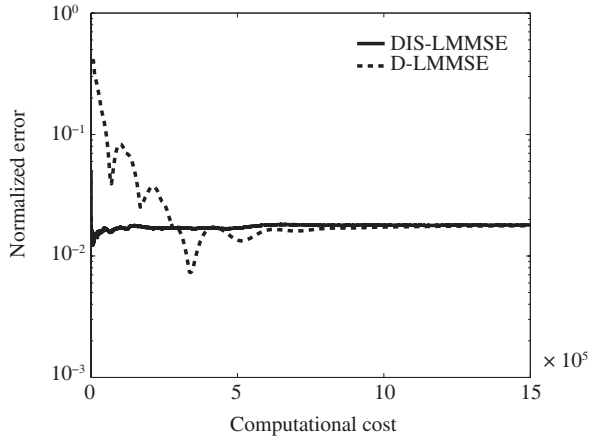


Figure 8 Normalized error vs. computational cost for DIS-LMMSE and D-LMMSE.

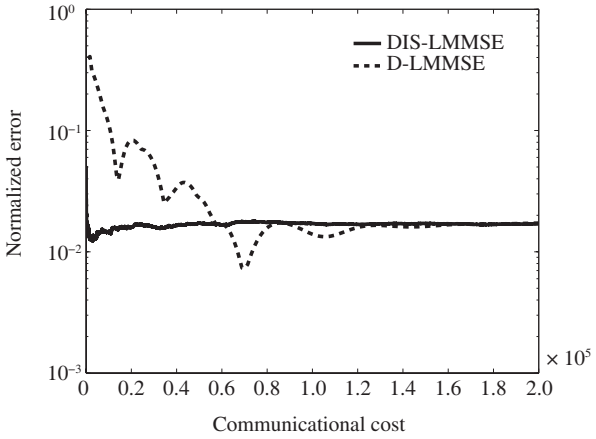


Figure 9 Normalized error vs. communicational cost for DIS-LMMSE and D-LMMSE.

and one addition and one multiplication are needed to update the local estimation value. As in previous algorithm, the whole network only needs to transmit the local estimation value and iteration index.

Figures 6 and 7 display the relationships between the normalized error and the computational or communicational cost for DIS-MAP and D-MAP [6]. It can be seen that the performance of DIS-MAP has the advantage at any computational and communicational costs.

6.4 Distributed LMMSE

Here we test the convergence of DIS-LMMSE, and compare it with D-LMMSE in [6]. In the model, the actual signal amplitude x is a random variable with Gaussian distribution $\mathcal{N}(1, 0.01)$, and observation noise w_i also obeys Gaussian distribution $\mathcal{N}(0, 2.5)$. In one iteration of DIS-LMMSE, the computation and communication cost are exactly the same as that of DIS-BLUE.

Figures 8 and 9 display the relationships between the normalized errors and the computational or communicational cost for DIS-LMMSE and D-LMMSE. The convergence performance of this case is similar to that of DIS-MLE in MLE case.

7 Conclusion

In this paper, we proposed a unified framework on distributed estimation in-network using the IS methods, as a lot of distributed estimation problems can be transformed into separable optimization problems. We

believe that the IS methods can also be used to build a distributed RLS, LMS and other popular estimation methods. The IS methods do not require the objection function to be differentiable, thus broadening the application areas in many practical scenarios, such as sensor network and collaborative robotics. The IS methods are also suitable for non-Gaussian noise models, e.g. the MLE and MAP cases in this paper. Although we only consider the undirected topology here, our method can be easily extended to directed topologies.

Although verified in many estimation contexts from the simulations, the IS methods converge not fast enough. The next step of our work is to improve the convergence rate by introducing recent progresses in optimization theory.

Acknowledgements

This work was supported in part by NSTMP of China (Grant No. 2013ZX03003006-003).

References

1 Giridhar A, Kumar P. Toward a theory of in-network computation in wireless sensor networks. IEEE Commun Mag, 2006, 44: 98–97
2 Olfati-Saber R, Fax J, Murray R. Consensus and cooperation in networked multi-agent systems. Proc IEEE, 2007, 95: 215–233
3 Casbeer D. Decentralized estimation using information consensus filters with a multi-static UAV radar tracking system. Dissertation for Doctoral Degree. Utah: Brigham Young University, 2009
4 Bertsekas D P, Tsitsiklis J N. Parallel and distributed computation: numerical methods. New Jersey: Printice-Hall, 1989
5 Schizas I, Ribeiro A, Giannakis G. Consensus in ad hoc WSNs with noisy links—part I: distributed estimation of deterministic signals. IEEE Trans Signal Process, 2008, 56: 350–364
6 Schizas I, Giannakis G, Roumeliotis S, et al. Consensus in ad hoc WSNs with noisy linkspart II: distributed estimation and smoothing of random signals. IEEE Trans Signal Process, 2008, 56: 1650–1666
7 Mateos G, Schizas I, Giannakis G. Distributed recursive least-squares for consensus-based in-network adaptive estimation. IEEE Trans Signal Process, 2009, 57: 4583–4588
8 Schizas I, Mateos G, Giannakis G. Distributed LMS for consensus-based in-network adaptive processing. IEEE Trans Signal Process, 2009, 57: 2365–2382
9 Wu J, Li H. A dominating-set-based routing scheme in ad hoc wireless networks. Telecommun Syst, 2001, 18: 13–36
10 Nedic A, Bertsekas D P. Incremental subgradient methods for nondifferentiable optimization. SIAM J Optim, 2001, 12: 109–138
11 Neto E S H, Pierro A R D. Incremental subgradients for constrained convex optimization: a unified framework and new methods. SIAM J Optim, 2009, 20: 1547–1572
12 Bertsekas D P. Incremental Gradient, Subgradient, and Proximal Methods for Convex Optimization: a Survey. Laboratory for Information and Decision Systems Report LIDS-P-2848, MIT, 2010
13 Shalev-Shwartz S, Singer Y, Srebro N, et al. Pegasos: primal estimated sub-gradient solver for SVM. Math Program, 2011, 127: 3–30
14 Predd J, Kulkarni S, Poor H. Distributed learning in wireless sensor networks. IEEE Signal Process Mag, 2006, 23: 56–69
15 Yousefian F, Nedić A, Shanbhag U V. On stochastic gradient and subgradient methods with adaptive steplength sequences. Automatica, 2012, 48: 56–67
16 Ram S S, Nedic A, Veeravalli V V. Incremental stochastic subgradient algorithms for convex optimization. SIAM J Optim, 2009, 20: 691–717
17 Johansson B, Rabi M, Johansson M. A randomized incremental subgradient method for distributed optimization in networked systems. SIAM J Optim, 2009, 20: 1157–1170
18 Kay S M. Fundamentals of Statistical Signal Processing, Volume I: Estimation Theory. Upper Saddle River: Prentice Hall, 1993