DISTRIBUTED RECURSIVE LEAST-SQUARES WITH DATA-ADAPTIVE CENSORING

Zifeng Wang[*] *Zheng Yu*[*] *Qing Ling*[†] *Dimitris Berberidis*[‡] *Georgios B. Giannakis*[‡]

[*] Special Class for the Gifted Young, University of Science and Technology of China
[†] Department of Automation, University of Science and Technology of China
[‡] Department of Electrical and Computer Engineering, University of Minnesota

ABSTRACT

The deluge of networked big data motivates the development of computation- and communication-efficient network information processing algorithms. In this paper, we propose two data-adaptive censoring strategies that significantly reduce the computation and communication costs of the distributed recursive least-squares (D-RLS) algorithm. Through introducing a cost function that underrates the importance of those observations with small innovations, we develop the first censoring strategy based on the alternating minimization algorithm and the stochastic Newton method. It saves computation when a datum is censored. The computation and communication costs are further reduced by the second censoring strategy, which prohibits a node updating and transmitting its local estimate to neighbors when its current innovation is less than a threshold. For both strategies, a simple criterion for selecting the threshold of innovation is given so as to reach a target ratio of data reduction. The proposed censored D-RLS algorithms guarantee convergence to the optimal argument in the mean-square deviation sense. Numerical experiments validate the effectiveness of the proposed algorithms.

Index Terms— Distributed networks, distributed recursive least-squares (D-RLS), data-adaptive censoring

1. INTRODUCTION

Nowadays, various networks are generating massive streaming data. Examples include a wireless sensor network, where a large number of inexpensive sensors cooperate to monitor environment, or a data center network, where a group of servers collaboratively serve user requests. Since a single node has limited computation and/or storage resources, distributed information processing is preferable over a large-scale network [1]. In this paper, we focus on the distributed linear regression problem and devote to developing computation- and communication-efficient distributed recursive least-squares (D-RLS) algorithms.

The main technique we adopt to reduce computation and communication costs is data-adaptive censoring, which utilizes the redundancy of big data. Upon receiving an observation, every node determines whether it is informative or not. Less informative observations are discarded; meanwhile, data exchange between neighboring nodes occurs only when it is necessary. We propose two censored D-RLS algorithms that are able to achieve the same regression accuracy as their non-censored counterpart, but incur significantly smaller computation and communication burdens.

1.1. Related Works

RLS algorithms are well celebrated in centralized linear regression problems [2]. When linear observations of an unknown signal are given in an online manner, an RLS algorithm is able to recursively update the least-squares estimate, mitigating the computation burden of resolving a batch least-squares. The computation cost can be further reduced through using the idea of data-adaptive censoring [3]. Therein, the usefulness of an observation is measured by its innovation and less informative data is discarded. On the other hand, the distributed versions of RLS, which solve linear regression problems defined over distributed networks, are proposed in [4]. In a D-RLS algorithm, a node updates its local estimate of the unknown signal, which is common to the whole network, from both its local observations and the local estimates of its neighbors. As time evolves, all the local estimates reach a consensual value, which is the same as the centralized RLS solution. This paper takes advantages of both [3] and [4] by proposing censored and distributed RLS algorithms that accommodate for linear regression applications over networks.

Different to our setting that the network is fully distributed and nodes are only able to communicate with their neighbors, most of the existing distributed censoring algorithms consider the network with a fusion center. Their basic idea is that every node determines a likelihood ratio for its local data, and only transmits the local data to the fusion center for further processing when the likelihood ratio exceeds a threshold. The thresholds are solved by minimizing the probability of error in [5]. Communication constrains are further taken into account in [6]. Fusion over fading channels in a wireless sensor network is considered in [7]. Practical issues such as joint dependence of sensor decision rules, randomization of decision strategies and and partially known distributions are considered in [8]. The work of [9] considers the impact of quantized communications on censoring and the resulting mean-square error.

Other than the star topology discussed in the above papers, [10] investigates the censoring strategy for a tree structure. If one node has a local likelihood ratio larger than a threshold, then it local datum is sent to its parent node. A fully distributed setting is considered in [11], where every node determines whether to transmit its local estimate to its neighbors by comparing its local estimate and the weighted average of the neighboring ones. This approach mitigates the communication cost. We further consider the reduction of the computation cost in this paper. We would also like to point out that the censored distributed linear regression algorithm in [12], though sharing a similar name as our work, considers how to handle partially known or noise-corrupted observations in order to correct bias. This is different to our goal of reducing computation and communication costs for distributed linear regression.

1.2. Our Contributions and Paper Organization

This paper proposes two data-adaptive online censoring strategies for distributed linear regression. The proposed censored D-RLS algorithms incur low computation and communication costs that are important to networked big data applications, while guarantee the quality of linear regression in theory.

In Section 2, we formulate the D-RLS problem (Section 2.1) and rewrite the D-RLS algorithm proposed in [4] to a new form (Section 2.2), which motives the development of two censoring strategies (Section 2.3). Section 3 provides theoretical results, including the derivation of the first censoring strategy (Section 3.1), convergence analysis of the two censoring strategies (Section 3.2), as well as the rule of setting the censoring threshold (Section 3.3). Numerical experiments in Section 4 demonstrate the effectiveness of the proposed censored D-RLS algorithms.

Notation. Lower- (upper-) case boldface letters denote column vectors (matrices). $(\cdot)^T$, $||\cdot||$ and $E[\cdot]$ stand for transpose, 2-norm and expectation, respectively. $tr(\mathbf{X})$, $\lambda_{\min}(\mathbf{X})$ and $\lambda_{\max}(\mathbf{X})$ are trace, minimum eigenvalue and maximum eigenvalue of matrix $\mathbf{X}$, respectively. $\mathcal{U}(a,b)$ denotes the uniform distribution within $[a,b]$. $\mathcal{N}(\mu,\sigma^2)$ denotes the Gaussian distribution with mean μ and variance σ^2. $\phi(t) = (1/\sqrt{2\pi})\exp(-t^2/2)$ denotes the standardized Gaussian probability density function, and $Q(z) = \int_z^{+\infty}\phi(t)dt$ is the associated complementary cumulative distribution function.

2. ALGORITHM DEVELOPMENT

This section introduces the online linear regression problem defined over networks and revisits the distributed recursive least-squares (D-RLS) algorithm. Two data-adaptive censoring strategies are proposed to reduce the computation and communication costs.

2.1. Problem Statement

Consider a bidirectionally connected network with J nodes, described by a graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, where $\mathcal{V}$ is the set of nodes with cardinality $|\mathcal{V}| = J$ and $\mathcal{E}$ is the set of edges. Each node j is only able to communicate with its one-hop neighbors where the neighbor set is denoted by $\mathcal{N}_j \subset \mathcal{V}$. The distributed network is deployed to estimate a real vector $\mathbf{s}_0 \in \mathbb{R}^p$. At each time instant $t = 0, 1, \ldots$, node j receives a scalar observation $x_j(t) \in \mathbb{R}$ by measuring $\mathbf{s}_0$ with a regression vector $\mathbf{h}_j(t) \in \mathbb{R}^p$. The measurement equation is $x_j(t) = \mathbf{h}_j^T(t)\mathbf{s}_0 + \epsilon_j(t)$, where $\epsilon_j(t)$ follows $\mathcal{N}(0, \sigma_j^2)$.

Our goal is to devise computation- and communication-efficient distributed online algorithms that solve the following exponentially-weighted least-squares (EWLS) problem:

$$\hat{\mathbf{s}}_{ewls}(t) := \arg\min_{\mathbf{s}} \sum_{\tau=0}^{t}\sum_{j=1}^{J} \lambda^{t-\tau}[x_j(\tau) - \mathbf{h}_j^T(\tau)\mathbf{s}]^2 \quad (1)$$

Here $\hat{\mathbf{s}}_{ewls}(t)$ is the EWLS estimator at time t and $\lambda \in (0,1]$ is a forgetting factor. Note that when $\lambda < 1$, the importance of past measurements is exponentially attenuated, which enables tracking of a non-stationary process.

2.2. D-RLS Revisited

The D-RLS algorithm based on the alternating minimization algorithm is given as follows [4]. At time t, node j receives a scalar observation $x_j(t)$ and a companion regression vector $\mathbf{h}_j(t)$. It starts by updating a covariance matrix $\mathbf{\Phi}_j^{-1}(t)$ from the previous one and the new regression vector:

$$\mathbf{\Phi}_j^{-1}(t) = \lambda^{-1}\mathbf{\Phi}_j^{-1}(t-1) - \frac{\lambda^{-1}\mathbf{\Phi}_j^{-1}(t-1)\mathbf{h}_j(t)\mathbf{h}_j^T(t)\mathbf{\Phi}_j^{-1}(t-1)}{\lambda + \mathbf{h}_j^T(t)\mathbf{\Phi}_j^{-1}(t-1)\mathbf{h}_j(t)} \quad (2)$$

Then node j updates a variable $\boldsymbol{\psi}_j(t)$ that stores the exponentially weighted summation of $\mathbf{h}_j(\tau)x_j(\tau)$, $\tau = 1, \cdots, t$:

$$\boldsymbol{\psi}_j(t) = \lambda\boldsymbol{\psi}_j(t-1) + \mathbf{h}_j(t)x_j(t) \quad (3)$$

Using $\mathbf{\Phi}_j^{-1}(t)$ and $\boldsymbol{\psi}_j(t)$, node j updates its local estimate $\mathbf{s}_j(t)$:

$$\mathbf{s}_j(t) = \mathbf{\Phi}_j^{-1}(t)\Big(\boldsymbol{\psi}_j(t) - \sum_{j'\in\mathcal{N}_j}\frac{\mathbf{v}_j^{j'}(t-1) - \mathbf{v}_{j'}^{j}(t-1)}{2}\Big) \quad (4)$$

$\mathbf{v}_j^{j'}(t-1)$ is the Lagrange multiplier of node j for its neighbor j' at time $t-1$. The Lagrange multiplier is given by the summation of the previous differences between node j and its neighbor j':

$$\mathbf{v}_j^{j'}(t-1) = \mathbf{v}_j^{j'}(t-2) + \frac{\rho}{2}[\mathbf{s}_j(t-1) - \mathbf{s}_{j'}(t-1)] \quad (5)$$

ρ is a positive constant.

Below we rewrite the D-RLS algorithm to an equivalent form, which reveals its connection with the centralized RLS algorithm, as well as motivates the idea of data-adaptive censoring. Detailed derivation of the equivalence is omitted due to the page limit. The update of $\mathbf{\Phi}_j^{-1}(t)$ remains the same as (2). However, the update of $\mathbf{s}_j(t)$ is rewritten to:

$$\begin{aligned}\mathbf{s}_j(t) =& \mathbf{s}_j(t-1) + \mathbf{\Phi}_j^{-1}(t)\mathbf{h}_j(t)[x_j(t) - \mathbf{h}_j^T(t)\mathbf{s}_j(t-1)] \\ &- \frac{\rho}{2}\mathbf{\Phi}_j^{-1}(t)\boldsymbol{\delta}_j(t-1)\end{aligned} \quad (6)$$

Here $\boldsymbol{\delta}_j(t)$ is a new Lagrange multiplier whose update is given by:

$$\begin{aligned}\boldsymbol{\delta}_j(t) =& \boldsymbol{\delta}_j(t-1) \\ &+ \sum_{j'\in\mathcal{N}_j}[\mathbf{s}_j(t) - \mathbf{s}_{j'}(t)] - \lambda\sum_{j'\in\mathcal{N}_j}[\mathbf{s}_j(t-1) - \mathbf{s}_{j'}(t-1)]\end{aligned} \quad (7)$$

Observe that $\boldsymbol{\delta}_j(t)$ stores the exponentially weighted summation of the differences between the local estimate of node j and all the neighboring local estimates. Interestingly, if the network is disconnected and the nodes are isolated, the Lagrange multiplier $\boldsymbol{\delta}_j(t) = 0$ and the update of $s_j(t)$ is very close to centralized RLS [2,13]. That is, the current estimate is modified from the previous one using the prediction error $x_j(t) - \mathbf{h}_j^T(t)\mathbf{s}_j(t-1)$, which is also termed as *innovation*. On the other hand, if the network is connected, nodes must consider their neighboring estimates, which provide new information from the network other than solely from their own observations. The term $(\rho/2)\mathbf{\Phi}_j^{-1}(t)\boldsymbol{\delta}_j(t-1)$ in (6) can also be treated as a Laplacian smoothness operator on the graph, which encourages all the nodes to reach a consensual estimate.

In D-RLS, (2) has a computational complexity of $O(3p^2/2)$, dominated by the multiplications in calculating $\mathbf{\Phi}_j^{-1}(t-1)\mathbf{h}_j(t)$ as well as the product of $\mathbf{\Phi}_j^{-1}(t-1)\mathbf{h}_j(t)$ and its transpose. Similarly, (6) has a computational complexity of $O(2p^2)$, dominated by the multiplications in calculating $\mathbf{\Phi}_j^{-1}(t)\mathbf{h}_j(t)$ and $\mathbf{\Phi}_j^{-1}(t)\boldsymbol{\delta}_j(t-1)$. The cost of computing (7) is minor. Regarding communication cost, after every iteration t, node j needs to transmit its local estimate $\mathbf{s}_j(t)$ to its neighbors and receive estimates $\mathbf{s}_{j'}(t)$ from all neighbors $j' \in \mathcal{N}_j$. The computation cost of the original D-RLS recursion (2), (3), (4) and (5) is close to that the new form (2), (6) and (7), except that (4) has a computation cost of $O(p^2)$ other than $O(2p^2)$ in (6). Meanwhile, the original form requires neighboring nodes j and j' to exchange $\mathbf{v}_j(t)$ and $\mathbf{v}_{j'}(t)$ in addition to $\mathbf{s}_j(t)$ and $\mathbf{s}_{j'}(t)$.

2.3. Censored D-RLS Strategies

The D-RLS algorithm has been shown as a powerful tool for distributed online linear regression [4]. However, its iteration-wise

Algorithm 1 Censored D-RLS-1 (CD-RLS-1)

Initialize $\boldsymbol{\delta}_j(-1)$, $\{\mathbf{s}_j(-1)\}_{j=1}^J$ and $\{\boldsymbol{\Phi}_j^{-1}(-1)\}_{j=1}^J$
for all $j \in \mathcal{V}$, $t = 0, 1, \ldots$ **do**
  if $|x_j(t) - \mathbf{h}_j^T(t)\mathbf{s}_j(t-1)| \leq \tau\sigma_j(t)$ **then**
    update $\boldsymbol{\Phi}_j^{-1}(t)$ using (9)
    update $\mathbf{s}_j(t)$ using (10)
  else
    update $\boldsymbol{\Phi}_j^{-1}(t)$ using (2)
    update $\mathbf{s}_j(t)$ using (6)
  end if
  compute $\boldsymbol{\delta}_j(t)$ using (7)
  transmit $\mathbf{s}_j(t)$ to and receive $\mathbf{s}_{j'}(t)$ from all $j' \in \mathcal{N}_j$
end for

Algorithm 2 Censored D-RLS-2 (CD-RLS-2)

Initialize $\boldsymbol{\delta}_j(-1)$, $\{\mathbf{s}_j(-1)\}_{j=1}^J$ and $\{\boldsymbol{\Phi}_j^{-1}(-1)\}_{j=1}^J$
for all $j \in \mathcal{V}$, $t = 0, 1, \ldots$ **do**
  if $|x_j(t) - \mathbf{h}_j^T(t)\mathbf{s}_j(t-1)| \leq \tau\sigma_j(t)$ **then**
    receive $\mathbf{s}_{j'}(t)$ from all $j' \in \mathcal{N}_j$
  else
    set $\mathbf{s}_{j'}(t-1)$ as recently received ones from all $j' \in \mathcal{N}_j$
    update $\boldsymbol{\Phi}_j^{-1}(t)$ using (2)
    update $\mathbf{s}_j(t)$ using (6)
    compute $\boldsymbol{\delta}_j(t)$ using (7)
    transmit $\mathbf{s}_j(t)$ to and receive $\mathbf{s}_{j'}(t)$ from all $j' \in \mathcal{N}_j$
  end if
end for

computation and communication costs are fixed, no matter the observations and/or the estimates from neighboring nodes are informative or not. This fact motivates us to introduce the idea of data-adaptive censoring to D-RLS, yielding two novel censored D-RLS strategies. They are different to the censored RLS algorithms proposed in [3], which focus on centralized online linear regression.

Our first censoring strategy comes from the following intuition: If a given datum $(x_j(t), \mathbf{h}_j(t))$ is not informative, we do not have to use it since its contribution to the local estimate of node j, as well as to those of the whole network, is limited. To be specific, define the censoring indicator $c_j(t)$ as

$$c_j(t) = \begin{cases} 0, & \text{if } |x_j(t) - \mathbf{h}_j^T(t)\mathbf{s}_j(t-1)| \leq \tau\sigma_j(t) \\ 1, & \text{if } |x_j(t) - \mathbf{h}_j^T(t)\mathbf{s}_j(t-1)| > \tau\sigma_j(t) \end{cases} \tag{8}$$

If the innovation $|x_j(t) - \mathbf{h}_j^T(t)\mathbf{s}_j(t-1)|$ is less than a threshold $\tau\sigma_j(t)$, then $c_j(t) = 0$ and $(x_j(t), \mathbf{h}_j(t))$ is censored; otherwise $c_j(t) = 1$ and $(x_j(t), \mathbf{h}_j(t))$ is used. Section 3.3 gives rules for tuning the positive constant threshold τ and the local estimate of noise variance $\sigma_j(t)^2$, whose computations are lightweight. If data censoring happens, we simply throw away the current datum by letting $\mathbf{h}_j(t) = \mathbf{0}$ in (2) and obtain:

$$\boldsymbol{\Phi}_j^{-1}(t) = \lambda^{-1}\boldsymbol{\Phi}_j^{-1}(t-1) \tag{9}$$

Likewise, letting $x_j(t) = 0$ and $\mathbf{h}_j(t) = \mathbf{0}$ in (6) yields:

$$\mathbf{s}_j(t) = \mathbf{s}_j(t-1) - \frac{\rho}{2}\boldsymbol{\Phi}_j^{-1}(t)\boldsymbol{\delta}_j(t-1) \tag{10}$$

The first censoring strategy is summarized in Algorithm 1. It reduces $5/7$ of the computation cost when the datum is censored at a certain iteration. To see so, observe that the scalar-matrix multiplications of $\lambda^{-1}\boldsymbol{\Phi}_j^{-1}(t-1)$ in (9) are not necessary since the update of $\boldsymbol{\Phi}_j^{-1}(t)$ can be merged to wherever it is needed (say, in (10) and in the next iteration). Meanwhile, in (4), the $O(p^2)$ multiplications $\boldsymbol{\Phi}_j^{-1}(t)\mathbf{h}_j(t)$ disappear, and the $O(p^2)$ multiplications $\boldsymbol{\Phi}_j^{-1}(t)\boldsymbol{\delta}_j(t-1)$ remain the same.

The first censoring strategy still requires every node to communicate with its neighbors at every iteration, and hence does not reduce message transmission. In order to also mitigate the communication cost, we propose the second censoring strategy, in which a node does not do any further computation and and only needs to receive neighboring iterates if its current datum is censored. The intuitive idea behind this strategy is that, if the datum is censored, then most likely the current local estimate is sufficiently accurate, and the node does not need to modify it using neighboring estimates. Meanwhile, its neighbors do not need its current estimate either, because they have received the same value previously. The second censoring strategy is summarized in Algorithm 2.

3. THEORETICAL ANALYSIS

This section sketches the derivation of the first censoring strategy. Convergence analysis of the two censoring strategies is given. We also address the practical issue of how to set the threshold $\tau\sigma_j(t)$.

3.1. Derivation of Censored D-RLS-1

To develop the first censoring strategy, we introduce a truncated quadratic cost function that is similar to the one used in the censored but centralized RLS [3]:

$$f_{j,t}(\mathbf{s}) := \begin{cases} 0, & |x_j(t) - \mathbf{h}_j^T(t)\mathbf{s}| \leq \tau\sigma_j(t) \\ \frac{[x_j(t) - \mathbf{h}_j^T(t)\mathbf{s}]^2 - \tau^2\sigma_j(t)^2}{2}, & |x_j(t) - \mathbf{h}_j^T(t)\mathbf{s}| > \tau\sigma_j(t) \end{cases}$$

This new cost function replaces the standard quadratic cost function $[x_j(\tau) - \mathbf{h}_j^T(\tau)\mathbf{s}]^2$ used in (1). Also, similar to [4], we introduce local estimate $\mathbf{s}_j$ for every node j as well as local estimates $\bar{\mathbf{z}}_j^{j'}$ and $\tilde{\mathbf{z}}_j^{j'}$ for every edge (j, j'). By forcing all neighboring local estimates to reach a consensus, at time t we have the following separable convex minimization problem:

$$\min_{\{\mathbf{s}_j\}_{j=1}^J} \sum_{\tau=1}^{t}\sum_{j=1}^{J} \lambda^{t-\tau} f_{j,\tau}(\mathbf{s}_j). \tag{11}$$
$$\text{s.t.} \quad \mathbf{s}_j = \bar{\mathbf{z}}_j^{j'}, \mathbf{s}_{j'} = \tilde{\mathbf{z}}_j^{j'}, \bar{\mathbf{z}}_j^{j'} = \tilde{\mathbf{z}}_j^{j'}, j \in \mathcal{V}, j' \in \mathcal{N}_j$$

Using the alternating minimization algorithm and the stochastic Newton method, we are able to derive the first censoring strategy. The techniques are close to those used in [3] and [4], but with modifications to handle the new cost function and simplify the update rules. We leave the detailed derivation to a longer report.

3.2. Convergence Analysis

In this section, we provide convergence properties of the two censoring strategies when the forgetting factor $\lambda = 1$. We make the following assumption on the linear regression model.

Assumption 1. *Observations obey the linear model $x_j(t) = \mathbf{h}_j(t)\mathbf{s}_0 + \epsilon_j(t)$, where the noises $\epsilon_j(t) \sim \mathcal{N}(0, \sigma_j^2)$ are independent over both nodes j and times t. Regression vectors $\mathbf{h}_j(t)$ are uniformly bounded and independent with $\epsilon_j(t)$. The covariance matrices of $\mathbf{h}_j(t)$ are constant and positive definite, namely, $\mathbf{R}_{h_j} := E[\mathbf{h}_j(t)\mathbf{h}_j(t)^T] \succ \mathbf{0}_{p\times p}$. Besides, $\{c_j(t)\mathbf{h}_j(t)\mathbf{h}_j^T(t)\}$ is assumed to be an ergodic process, while $\{\epsilon_j(t)\}$ and $\{c_j(t)\}$ are assumed to be irrelevant.*

We are interested in the global mean-square deviation (MSD) [14], which is the summation of the local MSDs [15, 16] defined by:

$$\text{MSD}_j(t) = E[||\mathbf{s}_j(t) - \mathbf{s}_0||^2], j = 1, \ldots, J$$

The main theorem is given below.

Theorem 1. *Consider the censored D-RLS strategies given by Algorithms 1 and 2. Suppose that for every node i the threshold $\sigma_j(t)$ is chosen to be σ_j and $\mathbf{\Phi}_j^{-1}(-1) = \delta \mathbf{I}_p$ where δ is a positive constant. If the step size ρ is sufficiently small, then there exits $t_0 > 0$, such that for all $t \geq t_0$ it holds*

$$\sum_{j=1}^{J} E[||\mathbf{s}_j(t) - \mathbf{s}_0||^2]$$
$$\leq \sum_{j=1}^{J} \frac{M_1}{t} ||\mathbf{s}_j(-1) - \mathbf{s}_0||^2 + \frac{M_2 \ln(t)}{t} \quad (12)$$

M_1 and M_2 are two constants determined by the parameters δ, τ and ρ, the network Laplacian, as well as the upper bound of $\mathbf{h}_j(t)$.

Theorem 1 shows that the global MSD defined in the left hand side of (12) converges to zero at the rate of $O(\ln(t)/t)$. It also indicates that the impact of the initial states $\mathbf{s}_j(0)$ vanishes at a faster rate of $O(1/t)$. The detailed proof is left to a longer version.

3.3. Threshold Setting and Variance Estimation

The threshold τ has significant influence on the performance of the censoring algorithms. The value of τ trades off the estimation accuracy and the computational/communication costs. Here a simple criterion of setting τ is determined by the expected censoring ratio π^*, which is defined as the quotient between the number of censored data and the number of total data [9]. The goal is to choose an appropriate τ such that the actual censoring ratio goes close to π^* when t goes to infinity – since we are dealing with streaming big data, the asymptotic property is of particular interest to us. When t is large enough, $\mathbf{s}$ is very close to $\mathbf{s}_0$, thus the innovation $x_j(t) - \mathbf{h}_j^T(t)\mathbf{s}_j(t-1) \approx x_j(t) - \mathbf{h}_j^T(t)\mathbf{s}_0 = \epsilon_j(t) \sim \mathcal{N}(0, \sigma_j^2)$. In consequence, $Pr(c_j(t) = 0) = Pr(|x_j(t) - \mathbf{h}_j^T(t)\mathbf{s}_j(t-1)| \leq \tau\sigma_j) \approx Pr(|\epsilon_j(t)| \leq \tau\sigma_j) = Pr(|\epsilon_j(t)/\sigma_j| \leq \tau) = 1 - 2Q(\tau)$, where the last equality comes from $\epsilon_j(t)/\sigma_j \sim \mathcal{N}(0,1)$. Therefore, $\pi^* = \lim_{t\to\infty} \frac{1}{t}\sum_{\tau=0}^{t} E[c_j(\tau)] \approx 1 - 2Q(\tau)$, which implies that $\tau = Q^{-1}((1-\pi^*)/2)$.

If the variances σ_j^2 are known, one can simply choose $\sigma_j(t) = \sigma_j$. However, in practical problems, σ_j are often unknown. In this case, we suggest to update $\sigma_j(t)$ in an online manner. Namely, $\sigma_j(t+1)^2 \approx (1/t)\sum_{\tau=1}^{t+1}(x_j(\tau) - \mathbf{h}_j^T(\tau)\mathbf{s}_0)^2 = (t-1)\sigma_j(t)^2/t + (x_j(t+1) - \mathbf{h}_j^T(t+1)\mathbf{s}_0)^2/t \approx (t-1)\sigma_j(t)^2/t + (x_j(t+1) - \mathbf{h}_j^T(t+1)\mathbf{s}_j(t))^2/t$.

4. NUMERICAL EXPERIMENTS

This section provides numerical results to validate the effectiveness of the proposed censoring strategies. We generate a network of $J = 15$ nodes, which are uniformly randomly deployed within a 1×1 square. Two nodes within the communication range of 0.3 are neighbors of each other. The observed unknown signal is p-dimensional and $p = 4$. The settings of the measurement equations are the same as those in [4]. Define an auxiliary sequence $r_j(t)$ that evolves according to $r_j(t) = (1-q)\beta_j r_j(t-1) + \sqrt{q}\omega_j(t)$. Starting from $r_j(t)$, the regression vector $\mathbf{h}_j(t)$ is formed by taking the next p elements, namely, $\mathbf{h}_j(t) = [r_j(t+p-1); \ldots; r_j(t)]$. Parameters are selected as $q = 0.5$, β_j follows $\mathcal{U}(0,1)$, and the diving white noise $\omega_j(t)$ follows $\mathcal{U}(-\sqrt{3}\sigma_{\omega_j}, \sqrt{3}\sigma_{w_j})$ where $\sigma_{\omega_j}^2$ follows $\mathcal{U}(0,2)$. Observation of node j is subject to additive white Gaussian noise, whose covariance is $\sigma_j^2 = 10^{-3}\alpha_j$ where α_j follows $\mathcal{U}(0,1)$.

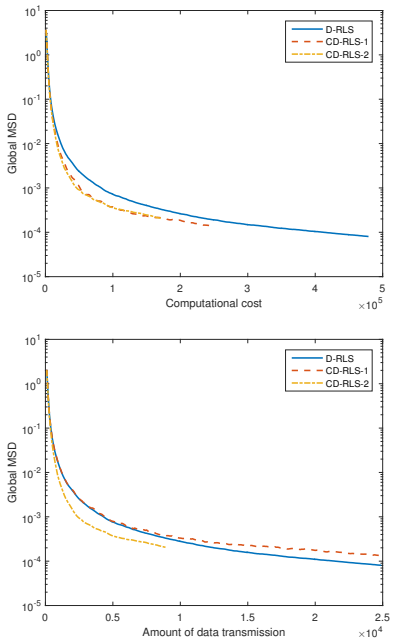

Fig. 1. Global MSD of the four algorithms versus computation cost (TOP) and amount of data transmission (BOTTOM).

We compare three approaches, D-RLS based on the alternating minimization algorithm [4] and the two censored D-RLS algorithms CD-RLS-1 and CD-RLS-2. The forgetting factor $\lambda = 1$. The common parameters are chosen as $\rho = 0.02$ and $\delta = 30$, which leads to the fastest convergence of D-RLS. For CD-RLS-1 and CD-RLS-2, we let the target censoring ratio be $\pi^* = 0.6$ such that the threshold $\tau = Q^{-1}((1-0.6)/2) \approx 0.84$. The variances σ_j^2 are estimated in an online manner as given by Section 3.3. For all curves obtained by running the algorithms, the ensemble averages are approximated via sample averaging over 100 runs of the experiment.

Fig. 1 shows global MSD, which is defined in the left hand side of (12), versus computation cost and amount of data transmission. It is easy to imagine that D-RLS is the fastest with respect to the number of iterations, while CD-RLS-2 is the slowest due to the aggressive censoring strategy. However, recall that for censored data, CD-RLS-1 only requires $2/7$ of the computation cost comparing to D-RLS, while DC-RLS-2 brings almost no computation cost. Taking into account of the target censoring ratio $\pi^* = 0.6$ (actual ratio is 0.6304 for CD-RLS-1 and 0.6285 for CD-RLS-2), the two censoring strategies significantly reduce the computation cost over D-RLS.

Regarding the amount of data transmission, which counts the number of transmitted local estimates in a unicast mode, CD-RLS-1 is the worst because every node needs to transmit its local estimate to neighbors, no matter its datum is censored or not. However, CD-RLS-2 shows significant improvement over D-RLS, demonstrating its potential of reducing both communication and computation costs in solving the distributed linear regression problem.

5. REFERENCES

[1] D. Bertsekas and J. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*, Second Edition, Athena Scientific, 1997

[2] H. J. Kushner and G. G. Yin, *Stochastic Approximation Algorithms and Applications*, Springer, New York, 1997.

[3] D. Berberidis, V. Kekatos, and G. B. 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.

[4] G. Mateos and G. B. Giannakis, "Distributed recursive least-squares: Stability and performance analysis," IEEE Transactions on Signal Processing, vol. 60, pp. 3740–3754, Jul. 2012.

[5] 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.

[6] R. Jiang, Y. Lin, B. Chen, and B. Suter, "Distributed sensor censoring for detection in sensor networks under communication constraints," Asilomar Conference on Signals, Systems and Computers, 2005.

[7] 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.

[8] S. Appadwedula, V. V. Veeravalli, and D. L. Jones, "Decentralized detection with censoring sensors," IEEE Transactions on Signal Processing, vol. 56, pp. 1362–1373, Apr. 2008.

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

[10] N. Patwari, and A. O. Hero, "Hierarchical censoring for distributed detection in wireless sensor networks," IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003.

[11] R. Arroyo-Valles, S. Maleki, and G. Leus, "A censoring strategy for decentralized estimation in energy-constrained adaptive diffusion networks," IEEE International Workshop on Signal Processing Advances in Wireless Communications, 2013.

[12] Z. Liu, C. Li, and Y. Liu, "Distributed Censored Regression Over Networks," IEEE Transactions on Signal Processing, vol. 63, pp. 5437–5449, Oct. 2015.

[13] 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.

[14] C. G. Lopes and A. H. Sayed, "Diffusion least-mean squares over adaptive networks: Formulation and performance analysis," IEEE Transactions on Signal Processing, vol. 56, pp. 3122–3136, Jul. 2008.

[15] A. H. Sayed, *Fundamentals of Adaptive Filtering*, John Wiley & Sons, 2003.

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