Abstract—The deluge of networked data motivates the development of algorithms for computation- and communication-efficient information processing. In this context, three data-adaptive censoring strategies are introduced to considerably reduce the computation and communication overhead of decentralized recursive least-squares (D-RLS) solvers. The first relies on alternating minimization and the stochastic Newton iteration to minimize a network-wide cost, which discards observations with small innovations. In the resultant algorithm, each node performs local data-adaptive censoring to reduce computations, while exchanging its local estimate with neighbors so as to consent on a network-wide solution. The communication cost is further reduced by the second strategy, which prevents a node from transmitting its local estimate to neighbors when the innovation it induces to incoming data is minimal. In the third strategy, not only transmitting, but also receiving estimates from neighbors is prohibited when data-adaptive censoring is in effect. For all strategies, a simple criterion is provided for selecting the threshold of innovation to reach a prescribed average data reduction. The novel censoring-based (C)D-RLS algorithms are proved convergent to the optimal argument in the mean-square deviation sense. Numerical experiments validate the effectiveness of the proposed algorithms in reducing computation and communication overhead.

Index Terms—Decentralized estimation, networks, recursive least-squares (RLS), data-adaptive censoring

I. INTRODUCTION

In our big data era, various networks generate massive amounts of streaming data. Examples include wireless sensor networks, where a large number of inexpensive sensors cooperate to monitor, e.g., the environment [17], [18], or data centers, where a group of servers collaboratively handles dynamic user requests [20]. Since a single node has limited computational resources, decentralized information processing is preferable as the network size scales up [5], [6]. In this paper, we focus on a decentralized linear regression setup, and develop computation- and communication-efficient decentralized recursive least-squares (D-RLS) algorithms.

The main tool we adopt to reduce computation and communication costs is data-adaptive censoring, which leverages the redundancy present especially in big data. Upon receiving an observation, nodes determine whether it is informative or not. Less informative observations are discarded, while messages among neighboring nodes are exchanged only when necessary. We propose three censoring-based (C)D-RLS algorithms that can achieve estimation accuracy comparable to D-RLS without censoring, while significantly reducing the computation and communication overhead.

A. Related works

The merits of RLS algorithms in solving centralized linear regression problems are well recognized [9], [21]. When streaming observations that depend linearly on a set of unknown parameters become available, RLS yields the least-squares parameter estimates online. RLS reduces the computational burden of finding a batch estimate per iteration, and can even allow for tracking time-varying parameters. The computational cost can be further reduced by data-adaptive censoring [4], where less informative data are discarded. On the other hand, decentralized versions of RLS without censoring have been advocated to solve linear regression tasks over networks [13]. In D-RLS, a node updates its estimate that is common to the entire network by fusing its local observations with the local estimates of its neighbors. As time evolves, all local estimates consent on the centralized RLS solution. This paper builds on both [4] and [13] by developing censoring-based decentralized RLS algorithms, thus catering to efficient online linear regression over large-scale networks.

Different from our in-network setting where operation is fully decentralized and nodes are only able to communicate with their neighbors, most of the existing distributed censoring algorithms apply to star topology networks that rely on a fusion center [2], [7], [8], [15], [19]. Their basic idea is that each node transmits data to the fusion center for further processing only when its local likelihood ratio exceeds a threshold [19]; see also [7] where communication constraints are also taken into account. Information fusion over fading channels is considered in [8]. Practical issues such as joint dependence of sensor decision rules, randomization of decision strategies and partially known distributions are reported.
in [2], while [15] also explores quantization jointly with censoring.

Other than the star topology studied in the aforementioned works, [16] investigates censoring for a tree structure. If a node’s local likelihood ratio exceeds a threshold, its local data is sent to its parent node for fusion. A fully decentralized setting is considered in [3], where each node determines whether to transmit its local estimate to its neighbors by comparing the local estimate with the weighted average of its neighbors. Nevertheless, [3] aims at mitigating only the communication cost, while the present work also considers reduction of the computational cost across the network. Furthermore, the censoring-based decentralized linear regression algorithm in [11] deals with optimal full-complexity estimation when observations are partially known or corrupted. This is different from our context, where censoring is deliberately introduced to reduce computation and communication costs for decentralized linear regression.

B. Our contributions and organization

The present paper introduces three data-adaptive online censoring strategies for decentralized linear regression. The resultant CD-RLS algorithms incur low computation and communication costs, and are thus attractive for large-scale network applications requiring decentralized solvers of linear regressions. Unlike most related works that specifically target wireless sensor networks (WSNs), the proposed algorithms may be used in a broader context of decentralized linear regression using multiple computing platforms. Of particular interest are cases where a regression dataset is not available at a single machine, but it is distributed over a network of computing agents that are interested in accurately estimating the regression coefficients in an efficient manner.

In Section II, we formulate the decentralized online linear regression problem (Section II-A), and recast the D-RLS in [13] into a new form (Section II-B) that prompts the development of three censoring strategies (Section II-C). Section III develops the first censoring strategy (Section III-A), analyzes all three censoring strategies (Section III-B), and discusses how to set the censoring thresholds (Section III-C). Numerical experiments in Section IV demonstrate the effectiveness of the novel CD-RLS algorithms.

Notation. Lower (upper) case boldface letters denote column vectors (matrices). $(\cdot)^T$, $|| \cdot ||_2$ and $E[\cdot]$ stand for transpose, 2-norm, induced matrix 2-norm and expectation, respectively. Symbols $\text{tr}(X)$, $\lambda_{\min}(X)$ and $\lambda_{\max}(X)$ are used for the trace, minimum eigenvalue and maximal eigenvalue of matrix $X$, respectively. Kronecker product is denoted by $\otimes$ and the uniform distribution over $[a, b]$ by $U(a, b)$, and the Gaussian probability distribution function (pdf) with mean $\mu$ and variance $\sigma^2$ by $\mathcal{N}(\mu, \sigma^2)$. The standardized Gaussian pdf is $\phi(t) = \frac{1}{\sqrt{2\pi}} \exp(-t^2/2)$, and its associated complementary cumulative distribution function is represented by $Q(z) = \int_z^{+\infty} \phi(t)dt$.

II. CONTEXT AND ALGORITHMS

This section outlines the online linear regression setup over networks, and takes a fresh look at the D-RLS algorithm. Three strategies are then developed using data-adaptive censoring to reduce the computation and communication costs of D-RLS.

A. Problem statement

Consider a bidirectionally connected network with $J$ nodes, described by a graph $\mathcal{G} := \{V, E\}$, where $V$ is the set of nodes with cardinality $|V| = J$, and $E$ denotes the set of edges. Each node $j$ only communicates with its one-hop neighbors, collected in the set $N_j \subset V$. The decentralized network is deployed to estimate a real vector $s_0 \in \mathbb{R}^p$. Per time slot $t = 1, 2, \ldots$, node $j$ receives a real scalar observation $x_j(t)$ involving the wanted $s_0$ with a regression row $h_j^T(t)$, where $(\cdot)^T$ stands for transposition, so that $x_j(t) = h_j^T(t)s_0 + \epsilon_j(t)$, with $\epsilon_j(t) \sim \mathcal{N}(0, \sigma_j^2)$.

Our goal is to devise efficient decentralized online algorithms to solve the following exponentially-weighted least-squares (EWLS) problem

$$\hat{s}_{\text{ewls}}(t) := \underset{s}{\text{arg min}} \frac{1}{2} \sum_{r=1}^{t} \sum_{j=1}^{J} \lambda^{t-r} ||x_j(r) - h_j^T(t)s||^2$$

where $\hat{s}_{\text{ewls}}(t)$ is the EWLS estimate at slot $t$, and $\lambda \in (0, 1]$ is a forgetting factor that de-emphasizes the importance of past measurements, and thus enables tracking of a non-stationary process. When $\lambda = 1$, (1) boils down to a standard decentralized online least-squares estimate.

B. D-RLS revisited

The D-RLS algorithm of [13] solves (1) as follows. Per time slot $t$, node $j$ receives $x_j(t)$ and $h_j^T(t)$ and uses them to update the per-node inverse $p \times p$ covariance matrix as

$$\Phi_j^{-1}(t) = \lambda^{1-t} \Phi_j^{-1}(t-1) - \frac{\lambda^{-1} \Phi_j^{-1}(t-1)h_j(t)h_j^T(t) \Phi_j^{-1}(t-1)}{\lambda + h_j^T(t) \Phi_j^{-1}(t-1) h_j(t)}$$

along with the per-node $p \times 1$ cross-covariance vector as

$$\psi_j(t) = \lambda \psi_j(t-1) + h_j(t)x_j(t).$$

Using $\Phi_j^{-1}(t)$ and $\psi_j(t)$, node $j$ then updates its local parameter estimate using

$$s_j(t) = \Phi_j^{-1}(t) \left[ \psi_j(t) - \frac{1}{2} \sum_{j' \in N_j} \left( v_{j'}^j(t-1) - v_{j'}^j(t-1) \right) \right]$$

where $v_{j'}^j(t-1)$ denotes the Lagrange multiplier of node $j$ corresponding to its neighbor $j'$ at slot $t-1$, that captures the
accumulated differences of neighboring estimates, recursively obtained as $(\rho > 0$ is a step-size)

$$v_j(t-1) = v_j(t-2) + \rho[s_j(t-1) - s_j(t-1)]. \tag{5}$$

Next, we develop an equivalent novel form of D-RLS recursions (2)–(5) that is convenient for our incorporation of data-adaptive censoring. Detailed derivation of the equivalence can be found in Appendix A. The inverse covariance matrix is updated as in (2). However, the update of $s_j(t)$ in (4) is replaced by

$$s_j(t) = s_j(t-1) + \Phi_j^{-1}(t)h_j(t)[x_j(t) - h_j^T(t)s_j(t-1)] - \rho\Phi_j^{-1}(t)\delta_j(t-1) \tag{6}$$

where $\delta_j(t)$ stands for a Lagrange multiplier conveying network-wide information that is updated as

$$\delta_j(t) = \delta_j(t-1) + \sum_{j' \in N_j} [s_j(t) - s_j(t)] - \lambda \sum_{j' \in N_j} [s_j(t-1) - s_j(t-1)]. \tag{7}$$

Observe that $\delta_j(t)$ stores the weighted sum of differences between the local estimate of node $j$, and all estimates of its neighbors. Interestingly, if the network is disconnected and the nodes are isolated, then $\delta_j(t) = 0$, and the update of $s_j(t)$ in (6) basically boils down to the centralized RLS one [9], [21]. That is, the current estimate is modified from its previous value using the prediction error $x_j(t) - h_j^T(t)s_j(t-1)$, which is known as the incoming data innovation. If on the other hand the network is connected, nodes can leverage estimates of their neighbors (captured by $\delta_j(t)$), which provide new information from the network other than its own observations $\{x_j(t)\}$. The term $\rho\Phi_j^{-1}(t)\delta_j(t-1)$ can be viewed as a Laplacian smoothing regularizer, which encourages all nodes of the graph to reach consensus on their estimates.

Remark 1. In D-RLS, (2) incurs computational complexity $O(p^2)$, since calculating the products $\Phi_j^{-1}(t-1)h_j(t)$ and $\Phi_j^{-1}(t-1)\psi_j(t)$ requires $O(p^2)$ multiplications. Similarly, (6) incurs computational complexity $O(p^2)$, that is dominated by the matrix-vector multiplications $\Phi_j^{-1}(t-1)h_j(t)$ and $\Phi_j^{-1}(t-1)\delta_j(t-1)$. The cost of carrying out (7) is relatively minor. Regarding communication per slot $t$, node $j$ needs to transmit its local estimate $s_j(t)$ to its neighbors and receive estimates $s_{j'}(t)$ from all neighbors $j' \in N_j$. The computational burden of D-RLS recursions (2)–(5) is comparable to that of (2), (6) and (7), with the cost of (4) being the same as what (6) requires. Meanwhile, the original form requires neighboring nodes $j$ and $j'$ to exchange $v_j(t)$ and $v_{j'}(t)$ in addition to $s_j(t)$ and $s_j(t)$, which doubles the communication cost relative to (6) and (7).

C. Censoring-based D-RLS strategies

The D-RLS algorithm has well documented merits for decentralized online linear regression [13]. However, its computation and communication costs per iteration are fixed, regardless of whether observations and/or the estimates from neighboring nodes are informative or not. This fact motivates our idea of permeating benefits of data-adaptive censoring to decentralized RLS, through three novel censoring-based (CD)-RLS strategies. They are different from the RLS algorithms in [4], where the focus is on centralized online linear regression.

Our first censoring strategy (CD-RLS-1) can be intuitively motivated as follows. If a given datum $(x_j(t), h_j(t))$ is not informative enough, we do not have to use it since its contribution to the local estimate of node $j$, as well as to those of all network nodes, is limited. With \{ $\tau\sigma_j(t)$ \} specifying proper thresholds to be discussed later, this intuition can be realized using a censoring indicator variable

$$c_j(t) := \begin{cases} 0, & \text{if } |x_j(t) - h_j^T(t)s_j(t-1)| < \tau\sigma_j(t) \\ 1, & \text{if } |x_j(t) - h_j^T(t)s_j(t-1)| > \tau\sigma_j(t). \end{cases} \tag{8}$$

If the absolute value of the innovation is less than \(\tau\sigma_j(t)\), then \((x_j(t), h_j(t))\) is censored; otherwise \((x_j(t), h_j(t))\) is used. Section III-C will provide rules for selecting the threshold $\tau$ along with the local noise variance $\sigma_j^2(t)$, whose computations are lightweight. If data censoring is in effect, we simply throw away the current datum by letting $h_j(t) = 0$ in (2), to obtain

$$\Phi_j^{-1}(t) = \lambda^{-1}\Phi_j^{-1}(t-1). \tag{9}$$

Likewise, letting $x_j(t) = 0$ and $h_j(t) = 0$ in (6), yields

$$s_j(t) = s_j(t-1) - \rho\Phi_j^{-1}(t)\delta_j(t-1). \tag{10}$$

CD-RLS-1 is summarized in Algorithm 1. If censoring is in effect, computation cost per node and per slot is a fraction 5/7 of the D-RLS in (4) and (7) without censoring. To recognize why, observe that the scalar-matrix multiplication $\lambda^{-1}\Phi_j^{-1}(t-1)$ in (9) is not necessary as the update of $\Phi_j^{-1}(t)$ can be merged to wherever it is needed, e.g., in (10) and the next slot. In addition, carrying out the $O(p^2)$ multiplications to obtain $\Phi_j^{-1}(t)h_j(t)$ is no longer necessary, while the $O(p^2)$ multiplications required to obtain $\Phi_j^{-1}(t)\delta_j(t-1)$ remain the same.

The first censoring strategy still requires nodes to communicate with neighbors per time slot; hence, the communication cost remains the same. Reducing this communication cost, motivates our second censoring strategy (CD-RLS-2), where each node does not perform extra computations relative to CD-RLS1, but only receives neighboring estimates if its current datum is censored. The intuition behind this strategy is that if a datum is censored, then very likely the current local estimate is sufficiently accurate, and the node does not need to account for estimates from its neighbors.
Algorithms 1 and 2 are summarized in Algorithm 2.

Algorithm 1 CD-RLS-1

1: Initialize $\delta_j(0)$, $\{s_j(0)\}_{j=1}^{J}$ and $\{\Phi_j^{-1}(0)\}_{j=1}^{J}$
2: for $t = 1, 2, \ldots$ do
3: All $j \in V$:
4: if $|x_j(t) - h_j^T(t)s_j(t-1)| \leq \tau \sigma_j(t)$ then
5: update $\Phi_j^{-1}$ using (9)
6: update $s_j(t)$ using (10)
7: else
8: update $\Phi_j^{-1}$ using (2)
9: update $s_j(t)$ using (6)
10: end if
11: transmit $s_j(t)$ to and receive $s_{j'}(t)$ from all $j' \in N_j$
12: compute $\delta_j(t)$ using (7)
13: end for

Algorithm 3 CD-RLS-3

1: Initialize $\delta_j(0)$, $\{s_j(0)\}_{j=1}^{J}$ and $\{\Phi_j^{-1}(0)\}_{j=1}^{J}$
2: for $t = 1, 2, \ldots$ do
3: All $j \in V$:
4: if $|x_j(t) - h_j^T(t)s_j(t-1)| \leq \tau \sigma_j(t)$ then
5: stay idle
6: else
7: set $s_{j'}(t-1)$ as recently received ones from all $j' \in N_j$
8: update $\Phi_j^{-1}$ using (2)
9: update $s_j(t)$ using (6)
10: transmit $s_j(t)$ to and receive $s_{j'}(t)$ from all $j' \in N_j$
11: compute $\delta_j(t)$ using (7)
12: end if
13: if do not receive from any $j' \in N_j$ for $d_{max}$ time then
14: receive $s_{j'}(t)$
15: end if
16: end for

Estimates from neighbors, are only stored for future usage. Likewise, $N_j$ neighbors do not need node $j$’s current estimate either, because they have already received a very similar estimate. CD-RLS-2 is summarized in Algorithm 2.

The third censoring strategy (CD-RLS-3) given by Algorithm 3 is more aggressive than the second one. If a node has its datum censored at a certain slot, then it neither transmits to nor receives from its neighbors, and in that sense it remains “isolated” from the rest of the network in this slot. Apparently, we should not allow any node to be forever isolated. To this end, we can force each node to receive the local estimate from any of its neighbors at least once every $d_{max}$ slots, which upper bounds the delay of information exchange to $d_{max}$. Interestingly, the ensuing section will prove convergence of all three strategies to the optimal argument in the mean-square deviation sense under mild conditions.

III. DEVELOPMENT AND PERFORMANCE ANALYSIS

This section starts with a criterion-based development of CD-RLS-1. Convergence analysis of all three censoring strategies will follow, before developing practical means of setting the censoring threshold $\tau \sigma_j(t)$.

A. Derivation of censoring-based D-RLS-1

Consider the following truncated quadratic cost that is similar to the one used in the censoring-based but centralized RLS [4]

$$f_{j,t}(s) := \begin{cases} 0, & |x_j(t) - h_j^T(t)s| \leq \tau \sigma_j(t) \\ \frac{1}{2}|x_j(t) - h_j^T(t)s|^2 - \frac{1}{2} \tau^2 \sigma_j(t)^2, & |x_j(t) - h_j^T(t)s| > \tau \sigma_j(t). \end{cases}$$

Using (11) to replace the quadratic loss $|x_j(t) - h_j^T(t)s|^2$ in (1), our CD-RLS-1 criterion is

$$\min_s \sum_{j=1}^{J} \sum_{r=1}^{J} \lambda^{t-r} f_{j,r}(s).$$

To solve (12) in a decentralized manner, we introduce a local estimate $s_j$ per node $j$, along with auxiliary vectors $\tilde{z}_j^j$ and $\tilde{z}_j^{j'}$ per edge $(j, j')$. By constraining all local estimates of neighbors to consent, we arrive at the following equivalent separable convex program per slot $t$

$$\min_{\{s_j\}_{j \in V}} \sum_{r=1}^{t} \sum_{j=1}^{J} \lambda^{t-r} f_{j,r}(s_j)$$

s.t. $s_j = \tilde{z}_j^j, s_{j'} = \tilde{z}_j^{j'}, j \in V, j' \in N_j.$

Next, we employ alternating minimization and the stochastic Newton iteration to derive our first censoring-based solver of (13). To this end, consider the Lagrangian of (13) that is given by

$$\mathcal{L}(s, z, v, u) = \sum_{j \in V} \sum_{r=1}^{t} \lambda^{t-r} f_{j,r}(s_j)$$

$$+ \sum_{j=1}^{J} \sum_{j' \in N_j} \left[(v_{j'}^T)^T(s_j - \tilde{z}_j^{j'}) + (u_{j'}^T)^T(s_{j'} - \tilde{z}_j^{j'}) \right]$$

where $s := \{s_j\}_{j \in V}$ and $z := \{\tilde{z}_j^j, \tilde{z}_j^{j'}\}_{j \in N_j}$ are primal variables, while $v := \{v_{j'} \in \mathbb{R}^{J\times J} : j' \in V\}$ and $u := \{u_{j'} \in \mathbb{R}^{J\times J} : j' \in V\}$ are dual variables.
\( \mathbb{R}^{p} J_{j} \in \mathcal{N} \) are dual variables. Consider also the augmented Lagrangian of (13), namely
\[
\mathcal{L}_{\rho}(s, z, v, u) = \mathcal{L}(s, z, u, v) + \frac{\rho}{2} \sum_{j=1}^{J} \sum_{j' \in \mathcal{N}} [||s_{j} - \bar{z}_{j}' ||^2 + ||s_{j'} - \bar{z}_{j}' ||^2]
\]
where \( \rho \) is a positive regularization scale. Note that the constraints on \( z \) are not dualized, but they are collected in the set \( \mathcal{C}_{z} := \{ z | \bar{z}_{j}' = \bar{z}_{j}, j \in \mathcal{N}, j \neq j' \} \).

To minimize (11) with the stochastic Newton iteration [1], eliminate \( v_{j}^{'} \) using (17) to obtain
\[
s_{j}(t) = \arg \min_{s_{j}} \sum_{r=1}^{t} \lambda^{t-r} f_{j,r}(s_{j}) + \rho \sum_{r=1}^{t-1} \sum_{j' \in \mathcal{N}} [s_{j}(r) - s_{j'}(r)]^{T} s_{j}.
\]
which after manipulating the double sum yields
\[
s_{j}(t) = \arg \min_{s_{j}} \sum_{r=1}^{t} \lambda^{t-r} f_{j,r}(s_{j}) + \rho \sum_{r=1}^{t-1} \sum_{j' \in \mathcal{N}} [s_{j}(r) - s_{j'}(r)]^{T} s_{j}.
\]

Moving on to [S1], observe that it can be split into \( J \) per-node subproblems
\[
s_{j}(t) = \arg \min_{s_{j}} \sum_{r=1}^{t} \lambda^{t-r} f_{j,r}(s_{j}) + \rho \sum_{r=1}^{t} \sum_{j' \in \mathcal{N}} [v_{j}'(t-r) - v_{j'}'(t-1)]^{T} s_{j}.
\]

Before solving (11) with the stochastic Newton iteration [1], eliminate \( v_{j}^{'} \) using (17) to obtain
\[
s_{j}(t) = \arg \min_{s_{j}} \sum_{r=1}^{t} \lambda^{t-r} f_{j,r}(s_{j}) + \rho \sum_{r=1}^{t-1} \sum_{j' \in \mathcal{N}} [s_{j}(r) - s_{j'}(r)]^{T} s_{j}.
\]
which after manipulating the double sum yields
\[
s_{j}(t) = \arg \min_{s_{j}} \sum_{r=1}^{t} \lambda^{t-r} f_{j,r}(s_{j}) + \rho \sum_{r=1}^{t} \sum_{j' \in \mathcal{N}} [s_{j}(r) - s_{j'}(r)]^{T} s_{j}.
\]

Moving on to [S1], observe that it can be split into \( J \) per-node subproblems
\[
s_{j}(t) = \arg \min_{s_{j}} \sum_{r=1}^{t} \lambda^{t-r} f_{j,r}(s_{j}) + \rho \sum_{r=1}^{t} \sum_{j' \in \mathcal{N}} [v_{j}'(t-r) - v_{j'}'(t-1)]^{T} s_{j}.
\]
Before solving (11) with the stochastic Newton iteration [1], eliminate \( v_{j}^{'} \) using (17) to obtain
\[
s_{j}(t) = \arg \min_{s_{j}} \sum_{r=1}^{t} \lambda^{t-r} f_{j,r}(s_{j}) + \rho \sum_{r=1}^{t} \sum_{j' \in \mathcal{N}} [s_{j}(r) - s_{j'}(r)]^{T} s_{j}.
\]
which after manipulating the double sum yields
\[
s_{j}(t) = \arg \min_{s_{j}} \sum_{r=1}^{t} \lambda^{t-r} f_{j,r}(s_{j}) + \rho \sum_{r=1}^{t} \sum_{j' \in \mathcal{N}} [s_{j}(r) - s_{j'}(r)]^{T} s_{j}.
\]
sample averaging. However, presence of $\lambda \neq 1$ affects attenuation of regressors, which leads to

$$M_j(t) = \frac{1}{t} \sum_{r=1}^{t} \lambda^{-r} c_j(r) h_j(r) h_j^T(r)$$

$$= \lambda \frac{t-1}{t} M_j(t-1) + \frac{1}{t} c_j(t) h_j(t) h_j^T(t).$$

Applying the matrix inversion lemma, we obtain

$$M_j^{-1}(t) = \frac{t}{t-1} \left[ \lambda^{-1} M_j^{-1}(t-1) - c_j(t) \frac{\lambda^{-1} M_j^{-1}(t-1) h_j(t) h_j^T(t) M_j^{-1}(t-1) h_j(t)}{(t-1) \lambda - h_j^T(t) M_j^{-1}(t-1) h_j(t)} \right]$$

and after adopting a diminishing step size $1/t$, the stochastic Newton update becomes

$$s_j(t) = s_j(t-1) - \frac{1}{t} M_j^{-1}(t) \nabla g_{j,t}(s_j(t-1)).$$

For rational convenience, let $\Phi_j^{-1}(t) := M_j^{-1}(t)/t$, and rewrite (21) as (cf. (2))

$$\Phi_j^{-1}(t) = \lambda^{-1} \Phi_j^{-1}(t-1) - c_j(t) \frac{\lambda^{-1} \Phi_j^{-1}(t-1) h_j(t) h_j^T(t) \Phi_j^{-1}(t-1)}{(t-1) \lambda + h_j^T(t) \Phi_j^{-1}(t-1) h_j(t)}.$$ 

Substituting $\nabla g_{j,t}(s_j(t-1))$ and $\Phi_j^{-1}(t)$ into the stochastic Newton iteration yields (cf. (6))

$$s_j(t) = s_j(t-1) + c_j(t) \Phi_j^{-1}(t) h_j(t) [x_j(t) - h_j^T(t) s_j(t-1)] - \rho \Phi_j^{-1}(t) \delta_j(t-1)$$

which completes the development of CD-RLS-1.

B. Convergence analysis

Here we establish convergence of all three novel strategies for $\lambda = 1$. With $\lambda < 1$, the EWLS estimator can even adapt to time-varying parameter vectors, but analyzing its tracking performance goes beyond the scope of this paper. For the time-invariant case ($\lambda = 1$), we will rely on the following assumption.

(a1) Observations obey the linear model $x_j(t) = h_j(t)s_0 + \epsilon_j(t)$, where $\epsilon_j(t) \sim N(0, \sigma^2_j)$ is independent across $j$ and $t$. Rows $h_j^T(t)$ are uniformly bounded and independent of $\epsilon_j(t)$. Covariance matrices $R_{h_j} := E[h_j(t) h_j^T(t)] > 0_{p \times p}$ are constant and positive definite. Process $\{c_j(t) h_j(t) h_j^T(t)\}$ is ergodic, while $\{\epsilon_j(t)\}$ and $\{c_j(t)\}$ are independent.

We are interested in the global mean-square deviation (GMSD) metric [12], [22], defined as

$$\text{GMSD}_j(t) := \frac{1}{J} \sum_{j=1}^{J} E[||s_j(t) - s_0||^2].$$

Under (a1), convergence of CD-RLS-1 and CD-RLS-2 is asserted as follows; see Appendix B for the proof.

**Theorem 1.** For CD-RLS-1 and CD-RLS-2 Algorithms 1 and 2, set $\sigma_j(t) = \sigma_j$ and $\Phi_j^{-1}(t) = \gamma_j I_p$ per node $j$. Let $\mu := \min\{\lambda_{\text{min}}(R_{h_j}), j \in V\}$, and suppose $0 < \rho < 1/(\gamma \lambda_{\text{max}}(L))$ for CD-RLS-1 and correspondingly $0 < \rho < \rho_0$ for CD-RLS-2, where $L$ is the network Laplacian and the constant $\rho_0$ depends on $\lambda_{\text{max}}(L)$, $\gamma$, $\tau$, $\mu$, and the upper bound of $h_j(t)$. Under (a1), there exists $t_0 > 0$ for which it holds for $t \geq t_0$ that

$$\sum_{j=1}^{J} E[||s_j(t) - s_0||^2] \leq \frac{\gamma \sigma^2_j \lambda_{\text{max}}(R_{h_j}^{-1}) tr(R_{h_j})}{4Q^2(\tau) \mu t} + \frac{\gamma \sigma^2_j \lambda_{\text{max}}(R_{h_j}^{-1}) tr(R_{h_j}) \ln(t)}{4Q^2(\tau) \mu t}.$$ 

**Theorem 2.** For CD-RLS-3 given by Algorithms 3, set $\sigma_j(t) = \sigma_j$ and $\Phi_j^{-1}(t) = \gamma_j I_p$ per node $j$. Under (a1) and (a2), with $0 < \rho < \rho_0$ as in Theorem 1, there exists $t_0 > 0$ for which it holds $\forall t \geq t_0$ that

$$\sum_{j=1}^{J} E[||s_j(t) - s_0||^2] \leq \frac{a + b \ln(t)}{t}$$

where $a$ and $b$ are positive constants that depend on the upper bounds of $h_j(t)$ and $s_j(t)$, parameters $\rho$ and $\tau$, the covariance $R_{h_j}(t)$, the Laplacian matrix $L$, and $t_0$. 

Theorem 1 establishes that the GMSD in (23) converges to zero at a rate $O(\ln(t)/t)$. The constant of the convergence rate is related to $R_{h_j}$ through $\lambda_{\text{max}}(R_{h_j}^{-1})$, $\text{tr}(R_{h_j})$ and $\mu$; the noise covariance $\sigma^2_j$, and the threshold $\tau$ through $Q(\tau)$. Theorem 1 also indicates the impact of the initial states (determined by $\gamma$ and $s_j(0)$), which disappears at a faster rate of $O(1/t)$. To guarantee convergence, the step size $\rho$ must be small enough.

The proof for CD-RLS-3 is more challenging. Because a node does not receive any information from its neighbors when censoring is in effect, it has to rely on outdated neighboring estimates when the incoming datum is not censored. This delay in percolating information may cause computational instability. For this reason, we will impose an additional constraint to guarantee that all local estimates do not grow unbounded. In practice, this can be realized by truncating local estimates when they exceed a certain threshold.

(a2) Local estimates $\{s_j(t)\}_{j=1}^{J}$ are uniformly bounded $\forall t \geq 0$.

Convergence of CD-RLS-3 is then asserted as follows. Similar to CD-RLS-1 and CD-RLS-2, the GMSD of CD-RLS-3 converges to zero with rate $O(\ln(t)/t)$, as stated in the following theorem. The proof is given in Appendix C.

**Theorem 2.** For CD-RLS-3 given by Algorithms 3, set $\sigma_j(t) = \sigma_j$ and $\Phi_j^{-1}(t) = \gamma_j I_p$ per node $j$. Under (a1) and (a2), with $0 < \rho < \rho_0$ as in Theorem 1, there exists $t_0 > 0$ for which it holds $\forall t \geq t_0$ that

$$\sum_{j=1}^{J} E[||s_j(t) - s_0||^2] \leq \frac{a + b \ln(t)}{t}$$

where $a$ and $b$ are positive constants that depend on the upper bounds of $h_j(t)$ and $s_j(t)$, parameters $\rho$ and $\tau$, the covariance $R_{h_j}(t)$, the Laplacian matrix $L$, and $t_0$. 


C. Threshold setting and variance estimation

The threshold $\tau$ influences considerably the performance of all CD-RLS algorithms. Its value trades off estimation accuracy for computation and communication overhead. We provide a simple criterion for setting $\tau$ using the average censoring ratio $\pi^*$, which is defined as the number of censored data over the total number of data [15]. The goal is to choose $\tau$ so that the actual censoring ratio approaches $\pi^*$ as $t$ goes to infinity – since we are dealing with streaming big data, such an asymptotic property is certainly relevant. When $t$ is large enough, $s$ is very close to $s_0$; thus, the innovation $x_j(t) - h_j^T(t)s_j(t-1) \approx x_j(t) - h_j^T(t)s_0 = \epsilon_j(t) \sim \mathcal{N}(0, \sigma_j^2)$. As a consequence, $\Pr(|\epsilon_j(t)| \leq \tau \sigma_j) \approx \Pr(|\epsilon_j(t)| \leq \tau s_j) = \Pr(|\epsilon_j(t)/\sigma_j| \leq \tau) = 1 - 2Q(\tau)$, where the last equality holds because $\epsilon_j(t)/\sigma_j \sim \mathcal{N}(0,1)$. Therefore, $\pi^* = \lim_{t \to \infty} \frac{1}{t} \sum_{\tau=0}^{t} E[|\epsilon_j(\tau)|] \approx 1 - 2Q(\tau)$, which implies that

$$\tau = Q^{-1}((1 - \pi^*)/2).$$

If the variances $\{\sigma_j^2\}$ were known, one could simply choose $\sigma_j(t) = \sigma_j$. However, $\sigma_j$ in practice is often unknown. In this case, we consider the running average $\sigma_j^2(t+1) \approx t^{-1} \sum_{\tau=1}^{t} |x_j(\tau) - h_j^T(\tau)s_0|^2 = (t-1)\sigma_j^2(t) + |x_j(t+1) - h_j^T(t+1)s_0|^2/t$, which suggests the recursive variance estimate

$$\sigma_j^2(t+1) = (t-1)\sigma_j^2(t) + |x_j(t+1) - h_j^T(t+1)s_0|^2/t.$$

IV. Numerical Experiments

This section provides numerical results to validate the effectiveness of our novel censoring strategies. We simulate a network of $J = 15$ nodes, which are uniformly randomly deployed over a $1 \times 1$ square. Two nodes within communication range 0.3 are deemed as being neighbors. The resultant network topology is depicted in Fig. 1. We compare five algorithms: the centralized adaptive censoring (AC)-RLS that runs in every node independently, D-RLS without censoring [14], and the three censoring-based D-RLS algorithms, namely CD-RLS-1, CD-RLS-2 and CD-RLS-3. All algorithms are evaluated on two data sets, one synthetic and one real. The empirical GMSD is used as a performance metric.

For the synthetic data set, the unknown $s_0$ is $p$-dimensional with $p = 4$. The setting is the one in [14], where WSN-based decentralized power spectrum estimation is sought for a signal modeled as an autoregressive process. In this context, consider an auxiliary sequence $r_j(t)$ that evolves according to $r_j(t) = (1-q)\beta_j r_j(t-1) + \sqrt{q} \sigma_j \omega_j(t)$. Starting from $r_j(t)$, the row $h_j^T(t)$ is formed by taking the next $p$ observations, namely $h_j^T(t) = [r_j(t+p-1); \ldots; r_j(t)]$. Parameters are selected as $q = 0.5$, $\beta_j \sim \mathcal{U}(0, 1)$, and also uniformly distributed driving white noise $\omega_j(t) \sim \mathcal{U}(-\sqrt{3\sigma_j}, \sqrt{3\sigma_j})$ with $\sigma_j \sim \mathcal{U}(0,2)$. Observation of node $j$ is subject to additive white Gaussian noise, with covariance $\sigma_j^2 = 10^{-3} \alpha_j$, where $\alpha_j \sim \mathcal{U}(0, 1)$. The true signal vector is $s_0 = 1_p$, for which $\lambda = 1$ is set for all algorithms. For D-RLS, CD-RLS-1, CD-RLS-2 and CD-RLS-3, the step size $\rho = 0.01$ and $\Phi_j^{-1}(0) = \gamma I_p$, where $\gamma = 30$, leading to fastest convergence of D-RLS. Regarding the four censoring-based algorithms AC-RLS, CD-RLS-1, CD-RLS-2 and CD-RLS-3, we set the average censoring ratio to $\pi^* = 0.6$, which is approached using $\tau = Q^{-1}((1 - \pi^*)/2) \approx 0.84$. The variances $\sigma_j^2$ are estimated in an online manner as described in Section III-C. AC-RLS uses $\Phi_j^{-1}(0) = \gamma I_p$, where $\gamma = 10^5$ leads to the fastest convergence. For all curves obtained by running the algorithms, the ensemble averages are approximated via sample averaging over 100 Monte Carlo runs.

Fig. 2 depicts the GMSD versus the number of iterations. Not surprisingly, since D-RLS does not censor data, its convergence rate with respect to the number of iterations is the fastest. Among the three proposed CD-RLS algorithms, CD-
RLS-2 and CD-RLS-3 are slower than CD-RLS-1, because the former two incur smaller communication cost than the latter. Though CD-RLS-3 adopts a more aggressive censoring strategy than CD-RLS-2, its convergence does not degrade as confirmed by Fig. 2. AC-RLS is the slowest among all, because it is run at all nodes independently, without sharing information over the network.

The merits of censoring are further appreciated when one considers computational costs. Recall that the target average censoring ratio is $\pi^* = 0.6$, meaning that 3/5 of the data are discarded (actual values are 0.6320 for AC-RLS, 0.6292 for CD-RLS-1, 0.6277 for CD-RLS-2, and 0.6237 for CD-RLS-3, averaged over 100 runs). As confirmed by Fig. 3, the three CD-RLS algorithms consume considerably less computational resources relative to D-RLS that does not censor data. Indeed, whenever a datum is censored, CD-RLS-1 only requires 2/7 of the computations relative to D-RLS, while CD-RLS-2 and CD-RLS-3 incur minimal computational overhead. Although AC-RLS is the most computationally efficient algorithm at the beginning, absence of collaboration undermines its performance in steady state.

Regarding the amount of data exchanged to communicate local estimates in a unicast mode, CD-RLS-1 is the worst because nodes need to transmit their local estimate to neighbors, no matter whether local data are censored or not. Fig. 4 corroborates that CD-RLS-2 and CD-RLS-3 show significant improvement over D-RLS, demonstrating their potential for reducing both communication and computation costs in solving decentralized linear regression problems over large-scale networks.

Next, we vary $\pi$ and evaluate its impact on GMSD, as shown in Fig. 5. When $\pi$ is close to 0.5, meaning about 1/2 of the data is censored, the three proposed CD-RLS algorithms are still able to reach GMSD of $10^{-4}$, which is the limit of D-RLS without censoring. Among the three algorithms, CD-RLS-1 exhibits the best GMSD curve, but its computation and communication costs are the highest. AC-RLS does not perform well especially for low censoring ratios due to the lack of network-wide collaboration. CD-RLS-2 and CD-RLS-3 perform comparably in this experiment.

The effectiveness of the novel censoring-based strategies is further assessed on a real data set of protein tertiary structures [10]. The premise here is that a given dataset is not available at a single location, but it is distributed over a network whose nodes are interested in obtaining accurate regression coefficients while suppressing the communication and computational overhead. Again, the graph in Fig. 1 is used to model the network of regression-performing agents. The number of control variables is $p = 9$. The first 45,720 (out of 45,730) observations are normalized and divided evenly into $J = 15$ parts, one per node. For CD-RLS-1, CD-RLS-2 and CD-RLS-3, we set $\rho = 0.05$ and $\Phi_j^{-1}(0) = 5I_p$, where $I_p$ is the identity matrix of size $p$. 
while for AC-RLS we choose $\gamma = 10$. The ground truth vector $s_0$ is estimated by solving a batch least-squares problem on the entire data set. Similar to what we deduced from Fig. 5 in the synthetic data set, the novel CD-RLS algorithms outperform AC-RLS in terms of GMSD, as one varies the average censoring ratio from 15% to nearly 100% in Fig. 6.

V. CONCLUDING REMARKS

This paper introduced three data-adaptive censoring strategies that significantly reduce the computation and communication costs of the RLS algorithm over large-scale networks. The basic idea behind these strategies is to avoid inefficient computation and communication when the local observations and/or the neighboring messages are not informative. We proved convergence of the resulting algorithms in the mean-square deviation sense. Numerical experiments validated the merits of the novel schemes.

The notion of identifying and discarding less informative observations can be widely used in various large-scale online machine learning tasks including nonlinear regression, matrix completion, clustering and classification, to name a few. These constitute our future research directions.

APPENDIX A

EQUIVALENT FORM OF D-RLS

Here we prove that D-RLS recursions (2) - (5) are equivalent to (2), (6) and (7). It follows from (4) that

$$\Phi_j(t)s_j(t) - \lambda \Phi_j(t - 1)s_j(t - 1) = \left[\psi_j(t) - \frac{1}{2} \sum_{j' \in N_j} (v_j'(t - 1) - v_j'(t - 1)) \right]$$

$$- \lambda \left[\psi_j(t - 1) - \frac{1}{2} \sum_{j' \in N_j} (v_j'(t - 2) - v_j'(t - 2)) \right].$$

Applying the matrix inversion lemma to (2) yields

$$\Phi_j(t) = \varphi_1 \Phi_j(t - 1) + \varphi_2(t)h_j(t)h_j^T(t).$$

Substituting $\psi_j(t - \lambda \varphi_1(t - 1) = h_j(t)x_j(t)$ from (3) and $\lambda \Phi_j(t - 1) = \Phi_j(t) - h_j(t)h_j^T(t)$ from (27) into (26), leads to

$$\Phi_j(t)[s_j(t) - s_j(t - 1)] = h_j(t)[x_j(t) - h_j^T(t)s_j(t - 1)]$$

$$- \frac{1}{2} \sum_{j' \in N_j} (v_j'(t - 1) - \varphi_1 v_j'(t - 2))$$

$$+ \frac{1}{2} \sum_{j' \in N_j} (v_j'(t - 1) - \varphi_1 v_j'(t - 2)).$$

Next, we will show that if $\varphi(t)$ is defined as

$$\varphi(t) := \frac{1}{2\rho} \sum_{j \in N_j} (v_j'(t) - \varphi v_j'(t - 1))$$

then its update is exactly (7). This can be done by taking the difference between slots $t$ and $t - 1$ for (29), and substituting the update of $v_j'$ in (5). Due to (29), it follows that (28) is equivalent to

$$\Phi_j(t)[s_j(t) - s_j(t - 1)] = h_j(t)[x_j(t) - h_j^T(t)s_j(t - 1)]$$

$$- \rho \varphi(t).$$

Left multiplying (30) with $\Phi_j^{-1}(t)$, yields the update of $s_j$ in (6), and completes the proof.

APPENDIX B

PROOF OF THEOREM 1

Proof: Starting with CD-RLS-1, the proof proceeds in five stages.

Stage 1. We first investigate the spectral properties of $\Phi_j(t)$ when $t$ is sufficiently large. Letting $\lambda = 1$ and applying the matrix inversion lemma to the censoring form (2), we have

$$\Phi_j(t) = \Phi_j(t - 1) + c_j(t)h_j(t)h_j^T(t).$$

Summing up from $r = 1$ to $r = t$ and using the telescopic cancellation, (31) yields

$$\Phi_j(t) = \sum_{r=1}^{t} c_j(r)h_j(r)h_j^T(r) + \gamma^{-1} I_p.$$
hand, given \( h_j(t), s_j(t-1) \) and \( c_j(t) \) defined in (8), the probability of \( c_j(t) = 1 \) is

\[
\Pr(c_j(t) = 1|h_j(t), s_j(t-1))
= 1 - \int_{-\tau}^{\tau} e^{-\gamma^{-1}h_j^2(t)} e(x)\,dx
\geq 1 - \int_{-\tau}^{\tau} e(x)\,dx = 2Q(\tau). \tag{33}
\]

Consequently, it holds that

\[
E[c_j(t)h_j(t)h_j^T(t)]
= E[h_j(t)h_j^T(t)E[c_j(t)|h_j(t), s_j(t-1)]]
= E[h_j(t)h_j^T(t)\Pr(c_j(t) = 1|h_j(t), s_j(t-1))]
\geq 2Q(\tau)E[h_j(t)h_j^T(t)] = 2Q(\tau)R_{h_j}.
\]

Thus, there exists \( t_0 > 0 \) for which it holds \( \forall t \geq t_0 \)

\[
2Q(\tau)trR_{h_j} \leq \Phi_j(t) \leq trR_{h_j}, \quad \forall j = 1, \ldots, J. \tag{34}
\]

As \( t \) grows, the eigenvalues of \( \Phi_j(t) \) are thus within the interval \([2Q(\tau)t\lambda_{\min}(R_{h_j}), t\lambda_{\max}(R_{h_j})] \). Consequently, the eigenvalues of \( \Phi^{-1}(t) \) are within the interval \([\lambda_{\min}(R_{h_j})^{-1}/t, \lambda_{\max}(R_{h_j})^{-1}/2Q(\tau)t] \).

**Stage 2.** Rewrite the update of \( s_j \) as

\[
s_j(t) = s_j(t-1) + c_j(t)\Phi_j^{-1}(t)h_j(t)[x_j(t) - h_j^T(t)s_j(t-1)]
- \rho\Phi_j^{-1}(t)\delta_j(t-1).
\]

Note also that for \( \lambda = 1 \), the update of \( \delta_j \) is equivalent to (cf. (18))

\[
\delta_j(t-1) = \sum_{j' \in N_j} [s_j(t-1) - s_{j'}(t-1)].
\]

Letting \( e_j(t) := s_j(t) - s_0 \), the estimation error obeys the recursion

\[
e_j(t) = e_j(t-1) + c_j(t)\Phi_j^{-1}(t)h_j(t)[x_j(t) - h_j^T(t)s_j(t-1)]
- \rho\Phi_j^{-1}(t)\sum_{j' \in N_j} [e_j(t-1) - e_{j'}(t-1)].
\]

Substituting \( x_j(t) = h_j(t)s_0 + e_j(t) \) to eliminate \( s_j(t-1) \), we obtain

\[
e_j(t) = e_j(t-1) - c_j(t)\Phi_j^{-1}(t)h_j(t)h_j^T(t)e_j(t-1)
+ c_j(t)\Phi_j^{-1}(t)h_j(t)e_j(t)
- \rho\Phi_j^{-1}(t)\sum_{j' \in N_j} [e_j(t-1) - e_{j'}(t-1)]. \tag{35}
\]

Left multiplying (35) with \( \Phi_j(t) \) yields

\[
\Phi_j(t)e_j(t)
= \Phi_j(t)e_j(t-1) - c_j(t)h_j(t)h_j^T(t)e_j(t-1)
+ c_j(t)h_j(t)e_j(t) - \rho\sum_{j' \in N_j} [e_j(t-1) - e_{j'}(t-1)]
= \Phi_j(t-1)e_j(t-1)
+ c_j(t)h_j(t)e_j(t) - \rho\sum_{j' \in N_j} [e_j(t-1) - e_{j'}(t-1)].
\]

Our convergence analysis result will rely on a matrix form of (36) that accounts for all nodes \( j \). Define vectors \( e(t) := [e_1(t); \ldots; e_J(t)]^T \in \mathbb{R}^{Jp} \), \( e(t) := [e_1^T(t); \ldots; e_J^T(t)]^T \in \mathbb{R}^{Jp} \), as well as diagonal matrices \( \Phi(t) := \text{diag}([\Phi_j(t)]) \in \mathbb{R}^{Jp \times Jp} \), \( C(t) := \text{diag}([c_j(t)]) \in \mathbb{R}^{J \times J} \), and \( H(t) := \text{diag}([h_j(t)]) \in \mathbb{R}^{Jp \times Jp} \). Then (36) can be written in matrix form as

\[
\Phi(t)e(t)
= \Phi(t-1) - \rho L \otimes I_p e(t-1) + H(t)C(t)e(t). \tag{37}
\]

which after left multiplication with \( \Phi^{-\frac{1}{2}}(t) \) yields

\[
\Phi(t)^{-\frac{1}{2}}e(t) = \Phi(t)^{-\frac{1}{2}}[\Phi(t-1) - \rho L \otimes I_p e(t-1)]
+ \Phi(t)^{-\frac{1}{2}}H(t)C(t)e(t). \tag{38}
\]

From (38), we have \((\otimes \text{ denotes Kronecker product})\)

\[
E[e^T(t)\Phi(t)e(t)]
= E[e^T(t-1)(\Phi(t-1) - \rho L \otimes I_p)^T\Phi^{-1}(t-1)
\times (\Phi(t-1) - \rho L \otimes I_p)e(t-1)]
+ 2E[e^T(t-1)(\Phi(t-1) - \rho L \otimes I_p)^T\Phi^{-1}(t-1)H(t)C(t)e(t)]
+ E[e^T(t)C^T(t)H^T(t)\Phi^{-1}(t)H(t)C(t)e(t)]. \tag{39}
\]

Since \( C(t) \) and \( e(t) \) are independent under (as1), the second term on the right hand side is zero; hence,

\[
E[e^T(t)\Phi(t)e(t)]
= E[e^T(t-1)(\Phi(t-1) - \rho L \otimes I_p)^T\Phi^{-1}(t-1)
\times (\Phi(t-1) - \rho L \otimes I_p)e(t-1)]
+ E[e^T(t)C^T(t)H^T(t)\Phi^{-1}(t)H(t)C(t)e(t)]]. \tag{39}
\]

**Stage 3.** Consider the first term on the right hand side of (39). Since \( L \) is positive semi-definite, we can find a matrix \( U = (L \otimes I_p)^{-\frac{1}{2}} \) such that \( L \otimes I_p = U^T U \). By the matrix inversion lemma, it holds that

\[
(\Phi(t-1) - \rho L \otimes I_p)^{-1}
= (\Phi(t-1) - \rho U^T U)^{-1}
= \Phi^{-1}(t-1) + \rho \Phi^{-1}(t-1)U^T
\times (I_p - \rho \Phi^{-1}(t-1)U)^{-1}U \Phi^{-1}(t-1). \tag{40}
\]
For $\lambda = 1$, it follows from (2) that $\Phi^{-1}(t-1) - \Phi^{-1}(t) \succeq 0_{Jp}$. Since $\Phi^{-1}(0) = \gamma L$, it holds that $\Phi^{-1}(t-1) \preceq \gamma I_p$ for all $t \geq 1$, and consequently

$$I_{Jp} - \rho U \Phi^{-1}(t-1) U^T \succeq I_{Jp} - \rho \gamma U U^T = I_{Jp} - \rho \gamma L \otimes I_p.$$  

If $0 < \rho < 1/(\gamma \lambda_{\max}(L))$, then for all $t \geq 1$ it follows that

$$I_{Jp} - \rho U \Phi^{-1}(t-1) U^T \succeq 0_{Jp}.$$  

This implies that the second term of (40) is positive definite. Thus, we have

$$\Phi^{-1}(t) \preceq \Phi^{-1}(t-1) \preceq (\Phi(t-1) - \rho \lambda \otimes I_p)^{-1}$$  

and hence, the first term on the right hand side of (39) is bounded by

$$E[\varepsilon(t)^T(\Phi(t-1) - \rho \lambda \otimes I_p)\Phi^{-1}(t)]$$

$$\times (\Phi(t-1) - \rho \lambda \otimes I_p)\varepsilon(t-1)]$$

$$\leq E[\varepsilon(t)^T(\Phi(t-1) - \rho \lambda \otimes I_p)^T\varepsilon(t-1)]$$

$$\leq E[\varepsilon(t)^T(\Phi(t-1))^T\varepsilon(t-1)].$$  

**Stage 4.** Now consider the second term on the right hand side of (39). Manipulating the expectation yields

$$E[\varepsilon(t)^T C(t) \Phi^{-1}(t) H(t) C(t) \varepsilon(t)]$$

$$= E[\varepsilon(t)^T C(t) \Phi^{-1}(t) H(t) C(t) \varepsilon(t)]$$

$$= E[\varepsilon(t)^T C(t) \Phi^{-1}(t) H(t) C(t) \varepsilon(t)]$$

$$= E[\varepsilon(t)^T C(t) \Phi^{-1}(t) H(t) C(t) \varepsilon(t)]$$

$$\leq \gamma \sum_j \sigma_j^2 \lambda_{\max}^{-1}(R_{h_j}) \text{tr}(R_{h_j}).$$  

While for $t < t_0$

$$E[\varepsilon(t)^T C(t) \Phi^{-1}(t) H(t) C(t) \varepsilon(t)]$$

$$\leq \gamma \sum_j \sigma_j^2 \lambda_{\max}^{-1}(R_{h_j}) \text{tr}(R_{h_j}).$$  

Summing (46) from $r = t_0$ to $r = t$ and (47) from $r = 1$ to $r = t_0 - 1$, applying telescopic cancellation, and noticing that $\Phi(0) = \gamma^{-1} I_p$, yields for $t \geq t_0$

$$E[\varepsilon(t)^T C(t) \Phi^{-1}(t) H(t) C(t) \varepsilon(t)]$$

$$\leq \gamma \sum_j \sigma_j^2 \lambda_{\max}^{-1}(R_{h_j}) \text{tr}(R_{h_j}).$$  

For $t < t_0$, we have $\Phi^{-1}(t) \preceq \Phi^{-1}(0) = \gamma I_p$ because to (41), and thus

$$\sum_j \sigma_j^2 E[|h_j(t)|^2 \Phi^{-1}(t) h_j(t)]$$

$$\leq \sum_j \gamma \sigma_j^2 \text{tr}(R_{h_j}).$$  

Therefore, for $t < t_0$ (43) yields

$$E[\varepsilon(t)^T C(t) \Phi^{-1}(t) H(t) C(t) \varepsilon(t)]$$

$$\leq \gamma \sum_j \sigma_j^2 \text{tr}(R_{h_j}).$$  

**Stage 5.** Substituting (42), (44) and (45) into (39) implies for $t \geq t_0$ that

$$E[\varepsilon(t)^T C(t) \Phi^{-1}(t) H(t) C(t) \varepsilon(t)]$$

$$\leq \gamma \sum_j \sigma_j^2 \lambda_{\max}^{-1}(R_{h_j}) \text{tr}(R_{h_j})$$  

which in turn implies that

$$2Q(\tau) \mu E[||e(t)||^2]$$

$$\leq \gamma \sum_j \sigma_j^2 \lambda_{\max}^{-1}(R_{h_j}) \text{tr}(R_{h_j})$$  

This completes the proof of CD-RLS-1.
Consider next CD-RLS-2. Stage 1 of the proof remains the same, while for Stage 2, $e_j(t-1) - e_j(t-1)$ is replaced by $c_j(t)[e_j(t-1) - e_{j'}(t-1)]$ in (36) to arrive at
\[
\Phi_j e_j(t)
\]
\[
= \Phi_j(t-1)e_j(t-1) - c_j(t)h_j(t)h_j^T(t)e_j(t-1) + c_j(t)h_j(t)e_j(t) - \rho \sum_{j' \notin \mathcal{N}_j} c_j(t)[e_j(t-1) - e_{j'}(t-1)].
\] (49)

Its matrix form (39) can be expressed as
\[
E[e^T(t)\Phi(t)e(t)] = E[e^T(t-1)(\Phi(t-1) - \rho(C(t)L) \otimes I_p)\Phi^{-1}(t)]
\]
\[
\times (\Phi(t-1) - \rho(C(t)L) \otimes I_p)e(t-1)]
\]
\[
+ E[e^T(t)C^T(t)H(t)\Phi^{-1}(t)H(t)C(t)e(t)].
\] (50)

Observe that the right hand sides of (39) and (50) are only different in their first terms. Similar to Stage 3 (cf. (42)), we need to show that the first term satisfies
\[
E[e^T(t-1)\Phi(t-1)e(t-1)]
\]
\[
\leq E[e^T(t-1)\Phi(t-1)e(t-1)].
\] (51)

Substituting the update (22) with $\lambda = 1$ into (51), it suffices to prove that
\[
E[e^T(t-1)C(t) \otimes I_p H(t)H^T(t)]
\]
\[
\times (I_J + H^T(t)\Phi^{-1}(t-1)H(t))^{-1} \otimes I_p e(t-1)]
\]
\[
\geq \rho E[e^T(t-1)We(t-1)]
\] (52)

where
\[
W := W_1 + W_1^T - W_2 - (LC(t)) \otimes I_p - (C(t)L) \otimes I_p
\]
\[
+ \rho L \otimes I_p \Phi^{-1}(t-1)/(C(t)L) \otimes I_p
\]
\[
W_1 := C(t) \otimes I_p H(t)H^T(t)\Phi^{-1}(t-1)
\]
\[
\times ((I_J + H^T(t)\Phi^{-1}(t-1)H(t))^{-1} L) \otimes I_p
\]
\[
W_2 := (LC(t)) \otimes I_p \Phi^{-1}(t-1)H(t)H^T(t)\Phi^{-1}(t-1)
\]
\[
\times ((I_J + H^T(t)\Phi^{-1}(t-1)H(t))^{-1} L) \otimes I_p.
\]

For the left hand side of (52), use the lower bound of the conditional expectation $2Q(\tau) \leq E[c_j(t)|h_j(t), s_j(t-1)]$ to eliminate $C(t)$, and arrive at
\[
E[e^T(t-1)C(t) \otimes I_p H(t)H^T(t)]
\]
\[
\times (I_J + H^T(t)\Phi^{-1}(t-1)H(t))^{-1} \otimes I_p e(t-1)]
\]
\[
\geq 2Q(\tau) E[e^T(t-1)H(t)H^T(t)]
\]
\[
\times (I_J + H^T(t)\Phi^{-1}(t-1)H(t))^{-1} \otimes I_p e(t-1)].
\] (53)

By (41), it holds that $\Phi^{-1}(t-1) \leq \Phi^{-1}(0) = \gamma I_{Jp}$, and thus
\[
[I_J + H^T(t)\Phi^{-1}(t-1)H(t)]^{-1} \geq [I_J + \gamma H^T(t)H(t)]^{-1}.
\]

By assumption $\{h_j(t)\}$ are uniformly bounded. If $h_j^T(t)h_j(t) \leq K$ for all $j = 1, \ldots, J$, we find
\[
[I_J + H^T(t)\Phi^{-1}(t-1)H(t)]^{-1} \geq \frac{1}{1 + \gamma K^2 I_J}.
\] (54)

Substituting (54) into (53), we obtain a lower bound for the left hand side of (52) given by
\[
E[e^T(t-1)C(t) \otimes I_p H(t)H^T(t)]
\]
\[
\times (I_J + H^T(t)\Phi^{-1}(t-1)H(t))^{-1} \otimes I_p e(t-1)]
\]
\[
\geq \frac{2Q(\tau)}{1 + \gamma K^2} E[e^T(t-1)H(t)H^T(t)e(t-1)]
\]
\[
= \frac{2Q(\tau)}{1 + \gamma K^2} E[||e(t-1)||^2].
\]

As for the right hand side of (52), it is upper bounded by
\[
\rho E[e^T(t-1)We(t-1)]
\]
\[
\leq \rho E[2||W_1||^2 + ||W_2||^2 + 2||L||^2 + \rho||L||^2 ||\Phi^{-1}(t-1)||^2 \otimes ||e(t-1)||^2]^2
\]
\[
= \rho \gamma \lambda_{max}(L) K^2 E[||e(t-1)||^2].
\] (56)

Noticing that $||C(t)||^2 \leq 1$, $||H(t)||^2 \leq K^2$ by assumption, $||\Phi^{-1}(t-1)||^2 \leq ||\Phi^{-1}(0)||^2 = \gamma$, $||I_J + H^T(t)\Phi^{-1}(t-1)H(t)||^2 \leq 1$ and $||L||^2 \leq \lambda_{max}(L)$, we find that
\[
||W_1||^2 \leq \gamma \lambda_{max}(L) K^2.
\]

Similarly, $||W_2||^2$ is upper bounded by
\[
||W_2||^2 \leq \gamma^2 \lambda_{max}(L) K^2.
\]

Therefore, (56) reduces to
\[
\rho E[e^T(t-1)We(t-1)]
\]
\[
\leq \rho(2\gamma \lambda_{max}(L) K^2 + \gamma^2 \lambda_{max}(L) K^2 + 2\lambda_{max}(L)
\]
\[
+ \rho \gamma \lambda_{max}(L) K^2 E[||e(t-1)||^2].
\] (57)

Considering a positive constant
\[
\rho_0 := \sqrt{\frac{2Q(\tau)\mu}{\gamma \lambda_{max}(L)^2 (1 + K^2) + (\gamma^2 K^2 + 1) K^2}}
\]
\[
\leq \frac{\gamma K^2}{2} + \frac{\gamma K^2 + \lambda_{max}(L)}{\gamma \lambda_{max}(L)}
\]

and combining (55) with (57), we see that if $\rho$ is chosen within $[0, \rho_0]$, then (52) holds for all $t \geq 1$; and so does (51).

Following Stages 4 and 5 in the proof for CD-RLS-1, we can show that (24) holds true for CD-RLS-2 $\forall t \geq t_0$. This completes the proof of the entire theorem. ■
APPENDIX C

PROOF OF THEOREM 2

Theorem 2 relies on the following lemma.

Lemma 1. There exist constants $M > 0$ and $t_0 > 0$ such that

$$E[||e_j(t) - e_j(t-1)||^2] \leq \frac{M^2}{t^2}, \forall j = 1, \cdots, J, \ t \geq t_0.$$ (58)

Proof of Lemma 1: The update of $e_j(t)$ for CD-RLS-3 is (cf. (35) for CD-RLS-1)

$$e_j(t) = e_j(t-1) - c_j(t)\Phi_j^{-1}(t)h_j(t)h_j^T(t)e_j(t-1) - c_j(t)\rho\Phi_j^{-1}(t)\sum_{j' \in N_j} (e_j(t-1) - e_{j'}(t-d_{j'}^t(t))).$$ (59)

Per time $t$, $t - d_{j'}^t(t)$ is the latest time slot when node $j$ received information from its neighbor $j'$. Therefore, $d_{j'}^t(t)$ can be viewed as network delay caused by the censoring strategy. Then we have

$$||e_j(t) - e_j(t-1)|| = ||c_j(t)\Phi_j^{-1}(t)h_j(t)h_j^T(t)e_j(t-1) - \rho\sum_{j' \in N_j} (e_j(t-1) - e_{j'}(t-d_{j'}^t(t)))||$$

$$\leq ||\Phi_j^{-1}(t)|| \sum_{j' \in N_j} ||e_j(t)|| + ||h_j(t)|| + \sum_{j' \in N_j} ||e_j(t-1)|| + ||e_{j'}(t-d_{j'}^t(t))||$$

In deriving the inequality we use the fact that $c_j(t) \in \{0, 1\}$.

According to (34) in the proof of Theorem 1, which also holds true for CD-RLS-3, there exists $t_0 > 0$, such that $2Q(\tau)R_{h_j} \leq \Phi_j(t)$ for all $t \geq t_0$ and $j = 1, \cdots, J$. Thus, $||\Phi_j^{-1}(t)||$ is upper bounded by $M_1/t$ where $M_1$ is a positive constant determined by $Q(\tau)$ and the smallest eigenvalue of $\Phi_j(t)$. By (as1) and (as2), $||h_j(t)||$, $||e_j(t-1)||$ and $||e_{j'}(t-d_{j'}^t(t))||$ are also upper bounded. Therefore, there exist constants $M_2, M_3, M_4 > 0$, such that

$$||e_j(t) - e_j(t-1)||^2 \leq \frac{1}{t^2}[M_2 + M_3\epsilon_j(t)] + M_4\epsilon_j(t)^2].$$

Taking expectations on both sides yields (58).

Now we turn to prove Theorem 2.

Proof of Theorem 2: Rewrite the update of $e_j(t)$ for CD-RLS-3 in (59) to

$$e_j(t) = e_j(t-1) - c_j(t)\Phi_j^{-1}(t)h_j(t)h_j^T(t)e_j(t-1) - c_j(t)\rho\Phi_j^{-1}(t)\sum_{j' \in N_j} (e_j(t-1) - e_{j'}(t-1))$$

$$- c_j(t)\rho\Phi_j^{-1}(t)\sum_{j' \in N_j} (e_j(t-1) - e_{j'}(t-d_{j'}^t(t)))$$

$$+ c_j(t)\Phi_j^{-1}(t)h_j(t)e_j(t).$$

Multiplying $\Phi_j(t)$ on both sides, we have

$$\Phi_j(t)e_j(t) = \Phi_j(t-1)e_j(t-1) - c_j(t)\rho\sum_{j' \in N_j} (e_j(t-1) - e_{j'}(t-1)) - c_j(t)\sum_{j' \in N_j} (e_{j'}(t-1) - e_{j'}(t-d_{j'}^t(t)))$$

$$+ c_j(t)h_j(t)e_j(t).$$

Using the same notations as in the proof of Theorem 1, we obtain an matrix form

$$\Phi_j(t)e_j(t) = (\Phi_j(t-1) - \rho(\mu(t)L \otimes I_p)e_j(t-1) + H(t)c(t)e_j(t) - \rho(\mu(t) \otimes I_p)e_j(t).$$

where $\tilde{e}(t) \in \mathbb{R}^d$ and its $j$th block is $\sum_{j' \in N_j} (e_{j'}(t-1) - e_{j'}(t-d_{j'}^t(t)))$. Observe that $\tilde{e}(t)$ contains the differences between the local estimates and their delayed values, and hence plays a critical role in the convergence proof. Below we look for an upper bound for $E[||\tilde{e}(t)||^2]$.

By the Cauchy-Schwarz inequality, we have

$$\frac{1}{t^2}\sum_{j' \in N_j} ||e_{j'}(t-1) - e_{j'}(t-d_{j'}^t(t))||^2$$

$$\leq \frac{1}{t^2}\sum_{j' \in N_j} ||e_{j'}(t-1)|| + ||e_{j'}(t-d_{j'}^t(t))||$$

$$\leq \sum_{j' \in N_j} ||e_{j'}(t-1)|| + ||e_{j'}(t-d_{j'}^t(t))||$$

$$\leq \sum_{j' \in N_j} (d_{j'}^t(t)-1)\sum_{k=1}^{d_{j'}^t(t)-1} ||e_{j'}(t-k) - e_{j'}(t-k-1)||^2$$

$$\leq \sum_{j' \in N_j} (d_{j'}^t(t)-1)\sum_{j' \in N_j} (d_{j'}^t(t)-1)\sum_{k=1}^{d_{j'}^t(t)-1} ||e_{j'}(t-k) - e_{j'}(t-k-1)||^2.$$
for some constant $M_0 > 0$. Because $(\cdot)^2$ is convex, we know that $(E[|\tilde{e}(t)|])^2 \leq E[|\tilde{e}(t)|^2]$. Thus, it holds
\[
E[|\tilde{e}(t)|^2] \leq \sqrt{E[|\tilde{e}(t)|^2]}^2 \leq \frac{M_0}{t}. \tag{61}
\]
Back to (60), multiplying $\Phi^{-1}(t)$ on both sides yields
\[
\Phi^{-1}(t)\tilde{e}(t) = \Phi^{-1}(t)(\Phi(t-1) - \rho(C(t)L) \otimes I_p)\tilde{e}(t-1) + \Phi^{-1}(t)(H(t)C(t)\tilde{e}(t) - \rho\Phi^{-1}(t)(C(t) \otimes I_p)\tilde{e}(t)).
\]
Since $H(t)$ and $e(t)$ are independent as given by (as1), we have
\[
E[\tilde{e}^T(t)\Phi(t)\tilde{e}(t)] = E[\tilde{e}^T(t-1)(\Phi(t-1) - \rho(C(t)L) \otimes I_p)\tilde{e}(t-1)]
\times (\Phi(t-1) - \rho(C(t)L) \otimes I_p)\tilde{e}(t-1)]
+ E[\tilde{e}^T(t)(H(t)C(t)\tilde{e}(t) - \rho\Phi^{-1}(t)(C(t) \otimes I_p)\tilde{e}(t)]
+ \rho^2 E[\tilde{e}^T(t)C(t) \otimes I_p \Phi^{-1}(t)C(t) \otimes I_p \tilde{e}(t)]
+ \rho E[\tilde{e}^T(t)C(t) \otimes I_p \Phi^{-1}(t)]
\times (\Phi(t-1) - \rho(C(t)L) \otimes I_p)\tilde{e}(t-1)].
\tag{62}
\]
Observe that (62) is different to (50) for having the last two terms at the right hand side. Because all the diagonal elements $c_j(t)$ in the diagonal matrix $C(t)$ are within $[0,1]$, \forall $t \geq t_0$
\[
\rho^2 E[\tilde{e}^T(t)C(t) \otimes I_p \Phi^{-1}(t)]
\times (\Phi(t-1) - \rho(C(t)L) \otimes I_p)\tilde{e}(t-1)]
\leq \rho^2 E[|\tilde{e}(t)|^2]|\Phi^{-1}(t)||\tilde{e}(t-1)|^2.
\]
The right hand side is in the order of $O(1/t^3)$ because $|\Phi^{-1}(t)||\tilde{e}(t-1)|^2$ is no larger than $\lambda_{max}(R_{h_j}^{-1})/(2Q(t)t)$ for all $t \geq t_0$ as we have shown in Step 1 of the proof of Theorem 1 (cf. (34)). Meanwhile, \forall $t \geq t_0$
\[
\rho E[\tilde{e}^T(t)C(t) \otimes I_p \Phi^{-1}(t)]
\times (\Phi(t-1) - \rho(C(t)L) \otimes I_p)\tilde{e}(t-1)]
\leq \rho E[|\tilde{e}(t)|^2]|\Phi^{-1}(t)||\tilde{e}(t-1)|^2 + \rho^2|L||\tilde{e}(t-1)|^2.
\]
Observe that $E[|\tilde{e}(t)|^2]$ and $E[|\Phi^{-1}(t)||\tilde{e}(t-1)|^2]$ are in the orders of $O(1/t)$ and $O(1/t^2)$, respectively, while $E[|\Phi^{-1}(t)||\tilde{e}(t-1)|^2 + \rho^2|L||\tilde{e}(t-1)|^2]$ is in the order of $O(t)$ because $\Phi_j(t) \leq \|R_{h_j}\|$ (cf. (34)). In addition, $|\tilde{e}(t-1)|$ is bounded by (as2). Therefore, the right hand side is in the order of $O(1/t)$.

For the first term at the right hand side of (62), similar to the proof for CD-RLS-2, if $\rho$ is chosen within $[0,\rho_0]$ we are able to show that (cf. (51))
\[
E[\tilde{e}^T(t-1)(\Phi(t-1) - \rho(C(t)L) \otimes I_p)^2 \Phi^{-1}(t)]
\times (\Phi(t-1) - \rho(C(t)L) \otimes I_p)\tilde{e}(t-1)]
\leq E[|\tilde{e}(t-1)|^2].
\]
Finally, following Step 4 of the proof of Theorem 1 to handle the second term at the right hand side of (62), we know that it is also in the order of $O(1/t)$. Therefore, for all $t \geq t_0$ (62) yields
\[
E[\tilde{e}^T(t)(\Phi(t)\tilde{e}(t))]
\leq E[|\tilde{e}(t-1)|^2 + \frac{K_1}{t} + \frac{K_2}{t^3}],
\tag{63}
\]
where $K_1, K_2 > 0$ are constants. Summing up both sides from time $t = r_0$ to $t = t$, we have
\[
E[\tilde{e}^T(t)(\Phi(t)\tilde{e}(t))]
\leq E[|\tilde{e}(t_0-1)|^2 + \frac{t}{r_0} + \sum_{r = r_0}^{t} \frac{K_1}{r} + \sum_{r = r_0}^{t} \frac{K_2}{r^3}].
\]
Observing that $E[\tilde{e}^T(t_0-1)(\Phi(t_0-1)\tilde{e}(t_0-1))$ is bounded because $|\tilde{e}(t_0-1)|$ is bounded by (as2), the right hand side of (63) is in the order of $O(1 + O(\ln(t)))$. Following the argument in Step 5 of the proof of Theorem 1, $|\Phi(t)||\tilde{e}(t)|$ is in the order of $O(1/t)$ when $t \geq t_0$. Therefore, $E[\tilde{e}^T(t)\Phi(t)\tilde{e}(t)]$ is in the order of $O(1/t) + O(\ln(t)/t)$, which completes the proof of Theorem 2.

\begin{thebibliography}{10}

[1] S. I. Amari, “Natural gradient works efficiently in learning,” Neural Computation, vol. 10, pp. 251–276, 1998.
[2] S. Appadwedula, V. V. Veeravalli, and D. L. Jones, “Decentralized detection and censoring sensors,” IEEE Transactions on Signal Processing, vol. 56, pp. 1362–1373, April 2008.
[3] R. Arroyo-Valles, S. Maleki, and G. Leus, “A censoring strategy for decentralized estimation in energy-constrained adaptive diffusion networks,” Proc. of Intl. Work. on Signal Processing Advances in Wireless Communications, Germany, June 2013.
[4] D. Berberidis, V. Kekatos, and G. G. Giannakis, “Online censoring for large-scale regressions with application to streaming big data,” IEEE Transactions on Signal Processing, vol. 64, pp. 3854–3867, Aug. 2016.
[5] V. Cevher, S. Becker, and M. Schmidt, “Convex optimization for big data: Scalable, randomized, and parallel algorithms for big data analytics,” IEEE Signal Processing Magazine, vol. 31, pp. 32–43, Sept. 2014.
[6] G. B. Giannakis, Q. Ling, G. Mateos, I. D. Schizas, and H. Zhu, “Decentralized learning for wireless communications and networking,” Splitting Methods in Communication and Imaging, Science and Engineering, R. Glowinski, S. Osher, and W. Yin (eds.), Springer, 2016.
[7] R. Jiang, Y. Lin, B. Chen, and B. Suter, “Distributed sensor censoring for detection in sensor networks under communication constraints,” Proc. of Asilomar Conf. on Signals, Systems and Computers, Pacific Grove, CA, Nov. 2005.
[8] R. Jiang and B. Chen, “Fusion of censored decisions in wireless sensor networks,” IEEE Transactions on Wireless Communications, vol. 4, pp. 2668–2673, Dec. 2005.
[9] H. J. Kushner and G. G. Yin, Stochastic Approximation Algorithms and Applications, Springer, 1997.
[10] M. Lichman, “UCI machine learning repository,” 2013. Available at: http://archive.ics.uci.edu/ml.
[11] Z. Liu, C. Li, and Y. Liu, “Distributed censored regression over networks,” IEEE Transactions on Signal Processing, vol. 63, pp. 5437–5449, Oct. 2015.
[12] C. G. Lopes and A. H. Sayed, “Diffusion least-mean squares over networks,” IEEE Transactions on Signal Processing, vol. 56, pp. 3122–3136, July 2008.
[13] G. Mateos and G. B. Giannakis, “Distributed recursive least-squares: Stability and performance analysis,” IEEE Transactions on Signal Processing, vol. 60, pp. 3740–3754, July 2012.

\end{thebibliography}
[14] G. Mateos, I. D. Schizas and G. B. Giannakis, “Distributed Recursive Least-Squares for Consensus-Based In-Network Adaptive Estimation,” *IEEE Transactions on Signal Processing*, vol. 57, pp. 4583–4588, Nov. 2009.

[15] E. Msechu and G. B. Giannakis, “Sensor-centric data reduction for estimation with WSNs via censoring and quantization,” *IEEE Transactions on Signal Processing*, vol. 60, pp. 400–414, Jan. 2012.

[16] N. Patwari, and A. O. Hero, “Hierarchical censoring for distributed detection in wireless sensor networks,” *Proc. of Intl. Conf. on Acoustics, Speech, and Signal Processing*, Hong Kong, 2003.

[17] J. Predd, S. Kulkarni, and H. V. Poor, “Distributed learning in wireless sensor networks,” *IEEE Signal Processing Magazine*, vol. 23, pp. 56–69, July 2006.

[18] M. Rabbat and R. Nowak, “Distributed optimization in sensor networks,” *Intl. Conf. on Information Processing in Sensor Networks*, pp. 20-27, Berkeley, CA, April 2004.

[19] C. Rago, P. Willett, and Y. Bar-Shalom, “Censoring sensors: A low-communication-rate scheme for distributed detection,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 32, pp. 554–568, Apr. 1996.

[20] M. A. Sharkh, M. Jammal, A. Shami, and A. Ouda, “Resource allocation in a network-based cloud computing environment: Design challenges,” *IEEE Communications Magazine*, vol. 51, pp. 46–52, Nov. 2013.

[21] K. Slavakis, S. J. Kim, G. Mateos, and G. B. Giannakis, “Stochastic approximation vis-a-vis online learning for big data analytics,” *IEEE Signal Processing Magazine*, vol. 31, pp. 124–129, Nov. 2014.

[22] V. Solo and X. Kong, *Adaptive Signal Processing Algorithms: Stability and Performance*, Prentice Hall, 1995.

[23] P. Tseng, “Applications of a splitting algorithm to decomposition in convex programming and variational inequalities,” *SIAM Journal on Control and Optimization*, vol. 29, pp. 119–138, Jan. 1991.

[24] Z. Wang, Z. Yu, Q. Ling, D. Berberidis, and G. B. Giannakis, “Distributed recursive least-squares with data-adaptive censoring,” *Proc. Conf. on Acoustics, Speech, and Signal Processing*, New Orleans, March 2017.