Online Censoring for Large-Scale Regressions with Application to Streaming Big Data

Dimitris Berberidis, *Student Member, IEEE,* Vassilis Kekatos, *Member, IEEE,*
and Georgios B. Giannakis*, *Fellow, IEEE*

***Abstract*—On par with data-intensive applications, the sheer size of modern linear regression problems creates an ever-growing demand for efficient solvers. Fortunately, a significant percentage of the data accrued can be omitted while maintaining a certain quality of statistical inference with an affordable computational budget. This work introduces means of identifying and omitting less informative observations in an online and data-adaptive fashion. Given streaming data, the related maximum-likelihood estimator is sequentially found using first- and second-order stochastic approximation algorithms. These schemes are well suited when data are inherently censored or when the aim is to save communication overhead in decentralized learning setups. In a different operational scenario, the task of joint censoring and estimation is put forth to solve large-scale linear regressions in a centralized setup. Novel online algorithms are developed enjoying simple closed-form updates and provable (non)asymptotic convergence guarantees. To attain desired censoring patterns and levels of dimensionality reduction, thresholding rules are investigated too. Numerical tests on real and synthetic datasets corroborate the efficacy of the proposed data-adaptive methods compared to data-agnostic random projection-based alternatives.**

***Index Terms*—Parameter estimation, least squares, stochastic approximation, big data, support vector machines, data reduction, censoring, LMS, RLS, random projections.**

I. Introduction

Nowadays omni-present monitoring sensors, search engines, rating sites, and Internet-friendly portable devices generate massive volumes of typically dynamic data [1]. The task of extracting the most informative, yet low-dimensional structure from high-dimensional datasets is thus of utmost importance. Fast-streaming and large in volume data, motivate well updating analytics rather than re-calculating new ones from scratch, each time a new observation becomes available. Redundancy is an attribute of massive datasets encountered in various applications [2], and exploiting it judiciously offers an effective means of reducing data processing costs.

In this regard, the notion of optimal design of experiments has been advocated for reducing the number of data required for inference tasks [3]. In recent works, the importance of sequential optimization along with random sampling of Big Data has been highlighted [1]. Specifically for linear regressions, random projection (RP)-based methods have been advocated for reducing the size of large-scale least-squares (LS) problems [4], [5], [6], [7], [8]; see also [9] for a randomized approach on constrained LS, and [10], [11] for randomized fast ridge regression. As far as online alternatives, the randomized Kaczmarz's (a.k.a. normalized least-mean-squares (LMS)) algorithm generates a sequence of linear regression estimates from projections onto affine subspaces [12], [13], [14]. Sequential optimization includes stochastic approximation, along with recent advances on online learning [15]. Frugal solvers of (possibly sparse) linear regressions are also available by estimating regression coefficients based on (severely) quantized data [16], [17]; see also [18] for decentralized sparse LS solvers.

In this context, the idea here draws on interval censoring to discard "less informative" observations. Censoring emerges naturally in several areas, and *batch* estimators relying on censored data have been used in econometrics, biometrics, and engineering tasks [19], including survival analysis [20], saturated metering [21], and spectrum sensing [22]. It has recently been employed to select data for distributed estimation of parameters and dynamical processes using resource-constrained wireless sensor networks, thus trading off performance for tractability [23], [24]. These works confirm that estimation accuracy achieved with censored measurements can be comparable to that based on uncensored data. Hence, censoring offers the potential to lower data processing costs, a feature certainly desirable in Big Data applications.

To this end, the present work employs interval censoring for large-scale *online* regression. Two censoring strategies are put forth, each tailored for different application scenarios (Section II). According to the first strategy, data are censored using a *data-nonadaptive rule.* This strategy is ideal when measurements are inherently censored (for example in survival analysis, saturated metering, and localization applications); and/or when censoring is introduced to reduce the cost of forwarding distributed data to a remote processing site. Relative to [23]–[24], the contribution here is first- and second-order stochastic approximation algorithms

Manuscript received Jul. 22, 2015; revised Nov. 30, 2015, and Feb. 8, 2016; accepted Mar. 7, 2016. Date of publication XXXXXX XX, 2016; date of current version XXXXXX XX, 20XX. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Sergio Netto.

Work in this paper was supported by NSF grants 1343860, 1442686, 1514056, and 1500713; and NIH grant 1R01GM104975-01. D. Berberidis and G. B. Giannakis are with the ECE Dept., University of Minnesota, Minneapolis, MN 55455, USA. V. Kekatos is with the ECE Dept., Virginia Tech, Blacksburg, VA 24061, USA. E-mails: bermp001@umn.edu; kekatos@vt.edu; georgios@umn.edu.

Color versions of one or more of the figures is this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier XXXXXX

for sequentially maximizing the likelihood of censored and uncensored observations (Section III). Error bounds as well as threshold selection rules for attaining prescribed data rejection ratios are provided. The second strategy employs censoring based on a *data-adaptive rule* to reduce the complexity of large-scale linear regressions (Section IV); see also [24] for innovation-based censoring. The difference with dimensionality-reduction alternatives, such as optimal design of experiments, randomized Kaczmarz's and RP-based methods, is that the devised technique discards observations in a data-driven manner. Judiciously designed threshold rules and robust versions of the algorithms are studied too. Section V compares the performance of the developed algorithms to competing alternatives on real and synthetic data, and the work is concluded in Section VI.

Notation. Lower- (upper-) case boldface letters denote column vectors (matrices). Calligraphic symbols are reserved for sets, while symbol ^{T} stands for transposition. Vectors $\mathbf{0}$, $\mathbf{1}$, and $\mathbf{e}_n$ denote the all-zeros, the all-ones, and the n-th canonical vector, respectively. Symbol $\mathbb{1}_E$ denotes the indicator for the event E. Notation $\mathcal{N}(\mathbf{m}, \mathbf{C})$ stands for the multivariate Gaussian distribution with mean $\mathbf{m}$ and covariance matrix $\mathbf{C}$; while $\phi(t) := (1/\sqrt{2\pi})\exp(-t^2/2)$ denotes the standardized Gaussian probability density function (pdf); and $Q(z) := \int_z^{+\infty} \phi(t)\mathrm{d}t$ the associated complementary cumulative distribution function. Symbols $\mathrm{tr}(\mathbf{X})$, $\lambda_{\min}(\mathbf{X})$, and $\lambda_{\max}(\mathbf{X})$ are reserved for the trace, the minimum and maximum eigenvalues of matrix $\mathbf{X}$, respectively.

II. Problem Statement and Preliminaries

Consider a $p \times 1$ vector of unknown parameters $\boldsymbol{\theta}_o$ generating scalar streaming observations

$$y_n = \mathbf{x}_n^T \boldsymbol{\theta}_o + \upsilon_n, \qquad n = 1, 2, \ldots, D \tag{1}$$

where $\mathbf{x}_n^T$ is the n-th row of the $D \times p$ regression matrix $\mathbf{X}$, and the noise samples υ_n are assumed independently drawn from $\mathcal{N}(0, \sigma^2)$. The high-level goal is to estimate $\boldsymbol{\theta}_o$ in an online manner, while meeting minimal resource requirements. The term resources here refers to the total number of utilized observations $\{y_n\}$ and/or rows $\{\mathbf{x}_n\}$, as well as the overall computational complexity of the estimation task. Furthermore, the sought data- and complexity-reduction schemes are desired to be data-adaptive, and thus scalable to the size of any given dataset $\{y_n, \mathbf{x}_n\}_{n=1}^D$. To meet such requirements, the proposed first- and second-order online estimation algorithms are based on the following two distinct censoring methods.

A. NAC and AC Rules

A generic censoring rule for the data in (1) is given by

$$z_n := \begin{cases} * & , \; y_n \in \mathcal{C}_n \\ y_n & , \; \text{otherwise} \end{cases}, \qquad n = 1, \ldots, D \tag{2}$$

where $*$ denotes an unknown value when the n-th datum has been censored (thus it is unavailable) – a case where we only know that $y_n \in \mathcal{C}_n$ for some set $\mathcal{C}_n$; otherwise, the actual measurement y_n is observed. Given $\{z_n, \mathbf{x}_n\}_{n=1}^D$, the goal is to estimate $\boldsymbol{\theta}_o$. Aiming to reduce the cost of storage and possible transmission, it is prudent to rely on innovation-based interval censoring of y_n. To this end, define per time n the binary censoring variable $c_n = 1$ if $y_n \in \mathcal{C}_n$; and zero otherwise. Each datum is decided to be censored or not based on a predictor $\hat{y}_n$ formed using a preliminary (e.g., LS) estimate of $\boldsymbol{\theta}_o$ as

$$\hat{\boldsymbol{\theta}}_K = (\mathbf{X}_K^T \mathbf{X}_K)^{-1} \mathbf{X}_K^T \mathbf{y}_K \tag{3}$$

from $K \geq p$ measurements $(K \ll D)$ collected in $\mathbf{y}_K$, and the corresponding $K \times p$ regression matrix $\mathbf{X}_K$. Given $\hat{y}_n = \mathbf{x}_n^T \hat{\boldsymbol{\theta}}_K$, the prediction error $\tilde{y}_n := y_n - \hat{y}_n$ quantifies the importance of datum n in estimating $\boldsymbol{\theta}_o$. The latter motivates what we term *non-adaptive censoring* (NAC) strategy:

$$(z_n, c_n) := \begin{cases} (y_n, 0) & , \; \text{if } \left| \frac{y_n - \mathbf{x}_n^T \hat{\boldsymbol{\theta}}_K}{\sigma} \right| \geq \tau_n \\ (*, 1) & , \; \text{otherwise} \end{cases} \tag{4}$$

where $\{\tau_n\}_{n=1}^D$ are censoring thresholds, and as in (2), $*$ signifies that the exact value of y_n is unavailable. The rule (4) censors measurements whose absolute normalized innovation is smaller than τ_n; and it is non-adaptive in the sense that censoring depends on $\hat{\boldsymbol{\theta}}_K$ that has been derived from a fixed subset of K measurements. Clearly, the selection of $\{\tau_n\}_{n=1}^D$ affects the percentage of censored data. Given streaming data $\{z_n, c_n, \mathbf{x}_n\}$, the next section will consider constructing a sequential estimator of $\boldsymbol{\theta}_o$ from censored measurements.

The efficiency of NAC in (4) in terms of selecting informative data depends on the initial estimate $\hat{\boldsymbol{\theta}}_K$. A data-adaptive alternative is to take into account all censored data $\{\mathbf{x}_i, z_i\}_{i=1}^{n-1}$ available up to time n. Predicting data through the most recent estimate $\hat{\boldsymbol{\theta}}_{n-1}$ defines our *data-adaptive censoring* (AC) rule:

$$(z_n, c_n) := \begin{cases} (y_n, 0) & , \; \text{if } \left| \frac{y_n - \mathbf{x}_n^T \boldsymbol{\theta}_{n-1}}{\sigma} \right| \geq \tau_n \\ (*, 1) & , \; \text{otherwise} \end{cases} . \tag{5}$$

In Section IV, (5) will be combined with first- and second-order iterations to perform joint estimation and censoring online. Implementing the AC rule requires feeding back $\boldsymbol{\theta}_{n-1}$ from the estimation to the censoring module, which may be undesirable in decentralized settings. Nonetheless, in centralized linear regression, AC is well motivated for reducing the problem dimension and computational complexity.

III. Online Estimation with NAC

Survival data analysis, saturated metering devices, and localization tasks, are examples where censored observations occur naturally [25]. On the other hand, data in distributed applications can be purposefully censored to save communication overhead between local agents collecting data $(y_n, \mathbf{x}_n)$ and a central processing agent. With $\mathbf{x}_n$'s known to the central agent, the NAC rule in (4) can be applied so that less informative y_n's are not forwarded to the central agent. Focusing on this mode of intentional censoring, this section develops stochastic approximation solvers for finding the associated maximum-likelihood estimator (MLE). Although the error and threshold analyses are geared towards intentional

Algorithm 1 First-order SA-MLE

```
Initialize $\boldsymbol{\theta}_1$ as the LSE $\hat{\boldsymbol{\theta}}_K$ in (3).
for $n = 1 : D$ do
    Measurement $y_n$ is possibly censored using (4).
    Estimator receives $(z_n, c_n, \mathbf{x}_n)$.
    Parameter $\boldsymbol{\theta}$ is updated via (8) and (9).
end for
```

censoring, the devised solvers can be coupled with any data-nonadaptive thresholding rule of the form of (2), including those emerging with non-deliberately censored data.

Since noise samples $\{v_n\}_{n=1}^{D}$ in (1) are independent and (4) applies independently over data, $\{z_n, c_n\}_{n=1}^{D}$ are independent too. With $\mathbf{z}_D := [z_1, \ldots, z_D]^T$ and $\mathbf{c}_D := [c_1, \ldots, c_D]^T$, the joint pdf is $p(\mathbf{z}_D, \mathbf{c}_D; \boldsymbol{\theta}) = \prod_{n=1}^{D} p(z_n, c_n; \boldsymbol{\theta})$ with

$$p(z_n, c_n; \boldsymbol{\theta}) = \left[\mathcal{N}\left(z_n; \mathbf{x}_n^T\boldsymbol{\theta}, \sigma^2\right)\right]^{1-c_n} \left[\Pr\{c_n = 1; \boldsymbol{\theta}\}\right]^{c_n} \tag{6}$$

since $c_n = 0$ means no censoring, and thus $z_n = y_n$ is Gaussian distributed; whereas $c_n = 1$ implies $|y_n - \hat{y}_n| \leq \tau_n\sigma$, that is $\Pr\{c_n = 1; \boldsymbol{\theta}\} = \Pr\{\hat{y}_n - \tau_n\sigma - \mathbf{x}_n^T\boldsymbol{\theta} \leq v_n \leq \hat{y}_n + \tau_n\sigma - \mathbf{x}_n^T\boldsymbol{\theta}\}$. Recalling that v_n is Gaussian yields

$$\Pr\{c_n = 1; \boldsymbol{\theta}\} = Q\left(z_n^l(\boldsymbol{\theta})\right) - Q\left(z_n^u(\boldsymbol{\theta})\right)$$

where $z_n^l(\boldsymbol{\theta}) := -\tau_n - \frac{\mathbf{x}_n^T\boldsymbol{\theta} - \hat{y}_n}{\sigma}$ and $z_n^u(\boldsymbol{\theta}) := \tau_n - \frac{\mathbf{x}_n^T\boldsymbol{\theta} - \hat{y}_n}{\sigma}$. Then, the maximum-likelihood estimator (MLE) of $\boldsymbol{\theta}_o$ is

$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \; \mathcal{L}_D(\boldsymbol{\theta}) := \sum_{n=1}^{D} \ell_n(\boldsymbol{\theta}) \tag{7}$$

where functions $\ell_n(\boldsymbol{\theta})$ are given by [cf. (6)]

$$\ell_n(\boldsymbol{\theta}) := \tfrac{1-c_n}{2\sigma^2}\left(y_n - \mathbf{x}_n^T\boldsymbol{\theta}\right)^2 - c_n \log\left[Q\left(z_n^l(\boldsymbol{\theta})\right) - Q\left(z_n^u(\boldsymbol{\theta})\right)\right].$$

If the entire dataset $\{z_n, c_n, \mathbf{x}_n\}_{n=1}^{D}$ is available, the *batch* MLE can be obtained via gradient descent or by using Newton's iterations; see for example [24] and [23].

A. First-Order SA-MLE

Targeting streaming big data applications, we resort to stochastic approximation solutions and process censored data sequentially. In particular, when datum n becomes available, the unknown parameter can be updated as

$$\boldsymbol{\theta}_n := \boldsymbol{\theta}_{n-1} - \mu_n \mathbf{g}_n(\boldsymbol{\theta}_{n-1}) \tag{8}$$

for a step size $\mu_n > 0$, and with $\mathbf{g}_n(\boldsymbol{\theta}) = \beta_n(\boldsymbol{\theta})\mathbf{x}_n$ denoting the gradient of $\ell_n(\boldsymbol{\theta})$, where

$$\beta_n(\boldsymbol{\theta}) := \frac{1-c_n}{\sigma^2}(y_n - \mathbf{x}_n^T\boldsymbol{\theta}) + \frac{c_n}{\sigma}\frac{\phi\left(z_n^u(\boldsymbol{\theta})\right) - \phi\left(z_n^l(\boldsymbol{\theta})\right)}{Q\left(z_n^u(\boldsymbol{\theta})\right) - Q\left(z_n^l(\boldsymbol{\theta})\right)}. \tag{9}$$

The overall scheme is tabulated as Algorithm 1.

When the n-th datum is not censored ($c_n = 0$), the second summand in the right-hand side (RHS) of (9) vanishes, and (8) reduces to an ordinary LMS update. When $c_n = 1$, the first summand in (9) disappears, while the second summand captures the fact that the unavailable y_n lies in the known interval, that is $|y_n - \mathbf{x}_n^T\hat{\boldsymbol{\theta}}_K| \leq \tau_n\sigma$. The latter information would have been ignored by an ordinary LMS algorithm using merely the uncensored data.

The SA-MLE is in fact a Robbins-Monro iteration on the sequence $\{\mathbf{g}(\boldsymbol{\theta})\}_{n=1}^{D}$; hence, it inherits SA-related convergence properties. Specifically, by selecting $\mu_n = 1/(nM)$ for an appropriate M, the SA-MLE algorithm is asymptotically efficient and Gaussian [26, pg. 197]. Performance guarantees also hold with finite samples. Indeed, with D finite, the *regret* attained by iterates $\{\boldsymbol{\theta}_n\}$ against a vector $\boldsymbol{\theta}$ is defined as

$$R_D(\boldsymbol{\theta}) := \sum_{n=1}^{D} \left[\ell_n(\boldsymbol{\theta}_n) - \ell_n(\boldsymbol{\theta})\right]. \tag{10}$$

Selecting μ properly, Algorithm 1 can afford bounded regret as asserted next and shown in the Appendix.

Proposition 1. *With $\boldsymbol{\theta}^*$ denoting the minimizer of (7), suppose $\|\boldsymbol{\theta}^* - \hat{\boldsymbol{\theta}}_K\|_2 < A$, $\|\mathbf{x}_n\|_2 \leq \bar{x}$, and $|\beta_n(\boldsymbol{\theta})| \leq \bar{\beta}$ for $n = 1, \ldots, D$, and let $\boldsymbol{\theta}^*$ be the minimizer of (7). By choosing $\mu = cA/(\sqrt{2D}\bar{\beta}\bar{x})$ for some $c > 0$, the regret of the SA-MLE against $\boldsymbol{\theta}^*$ is bounded as*

$$R_D(\boldsymbol{\theta}^*) \leq \sqrt{2D}A\bar{x}\bar{\beta}\max\{c, 1/c\}.$$

Apparently, setting $c = 1$ yields the step size with the tightest regret bound. Otherwise, parameter c quantifies the performance degradation for deviating from that value. Proposition 1 assumes bounded $\mathbf{x}_n$'s and noise. Furthermore, we assume that $\|\boldsymbol{\theta}^* - \hat{\boldsymbol{\theta}}_K\|_2 < A$, which for large enough A holds with high probability, since $\boldsymbol{\theta}^*$ is the MLE and $\hat{\boldsymbol{\theta}}_K$ is the LSE based on uncensored data, and are both close to $\boldsymbol{\theta}_o$.

B. Second-Order SA-MLE

If extra complexity can be afforded, one may consider incorporating second-order information in the SA-MLE update to improve its performance. In practice, this is possible by replacing scalar with matrix step-sizes $\mathbf{M}_n$. Thus, the first-order stochastic gradient descent (SGD) update in (8) is modified as follows

$$\boldsymbol{\theta}_n := \boldsymbol{\theta}_{n-1} - \mathbf{M}_n^{-1}\mathbf{g}_n(\boldsymbol{\theta}_{n-1}). \tag{11}$$

When solving $\min_{\boldsymbol{\theta}} \; \mathbb{E}[\ell_n(\boldsymbol{\theta})]$ using a second-order SA iteration, a desirable Newton-like matrix step size is $\mathbf{M}_n = \mathbb{E}[\nabla^2\ell_n(\boldsymbol{\theta}_n)]$. Given that the latter requires knowing the average Hessian that is not available in practice, it is commonly surrogated by its sample-average $(1/n)\sum_{i=1}^{n}\nabla^2\ell_i(\boldsymbol{\theta}_i)$ [27, Sec. 2.1]. In this direction, note first that $\nabla^2\ell_n(\boldsymbol{\theta}) = \gamma_n(\boldsymbol{\theta})\mathbf{x}_n\mathbf{x}_n^T$, where

$$\gamma_n(\boldsymbol{\theta}) := \frac{(1-c_n)}{\sigma^2} + \frac{c_n}{\sigma^2}\left[\left(\frac{\phi\left(z_n^u(\boldsymbol{\theta})\right) - \phi\left(z_n^l(\boldsymbol{\theta})\right)}{Q\left(z_n^u(\boldsymbol{\theta})\right) - Q\left(z_n^l(\boldsymbol{\theta})\right)}\right)^2 - \frac{z_n^u(\boldsymbol{\theta})\phi\left(z_n^u(\boldsymbol{\theta})\right) - z_n^l(\boldsymbol{\theta})\phi\left(z_n^l(\boldsymbol{\theta})\right)}{Q\left(z_n^u(\boldsymbol{\theta})\right) - Q\left(z_n^l(\boldsymbol{\theta})\right)}\right]. \tag{12}$$

Due to the rank-one update $\mathbf{M}_n = ((n-1)/n)\mathbf{M}_{n-1} + (1/n)\gamma_{n-1}(\boldsymbol{\theta}_{n-1})\;\mathbf{x}_{n-1}\mathbf{x}_{n-1}^T$, the matrix step size $\mathbf{C}_n :=$

Algorithm 2 Second-order SA-MLE

```
Initialize $\boldsymbol{\theta}_1$ as the LSE $\hat{\boldsymbol{\theta}}_K$ in (3).
Initialize $\mathbf{C}_0 = \sigma^2(\mathbf{X}_K^T\mathbf{X}_K)^{-1}$.
for $n = 1 : D$ do
    Measurement $y_n$ is possibly censored using (4).
    Estimator receives $(z_n, \mathbf{x}_n, c_n)$.
    Compute $\gamma_n(\boldsymbol{\theta}_{n-1})$ from (12).
    Update matrix step size from (13).
    Update parameter estimate as in (11).
end for
```

$\mathbf{M}_n^{-1}$ can be obtained efficiently using the matrix inversion lemma as

$$\mathbf{C}_n = \frac{n}{n-1}\left(\mathbf{C}_{n-1} - \frac{\mathbf{C}_{n-1}\mathbf{x}_n\mathbf{x}_n^T\mathbf{C}_{n-1}}{(n-1)\gamma_n^{-1}(\boldsymbol{\theta}_{n-1}) + \mathbf{x}_n^T\mathbf{C}_{n-1}\mathbf{x}_n}\right). \tag{13}$$

Similar to its first-order counterpart, the algorithm is initialized by the preliminary estimate $\boldsymbol{\theta}_0 = \hat{\boldsymbol{\theta}}_K$, and $\mathbf{C}_0 = \sigma^2(\mathbf{X}_K^T\mathbf{X}_K)^{-1}$. The second-order SA-MLE method is summarized as Algorithm 2, while the numerical tests of Section V-A confirm its faster convergence at the cost of $\mathcal{O}(p^2)$ complexity per update. Since $\mathbf{M}_n$ is updated regardless whether datum n is censored or not, Algorithm 2 incurs the same complexity order as the ordinary RLS.

C. Controlling Data Reduction via NAC

To apply the NAC rule of (4) for data reduction at a controllable rate, a link between thresholds $\{\tau_n\}$ and the censoring rate must be established. Furthermore, prior knowledge of the problem at hand (e.g., observations likely to contain outliers) may dictate a specific pattern of censoring probabilities $\{\pi_n^*\}_{n=1}^D$. If d is the number of uncensored data after NAC is applied on a dataset of size $D \geq d$, then $(D-d)/D$ is the censoring ratio, a metric introduced in [23], [24]. Since $\{y_n\}$ are generated randomly according to (1), it is clear that d is itself a random variable. The analysis is thus focused on the average censoring ratio

$$\bar{c} := \mathbb{E}\left[\frac{D-d}{D}\right] = \frac{1}{D}\sum_{n=1}^{D}\mathbb{E}[c_n] = \frac{1}{D}\sum_{n=1}^{D}\pi_n \tag{14}$$

where $\pi_n := \Pr(c_n = 1)$ is the probability of censoring datum n, that as a function of τ_n is given by [cf. (4)]

$$\begin{aligned}\pi_n(\tau_n) &= \Pr\{-\tau_n\sigma \leq y_n - \hat{y}_n \leq \tau_n\sigma\}\\ &= \Pr\{-\tau_n \leq \frac{\mathbf{x}_n^T(\boldsymbol{\theta}_o - \hat{\boldsymbol{\theta}}_K) + v_n}{\sigma} \leq \tau_n\}.\end{aligned} \tag{15}$$

Using the properties of LSE, $\hat{\boldsymbol{\theta}}_K \sim \mathcal{N}(\boldsymbol{\theta}_o, \sigma^2(\mathbf{X}_K^T\mathbf{X}_K)^{-1})$, it follows that

$$\frac{\mathbf{x}_n^T(\boldsymbol{\theta}_o - \hat{\boldsymbol{\theta}}_K) + v_n}{\sigma} \sim \mathcal{N}\left(0, \mathbf{x}_n^T(\mathbf{X}_K^T\mathbf{X}_K)^{-1}\mathbf{x}_n + 1\right).$$

Thus, the censoring probabilities in (15) simplify to

$$\pi_n(\tau_n) = 1 - 2Q\left(\tau_n\left[\mathbf{x}_n^T(\mathbf{X}_K^T\mathbf{X}_K)^{-1}\mathbf{x}_n + 1\right]^{-1/2}\right). \tag{16}$$

Solving (16) for τ_n, one arrives for a given $\pi_n^\star = \pi_n(\tau_n^\star)$ at

$$\tau_n^\star = \left[\mathbf{x}_n^T(\mathbf{X}_K^T\mathbf{X}_K)^{-1}\mathbf{x}_n + 1\right]^{1/2} Q^{-1}\left(\frac{1-\pi_n^\star}{2}\right). \tag{17}$$

Hence, for a prescribed $\bar{c}$, one can select a desired censoring probability pattern $\{\pi_n^\star\}_{n=1}^D$ to satisfy (14), and corresponding $\{\tau_n^\star\}_{n=1}^D$ in accordance with (17).

The threshold selection in (17) requires knowledge of all $\{\mathbf{x}_n\}_{n=1}^D$. In addition, implementing (17) for all D observations, requires $\mathcal{O}(Dp^2)$ computations that may not be affordable for $D \gg p$. To deal with this, the ensuing simple threshold selection rule is advocated. Supposing that $\{\mathbf{x}_n\}_{n=1}^D$ are generated independently and identically distributed (i.i.d.) according to some unknown distribution with known first- and second-order moments, a relation between a target *common* censoring probability $\pi^\star$ and a common threshold τ can be obtained in closed form. Suppose without loss of generality that $\mathbb{E}[\mathbf{x}_n] = \mathbf{0}$, and let $\mathbb{E}\left[\mathbf{x}_n\mathbf{x}_n^T\right] = \mathbf{R}_x$ and $\boldsymbol{\zeta}_K := (\boldsymbol{\theta}_o - \hat{\boldsymbol{\theta}}_K)/\sigma \sim \mathcal{N}(\mathbf{0}, (\mathbf{X}_K^T\mathbf{X}_K)^{-1})$. For sufficiently large K, it holds that $(\mathbf{X}_K^T\mathbf{X}_K)^{-1} \approx \mathbf{R}_x^{-1}/K$, and thus $\boldsymbol{\zeta}_K \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_x^{-1}/K)$. Next, using the standardized Gaussian random vector $\mathbf{u} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_p)$, one can write $\boldsymbol{\zeta}_K = \mathbf{R}_x^{-1/2}\mathbf{u}/\sqrt{K}$. Also, with an independent zero-mean random vector $\mathbf{u}_n$ with $\mathbb{E}[\mathbf{u}_n\mathbf{u}_n^T] = \mathbf{I}_p$, it is also possible to express $\mathbf{x}_n = \mathbf{R}_x^{1/2}\mathbf{u}_n$, which implies $\mathbf{x}_n^T\boldsymbol{\zeta}_K = \mathbf{u}_n^T\mathbf{u}/\sqrt{K}$. By the central limit theorem, $\mathbf{u}_n^T\mathbf{u}$ converges in distribution to $\mathcal{N}(0, p)$ as the inner dimension of the two vectors p grows; thus, $\mathbf{x}_n^T\boldsymbol{\zeta}_K \sim \mathcal{N}(0, p/K)$. Under this approximation, it holds that

$$\begin{aligned}\pi_n \approx \pi &= Q\left(-\frac{\tau}{\sqrt{p/K+1}}\right) - Q\left(\frac{\tau}{\sqrt{p/K+1}}\right)\\ &= 1 - 2Q\left(\frac{\tau}{\sqrt{p/K+1}}\right), \quad n = 1, \ldots, D.\end{aligned} \tag{18}$$

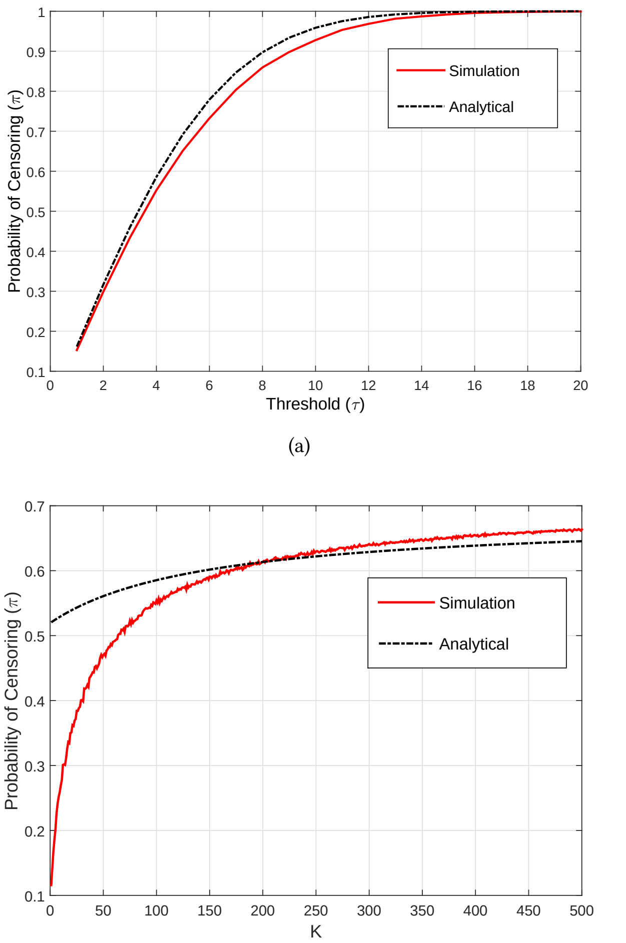
As expected, due to the normalization by σ in (4), π does not depend on σ. Interestingly, it does not depend on $\mathbf{R}_x$ either. Having expressed π as a function of τ, the latter can be tuned to achieve the desirable data reduction. Following the law of large numbers and given parameters p and K, to achieve an average censoring ratio of $\bar{c} = \pi^\star = (D-d)/D$, the threshold can be set to

$$\tau = \sqrt{1 + p/K}\, Q^{-1}\left(\frac{1-\pi^\star}{2}\right). \tag{19}$$

Figure 1(a) depicts π as a function of τ for $p = 100$ and $K = 200$. Function (18) is compared with the simulation-based estimate of π_n using 100 Monte Carlo runs, confirming that (18) offers a reliable approximation of π, which improves as p grows. However, for the approximation $(\mathbf{X}_K^T\mathbf{X}_K)^{-1} \approx \mathbf{R}_x^{-1}/K$ to be accurate, K should be large too. Figure 1(b) shows the probability of censoring for varying K with fixed $p = 100$ and $\tau = 1$. Approximation (18) yields a reliable value for π for as few as $K \approx 200$ preliminary data.

IV. Big Data Streaming Regression with AC

The algorithms devised and analyzed in Section III employ NAC rules. Data censoring there either occurred naturally,



(a)

(b)

Fig. 1. a) Censoring probability for varying threshold ($p = 100, K = 200$). b) Censoring probability for varying K ($p = 100, \tau = 1$).

or, it was introduced to reduce communication costs. This section employs data-adaptive censoring for data reduction. It should be understood that the NAC rule in (4) decouples censoring from estimation, whereas one intuitively expects improved performance with a *joint censoring-estimation* design. In this context, first- and second-order sequential algorithms are developed next for the AC rule of (5). Instead of $\hat{\boldsymbol{\theta}}_K$, censoring is here performed using the latest estimate of $\boldsymbol{\theta}$. Apart from being effective in handling streaming data, AC can markedly lower the complexity of a batch LS problem. Upon outlining an AC-based LMS algorithm, its RLS counterpart will be developed as a viable alternative to random projections and sampling.

A. AC-LMS

A first-order AC-based algorithm is presented here, inspired by the celebrated LMS algorithm. Originally developed for adaptive filtering, LMS is well motivated for low-complexity online estimation of (possibly slow-varying) parameters. Given $(y_n, \mathbf{x}_n)$, LMS entails the simple update

$$\boldsymbol{\theta}_n = \boldsymbol{\theta}_{n-1} + \mu\mathbf{x}_n e_n(\boldsymbol{\theta}_{n-1}) \tag{20}$$

where $e_n(\boldsymbol{\theta}) := y_n - \mathbf{x}_n^T\boldsymbol{\theta}$ can be viewed as the innovation of y_n, since $\hat{y}_n = \mathbf{x}_n^T\boldsymbol{\theta}_{n-1}$ is the prediction of y_n given $\boldsymbol{\theta}_{n-1}$. LMS can be regarded as an SGD method for minimizing $\mathbb{E}[f_n(\boldsymbol{\theta})]$ over $\boldsymbol{\theta}$ for the instantaneous costs $f_n(\boldsymbol{\theta}) = e_n^2(\boldsymbol{\theta})/2$.

To derive a first-order method for online censored regressions, consider minimizing $\mathbb{E}[f_n^{(\tau)}(\boldsymbol{\theta})]$ with the instantaneous cost selected as the *truncated* quadratic function

$$f_n^{(\tau)}(\boldsymbol{\theta}) := \begin{cases} \frac{1}{2}[e_n^2(\boldsymbol{\theta}) - \tau_n^2\sigma^2] & , \ |e_n(\boldsymbol{\theta})| \geq \tau_n\sigma \\ 0 & , \ |e_n(\boldsymbol{\theta})| < \tau_n\sigma \end{cases} \tag{21}$$

for a given $\tau_n > 0$. For the sake of analysis, a common threshold will be adopted; that is, $\tau_n = \tau \ \forall n$. The truncated cost can be also expressed as $f_n^{(\tau)}(\boldsymbol{\theta}) = \max\{0, (e_n^2(\boldsymbol{\theta}) - \tau^2\sigma^2)/2\}$. Being the pointwise maximum of two convex functions, $f_n^{(\tau)}(\boldsymbol{\theta})$ is convex, yet not everywhere differentiable. From standard rules of subdifferential calculus, its subgradient is

$$\partial f_n^{(\tau)}(\boldsymbol{\theta}) = \begin{cases} -\mathbf{x}_n e_n(\boldsymbol{\theta}) & , \ |e_n(\boldsymbol{\theta})| > \tau\sigma \\ \mathbf{0} & , \ |e_n(\boldsymbol{\theta})| < \tau\sigma \\ \{-\varphi\mathbf{x}_n e_n(\boldsymbol{\theta}) : 0 \leq \varphi \leq 1\} & , \ |e_n(\boldsymbol{\theta})| = \tau\sigma \end{cases} .$$

An SGD iteration for the instantaneous cost in (21) with $\tau_n = \tau$, performs the following AC-LMS update per datum n

$$\boldsymbol{\theta}_n := \begin{cases} \boldsymbol{\theta}_{n-1} + \mu\mathbf{x}_n e_n(\boldsymbol{\theta}_{n-1}) & , \ |e_n(\boldsymbol{\theta}_{n-1})| \geq \tau\sigma \\ \boldsymbol{\theta}_{n-1} & , \ \text{otherwise} \end{cases} \tag{22}$$

where $\mu > 0$ can be either constant for tracking a time-varying parameter, or, diminishing over time for estimating a time-invariant $\boldsymbol{\theta}_o$. Different from Alg. 1, AC-LMS does not update $\boldsymbol{\theta}$ if datum n is censored. The intuition is that if y_n can be closely predicted by $\hat{y}_n := \mathbf{x}_n^T\boldsymbol{\theta}_{n-1}$, then $(y_n, \mathbf{x}_n)$ can be censored; small innovation is indeed not much informative. Extracting interval information through a likelihood function as in Alg. 1 appears to be challenging here. This is because unlike NAC, the AC data $\{z_n\}_{n=1}^D$ are dependent across time.

Interestingly, upon invoking the so termed independent-data assumption of SA [26], following the same steps as in Section III, and substituting $\hat{\boldsymbol{\theta}}_K = \boldsymbol{\theta}_{n-1}$ into (9), the interval information term is eliminated. This is a strong indication that interval information from censored observations may be completely ignored without the risk of introducing bias. Indeed, one of the implications of the ensuing Proposition 2 is that the AC-LMS is asymptotically unbiased. Essentially, in AC-LMS as well as in the AC-RLS to be introduced later, both $\mathbf{x}_n$ and y_n are censored – an important feature effecting further data reduction and markedly lowering computational complexity of the proposed AC algorithms. A bound of the mean-square error (MSE) performance of AC-LMS is established in the next proposition proved in the Appendix.

Proposition 2. *Consider the observation model* (1)*, where* $\mathbf{x}_n$*'s are generated i.i.d. with* $\mathbb{E}[\mathbf{x}_n] = \mathbf{0}$*,* $\mathbb{E}\left[\mathbf{x}_n\mathbf{x}_n^T\right] = \mathbf{R}_x$*,* $\mathbb{E}\left[\|\mathbf{x}_n\|_2^3\right] = r_{3x}$*, and* $\mathbb{E}\left[\|\mathbf{x}_n\|_2^4\right] = r_{4x}$*. Assume that*

$$(A0) \quad \mathbb{E}_v\left[\|\boldsymbol{\zeta}_1\mathbb{1}_{\{|\mathbf{x}^T\boldsymbol{\zeta}_1+v|\geq\tau\sigma\}} - \boldsymbol{\zeta}_2\mathbb{1}_{\{|\mathbf{x}^T\boldsymbol{\zeta}_2+v|\geq\tau\sigma\}}\|_2^2\right] \leq \lambda_1(\tau)\|\boldsymbol{\zeta}_1 - \boldsymbol{\zeta}_2\|_2^2 \quad \forall\boldsymbol{\zeta}_1, \boldsymbol{\zeta}_2$$

$$(A1) \quad \mathbb{E}_v\left[v^2\left(\mathbb{1}_{\{|\mathbf{x}^T\boldsymbol{\zeta}_1+v|\geq\tau\sigma\}} - \mathbb{1}_{\{|\mathbf{x}^T\boldsymbol{\zeta}_2+v|\geq\tau\sigma\}}\right)^2\right] \leq \lambda_2(\tau)\|\boldsymbol{\zeta}_1 - \boldsymbol{\zeta}_2\|_2^2 \quad \forall\boldsymbol{\zeta}_1, \boldsymbol{\zeta}_2$$

and that the AC-LMS in (22) uses a fixed threshold τ and is initialized at $\boldsymbol{\theta}_1$. For a diminishing step size $\mu_n = 2/(\alpha n)$, the AC-LMS estimates $\boldsymbol{\theta}_n$ exhibit bounded MSE as

$$\mathbb{E}\left[\|\boldsymbol{\theta}_n - \boldsymbol{\theta}_o\|_2^2\right] \leq \frac{e^{4L^2/\alpha^2}}{n^2}\left(\|\boldsymbol{\theta}_1 - \boldsymbol{\theta}_o\|_2^2 + \frac{\Delta}{L^2}\right) + \frac{8\Delta \log n}{\alpha^2 n}$$

where $\alpha := 2Q(\tau)\lambda_{\min}(\mathbf{R}_x)$, $\Delta := 2\mathrm{tr}(\mathbf{R}_x)\sigma^2[1 - Q(\tau) + \tau\phi(\tau)]$, and $L^2 := r_{4x}\lambda_1(\tau) + \mathrm{tr}(\mathbf{R}_x)\lambda_2(\tau) + 2r_{3x}\sqrt{\lambda_1(\tau)\lambda_2(\tau)}$. Furthermore, for a constant μ, the estimates $\boldsymbol{\theta}_n$ converge exponentially to a bounded error as

$$\mathbb{E}\left[\|\boldsymbol{\theta}_n - \boldsymbol{\theta}_o\|_2^2\right] \leq 2\exp\left(-\left(\frac{\alpha\mu}{4} - 4L^2\mu^2\right)n - 4L^2\mu^2\right) \times \left(\|\boldsymbol{\theta}_1 - \boldsymbol{\theta}_o\|_2^2 + \frac{\Delta}{L^2}\right) + \frac{4\mu\Delta}{\alpha}$$

when $0 < \mu < \alpha/(16L^2)$.

Proposition 2 asserts that AC-LMS achieves a bounded MSE. It also links MSE to the censoring threshold τ, which can be used to adjust the censoring probability. Closer inspection reveals that the MSE bound decreases with τ. In par with intuition, lowering τ allows the estimator to access more data, thus enhancing estimation performance at the price of increasing the data volume processed.

Apparently, AC-LMS incurs the same order of complexity $\mathcal{O}(p)$ per iteration as the ordinary LMS. Thus, in a centralized estimation scenario, running AC-LMS rather than the ordinary LMS does not have a computational advantage. In a decentralized setup however, where data $(y_n, \mathbf{x}_n)$ are forwarded to a central agent, censoring via AC-LMS can significantly reduce the data forwarding overhead. With the central agent broadcasting $\boldsymbol{\theta}_n$'s to local agents, each local agent can separately decide whether its data should be censored or forwarded. This is a clear communication advantage of AC-LMS over Alg. 1, which presumed that $\mathbf{x}_n$'s are either transmitted to or they are known *a priori* by the central agent. Note also that $\boldsymbol{\theta}_n$ needs to be broadcast only when it is updated; hence, the central agent would be transmitting relatively infrequently. Beyond this decentralized setup, major computational savings can be harvested when introducing censoring into the RLS setup described next.

B. AC-RLS

A second-order AC algorithm is introduced here for sequential estimation and dimensionality reduction. It is closely related to the RLS algorithm, which per time n implements the updates; see e.g., [28]

$$\mathbf{C}_n = \frac{n}{n-1}\left[\mathbf{C}_{n-1} - \frac{\mathbf{C}_{n-1}\mathbf{x}_n\mathbf{x}_n^T\mathbf{C}_{n-1}}{n-1+\mathbf{x}_n^T\mathbf{C}_{n-1}\mathbf{x}_n}\right] \tag{23a}$$

$$\boldsymbol{\theta}_n = \boldsymbol{\theta}_{n-1} + \frac{1}{n}\mathbf{C}_n\mathbf{x}_n(y_n - \mathbf{x}_n^T\boldsymbol{\theta}_{n-1}) \tag{23b}$$

where $\mathbf{C}_n$ is the sample estimate for $\mathbf{R}_x^{-1}$, and it is typically initialized to $\mathbf{C}_0 = \epsilon\mathbf{I}$, for some small positive ϵ, e.g., [29]. The RLS estimate at time n can be also obtained as

$$\boldsymbol{\theta}_n = \arg\min_{\boldsymbol{\theta}} \sum_{i=1}^{n} (y_i - \mathbf{x}_i^T\boldsymbol{\theta})^2 + \epsilon\|\boldsymbol{\theta}\|_2^2. \tag{24}$$

Algorithm 3 Adaptive-Censoring (AC)-RLS

```
Initialize $\boldsymbol{\theta}_1 = \mathbf{0}$ and $\mathbf{C}_0 = \epsilon\mathbf{I}$.
for $n = 1 : D$ do
    if $|y_n - \mathbf{x}_n^T\boldsymbol{\theta}_{n-1}| \geq \tau\sigma$ then
        Estimator receives $(y_n, \mathbf{x}_n)$ while $c_n = 0$.
        Update inverse sample covariance from (25a).
        Update estimate from (25b).
    else
        Estimator receives no information ($c_n = 1$).
        Propagate inverse covariance as $\mathbf{C}_n = \frac{n}{n-1}\mathbf{C}_{n-1}$.
        Preserve estimate $\boldsymbol{\theta}_n = \boldsymbol{\theta}_{n-1}$.
    end if
end for
```

The bias introduced by the arbitrary choice of $\mathbf{C}_0$ vanishes asymptotically in n, while the RLS iterates converge to the batch LSE. RLS can be viewed as a second-order SGD method of the form $\boldsymbol{\theta}_n = \boldsymbol{\theta}_{n-1} - \mathbf{M}_n^{-1}\nabla f_n(\boldsymbol{\theta}_{n-1})$ for the quadratic cost $f_n(\boldsymbol{\theta}) = e_n^2(\boldsymbol{\theta})/2$. In this instance of SGD, the ideal matrix step size $\mathbf{M}_n := \mathbb{E}[\nabla^2 f_n(\boldsymbol{\theta}_{n-1})] = \mathbb{E}\left[(1-c_n)\mathbf{x}_n\mathbf{x}_n^T\right]$ is replaced by its running estimate $(1/n)\mathbf{C}_n^{-1}$; see e.g., [27].

To obtain a second-order counterpart of AC-LMS, we replace the quadratic cost of RLS with the truncated quadratic in (21). The matrix step-size is selected as

$$\begin{aligned}\mathbf{M}_n &= \frac{1}{n}\sum_{i=1}^{n}(1-c_i)\mathbf{x}_i\mathbf{x}_i^T \\ &= \frac{n-1}{n}\mathbf{M}_{n-1} + \frac{1}{n}(1-c_n)\mathbf{x}_n\mathbf{x}_n^T.\end{aligned}$$

Applying the matrix inversion lemma in finding $\mathbf{M}_n^{-1}$ yields the next AC-RLS updates

$$\mathbf{C}_n = \frac{n}{n-1}\left[\mathbf{C}_{n-1} - \frac{(1-c_n)\mathbf{C}_{n-1}\mathbf{x}_n\mathbf{x}_n^T\mathbf{C}_{n-1}}{n-1+\mathbf{x}_n^T\mathbf{C}_{n-1}\mathbf{x}_n}\right] \tag{25a}$$

$$\boldsymbol{\theta}_n = \boldsymbol{\theta}_{n-1} + \frac{1-c_n}{n}\mathbf{C}_n\mathbf{x}_n(y_n - \mathbf{x}_n^T\boldsymbol{\theta}_{n-1}) \tag{25b}$$

where c_n is decided by (5). When $c_n = 1$, not only the parameter vector is not updated, but costly updates of $\mathbf{C}_n$ are avoided too. In addition, different from the iterative expectation-maximization algorithm in [24], AC-RLS skips the covariance updates completely. Its performance is characterized by the following proposition proved in the Appendix.

Proposition 3. *If $\mathbf{x}_n$s are i.i.d. with $\mathbb{E}[\mathbf{x}_n] = \mathbf{0}$ and $\mathbb{E}\left[\mathbf{x}_n\mathbf{x}_n^T\right] := \mathbf{R}_x$, while observations y_n adhere to the model in (1), assuming $\{\mathbf{x}_n\mathbf{x}_n^T(1-c_n)\}$ is an ergodic process, for $\boldsymbol{\theta}_1 = \mathbf{0}$ and constant τ, there exists $k > 0$ such that AC-RLS estimates $\boldsymbol{\theta}_n$ yield bounded MSE*

$$\frac{1}{n}\mathrm{tr}\left(\mathbf{R}_x^{-1}\right)\sigma^2 \leq \mathbb{E}\left[\|\boldsymbol{\theta}_n - \boldsymbol{\theta}_o\|_2^2\right] \leq \frac{1}{n}\frac{\mathrm{tr}\left(\mathbf{R}_x^{-1}\right)\sigma^2}{2Q(\tau)}, \ \forall n \geq k.$$

As corroborated by Proposition 3, the AC-RLS estimates are guaranteed to converge to $\boldsymbol{\theta}_o$ for any choice of τ. Overall, the novel AC-RLS algorithm offers a computationally-efficient and accurate means of solving large-scale LS problems encountered with Big Data applications.

At this point, it is useful to contrast AC-RLS with RP and random sampling methods that have been advocated as fast LS solvers [30], [6]. In practice, RP-based schemes first premultiply data $(\mathbf{y}, \mathbf{X})$ with a random matrix $\mathbf{R} = \mathbf{HD}$, where $\mathbf{H}$ is a $D \times D$ Hadamard matrix and $\mathbf{D}$ is a diagonal matrix whose diagonal entries take values $\{-1/\sqrt{D}, +1/\sqrt{D}\}$ equiprobably. Intuitively, premultiplying by $\mathbf{R}$ yields measurements of approximately equal leverage scores (see [6], [7]), so that the ensuing random matrix $\mathbf{S}_d$ exhibits no preference in selecting uniformly a subset of d rows. Then, the reduced-size LS problem yields $\check{\boldsymbol{\theta}}_d = \arg\min_{\boldsymbol{\theta}} \|\mathbf{S}_d\mathbf{HD}(\mathbf{y} - \mathbf{X}\boldsymbol{\theta})\|_2^2$. For a general preconditioning matrix $\mathbf{HD}$, computing the product $\mathbf{HD}[\mathbf{y}\ \mathbf{X}]$ incurs prohibitive complexity of $\mathcal{O}(D^2p)$ computations. This is mitigated by choosing $\mathbf{H}$ to be the Hadamard matrix. Then, by using the fast Walsh-Hadamard transform one can reduce complexity of the sketch to $\mathcal{O}(Dp\log D)$, or, as low as $\mathcal{O}(Dp\log d)$, if one is only interested in d rows. Overall, the RP method reduces the computational complexity of LS from $\mathcal{O}(Dp^2)$ to $\mathcal{O}(Dp\log d + dp^2)$ operations.

By setting $\tau = Q^{-1}(d/(2D))$, our AC-RLS Algorithm 3 achieves an average reduction ratio d/D by scanning the observations, and selecting only the most informative ones. The same ratio can be achieved more accurately by choosing a sequence of data-adaptive thresholds $\{\tau_n\}_{n=1}^{D}$, as described in the next subsection. As will be seen in Section V-C, AC-RLS achieves significantly lower estimation error compared to RP-based solvers. Intuitively, this is because unlike RPs that are based solely on $\mathbf{X}$ and are thus *observation-agnostic*, AC extracts the most informative in terms of innovation subset of rows for a given problem instance $(\mathbf{y}, \mathbf{X})$.

Regarding the complexity of AC-RLS, if the pair $(y_n, \mathbf{x}_n)$ is not censored, the cost of updating $\boldsymbol{\theta}_n$ and $\mathbf{C}_n$ is $\mathcal{O}(p^2)$ multiplications. For a censored datum, there is no such cost. Thus, for d uncensored data the overall computational complexity is $\mathcal{O}(dp^2)$. Furthermore, evaluation of the absolute normalized innovation requires $\mathcal{O}(p)$ multiplications per iteration. Since this operation takes place at each of the D iterations, there are $\mathcal{O}(Dp)$ computations to be accounted for. Overall, AC-RLS reduces the complexity of LS from $\mathcal{O}(Dp^2)$ to $\mathcal{O}(Dp + dp^2)$. Evidently, the complexity reduction is more prominent for larger model dimension p. For $p \gg 1$, the first term may be neglected, yielding an $\mathcal{O}(dp^2)$ complexity for AC-RLS.

A couple of remarks are now in order.

Remark 1. The novel AC-LMS and AC-RLS algorithms bear structural similarities to sequential set-membership (SM)-based estimation [31], [32]. However, the model assumptions and objectives of the two are different. SM assumes that the noise distribution in (1) has bounded support, which implies that $\boldsymbol{\theta}_o$ belongs to a closed set. This set is sequentially identified by algorithms interpreted geometrically, while certain observations may be deemed redundant and thus discarded by the SM estimator. In our Big Data setup, an SA approach is developed to *deliberately* skip updates of low importance for reducing complexity regardless of the noise pdf.

Remark 2. Estimating regression coefficients relying on most informative data is reminiscent of support vector regression (SVR), which adopts an ϵ-insensitive cost (truncated ℓ_1 error norm). SVR has well-documented merits in robustness as well as generalization capability, both of which are attractive for (even nonlinear kernel-based) prediction tasks [33]. SVR solvers are typically based on nonlinear programming, and support vectors (SVs) are returned after *batch* processing. Inheriting the merits of SVRs, the novel AC-LMS and AC-RLS can be viewed as returning "causal SVs," which are different from the traditional (non-causal) batch SVs, but become available on-the-fly at complexity and storage requirements that are affordable for streaming Big Data. In fact, we conjecture that causal SVs returned by AC-RLS will approach their non-causal SVR counterparts if multiple passes over the data are allowed. Mimicking SVR costs, our AC-based schemes developed using the truncated ℓ_2 cost [cf. (21)] can be readily generalized to their counterparts based on the truncated ℓ_1 error norm.[1] Cross-pollinating in the other direction, our AC-RLS iterations can be useful for online learning from streaming large-scale data with second-order closed-form iterations.

C. Controlling Data Reduction via AC

A clear distinction between NAC and AC is that the latter depends on the estimation algorithm used. As a result, threshold design rules are estimation-driven rather than universal. In this section, threshold selection strategies are proposed for AC-RLS. Recall the average reduction ratio $\bar{c}$ in (14), and let $\boldsymbol{\zeta}_n := (\boldsymbol{\theta}_o - \boldsymbol{\theta}_n)/\sigma \sim \mathcal{N}(\mathbf{0}, \mathbf{K}_n)$ denote the normalized error at the $n-$th iteration. Similar to (16), it holds that

$$\pi_n(\tau_n) = 1 - 2Q\left(\tau_n\left[\mathbf{x}_n^T\mathbf{K}_{n-1}\mathbf{x}_n + 1\right]^{-1/2}\right). \qquad (26)$$

For $n \gg p$, estimates $\boldsymbol{\theta}_n$ are sufficiently close to $\boldsymbol{\theta}_o$ and thus $\mathbf{K}_n \approx \mathbf{0}$. Then, the data-agnostic $\tau_n \approx Q^{-1}(\frac{1-\pi_n}{2})$ attains an average censoring probability $\bar{\pi}$, while its asymptotic properties have been studied in [24]. For finite data, this simple rule leads to under-censoring by ignoring appreciable values of $\mathbf{K}_n$, which can increase computational complexity considerably. This consideration motivates well the data-adaptive threshold selection rules designed next.

AC-RLS updates can be seen as ordinary RLS updates on the subsequence of uncensored data. After ignoring the transient error due to initialization, it holds that $\mathbf{K}_n \approx \left[\sum_{i=1}^{n}(1 - c_i)\mathbf{x}_i\mathbf{x}_i^T\right]^{-1}$. The term $\mathbf{x}_n^T\mathbf{K}_{n-1}\mathbf{x}_n$ is encountered as $\mathbf{x}_n^T\mathbf{C}_{n-1}\mathbf{x}_n/n$ in the updates of Alg. 3, but it is not computed for censored measurements. Nonetheless, $\mathbf{x}_n^T\mathbf{C}_{n-1}\mathbf{x}_n/n$ can be obtained at the cost of $p(p+1)$ multiplications per censored datum. Then, the exact censoring probability at AC-RLS iteration n can be tuned to a prescribed $\pi_n^\star$ by selecting

$$\tau_n = \left(\mathbf{x}_n^T\mathbf{C}_{n-1}\mathbf{x}_n/n + 1\right)^{1/2} Q^{-1}\left(\frac{1 - \pi_n^\star}{2}\right). \qquad (27)$$

[1] In fact, any truncated error function taking value zero over the "dead zone" $[-\tau, \tau]$ will discard data leading to errors in $[-\tau, \tau]$, and thus reduce complexity of large-scale problems.

Given $\{\pi_n^\star\}_{n=1}^D$ satisfying (14), an average censoring ratio of $(D-d)/D$ is thus achieved in a controlled fashion.

Although lower than that of ordinary RLS, the complexity of AC-RLS using the threshold selection rule (27) is still $\mathcal{O}(Dp^2)$. To further lower complexity, a simpler rule is proposed that relies on averaging out the contribution of individual rows $\mathbf{x}_n^T$ in the censoring process. Suppose that $\mathbf{x}_n$'s are generated i.i.d. with $\mathbb{E}[\mathbf{x}_n] = \mathbf{0}$ and $\mathbb{E}[\mathbf{x}_n\mathbf{x}_n^T] = \mathbf{R}_x$. Similar to Section III-C, for p sufficiently large the inner product $\mathbf{x}_n^T\boldsymbol{\zeta}_n$ is approximately Gaussian. It then follows that the *a priori* error $e_n(\boldsymbol{\theta}_{n-1}) = \sigma\mathbf{x}_n^T\boldsymbol{\zeta}_{n-1} + v_n$ is zero-mean Gaussian with variance

$$\begin{aligned}\sigma_{e_n}^2 &= \sigma^2 \mathrm{tr}\left(\mathbb{E}\left[\mathbf{x}_n\mathbf{x}_n^T\boldsymbol{\zeta}_{n-1}\boldsymbol{\zeta}_{n-1}^T\right]\right) + \sigma^2\\ &= \sigma^2\mathrm{tr}\left(\mathbf{R}_x\mathbf{K}_{n-1}\right) + \sigma^2\end{aligned} \tag{28}$$

where the first equality follows from the independence of $\mathbf{x}_n^T\boldsymbol{\zeta}_{n-1}$ and v_n; and the third one from that of $\mathbf{x}_n$ with $\boldsymbol{\zeta}_{n-1}$. The censoring probability at time n is then expressed as

$$\pi_n = \Pr\{|e_n(\boldsymbol{\theta}_{n-1})| \leq \tau\sigma\} = 1 - 2Q\left(\tau_n\frac{\sigma}{\sigma_{e_n}}\right).$$

To attain $\pi_n^\star$, the threshold per datum n is selected as

$$\tau_n = \frac{\sigma_{e_n}}{\sigma}Q^{-1}\left(\frac{1-\pi_n^\star}{2}\right). \tag{29}$$

It is well known that for large n, the RLS error covariance matrix $\mathbf{K}_n$ converges to $(\sigma^2/n)\mathbf{R}_x^{-1}$. Specifying $\{\pi_n^\star\}_{n=1}^D$ is equivalent to selecting an average number of $\sum_{i=1}^n(1-\pi_i^\star)$ RLS iterations until time n. Thus, the AC-RLS with controlled selection probabilities yields an error covariance matrix $\mathbf{K}_n \approx \left(\sum_{i=1}^n(1-\pi_i^\star)\right)^{-1}\sigma^2\mathbf{R}_x^{-1}$. Combined with (28), the latter provides

$$\sigma_{e_n}^2 = \sigma^2 p\left(\sum_{i=1}^{n-1}(1-\pi_i^\star)\right)^{-1} + \sigma^2.$$

Plugging σ_{e_n} into (29) yields the simple threshold selection

$$\tau_n = \left[p\left(\sum_{i=1}^{n-1}(1-\pi_i^\star)\right)^{-1} + 1\right]^{1/2} Q^{-1}\left(\frac{1-\pi_n^\star}{2}\right). \tag{30}$$

Unlike (27), where thresholds are decided online at an additional computational cost, (30) offers an off-line threshold design strategy for AC-RLS. Based on (30), to achieve $\bar{c} = \pi^\star = (D-d)/D$, thresholds are chosen as

$$\tau_n = \left(\frac{p}{(n-1)(1-\pi^\star)} + 1\right)^{1/2} Q^{-1}\left(\frac{1-\pi^\star}{2}\right) \tag{31}$$

which attains a constant π^* across iterations.

D. Robust AC-LMS and AC-RLS

AC-LMS and AC-RLS were designed to adaptively select data with relatively large innovation. This is reasonable provided that (1) contains no outliers whose extreme values may give rise to large innovations too, and thus be mistaken for informative data. Our idea to gain robustness against outliers is to adopt the modified AC rule

$$(c_n, c_n^o) = \begin{cases}(1,0) & , |e_n(\boldsymbol{\theta}_{n-1})| < \sigma\tau\\ (0,0) & , \tau\sigma \leq |e_n(\boldsymbol{\theta}_{n-1})| < \tau_o\sigma\\ (0,1) & , |e_n(\boldsymbol{\theta}_{n-1})| \geq \tau_o\sigma\end{cases}. \tag{32}$$

Similar to (5), a nominal censoring variable c_n is activated here too for observations with absolute normalized innovation less than τ. To reveal possible outliers, a second censoring variable c_n^o is triggered when the absolute normalized innovation exceeds threshold τ_o with $\tau_o > \tau$.

Having separated data-censoring from outlier identification in (32), it becomes possible to robustify AC-LMS and AC-RLS against outliers. Toward this end, one approach is to completely ignore y_n when $c_n^o = 1$. Alternatively, the instantaneous cost function in (21) can be modified to a truncated Huber loss (cf. [34])

$$f^o(e_n) = \begin{cases}0 & , (c_n, c_n^o) = (1,0)\\ \left(\frac{1}{2}e_n^2 - \frac{1}{2}\tau^2\sigma^2\right) & , (c_n, c_n^o) = (0,0)\\ \tau_o\sigma\left(|e_n| - \frac{3}{2}\tau_o^2\sigma^2 - \frac{1}{2}\tau^2\sigma^2\right) & , (c_n, c_n^o) = (0,1)\end{cases}.$$

Applying the first-order SGD iteration on the cost $f^o(e_n)$, yields the robust (r) AC-LMS iteration

$$\boldsymbol{\theta}_n = \boldsymbol{\theta}_{n-1} + \mu_n\mathbf{g}_n(\boldsymbol{\theta}_{n-1}) \tag{33}$$

where the gradient vector is now provided as

$$\mathbf{g}_n(\boldsymbol{\theta}) = \begin{cases}\mathbf{0} & , \ (c_n, c_n^o) = (1,0)\\ \mathbf{x}_n(y_n - \mathbf{x}_n^T\boldsymbol{\theta}) & , \ (c_n, c_n^o) = (0,0)\\ \tau_o\sigma\mathbf{x}_n \,\mathrm{sign}\left(y_n - \mathbf{x}_n^T\boldsymbol{\theta}\right) & , \ (c_n, c_n^o) = (0,1)\end{cases}.$$

Similarly, the second-order SGD yields the rAC-RLS

$$\boldsymbol{\theta}_n = \boldsymbol{\theta}_{n-1} + \frac{1}{n}\mathbf{C}_n\mathbf{g}_n(\boldsymbol{\theta}_{n-1}) \tag{34a}$$

$$\mathbf{C}_n = \frac{n}{n-1}\left[\mathbf{C}_{n-1} - \frac{(1-c_n)(1-c_n^o)\mathbf{C}_{n-1}\mathbf{x}_n\mathbf{x}_n^T\mathbf{C}_{n-1}}{n-1+\mathbf{x}_n^T\mathbf{C}_{n-1}\mathbf{x}_n}\right]. \tag{34b}$$

Observe that when $c_n^o = 1$, only $\boldsymbol{\theta}_n$ is updated, and the computationally costly update of (34b) is avoided.

V. Numerical Tests

A. SA-MLE

The online SA-MLE algorithms presented in Section III are simulated using Gaussian data generated according to (1) with a time-invariant $\boldsymbol{\theta}_o \in \mathbb{R}^p$, where $p = 30$, $v_n \sim \mathcal{N}(0,1)$ and $\mathbf{x}_n \sim \mathcal{N}(\mathbf{0}_p, \mathbf{I}_p)$. The first $K = 50$ observations are used to compute $\hat{\boldsymbol{\theta}}_K$. The first-and second-order SA-MLE algorithms are then run for $D = 5,000$ time steps. The NAC rule in (4) was used with $\tau = 1.5$ to censor approximately 75% of the observations. To achieve a desirable tradeoff between convergence speed and asymptotic error performance, a diminishing step size $\mu_n = 0.1/\sqrt{n}$ was used for the SA-MLE. Plotted in Fig. 2 is the MSE $\mathbb{E}\left[\|\boldsymbol{\theta}_o - \hat{\boldsymbol{\theta}}_n\|_2^2\right]$ across time n, approximated by averaging over 100 Monte Carlo experiments. Also plotted is the Cramer-Rao lower bound (CRLB) of the observations, given by modifying the results of [23] to accommodate the NAC rule in (4). It can

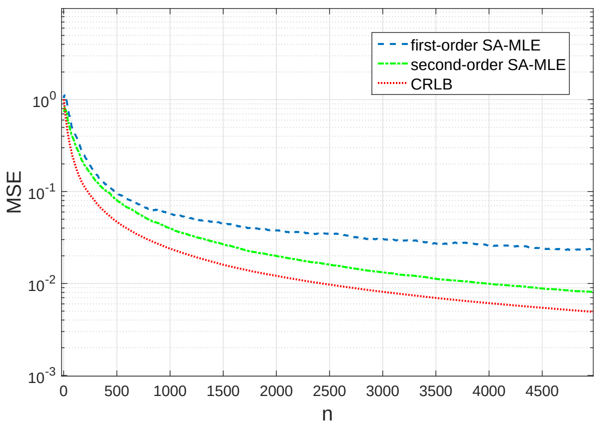


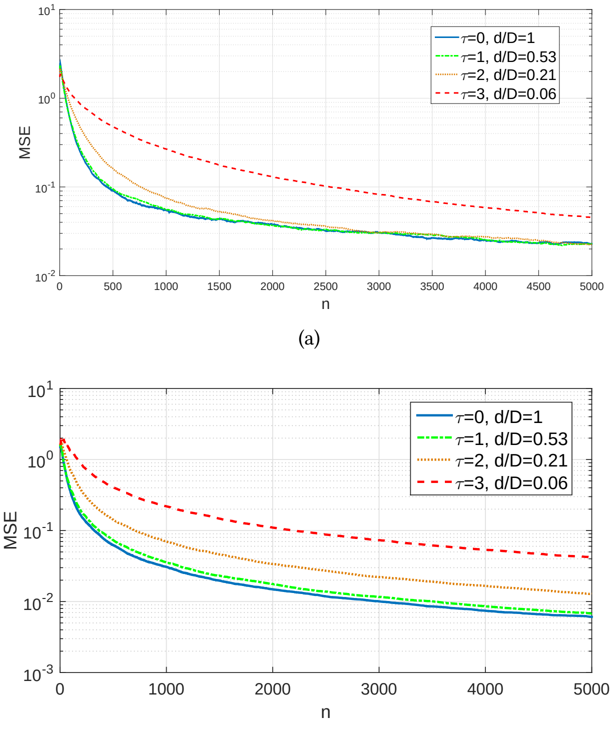
Fig. 2. Convergence of first- and second-order SA-MLE $(d/D = 0.25)$.

be inferred from the plot that the second-order SA-MLE exhibits markedly improved convergence rate compared to its first-order counterpart, at the price of minor increase in complexity. Furthermore, by performing a single pass over the data, the second-order SA-MLE performs close to the CRLB, thus offering an attractive alternative to the more computationally demanding batch Newton-based iterations in [23] and [24].

To further evaluate the efficacy of the proposed methods, additional simulations were run for different levels of censoring by adjusting τ. Plotted in Figs. 3(a) and 3(b) are the MSE curves of the first- and second-order SA-MLE respectively, for different values of τ. Notice that censoring up to 50% of the data incurs negligible estimation error compared to the full-data case (blue solid curve). In fact, even when operating on data reduced by 95% (red dashed curve) the proposed algorithms yield reliable online estimates.

B. AC-LMS

The AC-LMS algorithm introduced in Section IV-A was tested on synthetic data and compared to the normalized LMS (Kaczmarz) and the randomized Kaczmarz algorithm [12]. For this experiment, $D = 3{,}000$ observations y_n were generated as in (1) with $\sigma^2 = 1$, while $\mathbf{x}_n$'s of dimension $p = 200$ were generated i.i.d. following a standardized multivariate Gaussian distribution. For the randomized Kaczmarz algorithm, the probability of selecting the n-th row is $p_n = \|\mathbf{x}_n\|_2^2/\|\mathbf{X}\|_F^2$ [12]. In this experiment, $\mathbf{x}_n$'s were scaled with random weights to provide a favorable case for this magnitude-based random sampling. For AC-LMS, the step-size was set to $\mu = 0.004$ and the censoring threshold to $\tau = 2.3$ yielding a compression ratio of $d/D \approx 0.25$. Plotted in Fig. 4, are the relative MSE $\left(\mathbb{E}\left[\|\boldsymbol{\theta}_o - \hat{\boldsymbol{\theta}}_n\|_2^2/\|\boldsymbol{\theta}_o\|_2^2\right]\right)$ curves of the three algorithms w.r.t. n, averaged over 50 Monte Carlo runs. Interestingly, even after performing only 25% of the updates, AC-LMS achieves convergence speed and steady state error comparable to that of randomized Kaczmarz. Although AC-LMS does not provide major computational savings, a significant number of updates can be skipped thus harvesting communication savings in a decentralized setting.



(a)

(b)

Fig. 3. Convergence of (a) first-order SA-MLE; and (b) second-order SA-MLE for different values of τ.

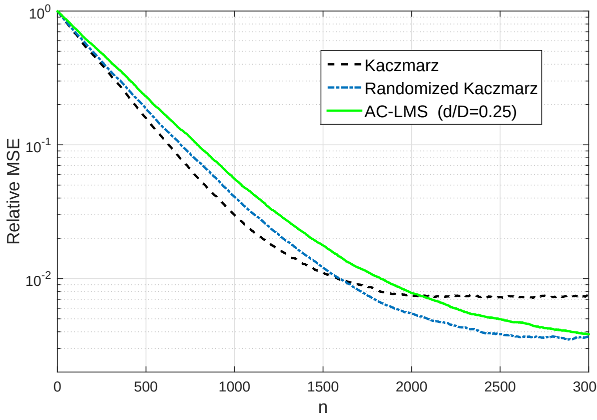


Fig. 4. Relative MSE for Kaczmarz, randomized Kaczmarz, and AC-LMS. AC-LMS used approximately 25% of the data (500 updates out of $3{,}000$ data).

C. AC-RLS

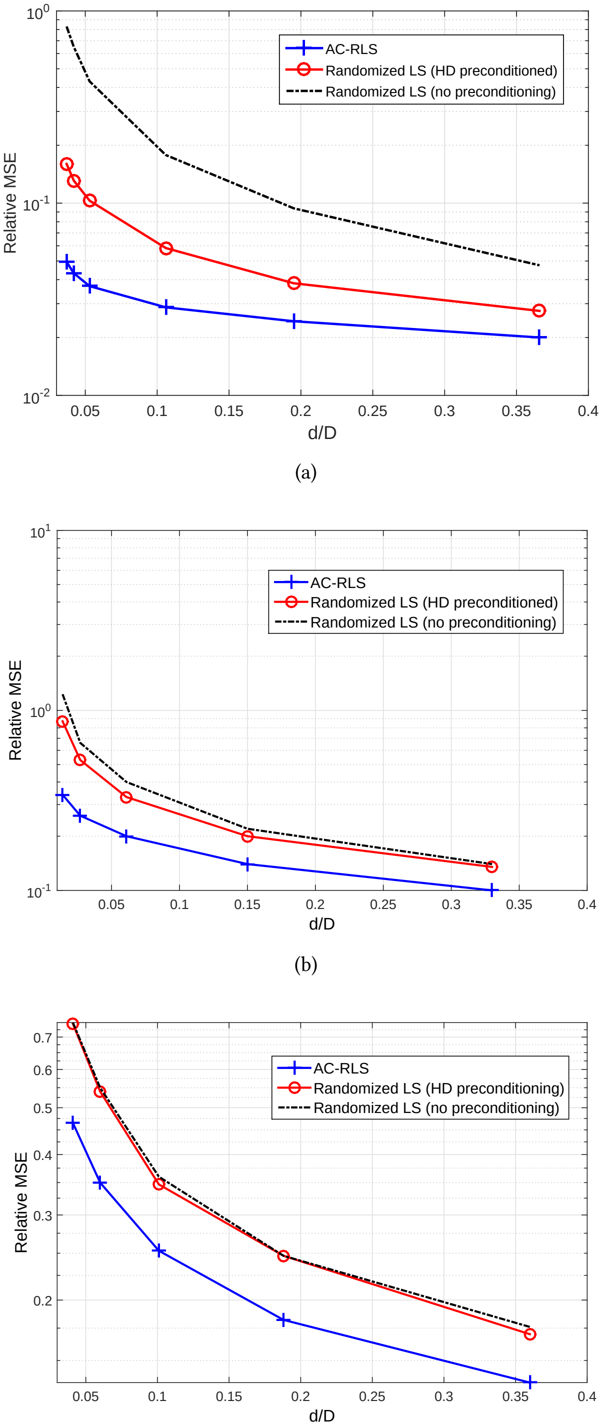
The AC-RLS algorithm developed in Section IV-B was tested on synthetic data. Specifically, the AC-RLS is treated here as an iterative method that sweeps once through the entire dataset, even though more sweeps can be performed at the cost of additional runtime. Its performance in terms of relative MSE was compared with the Hadamard (HD) preconditioned randomized LS solver, while plotted as a function of the compression ratio d/D. Parallel to the two methods, a uniform sampling randomized LSE was run as a simple benchmark. Measurements were generated according to (1) with $p = 300$, $D = 10{,}000$, and $v_n \sim \mathcal{N}(0, 9)$. Regarding the data distribution, three different scenario's were examined. In

Figure 5(a), $\mathbf{x}_n$'s were generated according to a heavy tailed multivariate $t-$distribution with one degree of freedom, and covariance matrix with (i,j)-th entry $\mathbf{\Sigma}_{i,j} = 2 \times 0.5^{|i-j|}$. Such a data distribution yields matrices $\mathbf{X}$ with highly non-uniform leverage scores, thus imitating the effect of a subset of highly "important" observations randomly scattered in the dataset. In such cases, uniform sampling without preconditioning performs poorly since many of those informative measurements are missed. As seen in the plot, preconditioning significantly improves performance, by incorporating "important" information through random projections. Further improvement is effected by our data-driven AC-RLS through adaptively selecting the most informative measurements and ignoring the rest, without overhead in complexity.

The experiment was repeated (Fig. 5(b)) for $\mathbf{x}_n$ generated from a multivariate $t-$distribution with 3 degrees of freedom, and $\mathbf{\Sigma}$ as before. Leverage scores for this dataset are moderately non-uniform, thus inducing more redundancy and resulting in lower performance for all algorithms, while closing the "gap" between preconditioned and non-preconditioned random sampling. Again, the proposed AC-RLS performs significantly better in estimating the unknown parameters for the entire range of data size reduction.

Finally, Fig. 5(c) depicts related performance for Gaussian $\mathbf{x}_n \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma})$. Compared to the previous cases, normally distributed rows yield a highly redundant set of measurements with $\mathbf{X}$ having almost uniform leverage scores. As seen in the plots, preconditioning offers no improvement in random sampling for this type data, whereas the AC-RLS succeeds in extracting more information on the unknown $\boldsymbol{\theta}$.

The efficacy of AC-RLS was further assessed using a real dataset regarding the physicochemical properties of protein tertiary structures [35]. In this linear regression dataset, $p = 9$ attributes of proteins are used to predict a value related to protein structure. A total of $D = 45,730$ observations are included. Data $(y_n, \mathbf{x}_n)$ were centered around the origin by compensating for their empirical mean. Since the true $\boldsymbol{\theta}_o$ is unknown, it is estimated by solving LS on the entire dataset. Subsequently, the noise variance is also estimated via sample averaging as $\sigma^2 = (D-1)^{-1}\sum_{n=1}^{D}(y_n - \mathbf{x}_n^T\boldsymbol{\theta}_o)^2$. In a realistic scenario where $\boldsymbol{\theta}_o$ is unknown, the variance σ^2 can be estimated recursively as $\hat{\sigma}_n^2 = \frac{n-1}{n}\hat{\sigma}_{n-1}^2 + \frac{1}{n}(y_n - \mathbf{x}_n^T\boldsymbol{\theta}_{n-1})^2$. Figure 6 depicts the relative squared error (RSE) with respect to the data reduction ratio d/D. The RSE curve for the HD-preconditioned LS corresponds to the RSE averaged over 50 runs, while the size of the vertical bars is proportional to its standard deviation. Different from RP-based methods, the RSE for AC-RLS does not entail standard deviation bars, because the algorithm output is deterministic for a given initialization and data order. It can be observed that for $d/D \geq 0.25$, the AC-RLS outperforms RPs in terms of estimating $\boldsymbol{\theta}$; while for very small d/D, RPs yield a lower average RSE, at the cost however of very high error uncertainty (variance). Heed that the dataset deviates from the assumed model; which demonstrates that AC-RLS is robust to model mismatch.



(a)

(b)

(c)

Fig. 5. Relative MSE of AC-RLS and randomized LS algorithms, for different levels of data reduction. Regression matrix $\mathbf{X}$ was generated with highly non-uniform (a), moderately non-uniform (b), and uniform leverage scores (c).

D. Robust AC-RLS

To test rAC-LMS and rAC-RLS of Section IV-D, datasets were generated with $D = 10,000$, $p = 30$ and $\mathbf{x}_n \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma})$, where $\mathbf{\Sigma}_{i,j} = 2 \times 0.5^{|i-j|}$; noise was i.i.d. Gaussian $v_n \sim \mathcal{N}(0, 9)$; meanwhile measurements y_n were generated

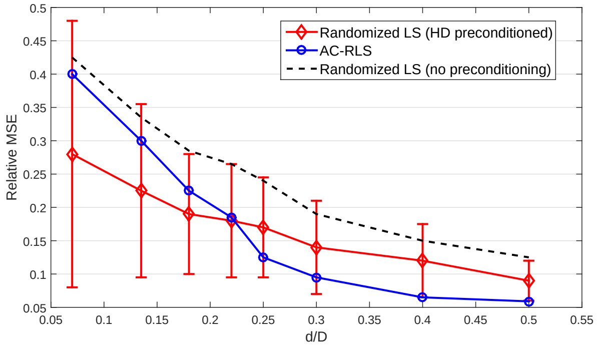


Fig. 6. Relative MSE of AC-RLS and randomized LS algorithms, for different levels of data reduction using the protein tertiary structure dataset.

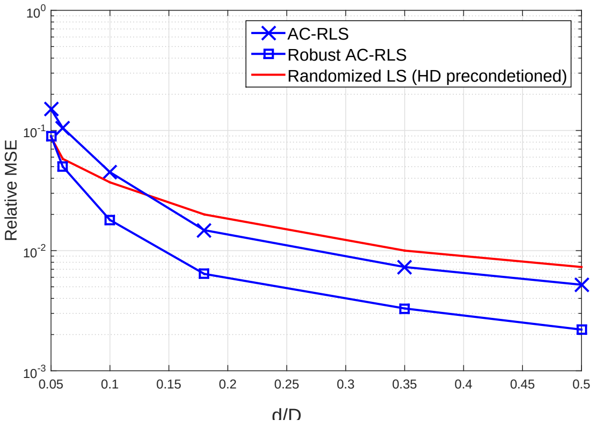


Fig. 7. Relative MSE of AC-RLS, rAC-RLS, and randomized LS algorithms, for different levels of data reduction using an outlier-corrupted dataset.

according to (1) with random and sporadic outlier spikes $\{o_n\}_{n=1}^D$. Specifically, we generated $o_n = \alpha_n \beta_n$, where $\alpha_n \sim \text{Bernoulli}(0.05)$, and $\beta_n \sim \mathcal{N}(0, 25 \times 9)$, thus resulting in approximately 5% of the data effectively being outliers. Similar to previous experiments, our novel algorithms were run once through the set selecting d out of D data to update $\boldsymbol{\theta}_n$. Plotted in Fig. 7 is the RSE averaged across 100 runs as a function of d/D for the HD-preconditioned LS, the plain AC-RLS, and the rAC-RLS with a Huber-like instantaneous cost. As expected, the performance of AC-RLS is severely undermined especially when tuned for very small d/D, exhibiting higher error than the RP-based LS. However, our rAC-RLS algorithm offers superior performance across the entire range of d/D values.

VI. Concluding Remarks

We developed online algorithms for large-scale LS linear regressions that rely on censoring for data-driven dimensionality reduction of streaming Big Data. First, a non-adaptive censoring setting was considered for applications where observations are censored – possibly naturally – and prior to estimation. Computationally efficient first- and second-order online algorithms were derived to estimate the unknown parameters, relying on stochastic approximation of the log-likelihood of the censored data. Performance was bounded analytically, while simulations demonstrated that the second-order method performs close to the CRLB.

Online data reduction occurring parallel to estimation was also explored. For this scenario, censoring is performed deliberately and adaptively based on estimates provided by first- and second-order algorithms. Robust versions were also developed for estimation in the presence of outliers. Introduced within the framework of stochastic approximation, the proposed algorithms were shown to enjoy guaranteed MSE performance. Moreover, the resulting recursive methods were advocated as low-complexity recursive solvers of large LS problems. Experiments run on synthetic and real datasets corroborated that the novel AC-LMS and AC-RLS algorithms outperformed competing randomized algorithms.

Our future research agenda includes approaches to nonlinear (e.g., kernel-based) (non)parametric large-scale regressions, along with estimation of dynamical (e.g., state-space) processes using adaptively censored measurements.

Appendix

Proof of Proposition 1: It can be verified that $\nabla^2 \ell_n(\boldsymbol{\theta}) \succeq \mathbf{0}$, which implies the convexity of $\ell_n(\boldsymbol{\theta})$ [23]. The regret of the SGD approach is then bounded as [15, Corollary 2.7]

$$\begin{aligned} R(D) &\leq \frac{1}{2\mu}\|\boldsymbol{\theta}^* - \boldsymbol{\theta}_1\|_2^2 + \mu \sum_{n=1}^{D} \|\nabla \ell_n(\boldsymbol{\theta}_{n-1})\|_2^2 \\ &= \frac{1}{2\mu}\|\boldsymbol{\theta}^* - \hat{\boldsymbol{\theta}}_K\|_2^2 + \mu \sum_{n=1}^{D} \|\mathbf{x}_n\|_2^2 \beta^2(\boldsymbol{\theta}_{n-1}) \\ &\leq \frac{1}{2\mu}\|\boldsymbol{\theta}^* - \hat{\boldsymbol{\theta}}_K\|_2^2 + \mu D (\bar{x}\bar{\beta})^2 \\ &\leq \frac{1}{2\mu} A^2 + \mu D (\bar{x}\bar{\beta})^2 \end{aligned}$$

where $\{\boldsymbol{\theta}_n\}_{n=1}^D$ is any sequence of estimates produced by the SA-MLE. Setting $\mu = \mu^* = A/(\sqrt{2D}\bar{\beta}\bar{x})$ minimizes the aforementioned upper bound. Otherwise, selecting $\mu = c\mu^*$ for some $c > 0$ readily provides the bound of Proposition 1 since $2(c + 1/c) \geq \max\{c, 1/c\}$. ■

Proof of Proposition 2: For the SGD update in (22), it is shown in [36] that by choosing a diminishing step size $\mu_n = C/n$, the MSE $\mathbb{E}_{\mathbf{x},v}\left[\|\boldsymbol{\theta}_n - \boldsymbol{\theta}_o\|_2^2\right]$, with $\boldsymbol{\theta}_o = \arg\min_{\boldsymbol{\theta}} F(\boldsymbol{\theta})$ and $F(\boldsymbol{\theta}) := \mathbb{E}_{\mathbf{x},v}\left[f^{(\tau)}(\boldsymbol{\theta}; \mathbf{y})\right]$ can be bounded as

$$\begin{aligned} \mathbb{E}\left[\|\boldsymbol{\theta}_n - \boldsymbol{\theta}_o\|_2^2\right] \leq \frac{e^{2L^2C^2}}{n^{\alpha C}} \left(\|\boldsymbol{\theta}_1 - \boldsymbol{\theta}_o\|_2^2 + \frac{\Delta}{L^2}\right) \\ + 2\frac{\Delta C^2 \omega_{\alpha C/2-1}(n)}{n^{\alpha C/2}} \end{aligned}$$

where $\omega_x(n) = \log n$ for $x = 0$. After setting $C = 2/\alpha$, the first bound in Proposition 2 is obtained with the parameters α, Δ, and L being identified as detailed next.

For the above bounds to hold, it is necessary to have: a1) the function gradient to be bounded at the optimum, i.e., $\mathbb{E}_{\mathbf{x},v}\left[\|\nabla f^{(\tau)}(\boldsymbol{\theta}_o, \mathbf{y})\|_2^2\right] \leq \Delta$; a2) the gradient to be $L-$smooth for any other $\boldsymbol{\theta}$; and, a3) $F(\boldsymbol{\theta})$ to be $\alpha-$strongly convex [36]. Note that for i.i.d. $\{(\mathbf{x}_n, v_n)\}$, α, Δ and L do not depend on n. Also, discontinuity points of $f^{(\tau)}(.)$ are zero-measure in expectation, and hence can be neglected.

Starting with a3), interchanging differentiation with expectation yields

$$\begin{aligned}\nabla^2 F(\boldsymbol{\theta}) &= \nabla^2_{\boldsymbol{\theta}} \mathbb{E}_{\mathbf{x},v}\left[f^{(\tau)}(\boldsymbol{\theta};\mathbf{x},v)\right]\\ &= \mathbb{E}_{\mathbf{x},v}\left[\nabla^2_{\boldsymbol{\theta}}\frac{e^2}{2}(1-c)\right] = \mathbb{E}_{\mathbf{x},v}\left[\mathbf{x}\mathbf{x}^T(1-c)\right]\\ &= \mathbb{E}_{\mathbf{x}}\left[\mathbf{x}\mathbf{x}^T\mathbb{E}_v\left[\mathbb{1}_{\{|\mathbf{x}^T(\boldsymbol{\theta}_o-\boldsymbol{\theta})+v|\geq\tau\sigma\}}\right]\right]\\ &= \mathbb{E}_{\mathbf{x}}\left[\mathbf{x}\mathbf{x}^T\Pr\left\{|\mathbf{x}^T(\boldsymbol{\theta}_o-\boldsymbol{\theta})+v|\geq\tau\sigma\right\}\right]\\ &= \mathbb{E}_{\mathbf{x}}\left[\mathbf{x}\mathbf{x}^T\left[Q\left(\tau+\frac{\mathbf{x}^T(\boldsymbol{\theta}_o-\boldsymbol{\theta})}{\sigma}\right)\right.\right.\\ &\qquad\left.\left.+\,Q\left(\tau-\frac{\mathbf{x}^T(\boldsymbol{\theta}_o-\boldsymbol{\theta})}{\sigma}\right)\right]\right].\end{aligned}$$

Observe that the function $g(z) := Q(\tau+z)+Q(\tau-z)$ is minimized at $z=0$ when $\tau>0$. To see this, heed that its derivative $g'(z) = -\phi(\tau+z)+\phi(\tau-z)$ vanishes when $|\tau+z| = |\tau-z|$. Thus, $g(z)\geq g(0) = 2Q(\tau)$ for all z, or

$$Q\left(\tau+\frac{\mathbf{x}^T(\boldsymbol{\theta}_o-\boldsymbol{\theta})}{\sigma}\right)+Q\left(\tau-\frac{\mathbf{x}^T(\boldsymbol{\theta}_o-\boldsymbol{\theta})}{\sigma}\right)\geq 2Q(\tau)$$

for all $\mathbf{x}$ and $\boldsymbol{\theta}$. The latter implies that $\nabla^2 F(\boldsymbol{\theta}) \succeq 2Q(\tau)\mathbb{E}_{\mathbf{x}}\left[\mathbf{x}\mathbf{x}^T\right]\succeq 2Q(\tau)\lambda_{\min}(\mathbf{R}_x)\mathbf{I}$ showing that function $F(\boldsymbol{\theta})$ is $\alpha-$strongly convex with $\alpha = 2Q(\tau)\lambda_{\min}(\mathbf{R}_x)$. As expected, α reduces for increasing τ.

Regarding the instantaneous gradient, it suffices to find L such that $\mathbb{E}_{\mathbf{x},v}\left[\|\nabla f^{(\tau)}(\boldsymbol{\theta}_1)-\nabla f^{(\tau)}(\boldsymbol{\theta}_2)\|_2^2\right] \leq L^2\|\boldsymbol{\theta}_1-\boldsymbol{\theta}_2\|_2^2$ for all n and $(\boldsymbol{\theta}_1,\boldsymbol{\theta}_2)$. Concerning the errors $\boldsymbol{\zeta}_i := \boldsymbol{\theta}_o-\boldsymbol{\theta}_i$ for $i=1,2$, it holds

$$\begin{aligned}&\mathbb{E}_{\mathbf{x},v}\left[\|\nabla f^{(\tau)}(\boldsymbol{\theta}_1)-\nabla f^{(\tau)}(\boldsymbol{\theta}_2)\|_2^2\right]\\ &= \mathbb{E}_{\mathbf{x},v}\left[\|\mathbf{x}e(\boldsymbol{\theta}_1)(1-c_1)-\mathbf{x}e(\boldsymbol{\theta}_2)(1-c_2)\|_2^2\right]\\ &= \mathbb{E}_{\mathbf{x},v}\left[\|\mathbf{x}(\mathbf{x}^T\boldsymbol{\zeta}_1+v)\mathbb{1}_{\{|\mathbf{x}^T\boldsymbol{\zeta}_1+v|\geq\tau\sigma\}}\right.\\ &\qquad\left.-\,\mathbf{x}(\mathbf{x}^T\boldsymbol{\zeta}_2+v)\mathbb{1}_{\{|\mathbf{x}^T\boldsymbol{\zeta}_2+v|\geq\tau\sigma\}}\|_2^2\right]\\ &= \mathbb{E}_{\mathbf{x},v}\big[\|\mathbf{x}\mathbf{x}^T\boldsymbol{\zeta}_1\mathbb{1}_{\{|\mathbf{x}^T\boldsymbol{\zeta}_1+v|\geq\tau\sigma\}}-\mathbf{x}\mathbf{x}^T\boldsymbol{\zeta}_2\mathbb{1}_{\{|\mathbf{x}^T\boldsymbol{\zeta}_2+v|\geq\tau\sigma\}}\\ &\qquad+\mathbf{x}v(\mathbb{1}_{\{|\mathbf{x}^T\boldsymbol{\zeta}_1+v|\geq\tau\sigma\}}-\mathbb{1}_{\{|\mathbf{x}^T\boldsymbol{\zeta}_2+v|\geq\tau\sigma\}})\|_2^2\big]\\ &\leq \mathbb{E}_{\mathbf{x},v}\big[\|\mathbf{x}\mathbf{x}^T(\boldsymbol{\zeta}_1\mathbb{1}_{\{|\mathbf{x}^T\boldsymbol{\zeta}_1+v|\geq\tau\sigma\}}-\boldsymbol{\zeta}_2\mathbb{1}_{\{|\mathbf{x}^T\boldsymbol{\zeta}_2+v|\geq\tau\sigma\}})\|_2^2\\ &\qquad+\|\mathbf{x}v(\mathbb{1}_{\{|\mathbf{x}^T\boldsymbol{\zeta}_1+v|\geq\tau\sigma\}}-\mathbb{1}_{\{|\mathbf{x}^T\boldsymbol{\zeta}_2+v|\geq\tau\sigma\}})\|_2^2\\ &\qquad+2\|\mathbf{x}\mathbf{x}^T(\boldsymbol{\zeta}_1\mathbb{1}_{\{|\mathbf{x}^T\boldsymbol{\zeta}_1+v|\geq\tau\sigma\}}-\boldsymbol{\zeta}_2\mathbb{1}_{\{|\mathbf{x}^T\boldsymbol{\zeta}_2+v|\geq\tau\sigma\}})\|_2\\ &\qquad\times\|\mathbf{x}v(\mathbb{1}_{\{|\mathbf{x}^T\boldsymbol{\zeta}_1+v|\geq\tau\sigma\}}-\mathbb{1}_{\{|\mathbf{x}^T\boldsymbol{\zeta}_2+v|\geq\tau\sigma\}})\|_2\big]\\ &\leq \mathbb{E}_{\mathbf{x},v}\big[\lambda_{\max}((\mathbf{x}\mathbf{x}^T)^2)\|\boldsymbol{\zeta}_1\mathbb{1}_{\{|\mathbf{x}^T\boldsymbol{\zeta}_1+v|\geq\tau\sigma\}}-\boldsymbol{\zeta}_2\mathbb{1}_{\{|\mathbf{x}^T\boldsymbol{\zeta}_2+v|\geq\tau\sigma\}}\|_2^2\\ &\qquad+\|\mathbf{x}\|_2^2v^2(\mathbb{1}_{\{|\mathbf{x}^T\boldsymbol{\zeta}_1+v|\geq\tau\sigma\}}-\mathbb{1}_{\{|\mathbf{x}^T\boldsymbol{\zeta}_2+v|\geq\tau\sigma\}})^2\\ &\qquad+2\lambda_{\max}(\mathbf{x}\mathbf{x}^T)\|\boldsymbol{\zeta}_1\mathbb{1}_{\{|\mathbf{x}^T\boldsymbol{\zeta}_1+v|\geq\tau\sigma\}}-\boldsymbol{\zeta}_2\mathbb{1}_{\{|\mathbf{x}^T\boldsymbol{\zeta}_2+v|\geq\tau\sigma\}}\|_2\\ &\qquad\times\|\mathbf{x}\|_2|v(\mathbb{1}_{\{|\mathbf{x}^T\boldsymbol{\zeta}_1+v|\geq\tau\sigma\}}-\mathbb{1}_{\{|\mathbf{x}^T\boldsymbol{\zeta}_2+v|\geq\tau\sigma\}})|\big]\\ &\leq \mathbb{E}_{\mathbf{x}}\big[\|\mathbf{x}\|_2^4\lambda_1(\tau)\|\boldsymbol{\zeta}_1-\boldsymbol{\zeta}_2\|_2^2+\|\mathbf{x}\|_2^2\lambda_2(\tau)\|\boldsymbol{\zeta}_1-\boldsymbol{\zeta}_2\|_2^2\\ &\qquad+2\|\mathbf{x}\|_2^3\sqrt{\lambda_1(\tau)\lambda_2(\tau)}\|\boldsymbol{\zeta}_1-\boldsymbol{\zeta}_2\|_2^2\big]\\ &= \left(r_{x^4}\lambda_1(\tau)+\operatorname{tr}(\mathbf{R}_x)\lambda_2(\tau)+2r_{x^3}\sqrt{\lambda_1(\tau)\lambda_2(\tau)}\right)\\ &\qquad\times\|\boldsymbol{\zeta}_1-\boldsymbol{\zeta}_2\|_2^2. \end{aligned} \tag{35}$$

where the first inequality comes from Cauchy-Schwarz inequality; the second inequality follows by bounding the Rayleigh quotient, and the last one uses $(A0)$ and $(A1)$. The identities $\lambda_{\max}((\mathbf{x}\mathbf{x}^T)^2) = \|\mathbf{x}\|_2^4$, $\lambda_{\max}(\mathbf{x}\mathbf{x}^T) = \|\mathbf{x}\|_2^2$, and $\mathbb{E}[\|\mathbf{x}\|_2^2] = \operatorname{tr}(\mathbf{R}_x)$, have also been used. Since $\|\boldsymbol{\zeta}_1-\boldsymbol{\zeta}_2\|_2 = \|\boldsymbol{\theta}_1-\boldsymbol{\theta}_2\|_2$, we have from (35) that $L^2 = r_{4x}\lambda_1(\tau)+\operatorname{tr}(\mathbf{R}_x)\lambda_2(\tau)+2r_{3x}\sqrt{\lambda_1(\tau)\lambda_2(\tau)}$.

Finally, the expected norm of the gradient at $\boldsymbol{\theta} = \boldsymbol{\theta}_o$ is bounded and equals

$$\begin{aligned}&\mathbb{E}\left[\|\nabla f^{(\tau)}(\boldsymbol{\theta}_o)\|_2^2\right] = \mathbb{E}\left[\|\mathbf{x}\|_2^2 e(\boldsymbol{\theta}_o)(1-c)\right]\\ &= \mathbb{E}_{\mathbf{x}}\left[\|\mathbf{x}\|_2^2\right]\mathbb{E}_v\left[v^2\mathbb{1}_{\{|v|>\tau\sigma\}}\right]\\ &= \operatorname{tr}(\mathbf{R}_x)\left[\sigma^2-\int_{-\tau\sigma}^{\tau\sigma} v^2\frac{e^{-\frac{v^2}{2\sigma^2}}}{\sqrt{2\pi\sigma^2}}dv\right]\\ &= \operatorname{tr}(\mathbf{R}_x)\left[\sigma^2-\sigma^2\left[Q\left(\frac{v}{\sigma}\right)-\frac{v}{\sigma}\phi\left(\frac{v}{\sigma}\right)\right]_{-\tau\sigma}^{\tau\sigma}\right]\\ &= 2\sigma^2\operatorname{tr}(\mathbf{R}_x)\left(1-Q(\tau)+\tau\phi(\tau)\right)\end{aligned}$$

which completes the proof. ■

Proof of Proposition 3: For the error vector $\boldsymbol{\zeta}_n := \boldsymbol{\theta}_n-\boldsymbol{\theta}_o$, AC-RLS satisfies $\boldsymbol{\zeta}_n = \mathbf{C}_n\sum_{i=1}^n\mathbf{x}_iv_i(1-c_i)$. If $\{c_i\}_{i=1}^n$ are deterministic and given, the error covariance matrix $\mathbf{K}_n := \mathbb{E}[\boldsymbol{\zeta}_n\boldsymbol{\zeta}_n^T]$ becomes

$$\begin{aligned}\mathbf{K}_n &= \mathbb{E}_{\mathbf{x},v}\left[\mathbf{C}_n\sum_{i=1}^n\sum_{j=1}^n\mathbf{x}_i\mathbf{x}_j^Tv_iv_j(1-c_i)(1-c_j)\mathbf{C}_n\right]\\ &= \mathbb{E}_{\mathbf{x}}\left[\mathbf{C}_n\sum_{i=1}^n\sum_{j=1}^n\mathbf{x}_i\mathbf{x}_j^T\mathbb{E}_v\left[v_iv_j\right](1-c_i)(1-c_j)\mathbf{C}_n\right]\\ &= \sigma^2\mathbb{E}_{\mathbf{x}}\left[\mathbf{C}_n\sum_{i=1}^n\mathbf{x}_i\mathbf{x}_i^T(1-c_i)\mathbf{C}_n\right]\\ &= \sigma^2\mathbb{E}_{\mathbf{x}}\left[\mathbf{C}_n\mathbf{C}_n^{-1}\mathbf{C}_n\right] = \sigma^2\mathbb{E}_{\mathbf{x}}[\mathbf{C}_n]\end{aligned}$$

Having assumed $\mathbf{x}_n\mathbf{x}_n^T(1-c_n)$ to be ergodic and for large enough n, then $\mathbf{C}_n^{-1} = \sum_{i=1}^n\mathbf{x}_i\mathbf{x}_i^T(1-c_i)$ can be approximated by $n\mathbb{E}_{\mathbf{x},v}\left[\mathbf{x}\mathbf{x}^T(1-c)\right] = n\mathbb{E}_{\mathbf{x}}\left[\mathbf{x}\mathbf{x}^T\mathbb{E}_v[1-c]\right] = n\mathbb{E}_{\mathbf{x}}\left[\mathbf{x}\mathbf{x}^T\Pr\{c=0|\mathbf{x}\}\right] = \bar{\mathbf{C}}_n^{-1}$. Given that $2Q(\tau)\leq\Pr\{c=0|\mathbf{x}\}\leq 1\,\forall\mathbf{x}$, we obtain

$$2Q(\tau)n\mathbf{R}_x\preceq\bar{\mathbf{C}}_n^{-1}\preceq n\mathbf{R}_x.$$

From [37, Cor. 7.7.4(a)] and since $\mathbf{C}_n$ converges monotonically to $\bar{\mathbf{C}}_n$, there exists $k>0$ such that for all $n>k$

$$\frac{1}{n}\mathbf{R}_x^{-1}\preceq\mathbf{C}_n\preceq\frac{1}{2Q(\tau)n}\mathbf{R}_x^{-1}.$$

The result follows since $\mathbb{E}\left[\|\boldsymbol{\theta}_n-\boldsymbol{\theta}_o\|_2^2\right] = \operatorname{tr}(\mathbf{K}_n) = \sigma^2\operatorname{tr}(\mathbb{E}\left[\mathbf{C}_n\right])$. ■

References

[1] K. Slavakis, G. B. Giannakis, and G. Mateos, "Modeling and optimization for big data analytics: Learning tools for our era of data deluge," *IEEE Sig. Proc. Mag.*, vol. 31, no. 5, pp. 18–31, Sept. 2014.

[2] D. Donoho, "Compressed sensing," *IEEE Trans. Info. Theory*, vol. 52, no. 4, pp. 1289–1306, Apr. 2006.

[3] F. Pukelsheim, *Optimal Design of Experiments*. SIAM, 1993.

[4] P. Drineas, M. W. Mahoney, and S. Muthukrishnan, "Sampling algorithms for l 2 regression and applications," in *SIAM-ACM Symp. on Discrete Algorithms*, 2006, pp. 1127–1136.

[5] C. Boutsidis and P. Drineas, "Random projections for the nonnegative least-squares problem," *Linear Algebra and its Applications*, vol. 431, no. 5, pp. 760–771, 2009.

[6] M. Mahoney, "Randomized algorithms for matrices and data," *Found. Trends. in Mach. Learn.*, vol. 3, no. 2, pp. 123–224, 2011.

[7] D. P. Woodruff, "Sketching as a tool for numerical linear algebra," *Found. and Trends in Theor. Computer Science*, vol. 10, pp. 1–157, 2014.

[8] G. Raskutti and M. Mahoney, "A statistical perspective on randomized sketching for ordinary least-squares," in *Inter. Conf. on Machine Learning*, Lille, France, 2015.

[9] M. Pilanci and M. J. Wainwright, "Iterative Hessian sketch: Fast and accurate solution approximation for constrained least-squares," *arXiv preprint arXiv:1411.0347*, 2014.

[10] Y. Lu, P. Dhillon, D. P. Foster, and L. Ungar, "Faster ridge regression via the subsampled randomized Hadamard transform," in *Advances in Neural Inform. Proces. Systems*, Lake Tahoe, CA, Dec. 2013, pp. 369–377.

[11] Y. Lu and D. P. Foster, "Fast ridge regression with randomized principal component analysis and gradient descent," *arXiv preprint arXiv:1405.3952*, 2014.

[12] T. Strohmer and R. Vershynin, "A randomized Kaczmarz algorithm with exponential convergence," *J. of Fourier Analysis and Applications*, vol. 15, no. 2, pp. 262–278, 2009.

[13] D. Needell, N. Srebro, and R. Ward, "Stochastic gradient descent and the randomized Kaczmarz algorithm," *ArXiv e-prints. [Online]. Available: arXiv:1310.5715v2.*, 2014.

[14] A. Agaskar, C. Wang, and Y. M. Lu, "Randomized Kaczmarz algorithms: Exact MSE analysis and optimal sampling probabilities," in *Proc. of Global Conf. on Signal and Info. Proc.*, Atlanta, Dec. 2014, pp. 389–393.

[15] S. Shalev-Shwartz, "Online learning and online convex optimization," *Foundations and Trends in Machine Learning*, pp. 107–194, 2011.

[16] A. Ribeiro and G. B. Giannakis, "Bandwidth–constrained distributed estimation for wireless sensor networks–part I: Gaussian case," *IEEE Trans. Sig. Proc.*, vol. 54, no. 3, pp. 1131–1143, Mar. 2006.

[17] Y. Plan and R. Vershynin, "One-bit compressed sensing by linear programming," *IEEE Trans. Sig. Proc.*, vol. 66, no. 8, pp. 1275–1297, Aug. 2013.

[18] G. Mateos, J. A. Bazerque, and G. B. Giannakis, "Distributed sparse linear regression," *IEEE Trans. Sig. Proc.*, vol. 58, no. 10, pp. 5262–5276, Oct. 2010.

[19] T. Amemiya, "Tobit models: A survey," *J. Econom.*, vol. 24, no. 1, pp. 3–61, 1984.

[20] L. Evers and C. M. Messow, "Sparse kernel methods for high-dimensional survival data," *Bioinformatics*, vol. 14, no. 2, pp. 1632–1638, July 2008.

[21] J. Tobin, "Estimation of relationships for limited dependent variables," *Econometrica: J. Econometric Soc.*, vol. 26, no. 1, pp. 24–36, 1958.

[22] S. Maleki and G. Leus, "Censored truncated sequential spectrum sensing for cognitive radio networks," *IEEE J. Sel. Areas Commun.*, vol. 31, no. 3, pp. 364–378, 2013.

[23] E. Msechu and G. B. Giannakis, "Sensor–centric data reduction for estimation with WSNs via censoring and quantization," *IEEE Trans. Sig. Proc.*, vol. 60, no. 1, pp. 400–414, 2012.

[24] K. You, L. Xie, and S. Song, "Asymptotically optimal parameter estimation with scheduled measurements," *IEEE Trans. Sig. Proc.*, vol. 61, no. 14, pp. 3521–3531, July 2013.

[25] T. Amemiya, "Regression analysis when the dependent variable is truncated normal," *Econometrica*, vol. 41, no. 6, pp. 997–1016, 1973.

[26] T. Y. Young and T. W. Calvert, *Classification, Estimation and Pattern Recognition.* North-Holland, 1974.

[27] D. Bertsekas, *Convex Optimization Algorithms.* Athena Scientific, United States, 2015.

[28] K. Slavakis, S.-J. Kim, G. Mateos, and G. B. Giannakis, "Stochastic approximation vis-a-vis online learning for big data analytics [lecture notes]," *IEEE Sig. Proc. Mag.*, vol. 31, no. 6, pp. 124–129, 2014.

[29] S. M. Kay, *Fundamentals of Statistical Signal Processing, Vol. I: Estimation Theory.* Englewood Cliffs: Prentice Hall PTR, 1993.

[30] M. Mahoney, "Algorithmic and statistical perspectives on large-scale data analysis," *Combinatorial Scientific Computing*, pp. 427–469, 2012.

[31] D. P. Bertsekas and I. B. Rhodes, "Recursive state estimation for a set-membership description of uncertainty," *IEEE Trans. Autom. Control*, vol. 16, no. 2, pp. 117–128, 1971.

[32] S. Gollamudi, S. Nagaraj, S. Kapoor, and Y.-F. Huang, "Set-membership filtering and a set-membership normalized LMS algorithm with an adaptive step size," *IEEE Signal Processing Letters*, vol. 5, no. 5, pp. 111–114, 1998.

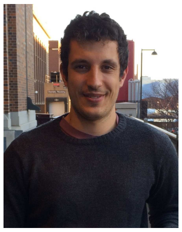
[33] Y. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning.* Springer, 2009.

[34] P. J. Huber, "Robust estimation of a location parameter," *The Annals of Mathematical Statistics*, vol. 35, no. 1, pp. 73–101, 1964.

[35] M. Lichman, "UCI machine learning repository," 2013. [Online]. Available: http://archive.ics.uci.edu/ml

[36] E. Moulines and F. R. Bach, "Non-asymptotic analysis of stochastic approximation algorithms for machine learning," in *Proc. of Advances in Neural Info. Proc. Sys. Conf.*, Granada, Spain, 2011, pp. 451–459.

[37] R. A. Horn and C. R. Johnson, *Matrix Analysis*, 2nd ed. Cambridge University Press, 2013.



Dimitris Berberidis (S'15) received his Diploma in Electrical and Computer Engineering (ECE) from the University of Patras, Greece, in 2012. In 2015, he obtained his M.Sc. degree in ECE from the University of Minnesota, Twin Cities, where he is currently working towards his Ph.D. degree. His research interests include statistical signal processing, big data analytics, tracking, and measurement selection.

Vassilis Kekatos (M'10) obtained his Diploma, M.Sc., and Ph.D. in Computer Science and Engr. from the Univ. of Patras, Greece, in 2001, 2003, and 2007, respectively. He was a recipient of a Marie Curie Fellowship during 2009-2012. During the summer of 2012, he worked for Windlogics Inc. After that, he was a research associate with the Dept. of Electrical and Computer Engr. of the Univ. of Minnesota. During 2014, he stayed with the University of Texas at Austin and the Ohio State University as a visiting researcher, and he received the postdoctoral career development award (honorable mention) by the University of Minnesota. In August 2015, he joined the Dept. of Electrical and Computer Engr. of Virginia Tech as an Assistant Professor. His current research focus is on optimization, learning, and management of future energy systems.

Georgios B. Giannakis (F'97) received his Diploma in Electrical Engnr. from the Ntl. Tech. Univ. of Athens, Greece, 1981. From 1982 to 1986 he was with the Univ. of Southern California, where he received his MSc. in Electrical Engnr. (1983), MSc. in Mathematics (1986), and Ph.D. in Electrical Engineering (1986). He became a Fellow of the IEEE in 1997. Since 1999, he has been a Professor with the Univ. of Minnesota where he now holds an ADC Chair in Wireless Telecommunications in the ECE Department, and serves as director of the Digital Technology Center.

His general interests span the areas of communications, networking and statistical signal processing subjects on which he has published more than 390 journal papers, 660 conference papers, 20 book chapters, two edited books and two research monographs (h-index 116). Current research focuses on sparsity and big data analytics, wireless cognitive radios, mobile ad hoc networks, renewable energy, power grids, gene-regulatory, and social networks. He is the (co-) inventor of 22 patents issued, and the (co-) recipient of 8 best paper awards from the IEEE Signal Processing (SP) and Communications Societies, including the G. Marconi Prize Paper Award in Wireless Communications. He also received Technical Achievement Awards from the SP Society (2000), from EURASIP (2005), a Young Faculty Teaching Award, the G. W. Taylor Award for Distinguished Research from the University of Minnesota, and the IEEE Fourier Technical Field Award (2014). He is a Fellow of EURASIP, and has served the IEEE in a number of posts, including that of a Distinguished Lecturer for the IEEE-SP Society.