Online Censoring for Large-Scale Regressions and Dynamical Processes with Application to Big Data

---

A THESIS

SUBMITTED TO THE FACULTY OF THE GRADUATE SCHOOL

OF THE UNIVERSITY OF MINNESOTA

BY

Dimitris Berberidis

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

MASTER OF SCIENCE

Prof. Georgios B. Giannakis

Prof. Jarvis Haupt

Prof. William L. Cooper

July 2015

Acknowledgments

Foremost, I would like to express my sincere gratitude to my advisor Prof. Georgios B. Giannakis for giving me the opportunity to be a part of the prestigious SPiNCOM research group. With his continued guidance and support, he has greatly helped me in taking my first steps into the world of research and engineering, leading to the completion of this thesis. In addition, he has and continues to aid me in developing clear scientific thought and expression.

In a special way I would like to extend my appreciation to Prof. Jarvis Haupt and Prof. William Cooper who agreed to serve on my thesis committee. Thanks go to a number of other professors in the departments of Electrical Engineering and Computer Science whose graduate level courses opened up my mind and helped me build the necessary background to embark on this area of research.

Special thanks also go to Prof. Vassilis Kekatos and Gang Wang for their significant contribution in the content of this thesis. I thank all my fellow labmates and good friends in SPINCOM for their support and encouragement in overcoming all difficulties.

Last but not the least, I would like to express by deepest thanks to my family: my parents Kostas and Maria for raising me and my dear brother Theodoros, and for their continued love and support throughout my life.

Dimitris Berberidis, Minneapolis, July 27, 2015

Work in this thesis was supported by the Institute of Renewable Energy and the Environment grant no. RL-0010-13, University of Minnesota, and NSF grants 1343860, 1442686, and 1500713.

Abstract

In an age of exponentially increasing data availability, performing inference tasks by utilizing the available information in its entirety is not always an affordable option. In this context, the present thesis introduces different methods for rendering large-scale linear regression and tracking of dynamic processes affordable, by processing a reduced number of data. The proposed algorithms utilize interval censoring of observations, in order to judiciously discard those deemed to have relatively small contribution towards enhancing the estimation or tracking accuracy. For linear regression, two groups of first- and second-order iterative algorithms are proposed: the first one focuses on reducing data storage and transmission costs, while the second is tailored for reducing the overall problem complexity. Leveraging principles of stochastic approximation, the introduced methods entail simple, closed-form updates, provable convergence guarantees, and can afford online processing of the data. As far as the tracking of dynamical processes, two distinct methods are put forth for reducing the number of data involved per time step. The first method builds on pre-processing the data for dimensionality reduction using low-complexity random projections, while the second performs censoring for data-adaptive measurement selection. Simulations on real and synthetic data, compare the proposed methods with competing alternatives and corroborate their efficacy in terms of estimation accuracy over complexity reduction.

Contents

List of Figures

Chapter 1

Introduction

1.1 Context, motivation and related work

Nowadays omni-present monitoring sensors, search engines, rating sites, and Internet-friendly portable devices generate massive volumes of typically dynamic data [37]. 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 [11], and exploiting it judiciously offers an effective means of reducing data processing, storage and communication costs.

In this regard, the notion of optimal design of experiments has been advocated for reducing the number of data required for inference tasks [30]. In recent works, the importance of sequential optimization along with random sampling of Big Data has also been highlighted [37]. Specifically for linear regressions, random projection (RP)-based methods have been popular for reducing the size of large-scale least-squares (LS) problems [10,12,22]. 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 convex subsets of the data [1,28,38]. Sequential optimization includes stochastic approximation, along with recent advances in online learning [34]. Frugal solvers of (pos-

sibly sparse) linear regressions are available by estimating regression coefficients based on (severely) quantized data [29, 33]; see also [25] for decentralized sparse LS solvers. With regards to tracking dynamical processes entailing time-varying parameters, channel-aware dimensionality reduction of observations was proposed in [47] and [21] for distributed wireless sensor networks (WSNs). A posterior-CRLB-based method for sensor selection in tracking was introduced in [48], and a greedy algorithm leveraging submodularity was proposed in [35] for measurement selection in sequential estimation.

In this context, the present thesis 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 [2], including survival analysis [13], saturated metering [39], and spectrum sensing [24]. It has recently been employed to select data for distributed parameter estimation using resource-constrained WSNs, thus trading off performance for tractability [27, 44]. Furthermore, censoring has been proposed for signal estimation using WSNs, tracking, and control of dynamical processes [4, 19, 41, 46]. Existing works on censoring mostly focus on reducing the rate at which sensors communicate their observations, and pertinent methods exhibit large computational complexity and storage requirements.

Nonetheless, it is by now well documented 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, which constitutes the main objective of this thesis. Envisioned applications for large-scale regressions include a wide range of parametric model identification tasks with large number of observations, encountered for instance in the analysis and modeling of biological data [18]. For dynamical processes, the contents of this thesis are expected to impact areas such as weather prediction [20], delay cartography in dynamic networks [31], as well as modeling and prediction of processes evolving over general network graphs [17, Ch. 8].

1.2 Thesis contributions

The present work employs interval censoring both for large-scale *online* regressions as well as for tracking dynamical processes. In the time-invariant case, the key novelty is to sequentially test and update regression estimates using censored data. Two censoring strategies are put forth, each tailored for mitigating different costs. In the first one, stochastic approximation algorithms are developed for sequentially updating the regression coefficients with low-complexity first- or second-order iterations to maximize the likelihood of censored and uncensored observations. This strategy aims at reducing the number of observations, in order to lower the cost of storage or transmission to a remote estimation site (a.k.a. fusion center (FC)). Relative to [27,44], the contribution here is a novel online scheme that greatly reduces storage requirements without requiring feedback from the FC to sensors. Error bounds are derived, while simulations demonstrate performance close to estimation error limits. The second censoring strategy focuses on reducing the complexity of large-scale linear regressions. The proposed methods are online by design, but may also be readily employed to reduce the complexity of solving a batch linear regression problem. The key difference with dimensionality-reducing alternatives, such as optimal design of experiments, randomized Kaczmarz's and RP-based methods, is that the introduced technique reduces complexity in a data-driven manner.

With regards to tracking dynamical processes with time-varying parameters, the estimation task is treated as a sequence of regularized linear regression problems (see Fig. 1.1). Thus, methods based on RPs and censoring can be sequentially leveraged to reduce the dimensionality of data associated with individual regression problems. The proposed censoring-based method supports online processing of observations, it is simple to implement, and simulations demonstrate that its estimation performance is close to the greedy method in [35], which is computationally much more complex. Finally, capitalizing on the state-space dynamical model, data-independent forward-backward smoothing iterations are developed to mitigate "on-a-budget" the performance degradation caused by dimensionality reduction.

Results of this thesis have been reported in journal and conference publications [5–7,40].

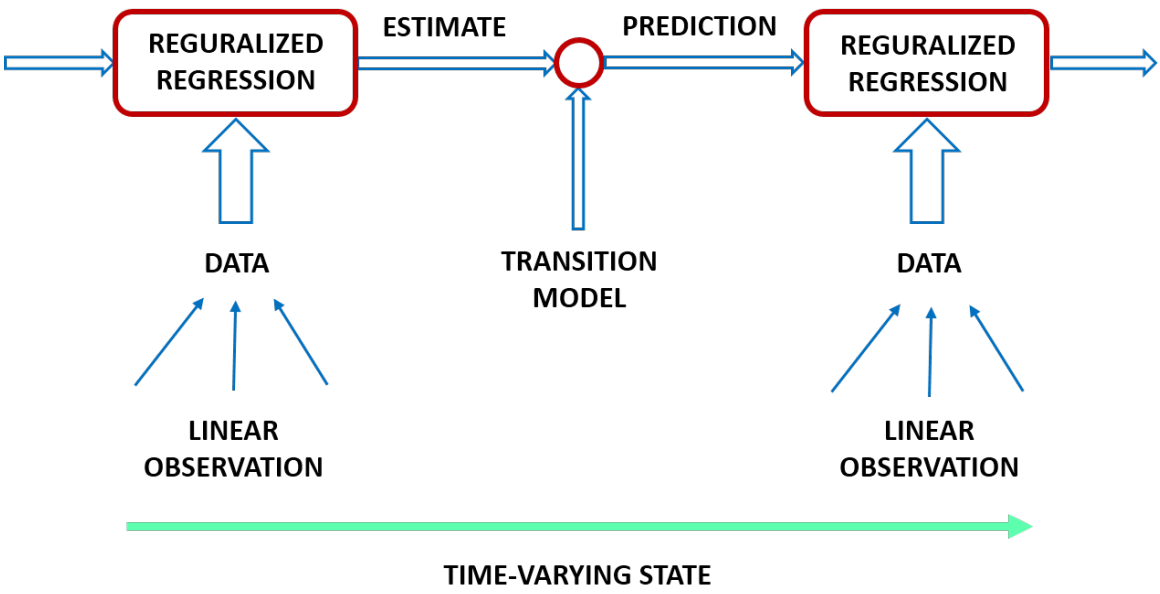


Figure 1.1: Visual interpretation of dynamical process estimation as a sequence or regularized regression problems.

1.3 Thesis outline

The rest of the thesis is as follows. Chapter 2 introduces censoring for linear regression with time-invariant parameters, and proposes first- and second-order estimation algorithms at reduced complexity. Chapter 3 develops RP- and censoring-based methods for dimensionality reduction and measurement selection to render the complexity of tracking dynamical processes, affordable. Finally, concluding remarks and future research directions are outlined in Chapter 4.

1.4 Notational convensions

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. Notation

$\mathcal{N}(\mathbf{m}, \mathbf{C})$ stands for the multivariate Gaussian distribution with mean $\mathbf{m}$ and covariance matrix $\mathbf{C}$. The ℓ_1- and ℓ_2-norms of a vector $\mathbf{y} \in \mathbb{R}^d$ are defined as $\|\mathbf{y}\|_1 := \sum_{i=1}^{d} |y(i)|$ and $\|\mathbf{y}\|_2 := \sqrt{\sum_{i=1}^{d} |y(i)|^2}$, respectively; $\phi(t) := (1/\sqrt{2\pi})\exp(-t^2/2)$ 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. Finally, for a matrix $\mathbf{X}$, let $\mathrm{tr}(\mathbf{X})$, $\lambda_{\min}(\mathbf{X})$ and $\lambda_{\max}(\mathbf{X})$ denote the trace, minimum, and maximum eigenvalue, respectively.

Chapter 2

Censoring for Linear Regression

Linear regression is arguably the most prominent among statistical inference methods, popular both for its simplicity as well as its broad applicability. On par with data-intensive applications, the sheer size of linear regression problems creates an ever growing demand for quick and cost efficient solvers. For instance, regression analysis on biological data (e.g.n protein tertiary structure prediction) may involve a prohibitively large number of observations. Fortunately, a significant percentage of the data accrued can be omitted while maintaining a certain quality of statistical inference with a limited computational budget. The present chapter proposes different methods for identifying and omitting uninformative observations in an online and data-adaptive fashion, built on principles of stochastic approximation and recent advances in data censoring. First- and second-order stochastic approximation maximum likelihood-based algorithms for censored observations are proposed for estimating the linear regression coefficients. Moreover, online algorithms are introduced to reduce the overall complexity by adaptively performing censoring along with estimation. The proposed algorithms entail simple closed-form updates, and have provable non-asymptotic guarantees. Furthermore, specific rules are developed for tuning to desired censoring patterns and levels of dimensionality reduction. Simulated tests on real and synthetic datasets corroborate the efficacy of the proposed data-adaptive methods compared to data-agnostic random projection based alternatives.

2.1 Problem Statement and Preliminaries

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

$$y_n = \mathbf{x}_n^T \boldsymbol{\theta}_o + \upsilon_n, \quad n = 1, 2, \ldots, D \tag{2.1}$$

where $\mathbf{x}_n$ 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 fashion, while meeting minimal resource requirements. The term resources here refers to the total number of utilized observations and/or regression rows, 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.

2.1.1 NAC and AC Rules

A generic censoring rule for the data in (2.1) is given by

$$z_n := \begin{cases} * & , \ y_n \in \mathcal{C}_n \\ y_n & , \ \text{otherwise} \end{cases}, \quad n = 1, \ldots, D \tag{2.2}$$

where $*$ denotes an unknown value when the n-th datum has been censored (thus it is unavailable) - a case when 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 using a predictor $\hat{y}_n$ formed using a preliminary (e.g., LS) estimate of $\boldsymbol{\theta}_o$ as

$$\hat{\boldsymbol{\theta}}_K = (\mathbf{X}_K^T \mathbf{X}_K)^{-1} \mathbf{X}_K^T \mathbf{y}_K \tag{2.3}$$

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:

$$(z_n, c_n) := \begin{cases} (y_n, 0) & , \text{ if } \left|\frac{y_n - \mathbf{x}_n^T \hat{\boldsymbol{\theta}}_K}{\sigma}\right| \geq \tau_n \\ (*, 1) & , \text{ otherwise} \end{cases} \tag{2.4}$$

where $\{\tau_n\}_{n=1}^D$ are censoring thresholds, and as in (2.2), $*$ signifies that the exact value of y_n is unavailable. The rule (2.4) 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 proportion 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 (2.4) 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:

$$(z_n, c_n) := \begin{cases} (y_n, 0) & , \text{ if } \left|\frac{y_n - \mathbf{x}_n^T \boldsymbol{\theta}_{n-1}}{\sigma}\right| \geq \tau_n \\ (*, 1) & , \text{ otherwise} \end{cases} . \tag{2.5}$$

In Section 2.3, (2.5) 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 estimator to the censor, a feature that may be undesirable in distributed estimation setups. Nonetheless, in centralized linear regression, AC is well motivated for reducing the problem dimension and computational complexity.

2.2 Online Estimation with NAC

Since noise samples $\{\upsilon_n\}_{n=1}^D$ in (2.1) are independent and (2.4) 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

$$p(z_n, c_n; \boldsymbol{\theta}) = \left[\mathcal{N}\left(z_n; \mathbf{x}_n^T \boldsymbol{\theta}, \sigma^2\right)\right]^{1-c_n} \left[\Pr\{c_n = 1\}\right]^{c_n} \tag{2.6}$$

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 $\Pr\{c_n = 1\} = \Pr\{\hat{y}_n - \tau_n\sigma - \mathbf{x}_n^T\boldsymbol{\theta}_0 \leq v_n \leq \hat{y}_n + \tau_n\sigma - \mathbf{x}_n^T\boldsymbol{\theta}_0\}$, and after recalling that v_n is Gaussian

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

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

$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \ \mathcal{L}_D(\boldsymbol{\theta}) := \sum_{n=1}^{D} \ell_n(\boldsymbol{\theta}) \tag{2.7}$$

where functions $\ell_n(\boldsymbol{\theta})$ are given by (cf. (2.6))

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

If the entire dataset $\{z_n, c_n, \mathbf{x}_n\}_{n=1}^D$ were available, the MLE could be obtained via gradient descent or Newton iterations.

Considering Big Data applications where storage resources are scarce, we resort to a stochastic approximation solution and process censored data sequentially. In particular, when datum n becomes available, the unknown parameter is updated as

$$\boldsymbol{\theta}_n := \boldsymbol{\theta}_{n-1} - \mu_n \mathbf{g}_n(\boldsymbol{\theta}_{n-1}) \tag{2.8}$$

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

$$\beta_n(\boldsymbol{\theta}) := \tfrac{1-c_n}{\sigma^2}(y_n - \mathbf{x}_n^T\boldsymbol{\theta}) + \frac{c_n}{\sigma}\frac{\phi\left(z_n^u(\boldsymbol{\theta})\right) - \phi\left(z_n^l(\boldsymbol{\theta})\right)}{Q\left(z_n^u(\boldsymbol{\theta})\right) - Q\left(z_n^l(\boldsymbol{\theta})\right)}. \tag{2.9}$$

The overall scheme is tabulated as Algorithm 1.

Observe that when the n-th datum is not censored ($c_n = 0$), the second summand in the right-hand side (RHS) of (2.9) vanishes, and (2.8) reduces to an ordinary LMS update. When $c_n = 1$, the first summand disappears, and the update in (2.8) exploits the fact that the unavailable y_n lies in a known interval ($|y_n - \mathbf{x}_n^T\hat{\boldsymbol{\theta}}_K| \leq \tau_n\sigma$), information that would have been ignored by an ordinary LMS algorithm.

Since the SA-MLE is in fact a Robbins-Monroe iteration on the sequence $\{\mathbf{g}(\boldsymbol{\theta})\}_{n=1}^D$, it inherits related convergence properties. Specifically, by selecting $\mu_n = 1/(nM)$ (for an

Algorithm 1 Stochastic Approximation (SA)-MLE

Initialize $\boldsymbol{\theta}_0$ as the LSE $\hat{\boldsymbol{\theta}}_K$ in (2.3).

for $n = 1 : D$ **do**

Measurement y_n is possibly censored using (2.4).

Estimator receives $(z_n, c_n, \mathbf{x}_n)$.

Parameter $\boldsymbol{\theta}$ is updated via (2.8) and (2.9).

end for

appropriate M), the SA-MLE algorithm is asymptotically efficient and Gaussian [45, 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

$$R(D) := \sum_{n=1}^{D} [\ell_n(\boldsymbol{\theta}_n) - \ell_n(\boldsymbol{\theta})] \,. \tag{2.10}$$

Selecting μ properly, Algorithm 1 can afford bounded regret as asserted next; see Appendix for the proof.

Proposition 1. *Suppose* $\|\mathbf{x}_n\|_2 \leq \bar{x}$ *and* $|\beta_n(\boldsymbol{\theta})| \leq \bar{\beta}$ *for* $n = 1, \ldots, D$, *and let* $\boldsymbol{\theta}^*$ *be the minimizer of (2.7). By choosing* $\mu = \|\boldsymbol{\theta}^* - \hat{\boldsymbol{\theta}}_K\|_2/(\sqrt{2D}\bar{\beta}\bar{x})$, *the regret of the SA-MLE satisfies*

$$R(D) \leq \sqrt{2D}\|\boldsymbol{\theta}^* - \hat{\boldsymbol{\theta}}_K\|_2 \bar{x}\bar{\beta} \;.$$

Proposition 1 assumes bounded $\mathbf{x}_n$'s and noise. Although the latter is not satisfied by e.g., the Gaussian distribution, appropriate bounds ensure that (1) holds with high probability.

2.2.1 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 (2.8) is modified as follows

$$\boldsymbol{\theta}_n := \boldsymbol{\theta}_{n-1} - \mathbf{M}_n^{-1}\mathbf{g}_n(\boldsymbol{\theta}_{n-1}). \tag{2.11}$$

Algorithm 2 Second-order SA-MLE

Initialize $\boldsymbol{\theta}_0$ as the LSE $\hat{\boldsymbol{\theta}}_K$ in (2.3).

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 (2.4).

Estimator receives $(z_n, \mathbf{x}_n, c_n)$.

Compute $\gamma_n(\boldsymbol{\theta}_{n-1})$ from (2.12).

Update matrix step size from (2.13).

Update parameter estimate as in (2.11).

end for

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 $(1/n)\sum_{i=1}^n \nabla^2\ell_i(\boldsymbol{\theta}_i)$ [8]. To this end, note first that $\nabla^2\ell_n(\boldsymbol{\theta}) = \gamma_n(\boldsymbol{\theta})\mathbf{x}_n\mathbf{x}_n^T$, where

$$\gamma_n(\boldsymbol{\theta}) := -\frac{(1-c_n)}{\sigma^2} - \frac{c_n}{\sigma^2}\left[\left(\frac{\phi\left(z_n^u(\boldsymbol{\theta})\right) - \phi\left(z_n^l(\boldsymbol{\theta})\right)}{Q\left(z_n^u(\boldsymbol{\theta})\right) - Q\left(z_n^l(\boldsymbol{\theta})\right)}\right)^2 - \frac{z_n^u(\boldsymbol{\theta}\phi\left(z_n^u(\boldsymbol{\theta})\right) - z_n^l(\boldsymbol{\theta}\phi\left(z_n^l(\boldsymbol{\theta})\right)}{Q\left(z_n^u(\boldsymbol{\theta})\right) - Q\left(z_n^l(\boldsymbol{\theta})\right)}\right]. \tag{2.12}$$

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

$$\mathbf{C}_n = \frac{n}{n-1}\left(\mathbf{C}_{n-1} - \frac{\mathbf{C}_{n-1}\mathbf{x}_n\mathbf{x}_n^T\mathbf{C}_{n-1}}{(n-1)\gamma_n^{-1}(\boldsymbol{\theta}_{n-1}) + \mathbf{x}_n^T\mathbf{C}_{n-1}\mathbf{x}_n}\right). \tag{2.13}$$

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 2.4.1 confirm its faster convergence at the cost of $\mathcal{O}(p^2)$ complexity per update.

2.2.2 Controlling Data Reduction via NAC

To apply the NAC rule of (2.4) for data reduction at a controllable rate, a relation between thresholds $\{\tau_n\}$ and the censoring rate must be derived. 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. Since $\{y_n\}$ are generated randomly according to (2.1), it is clear that d is itself a random variable. The analysis is thus focused on the average censoring ratio

$$\bar{c} := \mathbb{E}\left[\frac{D-d}{D}\right] = \frac{1}{D}\sum_{n=1}^{D}\mathbb{E}[c_n] = \frac{1}{D}\sum_{n=1}^{D}\pi_n \tag{2.14}$$

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. (2.4)]

$$\begin{aligned}\pi_n(\tau_n) &= \Pr\{-\tau_n\sigma \leq y_n - \hat{y}_n \leq \tau_n\sigma\} \\ &= \Pr\{-\tau_n \leq \frac{\mathbf{x}_n^T(\boldsymbol{\theta}_o - \hat{\boldsymbol{\theta}}_K) + v_n}{\sigma} \leq \tau_n\}.\end{aligned} \tag{2.15}$$

By the properties of the 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 (2.15) simplify to

$$\pi_n(\tau_n) = 1 - 2Q\left(\tau_n\left[\mathbf{x}_n^T(\mathbf{X}_K^T\mathbf{X}_K)^{-1}\mathbf{x}_n + 1\right]^{-1/2}\right). \tag{2.16}$$

Solving (2.16) for τ_n, one arrives for a given $\pi_n^\star = \pi_n(\tau_n^\star)$ at

$$\tau_n^\star = \left[\mathbf{x}_n^T(\mathbf{X}_K^T\mathbf{X}_K)^{-1}\mathbf{x}_n + 1\right]^{1/2} Q^{-1}\left(\frac{1-\pi_n^\star}{2}\right). \tag{2.17}$$

Hence, for a prescribed $\bar{c}$, one can select a desired censoring probability pattern $\{\pi_n^\star\}_{n=1}^D$ to satisfy (2.14), and corresponding $\{\tau_n^\star\}_{n=1}^D$ in accordance with (2.17).

The threshold selection (2.17) requires knowledge of all $\{\mathbf{x}_n\}_{n=1}^D$. In addition, implementing (2.17) for all D observations, requires $\mathcal{O}(Dp^2)$ computations that may not

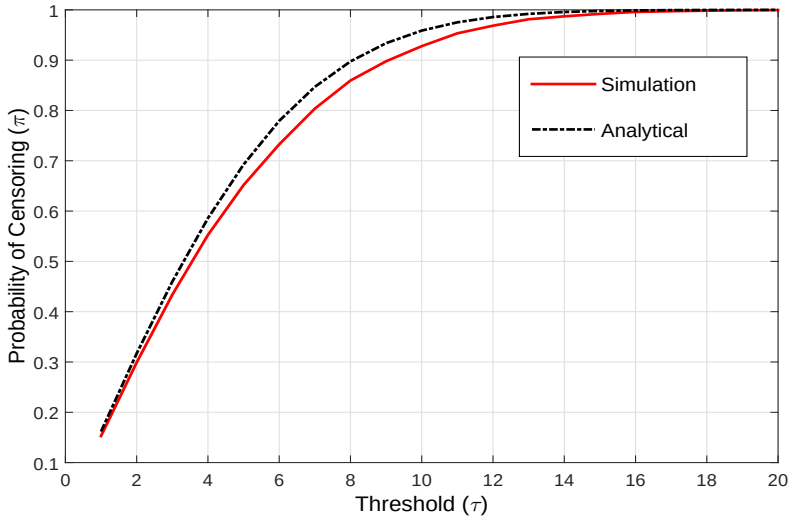
be affordable for $D \gg p$. To deal with this, the ensuing simple threshold selection rule is advocated. Supposing that $\{\mathbf{x}_n\}_{n=1}^D$ are generated i.i.d. according to some unknown distribution with known first- and second-order moments, a relation between a target *common* censoring probability $\pi^\star$ and a common threshold τ can be obtained in closed form. Assume 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 (CLT), $\mathbf{u}_n^T\mathbf{u}$ 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

$$\begin{aligned}\pi_n \approx \pi &= Q\left(-\frac{\tau}{\sqrt{p/K+1}}\right) - Q\left(\frac{\tau}{\sqrt{p/K+1}}\right) \\ &= 1 - 2Q\left(\frac{\tau}{\sqrt{p/K+1}}\right), \quad n = 1, \ldots, D. \end{aligned} \tag{2.18}$$

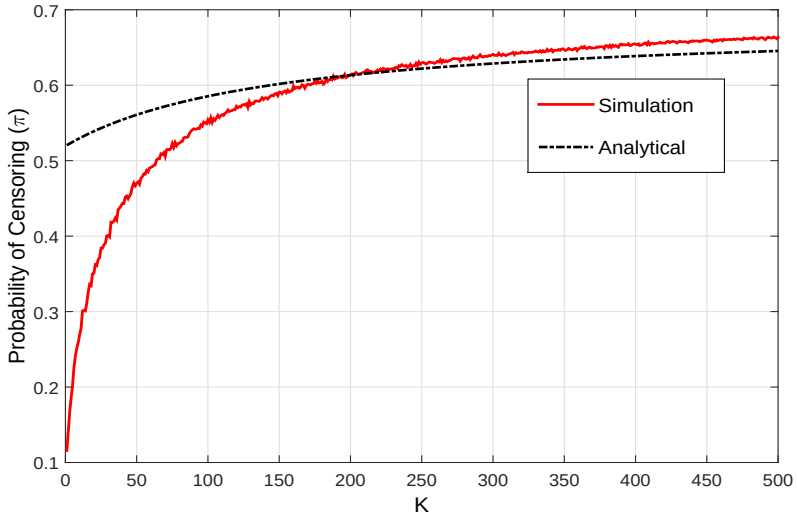
As expected, due to the normalization by σ in (2.4), π does not depend on σ. Interestingly, it does not depend on $\mathbf{R}_x$ either. Having expressed π as a function of τ, the latter can be tuned to achieve the desirable data reduction. Following the law of large numbers and given parameters p and K, to achieve an average censoring ratio of $\bar{c} = \pi^\star = (D-d)/D$, the threshold can be set to

$$\tau = \sqrt{1 + p/K}\, Q^{-1}\left(\tfrac{1-\pi^\star}{2}\right). \tag{2.19}$$

Figure 2.1(a) depicts π as a function of τ for $p = 100$ and $K = 200$. Function (2.18) is compared with the simulation-based estimate of π_n using 100 Monte Carlo runs, confirming that (2.18) 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 2.1(b) shows the probability of censoring for varying K with fixed $p = 100$ and $\tau = 1$. Approximation (2.18) yields a reliable value for π for as few as $K \approx 200$ preliminary data.



(a)



(b)

Figure 2.1: a) Censoring probability for varying threshold ($p = 100, K = 200$). b) Censoring probability for varying K ($p = 100, \tau = 1$).

2.3 Big Data Streaming Regression with AC

The NAC-based algorithms of Section 2.2 emerge in a wide range of applications for which censoring occurs naturally as part of the data acquisition process; see e.g., the Tobit model

in economics [2], and survival data analytics in [13]. Apart from these applications where data are inherently censored, our idea is to *employ censoring deliberately* for data reduction. Leveraging NAC for data reduction decouples censoring from estimation, and thus eliminates the need for obtaining further information. However, one intuitively expects improved performance with a *joint censoring-estimation* design.

In this context, first- and second-order sequential algorithms will be developed in this section for the AC in (2.5). Instead of $\hat{\boldsymbol{\theta}}_K$, AC is 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. Section 2.3.1 introduces an AC-based LMS algorithm for large-scale streaming regressions, while Section 2.3.2 puts forth an AC-based recursive least-squares (RLS) algorithm as a viable alternative to random projections and sampling.

2.3.1 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

$$\boldsymbol{\theta}_n = \boldsymbol{\theta}_{n-1} + \mu \mathbf{x}_n e_n(\boldsymbol{\theta}_{n-1}) \tag{2.20}$$

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 $\min_{\boldsymbol{\theta}} \mathbb{E}[f_n(\boldsymbol{\theta})]$, where the instantaneous cost is $f_n(\boldsymbol{\theta}) = e_n^2(\boldsymbol{\theta})/2$.

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

$$f_n^{(\tau)}(\boldsymbol{\theta}) := \begin{cases} \frac{e_n^2(\boldsymbol{\theta}) - \tau_n^2\sigma^2}{2} & , \ |e_n(\boldsymbol{\theta})| \geq \tau_n\sigma \\ 0 & , \ |e_n(\boldsymbol{\theta})| < \tau_n\sigma \end{cases} \tag{2.21}$$

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 (2.21) with $\tau_n = \tau$, performs the following AC-LMS update per datum n

$$\boldsymbol{\theta}_n := \begin{cases} \boldsymbol{\theta}_{n-1} + \mu \mathbf{x}_n e_n(\boldsymbol{\theta}_{n-1}) & , \ |e_n(\boldsymbol{\theta}_{n-1})| \geq \tau\sigma \\ \boldsymbol{\theta}_{n-1} & , \ \text{otherwise} \end{cases} \tag{2.22}$$

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 SA-MLE, the 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 Algorithm 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 "independent-data assumption" of SA [45], following the same steps as in Section 2.2, and substituting $\hat{\boldsymbol{\theta}}_K = \boldsymbol{\theta}_{n-1}$ into (2.9), 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 lowering computational complexity of the proposed AC algorithms. The mean-square error (MSE) performance of AC-LMS is established in the next proposition proved in the Appendix.

Proposition 2. *Assume* $\mathbf{x}_n$*'s are generated i.i.d. with* $\mathbb{E}[\mathbf{x}_n] = \mathbf{0}$, $\mathbb{E}\left[\mathbf{x}_n \mathbf{x}_n^T\right] = \mathbf{R}_x$, $\mathbb{E}\left[\mathbf{x}_n^T \mathbf{x}_n \mathbf{x}_n^T\right] = \mathbf{0}^T$, *and* $\mathbb{E}\left[\left(\mathbf{x}_n \mathbf{x}_n^T\right)^2\right] = \mathbf{R}_x^2$, *while observations* y_n *are obtained according to model (2.1). For a diminishing* $\mu_n = \mu/n$ *with* $\mu = 2/\alpha$, *initial estimate* $\boldsymbol{\theta}_1$, *and*

censoring-controlling threshold τ, the AC-LMS in (2.22) yields an estimate $\boldsymbol{\theta}_n$ with MSE bounded 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 $\alpha:=2Q(\tau)\lambda_{\min}(\mathbf{R}_x)$, $\Delta:=2\mathrm{tr}(\mathbf{R}_x)\sigma^2(1-Q(\tau)+\tau p(\tau))$, *and* $L^2:=\lambda_{\max}\left(\mathbf{R}_x^2\right)$. *Further, for* $\mu<\alpha/(16L^2)$, *AC-LMS converges exponentially to a bounded error*

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

Proposition 2 asserts that AC-LMS achieves a bounded MSE. It also links MSE with the AC threshold τ that 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.

2.3.2 AC-RLS

A second-order AC algorithm is introduced here for the purpose of sequential estimation and dimensionality reduction. It is closely related to the RLS algorithm, which per time n implements the updates; see e.g., [36]

$$\mathbf{C}_n=\frac{n}{n-1}\left[\mathbf{C}_{n-1}-\frac{\mathbf{C}_{n-1}\mathbf{x}_n\mathbf{x}_n^T\mathbf{C}_{n-1}}{n-1+\mathbf{x}_n^T\mathbf{C}_{n-1}\mathbf{x}_n}\right] \tag{2.23a}$$

$$\boldsymbol{\theta}_n=\boldsymbol{\theta}_{n-1}+\frac{1}{n}\mathbf{C}_n\mathbf{x}_n(y_n-\mathbf{x}_n^T\boldsymbol{\theta}_{n-1}) \tag{2.23b}$$

where $\mathbf{C}_n$ is the sample estimate for $\mathbf{R}_x^{-1}$ and is typically initialized to $\mathbf{C}_0=\epsilon\mathbf{I}$, for some small positive ϵ, e.g., [16]. The RLS estimate at time n can be also obtained as

$$\boldsymbol{\theta}_n=\arg\min_{\boldsymbol{\theta}}\sum_{i=1}^{n}(y_i-\mathbf{x}_i^T\boldsymbol{\theta})^2+\epsilon\|\boldsymbol{\theta}\|_2^2. \tag{2.24}$$

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., [8].

To obtain a second-order counterpart of AC-LMS, we replace the quadratic instantaneous cost of RLS with the truncated quadratic in (2.21). The matrix step-size is further surrogated by

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

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

$$\mathbf{C}_n = \frac{n}{n-1}\left[\mathbf{C}_{n-1} - \frac{(1-c_n)\mathbf{C}_{n-1}\mathbf{x}_n\mathbf{x}_n^T\mathbf{C}_{n-1}}{n-1+\mathbf{x}_n^T\mathbf{C}_{n-1}\mathbf{x}_n}\right] \tag{2.25a}$$

$$\boldsymbol{\theta}_n = \boldsymbol{\theta}_{n-1} + \frac{1-c_n}{n}\mathbf{C}_n\mathbf{x}_n(y_n - \mathbf{x}_n^T\boldsymbol{\theta}_{n-1}) \tag{2.25b}$$

where c_n is decided by (2.5). For $c_n = 1$, the parameter vector is not updated, while costly updates of $\mathbf{C}_n$ are also avoided. In addition, different from the iterative expectation-maximization algorithm in [44], AC-RLS skips completely covariance updates. Its performance is characterized by the following proposition shown 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 (2.1), then for $\boldsymbol{\theta}_1 = \mathbf{0}$ and constant τ, there exists $k > 0$ such that AC-RLS estimates $\boldsymbol{\theta}_n$ yield bounded MSE*

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

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 and compare AC-RLS with RP and random sampling methods that have been advocated as fast LS solvers [22, 23]. 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

Algorithm 3 Adaptive-Censoring (AC)-RLS

Initialize $\boldsymbol{\theta}_0 = \mathbf{0}$ and $\mathbf{C}_0 = \epsilon\mathbf{I}$.

for $n = 1 : D$ **do**

if $\left|y_n - \mathbf{x}_n^T\boldsymbol{\theta}_{n-1}\right| \geq \tau\sigma$ **then**

Estimator receives $(y_n, \mathbf{x}_n)$ while $c_n = 0$.

Update inverse sample covariance from (2.25a).

Update estimate from (2.25b).

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

$\{-1/\sqrt{D}, +1/\sqrt{D}\}$ equiprobably. Intuitively, $\mathbf{R}$ renders all rows of "comparable importance" (quantified by the leverage scores [22, 23]), 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 can be solved as $\check{\boldsymbol{\theta}}_d = \arg\min_{\boldsymbol{\theta}} \|\mathbf{S}_d\mathbf{H}\mathbf{D}(\mathbf{y} - \mathbf{X}\boldsymbol{\theta})\|_2^2$. For a general preconditioning matrix $\mathbf{HD}$, computing the products $\mathbf{HDy}$ and $\mathbf{HDX}$ requires a prohibitive number of $\mathcal{O}(D^2p)$ computations. This is mitigated by the fact that $\mathbf{H}$ has binary $\{+1, -1\}$ entries and thus multiplications can be implemented as simple sign flips. Overall, the RP method reduces the computational complexity of the LS problem from $\mathcal{O}(Dp^2)$ to $\mathcal{O}(Dp^2)$ 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 data 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 2.4.3, 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 $\mathcal{O}(dp^2) + \mathcal{O}(Dp)$. Evidently, the complexity reduction is more prominent for larger model dimension p. For $p \gg 1$, the second 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 [9, 14]. However, the model assumptions and objectives of the two are different. SM assumes that the noise distribution in (2.1) 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 typically 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 [43]. Solvers are typically based on nonlinear programming, and support vectors (SVs) are returned after *batch* processing that does not scale well with the data size. 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. (2.21)] can be readily generalized to their counterparts based on the truncated ℓ_1 error norm. Cross-pollinating in the other direction, our AC-RLS iterations can be useful for online support vector machines capable of learning from streaming large-scale data with second-order closed-form iterations.

2.3.3 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 (2.14), 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 (2.14)–(2.15), it holds that

$$\pi_n(\tau_n) = 1 - 2Q\left(\tau_n \left[\mathbf{x}_n^T \mathbf{K}_{n-1} \mathbf{x}_n + 1\right]^{-1/2}\right). \tag{2.26}$$

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 [44]. 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

$$\tau_n = \left(\mathbf{x}_n^T\mathbf{C}_{n-1}\mathbf{x}_n/n + 1\right)^{1/2} Q^{-1}\left(\frac{1-\pi_n^\star}{2}\right). \tag{2.27}$$

Given $\{\pi_n^\star\}_{n=1}^D$ satisfying (2.14), 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 (2.27) 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 2.2.2, 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 $\sigma_{e_n}^2=\sigma^2\mathbb{E}\left[\mathbf{x}_n^T\boldsymbol{\zeta}_{n-1}\boldsymbol{\zeta}_{n-1}^T\mathbf{x}_n\right]+\sigma^2=\sigma^2\mathrm{tr}\left(\mathbb{E}\left[\mathbf{x}_n\mathbf{x}_n^T\boldsymbol{\zeta}_{n-1}\boldsymbol{\zeta}_{n-1}^T\right]\right)+\sigma^2=\sigma^2\mathrm{tr}\left(\mathbf{R}_x\mathbf{K}_{n-1}\right)+\sigma^2$, 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

$$\tau_n=\frac{\sigma_{e_n}}{\sigma}Q^{-1}\left(\frac{1-\pi_n^\star}{2}\right). \tag{2.28}$$

It is well known that for large n, the RLS error covariance matrix $\mathbf{K}_n$ converges to $\frac{\sigma^2}{n}\mathbf{R}_x^{-1}$. 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 $\mathbf{K}_n\approx\left(\sum_{i=1}^n(1-\pi_i^\star)\right)^{-1}\sigma^2\mathbf{R}_{\mathbf{x}}^{-1}$. Combined with (2.28), the latter leads to

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

Plugging σ_{e_n} into (2.28) yields the simple threshold selection

$$\tau_n=\left[p\left(\sum_{i=1}^{n-1}(1-\pi_i^\star)\right)^{-1}+1\right]^{1/2}Q^{-1}\left(\frac{1-\pi_n^\star}{2}\right). \tag{2.29}$$

Unlike (2.27), where thresholds are decided online at an additional computational cost, (2.29) offers an off-line threshold design strategy for AC-RLS. Based on (2.29), to achieve

$\bar{c} = \pi^\star = (D-d)/D$, thresholds are chosen as

$$\tau_n = \left(\frac{p}{(n-1)(1-\pi^\star)} + 1\right)^{1/2} Q^{-1}\left(\frac{1-\pi^\star}{2}\right) \tag{2.30}$$

which attains a constant π^* across iterations.

2.3.4 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 (2.1) 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

$$(c_n, c_n^o) = \begin{cases} (1,0) & , \ |e_n(\boldsymbol{\theta}_{n-1})| < \sigma\tau \\ (0,0) & , \ \tau\sigma \leq |e_n(\boldsymbol{\theta}_{n-1})| < \tau_o\sigma \\ (0,1) & , \ |e_n(\boldsymbol{\theta}_{n-1})| \geq \tau_o\sigma \end{cases} . \tag{2.31}$$

Similar to (2.5), 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 $\tau_o > \tau$.

Having separated data-censoring from outlier identification in (2.31), it becomes possible to robustify AC-LMS and AC-RLS against outliers. Towards this end, one approach is to completely ignore y_n when $c_n^o = 1$. Alternatively, the instantaneous cost function in (2.21) can be modified to a truncated Huber loss (cf. [15])

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

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

$$\boldsymbol{\theta}_n = \boldsymbol{\theta}_{n-1} + \mu_n \mathbf{g}_n(\boldsymbol{\theta}_{n-1}) \tag{2.32}$$

where

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

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

$$\boldsymbol{\theta}_n = \boldsymbol{\theta}_{n-1} + \frac{1}{n} \mathbf{C}_n \mathbf{g}_n(\boldsymbol{\theta}_{n-1}) \tag{2.33a}$$

$$\mathbf{C}_n = \frac{n}{n-1} \left[\mathbf{C}_{n-1} - \frac{(1-c_n)(1-c_n^o) \mathbf{C}_{n-1} \mathbf{x}_n \mathbf{x}_n^T \mathbf{C}_{n-1}}{n - 1 + \mathbf{x}_n^T \mathbf{C}_{n-1} \mathbf{x}_n} \right] . \tag{2.33b}$$

Observe that when $c_n^o = 1$, only $\boldsymbol{\theta}_n$ is updated, and the computationally costly update of (2.33b) is avoided.

2.4 Numerical Tests

2.4.1 SA-MLE

The online SA-MLE algorithms presented in Section 2.2 are simulated using Gaussian data generated according to (2.1) with a time-invariant $\boldsymbol{\theta}_o \in \mathbb{R}^p$, where $p = 30$, $\upsilon_n \sim \mathcal{N}(0,1)$ 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 (2.4) was used with $\tau = 1.5$ to censor approximately 75% of the observations. Plotted in Fig. 2.2 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 [27] to accommodate the NAC rule in (2.4). It can 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 [44] and [27].

To further evaluate the efficacy of the proposed methods, additional simulations were run for different levels of censoring by adjusting τ. Plotted in Figs. 2.3(a) and 2.3(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 (green solid curve) 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.

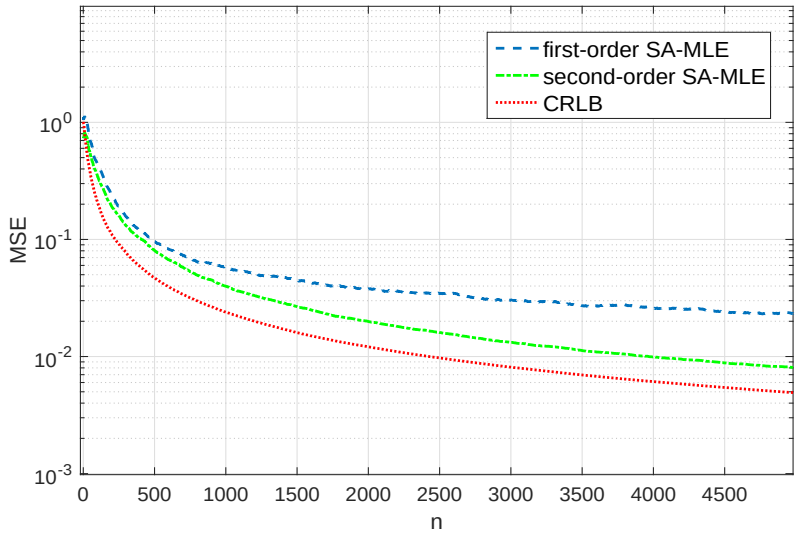
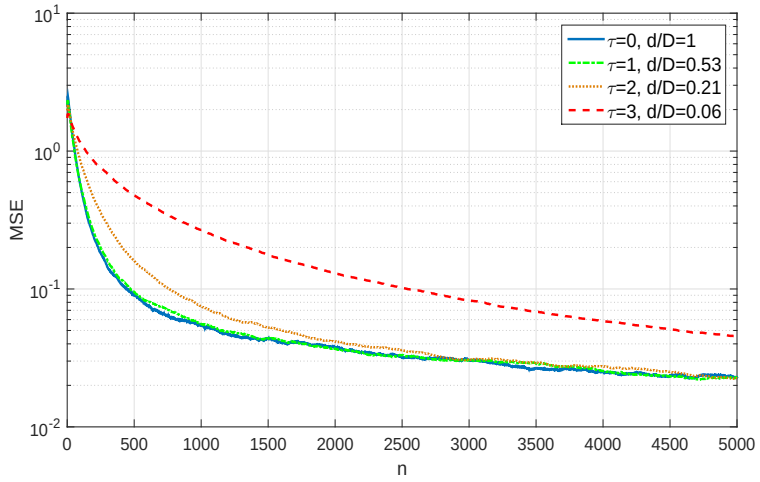


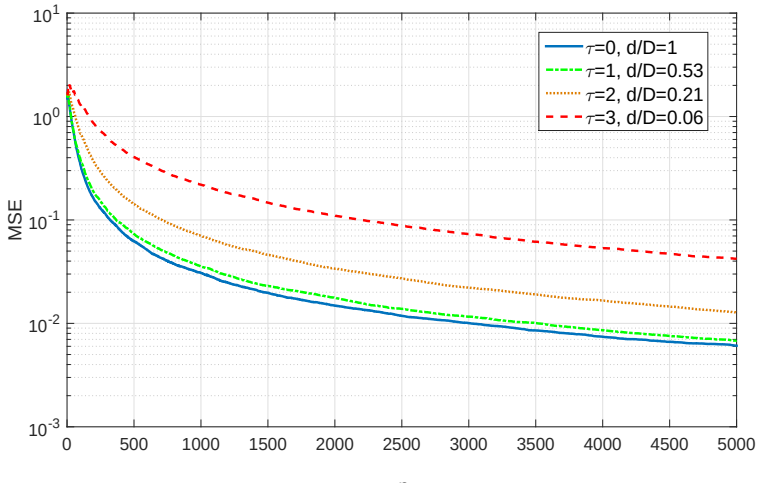
Figure 2.2: Convergence of first- and second-order SA-MLE ($d/D = 0.25$) .

2.4.2 AC-LMS comparison with Randomized Kaczmarz

The AC-LMS algorithm introduced in Section 2.3.1 was tested on synthetic data as an alternative to the randomized Kaczmarz's algorithm. For this experiment, $D = 30,000$ observations y_n were generated as in (2.1) with $\sigma^2 = 0.25$, while the $\mathbf{x}_n$'s of dimension $p = 100$ were generated i.i.d. following a multivariate Gaussian distribution. For the randomized Kaczmarz's algorithm, the probability of selecting the $i-$th row is $p_n = \|\mathbf{x}_n\|_2^2/\|\mathbf{X}\|_F^2$ [38]. Since the computational complexity of the two methods is roughly the same, the comparison was done in terms of the relative MSE, namely $\mathbb{E}\left[\|\boldsymbol{\theta}_o - \hat{\boldsymbol{\theta}}_n\|_2^2 \big/ \|\boldsymbol{\theta}_o\|_2^2\right]$. Plotted in Fig. 2.4, are the relative MSE curves of the two algorithms w.r.t. the number of data $\{\mathbf{x}_n, y_n\}$ that were used to estimate $\boldsymbol{\theta}_o$ (50 Monte Carlo runs). While the AC-LMS scans the entire dataset updating only informative data, the randomized Kaczmarz's algorithm needs access



(a)



(b)

Figure 2.3: Convergence of (a) first-order SA-MLE; and (b) second-order SA-MLE for different values of τ.

only to the data used for its updates. This is only possible if the data-dependent selection probabilities p_n are given a-priori, which may not always be the case. Regardless, two more experiments were run, in which the AC-LMS had limited access to 3,000 and 1,400 data. Overall, it can be argued that when the sought reduced dimension is small, the AC-LMS offers a simple and reliable first-order alternative to the randomized Kaczmarz's algorithm.

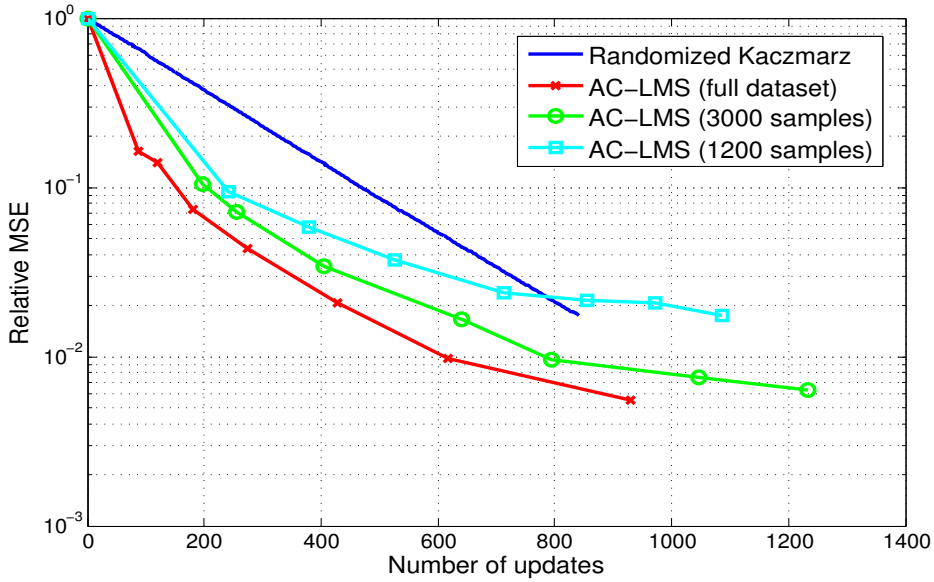


Figure 2.4: Relative MSE for AC-LMS and randomized Kaczmarz's algorithms.

2.4.3 AC-RLS

The AC-RLS algorithm developed in Section 2.3.2 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 (2.1) 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 2.5(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.

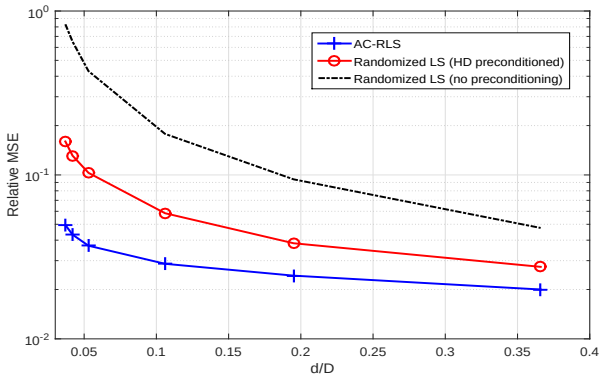
The experiment was repeated (Fig. 2.5(b)) for $\mathbf{x}_n$ generated from a multivariate $t-$distribution with 3 degrees of freedom, and $\mathbf{\Sigma}$ as before. Leverage scores for this dataset are moderately non-uniform, thus inducing more redundancy and resulting in lower performance for all 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. 2.5(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 $\boldsymbol{\theta}$.

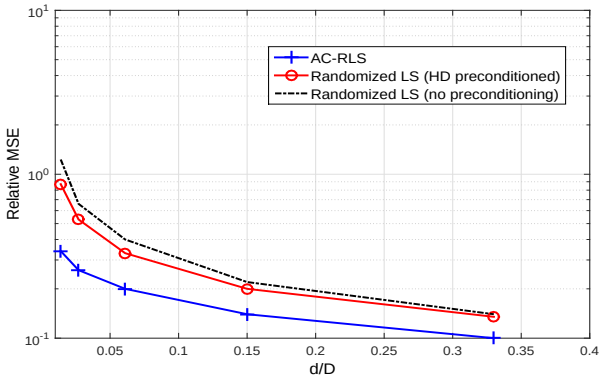
To further assess efficacy of the AC-RLS algorithm, real data tests were performed. The Protein Tertiary Structure dataset from the UCI Machine Learning Repository was tested. In this linear regression dataset, $p = 9$ attributes of proteins are used to predict a value related to protein structure. A total of $D = 45,730$ observations are included. 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 $\sigma^2 = (1/D)\sum_{n=1}^{D}(y_n - \mathbf{x}_n^T\boldsymbol{\theta}_o)^2$. Figure 2.6 depicts relative squared-error (RSE) with respect to the data reduction ratio d/D. The RSE curve for the HD-preconditioned LS corresponds to the average RSE across 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 for a given initialization and data order, the output of the algorithm is deterministic. 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).

2.4.4 Robust AC-RLS

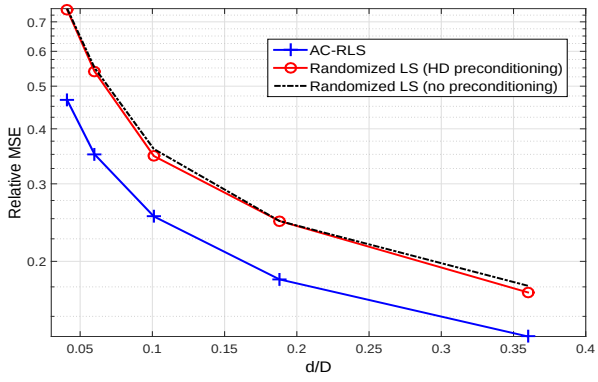
To test rAC-LMS and rAC-RLS of Section 2.3.4, datasets were generated with $D = 10,000$, $p = 30$ and $\mathbf{x}_n \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma})$, where $\mathbf{\Sigma}_{i,j} = 2 \times 0.5^{|i-j|}$; noise was i.i.d. Gaussian $v_n \sim \mathcal{N}(0, 9)$; meanwhile measurements y_n were generated according to (2.1) with random and sporadic outlier spikes $\{o_n\}_{n=1}^{D}$. Specifically, we generated $o_n = \alpha_n \beta_n$, where $\alpha_n \sim$ 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. 2.7 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.



(a)



(b)



(c)

Figure 2.5: 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).

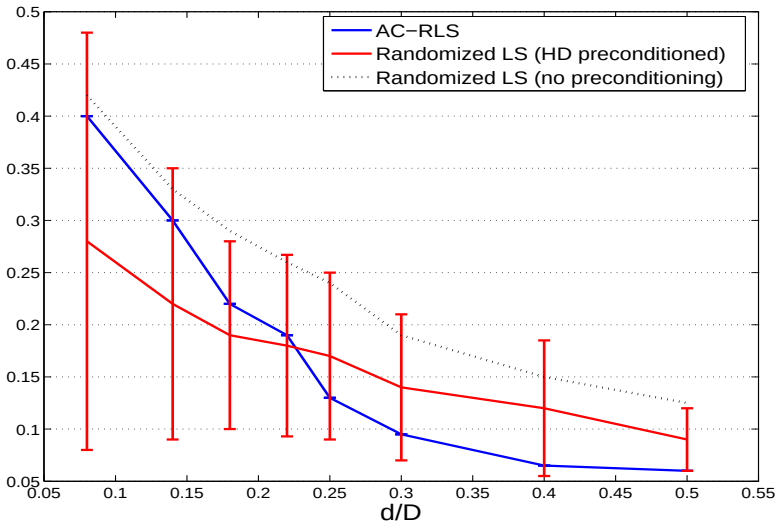


Figure 2.6: Relative MSE of AC-RLS and randomized LS algorithms, for different levels of data reduction using the protein tertiary structure dataset.

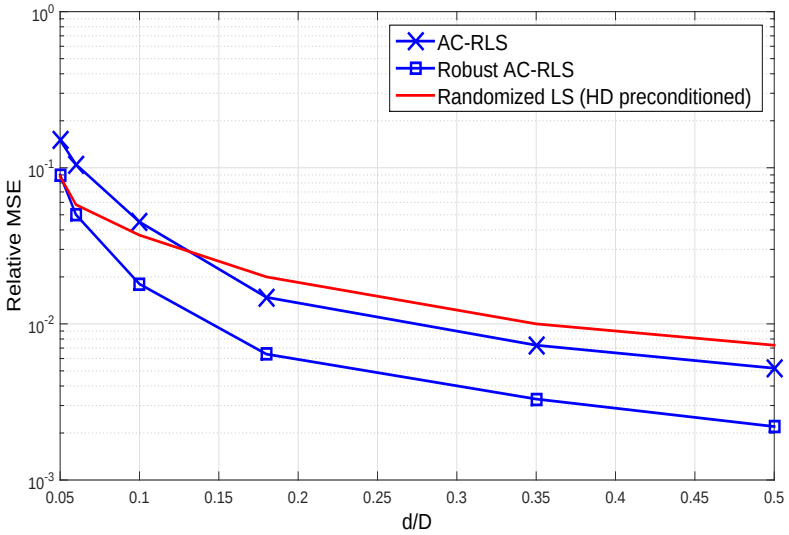


Figure 2.7: Relative MSE of AC-RLS, rAC-RLS, and randomized LS algorithms, for different levels of data reduction using an outlier-corrupted dataset.

Chapter 3

Censoring for Reduced-Complexity Tracking of Dynamical Processes

Tracking nonstationary dynamic processes is of paramount importance in various applications. In the context of big data, being able to perform accurate and economical state estimation may render problems of prohibitive scale feasible. Weather prediction is an example of tracking a slowly-varying dynamic process, from a massive volume of observations acquired from fast-sampling sensors at each time interval. Monitoring large and dynamically evolving networks, where nodes may join or leave and connections may be established or lost as time progresses, provides an exciting domain in which the acquisition and processing of network-wide performance metrics becomes challenging as the network-size increases. For instance, monitoring path metrics such as delays or loss rates is challenging primarily because the number of paths generally grows as the square of the number of nodes in the network. Therefore, measuring and storing the delays of all possible source-destination pairs is hard in practice even for moderate-size networks [17].

The objective of this chapter is to perform reliable tracking while reducing the amount of data and the computational complexity involved. Towards this goal, two algorithms are proposed for dimensionality reduction and tracking. The first, is based on RPs and thus is data-agnostic, while the second adopts censoring for joint tracking and rejection of "uninformative" data. Corroborating simulations compare with the state-of-the-art greedy

measurement selection algorithm, and illustrate the efficacy of the novel schemes.

3.1 Preliminaries

In the present chapter, a more general version of model (2.1) is considered, where parameters are allowed to vary across time while obeying known dynamics. Specifically, consider the following linear dynamical system model

$$\boldsymbol{\theta}_n = \mathbf{F}_n \boldsymbol{\theta}_{n-1} + \mathbf{w}_n \tag{3.1}$$

$$\mathbf{y}_n = \mathbf{X}_n \boldsymbol{\theta}_n + \mathbf{v}_n \tag{3.2}$$

where $\boldsymbol{\theta}_n \in \mathbb{R}^p$ denotes the state vector at time n; $\mathbf{F}_n$ is the known state-transition matrix; $\mathbf{y}_n \in \mathbb{R}^D$ the measurement vector and $\mathbf{X}_n$ is the known $D \times p$ measurement matrix; while $\mathbf{w}_n$ and $\mathbf{v}_n$ are zero-mean, mutually uncorrelated and individually uncorrelated across time random noise vectors, with respective covariance matrices $\mathbf{Q}_n$ and $\mathbf{R}_n$. The initial state $\boldsymbol{\theta}_0$ has mean $\mathbf{m}_0$ and covariance $\mathbf{P}_0$.

Given the information-bearing data $\mathcal{I}_n := \{\mathbf{y}_n, \mathbf{X}_n, \mathbf{R}_n\}$ at time n, the most recent estimate $\hat{\boldsymbol{\theta}}_{n-1|n-1}$ and its covariance matrix $\mathbf{P}_{n-1|n-1}$, the celebrated Kalman Filter (KF) yields the MMSE optimal estimate $\hat{\boldsymbol{\theta}}_{n|n}$ in two steps. First, the state prediction $\hat{\boldsymbol{\theta}}_{n|n-1}$ and its covariance matrix $\mathbf{P}_{n|n-1}$ are obtained using the model dynamics $\{\mathbf{F}_n, \mathbf{Q}_n\}$ as

$$\hat{\boldsymbol{\theta}}_{n|n-1} = \mathbf{F}_n \hat{\boldsymbol{\theta}}_{n-1|n-1} \tag{3.3a}$$

$$\mathbf{P}_{n|n-1} = \mathbf{F}_n \mathbf{P}_{n-1|n-1} \mathbf{F}_n^T + \mathbf{Q}_n. \tag{3.3b}$$

Subsequently, when data $\mathcal{I}_n$ become available, $\hat{\boldsymbol{\theta}}_{n|n}$ is obtained as

$$\hat{\boldsymbol{\theta}}_{n|n} = \arg\min_{\boldsymbol{\theta}} \|\mathbf{y}_n - \mathbf{X}_n \boldsymbol{\theta}\|^2_{\mathbf{R}_n^{-1}} + \|\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}_{n|n-1}\|^2_{\mathbf{P}^{-1}_{n|n-1}}. \tag{3.4}$$

The first term of the cost function in (3.4) is a weighted least-squares (WLS) term fitting the state $\boldsymbol{\theta}$ with $\mathcal{I}_n$ that arises from the linear observation model in (3.2); while the second regularization term corresponds to treating $\hat{\boldsymbol{\theta}}_{n|n-1}$ as a prior of $\boldsymbol{\theta}_n$.

Solving (3.4) and applying the matrix inversion lemma (MIL) yields the well-known KF correction step

$$\hat{\boldsymbol{\theta}}_{n|n} = \hat{\boldsymbol{\theta}}_{n|n-1} + \mathbf{K}_n(\mathbf{y}_n - \mathbf{X}_n \hat{\boldsymbol{\theta}}_{n|n-1})$$

where the so-termed KF gain $\mathbf{K}_n$ and the state covariance update are given by

$$\mathbf{K}_n = \mathbf{P}_{n|n-1}\mathbf{X}_n^T \left(\mathbf{X}_n \mathbf{P}_{n|n-1}\mathbf{X}_n^T + \mathbf{R}_n\right)^{-1}$$
$$\mathbf{P}_{n|n} = \left(\mathbf{I}_p - \mathbf{K}_n \mathbf{X}_n\right) \mathbf{P}_{n|n-1}.$$

A dual form of the KF known as the Information Filter (IF) has also been proposed as a more efficient solver of (3.4) as D grows large [3, Ch. 7]. Nevertheless, even the low-complexity IF requires $\mathcal{O}(Dp^2)$ multiplications to solve (3.4) in the case of uncorrelated observations ($\mathbf{R}_n$ diagonal), and $\mathcal{O}(D^2 p)$ in general. Therefore, for large-scale problems where $D \gg p$, dimensionality reduction of the datasets $\mathcal{I}_n$ is an attractive tool for rendering the solution of (3.4) computationally tractable, while also reducing other data-related costs, such as storage and transmission.

In this context, our goal in this chapter is to design computationally efficient methods for extracting a small (size $d < D$), yet informative dataset $\mathcal{I}_n^d := \{\check{\mathbf{y}}_n, \check{\mathbf{X}}_n, \check{\mathbf{R}}_n\}$ from the original $\mathcal{I}_n$; where $\check{\mathbf{y}}_n \in \mathbb{R}^d$, $\check{\mathbf{X}}_n \in \mathbb{R}^{d\times p}$ and $\check{\mathbf{R}}_n \in \mathbb{R}^{d\times d}$ are the corresponding reduced-dimension observation vector, measurement matrix, and covariance matrix; see also Fig. 3.1. In the ensuing two sections, a *data-agnostic* method based on RPs followed by a *data-adaptive* method based on censoring are proposed for reduced-complexity tracking of dynamical processes obeying (3.1) and (3.2).

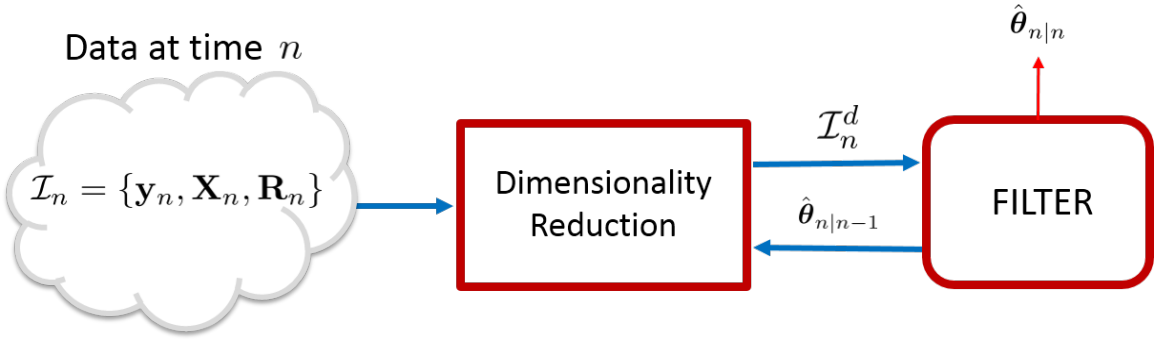


Figure 3.1: Dimensionality reduction for model-based estimation.

3.2 KF based on Random Projections

As briefly mentioned in Section 2.3.2, RP-based dimensionality reduction for LS amounts to premultiplying measurements and regressors $\{\mathbf{y}_n, \mathbf{X}_n\}$ with a random matrix $\mathbf{H}$, and a diagonal matrix $\mathbf{D}$, whose entries take the values $\{+1/\sqrt{D}, -1/\sqrt{D}\}$ equiprobably. The net result is to obtain a linear transformation of the system of equations in which all rows are of approximately equal importance. A subset of d rows of the transformed system is then extracted by simple random sampling, implemented by multiplication with a random $d \times D$ selection matrix $\mathbf{S}_d$.

Originally developed in the context of LS regression for time-invariant parameters, RPs can be readily adapted to reduce dimensionality in tracking dynamical processes too. Applying the Hadamard preconditioning and random sampling on (3.2) yields the reduced-dimension observation model

$$\check{\mathbf{y}}_n = \mathbf{S}_d\mathbf{H}\mathbf{D}\mathbf{y}_n = \mathbf{S}_d\mathbf{H}\mathbf{D}(\mathbf{X}_n\boldsymbol{\theta}_n + \mathbf{v}_n) = \check{\mathbf{X}}_n\boldsymbol{\theta}_n + \check{\mathbf{v}}_n \tag{3.5}$$

where $\check{\mathbf{v}}_n := \mathbf{S}_d\mathbf{H}\mathbf{D}\mathbf{v}_n$ is zero-mean with covariance $\check{\mathbf{R}}_n = \mathbf{S}_d\mathbf{H}\mathbf{D}\mathbf{R}_n(\mathbf{S}_d\mathbf{H}\mathbf{D})^T$. Given $\hat{\boldsymbol{\theta}}_{n|n-1}$ and the reduced data $\mathcal{I}_n^d$, state estimate $\hat{\boldsymbol{\theta}}_{n|n}$ can be obtained similar to (3.4) as

$$\hat{\boldsymbol{\theta}}_{n|n} = \arg\min_{\boldsymbol{\theta}} \|\check{\mathbf{y}}_n - \check{\mathbf{X}}_n\boldsymbol{\theta}\|^2_{\check{\mathbf{R}}_n^{-1}} + \|\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}_{n|n-1}\|^2_{\mathbf{P}^{-1}_{n|n-1}}. \tag{3.6}$$

Solving (3.6) and applying the MIL gives rise to the novel random-projections (RP)-KF tabulated as Algorithm 4. As mentioned in Section 2.3.2, implementing RPs can have affordable complexity if $\mathbf{H}$ is chosen to be a pseudo-random Hadamard matrix of size D. Different from the more elaborate approaches in [47] and [21], the proposed RP-KF is an easy-to-implement, "one-size-fits-all" reduced-complexity tracker, using data-agnostic dimensionality reduction. Furthermore, the RP-KF's estimation performance provides a benchmark for the data-driven censoring-based methods introduced in the following section.

Algorithm 4 RP-KF

$\{\mathbf{F}_n, \mathbf{Q}_n\}_{n=1}^N$ known to the FC

$\{\mathbf{X}_n, \mathbf{R}_n\}_{n=1}^N$ unknown

Initialization: $\hat{\boldsymbol{\theta}}_{0|0} = \mathbf{m}_0,\ \mathbf{P}_{0|0} = \mathbf{P}_0$

for $n = 1 : N$ **do**

Prediction Step:

$\hat{\boldsymbol{\theta}}_{n|n-1} = \mathbf{F}_n \hat{\boldsymbol{\theta}}_{n-1|n-1}$

$\mathbf{P}_{n|n-1} = \mathbf{F}_n \mathbf{P}_{n-1|n-1} \mathbf{F}_n^T + \mathbf{Q}_n$

Data Reduction with RPs:

$\check{\mathbf{y}}_n = \mathbf{S}_d \mathbf{H} \mathbf{D} \mathbf{y}_n$

$\check{\mathbf{X}}_n = \mathbf{S}_d \mathbf{H} \mathbf{D} \mathbf{X}_n$

$\check{\mathbf{R}}_n = \mathbf{S}_d \mathbf{H} \mathbf{D} \mathbf{R}_n (\mathbf{S}_d \mathbf{H} \mathbf{D})^T$

Correction Step:

$\hat{\boldsymbol{\theta}}_{n|n} = \hat{\boldsymbol{\theta}}_{n|n-1} + \mathbf{K}_n (\check{\mathbf{y}}_n - \check{\mathbf{X}}_n \hat{\boldsymbol{\theta}}_{n|n-1})$

$\mathbf{K}_n = \mathbf{P}_{n|n-1} \check{\mathbf{X}}_n^T \left(\check{\mathbf{X}}_n \mathbf{P}_{n|n-1} \check{\mathbf{X}}_n^T + \check{\mathbf{R}}_n \right)^{-1}$

$\mathbf{P}_{n|n} = \left(\mathbf{I}_p - \mathbf{K}_n \check{\mathbf{X}}_n \right) \mathbf{P}_{n|n-1}$

$\{\hat{\boldsymbol{\theta}}_{n|n}, \mathbf{P}_{n|n}\}$ are (possibly) stored

end for

3.3 BC-KF and AC-KF algorithms

Measurement censoring for estimating dynamical processes has recently been advocated as a means of reducing the inter-sensor transmission rate when wireless sensor networks are employed for distributed tracking [4, 46]; see also [19, 42], where censoring is employed for event-based estimation. Since the goal in the aforementioned applications is to reduce communication requirements, censoring is performed solely on measurements $\mathbf{y}_n$, with $\mathbf{X}_n$ and $\mathbf{R}_n$ assumed to be known; thus, in our notation $\mathcal{I}_n := \{\mathbf{y}_n\}$. A set of d observations $\mathcal{I}_n^d := \{[\mathbf{y}_n]_{\mathcal{S}_n}\}$ is obtained, where $[\mathbf{y}_n]_i$ denotes the $i-$th element of $\mathbf{y}_n$, and $\mathcal{S}_n \subseteq \{1, \ldots, D\}$ denotes a set collecting the indices of uncensored observations. Given $[\mathbf{y}_n]_{\mathcal{S}_n}, \mathbf{X}_n$ and $\mathbf{R}_n$, sequential estimators are then designed to optimally estimate $\boldsymbol{\theta}_n$. Interestingly, optimal (in

the ML or MMSE sense) estimation from censored observations comes with computational complexity that is comparable to that of using the full number of measurements.

As the goal of the present thesis is dimensionality and complexity reduction, the starting point is on censoring entire rows of the full dataset $\mathcal{I}_n^d := \{\mathbf{y}_n, \mathbf{X}_n, \mathbf{R}_n\}$, in order to obtain a reduced set $\mathcal{I}_n^d := \{[\mathbf{y}_n]_{\mathcal{S}_n}, [\mathbf{X}_n]_{\mathcal{S}_n}, [\mathbf{R}_n]_{\mathcal{S}_n}\}$, where $[\mathbf{X}_n]_i$ denotes the $i-$th row of $\mathbf{X}_n$ and $[\mathbf{R}_n]_{\mathcal{S}_n} := \mathrm{cov}([\mathbf{v}_n]_{\mathcal{S}_n})$. In this context, the objective is to develop censoring rules in order to obtain $\mathcal{S}_n$, so that $\mathcal{I}_n^d$ is an "informative" subset of $\mathcal{I}_n$.

Most existing censoring strategies consider the innovation $\tilde{\mathbf{y}}_n := \mathbf{y}_n - \mathbf{X}_n\hat{\boldsymbol{\theta}}_{n|n-1}$ as a measure of information contained in $\mathbf{y}_n$. One approach –henceforth referred to as *block censoring* (BC)– is to censor the entire vector $\mathbf{y}_n$. From an information-theoretic viewpoint [46], the optimal BC rule is based on the magnitude of the prewhitened innovation $\boldsymbol{\Sigma}_n^{-1/2}\tilde{\mathbf{y}}_n$, where $\boldsymbol{\Sigma}_n := \mathrm{cov}(\tilde{\mathbf{y}}_n) = \mathbf{X}_n\mathbf{P}_{n|n-1}\mathbf{X}_n^T + \mathbf{R}_n$; thus, $\mathcal{S}_n$ is obtained as

$$\mathcal{S}_n := \begin{cases} \{1, \ldots, D\}, & \|\boldsymbol{\Sigma}_n^{-1/2}\tilde{\mathbf{y}}_n\|_2 > \tau_n \\ \emptyset, & \text{otherwise} \end{cases}. \tag{3.7}$$

Clearly, having $\mathcal{S}_n = \emptyset$ corresponds to skipping the correction step of the KF. A major shortcoming of (3.7) is the cubic complexity $\mathcal{O}(D^3)$ associated with calculating $\boldsymbol{\Sigma}_n$. Furthermore, BC-KF may only reduce the data cost *on average* across iterations by skipping correction steps. Within a singe iteration however, it either incurs full complexity by using all D observations, or, no complexity when $\mathcal{I}_n$ is censored.

Our idea of a more attractive alternative is to censor each element of $\mathcal{I}_n$ separately. Such *entry-wise* censoring rules naturally arise in applications where the entries of $\mathbf{y}_n$ are observations collected from distributed and often uncorrelated sensors. In our context, entry-wise censoring yields $\mathcal{S}_n$ according to

$$\mathcal{S}_n := \{1 \leq i \leq D \;\Big|\; |[\tilde{\mathbf{y}}_n]_i| > \tau_n\} \tag{3.8}$$

where τ_n can be designed so that the set cardinality $|\mathcal{S}_n| \approx d$. Compared to BC-KF, the innovation-based entry-wise rule in (3.8) is not only more flexible in reducing the available data, but also simpler to implement and closer to the adaptive censoring of Chapter 2. However, it comes with the following limitation.

Proposition 4. *The KF using censoring rule (3.8) yields biased estimates.*

Proof: See Appendix.

As asserted by Proposition 4, naively using the rule in (3.8) to discard entries of $\mathcal{I}_n$ causes the KF to return biased estimates. However, the proof of Proposition 4 shows that the bias at time n is proportional to the factor $\tau_n(\boldsymbol{\theta}_n - \hat{\boldsymbol{\theta}}_{n|n-1})$. Since $\tau_n > 0$ is necessary to implement censoring, this bias can only be reduced by decreasing the prediction error $\boldsymbol{\theta}_n - \hat{\boldsymbol{\theta}}_{n|n-1}$. Towards this goal, the AC-LMS algorithm introduced in Section 2.3.1 can be used to adaptively censor uninformative rows of $\mathcal{I}_n$ (see Algorithm 6). By using an increasingly accurate tentative estimate to construct innovations $[\tilde{\mathbf{y}}_n]_i$, the proposed adaptive-censoring (AC)-KF, tabulated as Algorithm 5, yields reduced bias and estimation error.

Simulations in Section 3.5 will demonstrate that the proposed AC-KF attains estimation accuracy close to that of the KF using the greedy measurement selection method in [35]. More importantly, the proposed AC performs a single pass over the data, and requires $\mathcal{O}(Dp)$ computations, which is significantly less than the $\mathcal{O}(Ddp^2)$ order required to perform greedy selection. Furthermore, AC-KF is suitable for online implementation by processing rows of $\mathcal{I}_n$ sequentially devoid of the need for storage, while it can be readily modified for robustness to outliers, as outlined in Section 2.3.4.

Algorithm 5 AC-KF

$\{\mathbf{F}_n, \mathbf{Q}_n\}_{n=1}^N$ known to the FC

$\{\mathbf{X}_n, \mathbf{R}_n\}_{n=1}^N$ unknown

Initialization: $\hat{\boldsymbol{\theta}}_{0|0} = \mathbf{m}_0,\ \mathbf{P}_{0|0} = \mathbf{P}_0$

for $n = 1 : N$ **do**

Prediction Step:

$\hat{\boldsymbol{\theta}}_{n|n-1} = \mathbf{F}_n \hat{\boldsymbol{\theta}}_{n-1|n-1}$

$\mathbf{P}_{n|n-1} = \mathbf{F}_n \mathbf{P}_{n-1|n-1} \mathbf{F}_n^T + \mathbf{Q}_n$

Data Reduction with AC:

$\{\check{\mathbf{y}}_n, \check{\mathbf{X}}_n, \check{\mathbf{R}}_n\} = \text{Sketching}\Big(\{\mathbf{y}_n, \mathbf{X}_n, \mathbf{R}_n\},\ \hat{\boldsymbol{\theta}}_{n|n-1}\Big)$; as in Algorithm 6.

Correction Step:

$\hat{\boldsymbol{\theta}}_{n|n} = \hat{\boldsymbol{\theta}}_{n|n-1} + \mathbf{K}_n(\check{\mathbf{y}}_n - \check{\mathbf{X}}_n \hat{\boldsymbol{\theta}}_{n|n-1})$

$\mathbf{K}_n = \mathbf{P}_{n|n-1} \check{\mathbf{X}}_n^T \left(\check{\mathbf{X}}_n \mathbf{P}_{n|n-1} \check{\mathbf{X}}_n^T + \check{\mathbf{R}}_n\right)^{-1}$

$\mathbf{P}_{n|n} = \left(\mathbf{I}_p - \mathbf{K}_n \check{\mathbf{X}}_n\right) \mathbf{P}_{n|n-1}$

$\{\hat{\boldsymbol{\theta}}_{n|n}, \mathbf{P}_{n|n}\}$ are (possibly) stored

end for

Algorithm 6 Sketching module

Measurement selection with AC-LMS

Input: $\hat{\boldsymbol{\theta}}_{n|n-1},\ \ \{\mathbf{y}_n, \mathbf{X}_n, \mathbf{R}_n\}$

Initialization: $\hat{\boldsymbol{\theta}}_{n|n-1}^{(0)} = \hat{\boldsymbol{\theta}}_{n|n-1},\ \mathcal{S}_n^{(0)} = \emptyset$

for $i = 1 : D$ **do**

$c_i = \mathbb{1}\{\left|[\mathbf{y}_n]_i - [\mathbf{X}_n]_{i,:} \hat{\boldsymbol{\theta}}_{n|n-1}^{(i-1)}\right| \leq \tau_n\}$

if $c_i = 0$, **then**

$\mathcal{S}_n^{(i)} = \mathcal{S}_n^{(i-1)} \cup \{i\}$

end if

$\hat{\boldsymbol{\theta}}_{n|n-1}^{(i)} = \hat{\boldsymbol{\theta}}_{n|n-1}^{(i-1)} + \mu [\mathbf{X}_n]_{i,:} \left([\mathbf{y}_n]_i - [\mathbf{X}_n]_{i,:} \hat{\boldsymbol{\theta}}_{n|n-1}^{(i-1)}\right)$

end for

Return: $\{\check{\mathbf{y}}_n, \check{\mathbf{X}}_n, \check{\mathbf{R}}_n\} = \{[\mathbf{y}_n]_{\mathcal{S}_n^{(D)}}, [\mathbf{X}_n]_{\mathcal{S}_n^{(D)},:}, [\mathbf{R}_n]_{\mathcal{S}_n^{(D)}}\}$

3.4 Budgeted Fixed-Interval Smoothing

Given $\{\mathbf{F}_n, \mathbf{X}_n, \mathbf{Q}_n, \mathbf{R}_n\}_{n=1}^N$, the goal of *fixed-interval Kalman smoothing* (KS) is to estimate $\{\boldsymbol{\theta}_n\}_{n=1}^N$. To this end, consider expressing (3.2) for $n = 0, 1, \ldots, N$ as

$$\left[\begin{array}{cccc} -\mathbf{I}_p & & & \\ \mathbf{F}_1 & -\mathbf{I}_p & & \\ & \ddots & \ddots & \\ & & \mathbf{F}_N & -\mathbf{I}_p \\ \hline \mathbf{0} & \mathbf{X}_1 & & \\ \vdots & & \ddots & \\ \mathbf{0} & & & \mathbf{X}_N \end{array}\right] \left[\begin{array}{c} \boldsymbol{\theta}_0 \\ \boldsymbol{\theta}_1 \\ \boldsymbol{\theta}_2 \\ \vdots \\ \boldsymbol{\theta}_N \end{array}\right] + \left[\begin{array}{c} \mathbf{w}_0 \\ \mathbf{w}_1 \\ \vdots \\ \mathbf{w}_N \\ \hline \mathbf{v}_1 \\ \vdots \\ \mathbf{v}_N \end{array}\right] = \left[\begin{array}{c} -\mathbf{m}_0 \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \\ \hline \mathbf{y}_1 \\ \vdots \\ \mathbf{y}_N \end{array}\right]$$

or more compactly as the block-matrix-vector linear regression model

$$\mathbf{A}\boldsymbol{\theta} + \mathbf{w} = \mathbf{y} \tag{3.9}$$

where $\mathbf{A}, \boldsymbol{\theta}, \mathbf{w}, \mathbf{y}$ definitions are immediate, and $\mathbf{w}$ has block-diagonal covariance matrix

$$\mathbf{Q}_w := \mathrm{diag}\left(\mathbf{P}_0, \mathbf{Q}_1, \ldots, \mathbf{Q}_N, \mathbf{R}_1, \ldots, \mathbf{R}_N\right).$$

The WLS solution of (3.9) yields the batch KS estimate

$$\hat{\boldsymbol{\theta}}^{\mathrm{KS}} = \arg\min_{\boldsymbol{\theta}} (\mathbf{y} - \mathbf{A}\boldsymbol{\theta})^T \mathbf{Q}_w^{-1} (\mathbf{y} - \mathbf{A}\boldsymbol{\theta})$$

which can also be written explicitly as

$$\hat{\boldsymbol{\theta}}^{\mathrm{KS}} = \arg\min_{\boldsymbol{\theta}} \frac{1}{2} \sum_{n=1}^{N} \|\mathbf{y}_n - \mathbf{X}_n\boldsymbol{\theta}_n\|^2_{\mathbf{R}_n^{-1}} + \|\boldsymbol{\theta}_n - \mathbf{F}_n\boldsymbol{\theta}_{n-1}\|^2_{\mathbf{Q}_n^{-1}} + \|\boldsymbol{\theta}_0 - \mathbf{m}_0\|^2_{\mathbf{P}_n^{-1}} \tag{3.10}$$

and it is optimal in the linear minimum mean-square error (LMMSE) sense.

Aiming at a recursive solver of (3.10), one can rely on the Rauch-Tung-Stribel (RTS) forward-backward KS algorithm [32]. In its forward pass, the RTS algorithm is identical to the KF. The KF estimates $\{\hat{\boldsymbol{\theta}}_{n|n}\}_{n=1}^N$ are then stored and processed by the backward pass of the KS, while the error covariance matrices $\{\mathbf{P}_{n|n}\}_{n=1}^N$ are typically computed off-line.

Algorithm 7 The budgeted Kalman smoother (Bud-KS)

for $n = N-1:0$ **do**

  if $\hat{\boldsymbol{\theta}}_{n|n} \in \Theta_n^S$ **then**

    $\hat{\boldsymbol{\theta}}_{n|N} = \hat{\boldsymbol{\theta}}_{n|n}$

    $\mathbf{P}_{n|N} = \mathbf{P}_{n|n}$

  else

    $\hat{\boldsymbol{\theta}}_{n|N} = \hat{\boldsymbol{\theta}}_{n|n} + \mathbf{B}_n \left(\hat{\boldsymbol{\theta}}_{n+1|N} - \mathbf{F}_n \hat{\boldsymbol{\theta}}_{n|n} \right)$

    $\mathbf{B}_n = \mathbf{P}_{n|n} \mathbf{F}_n^T \mathbf{P}_{n+1|n}^{-1}$

    $\mathbf{P}_{n|N} = \mathbf{P}_{n|n} + \mathbf{B}_n \left(\mathbf{P}_{n+1|N} - \mathbf{P}_{n+1|n} \right) \mathbf{B}_n^T$

  end if

end for

Given $\hat{\boldsymbol{\theta}}_{n+1|N}$, the backward iteration solves

$$\hat{\boldsymbol{\theta}}_{n|N} := \arg\min_{\boldsymbol{\theta}} \|\hat{\boldsymbol{\theta}}_{n+1|N} - \mathbf{F}_n \boldsymbol{\theta}\|^2_{\mathbf{Q}_n^{-1}} + \|\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}_{n|n}\|^2_{\mathbf{P}_{n|n}^{-1}}. \tag{3.11}$$

Similar to filtering, the minimizer of (3.11) is also given in closed form as

$$\hat{\boldsymbol{\theta}}_{n|N} = \left(\mathbf{F}_n^T \mathbf{Q}_n^{-1} \mathbf{F}_n + \mathbf{P}_{n|n}^{-1} \right)^{-1} \left(\mathbf{F}_n^T \mathbf{Q}_n^{-1} \hat{\boldsymbol{\theta}}_{n+1|N} + \mathbf{P}_{n|n}^{-1} \hat{\boldsymbol{\theta}}_{n|n} \right).$$

After invoking the matrix inversion lemma and letting $\mathbf{B}_n = \mathbf{P}_{n|n} \mathbf{F}_n^T \mathbf{P}_{n+1|n}^{-1}$, the KS estimate $\hat{\boldsymbol{\theta}}_{n|N}$ can be given in the form of correction of $\hat{\boldsymbol{\theta}}_{n|n}$ as

$$\hat{\boldsymbol{\theta}}_{n|N} = \hat{\boldsymbol{\theta}}_{n|n} + \mathbf{B}_n \left(\hat{\boldsymbol{\theta}}_{n+1|N} - \mathbf{F}_n \hat{\boldsymbol{\theta}}_{n|n} \right) \tag{3.12a}$$

with corresponding error covariance matrix

$$\mathbf{P}_{n|N} = \mathbf{P}_{n|n} + \mathbf{B}_n \left(\mathbf{P}_{n+1|N} - \mathbf{P}_{n+1|n} \right) \mathbf{B}_n^T. \tag{3.13}$$

A key property of the backward KS iteration, is that it improves KF performance using from $\{\mathcal{I}_n\}_{n=1}^N$ only the information encapsulated in the output $\hat{\boldsymbol{\theta}}_{n|n}$ of the forward filter. Therefore, backward iterations can be readily applied on filtered estimates of RP-KF or AC-KF to limit the tracker's performance loss caused by the measurement reduction.

In addition, the backward iteration can also be modified to operate within a limited computational budget. Given the smoothed estimate at time $n+1$, let us define the set

$$\Theta_n^b := \left\{ \boldsymbol{\theta} \middle| \|\hat{\boldsymbol{\theta}}_{n+1|N} - \mathbf{F}_n \boldsymbol{\theta}\|^2_{\mathbf{Q}_n^{-1}} \leq \tau_b \right\} \tag{3.14}$$

of states at time n that are consistent enough with the transition model, in the WLS sense. Based on (3.14), the Bud-KS estimate at time n is given as

$$\hat{\boldsymbol{\theta}}_{n|N} = \begin{cases} \hat{\boldsymbol{\theta}}_{n|n}, & \hat{\boldsymbol{\theta}}_{n|n} \in \Theta_n^b \\ \hat{\boldsymbol{\theta}}_{n|n} + \mathbf{B}_n \left(\hat{\boldsymbol{\theta}}_{n+1|N} - \mathbf{F}_n \hat{\boldsymbol{\theta}}_{n|n} \right), & \hat{\boldsymbol{\theta}}_{n|n} \notin \Theta_n^b. \end{cases} \tag{3.15}$$

Clearly, for $\hat{\boldsymbol{\theta}}_{n|n} \in \Theta_n^b$, it holds that $\mathbf{P}_{n|N} = \mathbf{P}_{n|n}$; while for $\hat{\boldsymbol{\theta}}_{n|n} \notin \Theta_n^b$, the error covariance is given by (3.13). Essentially, KS estimates that are consistent enough with the system model are not smoothed, thus saving the computations required. Here, τ_b can be tuned to control the amount of "acceptable" deviation from the model. The novel economical, fixed-interval smoothing algorithm that we abbreviate as Bud-KS, is tabulated as Algorithm 7.

3.5 Numerical Tests

The novel AC-KF and RP-KF algorithms are tested here on a simulated linear dynamical system modeling a random spiral trajectory, which consists of a rotation on the x $-$ y plane and a downward movement along the z axis. The state transition matrix of such a model is

$$\mathbf{F}_n = \begin{bmatrix} \cos(\phi) & \sin(\phi) & 0 \\ -\sin(\phi) & \cos(\phi) & 0 \\ 0 & 0 & a_z \end{bmatrix}, \quad \forall n$$

where ϕ determines the angular speed of rotation set to $\pi/60$, and a_z the rate of descent set to 0.997. The state noise $\{\mathbf{w}_n\}_{n=1}^N$ was generated i.i.d. with $\mathbf{w}_n \sim \mathcal{N}(\mathbf{0}, \sigma_w^2 \mathbf{Q}_n)$, where $[\mathbf{Q}_n]_{i,j} = 0.5^{|i-j|}$ and $\sigma_w = 0.02$. Finally, the initial state is $\boldsymbol{\theta}_0 \sim \mathcal{N}(\mathbf{m}_0, \mathbf{P}_0)$, with $\mathbf{m}_0 = [1, 1, 10]^T$ and $\mathbf{P}_0 = 0.09\mathbf{I}$.

Per time instant $n \in \{1, \ldots, N\}$ with $N = 100$, $D = 1000$ measurements are obtained and concatenated in vector $\mathbf{y}_n = \mathbf{X}_n^T \boldsymbol{\theta}_n + \mathbf{v}_n$, where rows of $\mathbf{X}_n$ are generated as i.i.d.

standardized Gaussian vectors. For this experiment, observations are assumed correlated; thus, $\mathbf{v}_n \sim \mathcal{N}(0, \sigma_v^2 \mathbf{R}_n)$ where $[\mathbf{R}_n]_{i,j} = 0.5^{|i-j|}$. In Fig. 3.2, the true system trajectory is shown by the solid blue line, while the trajectory of the AC-KF estimates $\{\boldsymbol{\theta}_{n|n}\}_{n=1}^N$ with $d \approx 50$ observations per time slot is the dashed green line. Also, plotted with dotted red is the trajectory of the smoothed estimates $\{\boldsymbol{\theta}_{n|N}\}_{n=1}^N$. Evidently, by judiciously censoring a large number of measurements, the trajectory may still be recovered with relatively high accuracy.

3.5.1 AC-KF and RP-KF

To determine the average performance in terms of estimation error and computational complexity of AC-KF and RP-KF for different values of d/D, 20 Monte Carlo realizations were run on the same simulated linear dynamical system. The experiment was repeated for three different levels of signal-to-noise-ratio (SNR) at the observation model. High ($\sigma_v^2 = 25 \times 10^{-4}$), average ($\sigma_v^2 = 2 \times 10^{-2}$) and low ($\sigma_v^2 = 1$) SNR cases were considered. The estimation performance was measured in terms of, averaged across realizations, root-mean-square error (RMSE) of the estimates across iterations; that is,

$$\text{RMSE} = \frac{1}{N} \sqrt{\sum_{n=1}^{N} \|\hat{\boldsymbol{\theta}}_{n|n} - \boldsymbol{\theta}_n\|}.$$

AC-KF was run first, with $\tau_n = \tau$ tuned so that a constant number of approximately d observations were selected per time slot; RP-KF and the greedy algorithm were then set to obtain d measurements per time slot. As a performance benchmark for the three algorithms, KF was also run with d randomly sampled observations per time step.

The average RMSE of the four methods as a function of d/D is plotted in Figs. 3.3, 3.4 and 3.5, for high, average and low SNR, respectively. These plots confirm that the proposed data-agnostic RP-KF is useful for increasing the accuracy (compared to plain random sampling) when estimating dynamic processes. With regards to the more elaborate algorithms, the proposed AC-KF has comparable performance with the KF using greedy measurement selection, while being orders of magnitude faster in terms of runtime. Furthermore, the gap between the estimation accuracy of the two methods closes as SNR decreases, indicating

that the AC-KF is more robust to noisy observations.

3.5.2 Bud-KS

In the last experiment, the extent to which backward smoothing iterations can improve reduced-observation filtering was examined. The AC-KF algorithm was first run for the low SNR model with d/D ranging from 0.0095 up to 0.65; Bud-KF was then run with $\tau_b = 0$ in order to smooth all N filtered estimates. Figure 3.6 depicts the average RMSE of the AC-KF with and without smoothing. Evidently, smoothing can significantly reduce RMSE over the entire range of dimensionality reduction, while its effect becomes more prominent as d/D decreases. Upon examining Fig. 3.6, the AC-KF using $< 1\%$ of the data followed by Bud-KS, attains the same RMSE as the AC-KF using 5% of the data; a surprising five-fold decrease. Thus, at the cost of introducing non-causality (or delay if a fixed-lag KS is used), smoothing offers room for significant decrease in the data requirements and complexity of tracking.

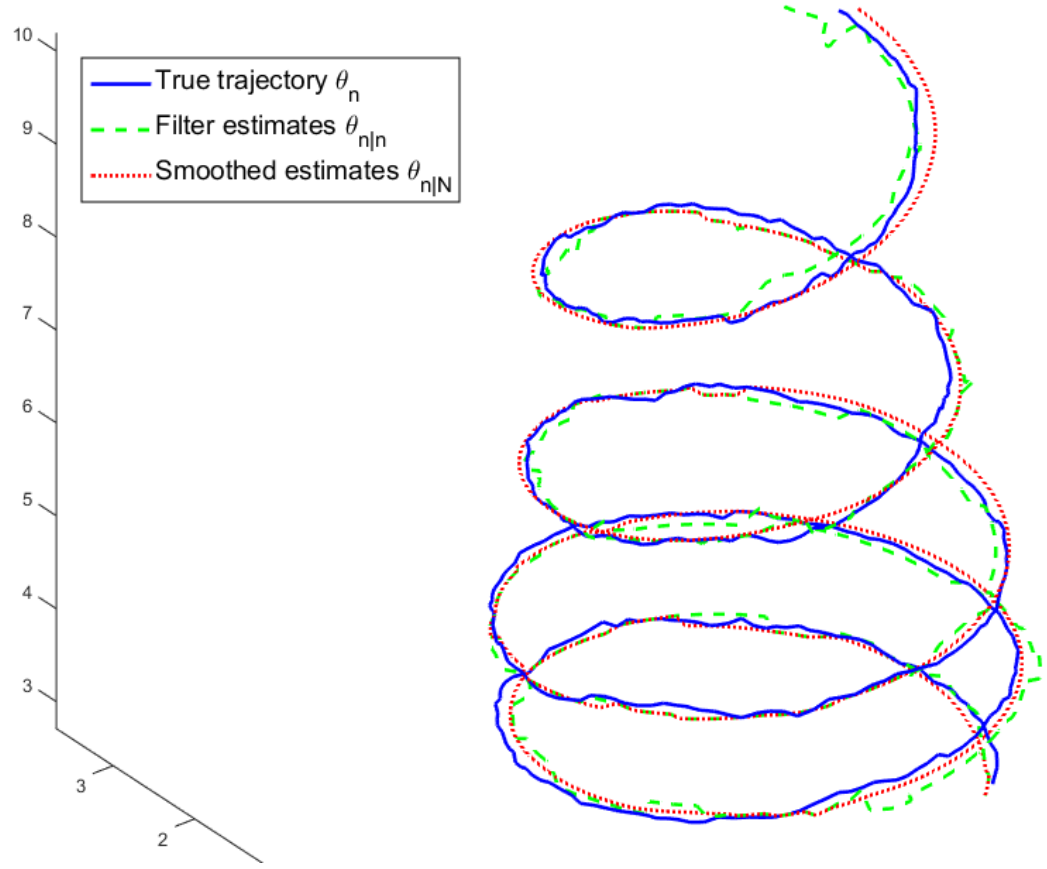


Figure 3.2: Tracking trajectory of a linear dynamical process (**solid blue**). Filtered (**dashed green**) and smoothed estimates (**red dotted**).

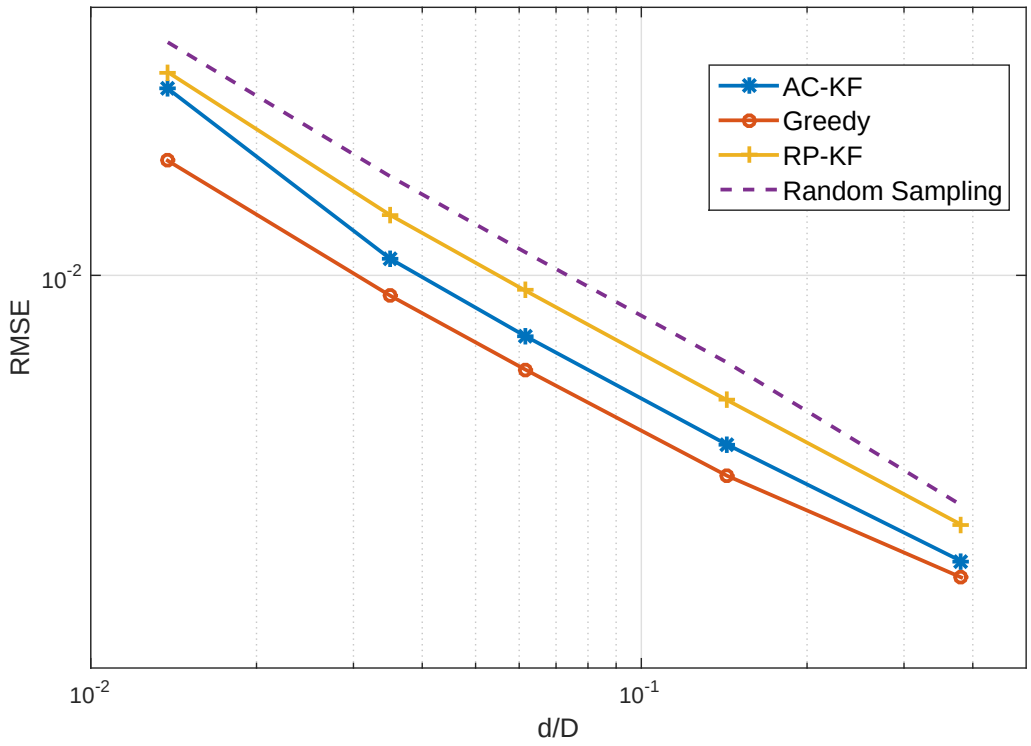


Figure 3.3: Average RMSE for AC-KF, Greedy algorithm, RP-KF and random sampling as a function of data reduction ratio d/D. High SNR case with $\sigma_v^2 = 25 \times 10^{-4}$.

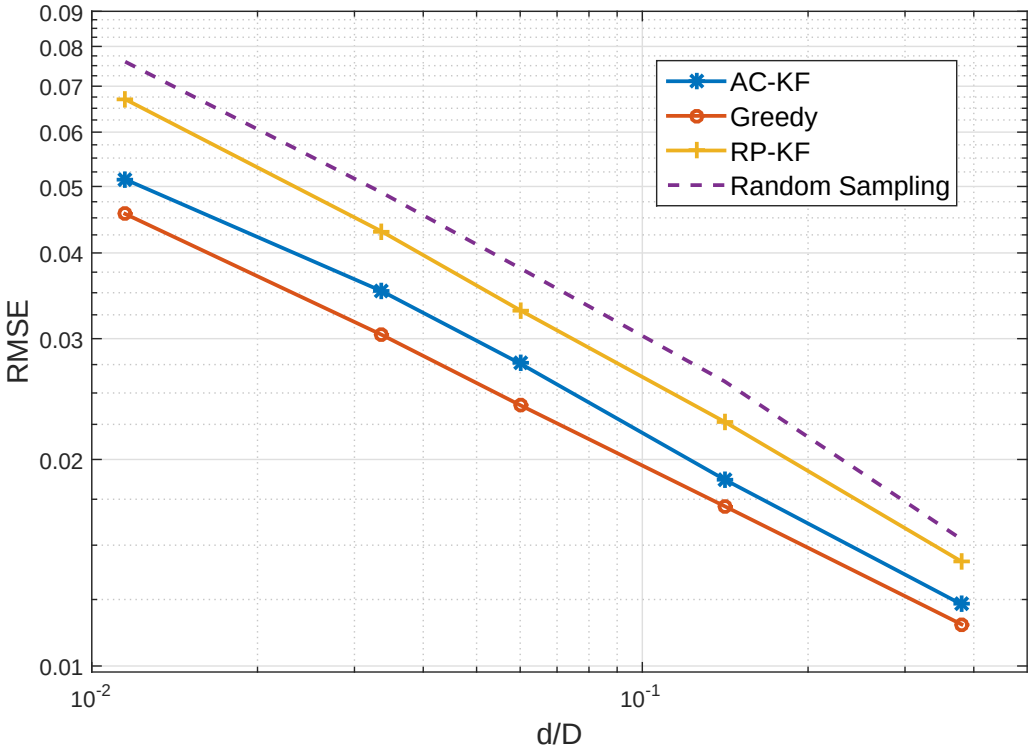


Figure 3.4: Average RMSE for AC-KF, Greedy algorithm, RP-KF and random sampling as a function of data reduction ratio d/D. Average SNR case with $\sigma_v^2 = 4 \times 10^{-2}$.

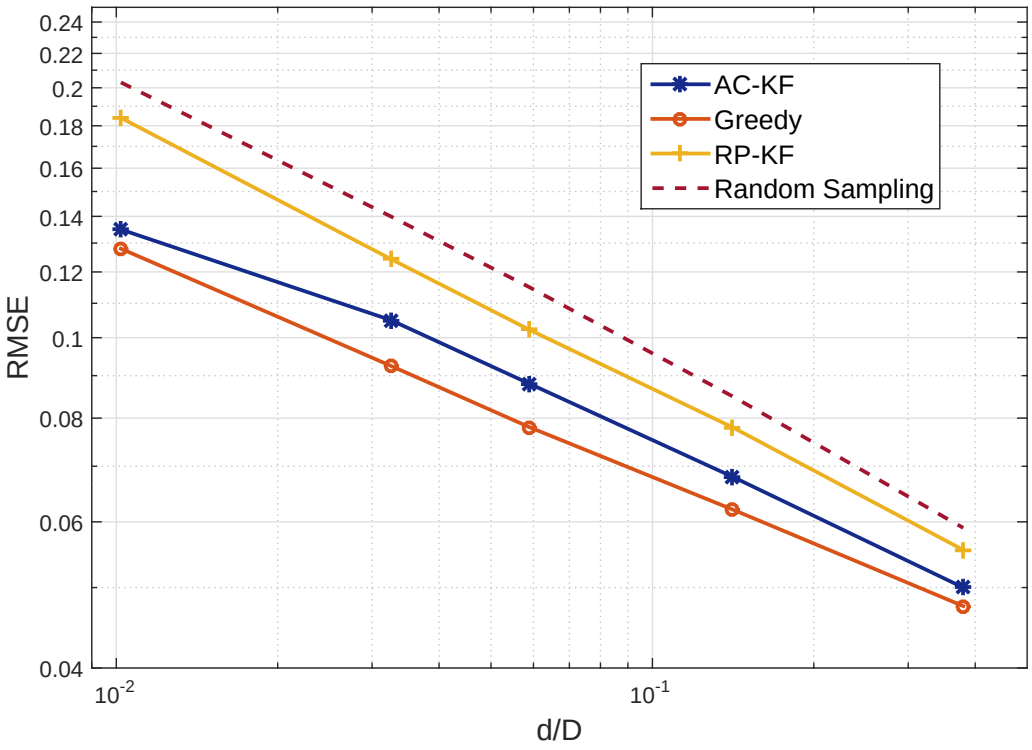


Figure 3.5: Average RMSE for AC-KF, Greedy algorithm, RP-KF and random sampling as a function of data reduction ratio d/D. High SNR case with $\sigma_v^2 = 1$.

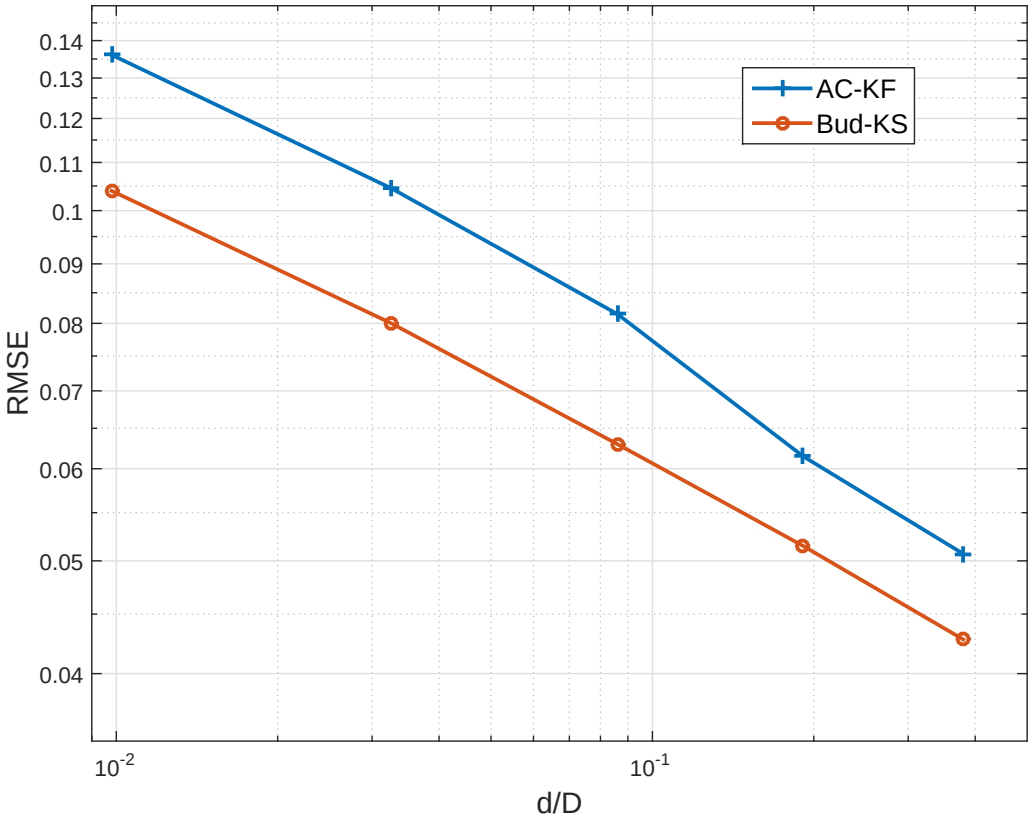


Figure 3.6: RMSE of AC-KF versus Bud-KS, as a function of the data reduction ratio d/D.

Chapter 4

Concluding remarks and outlook

4.1 Summary

In this thesis, online censoring was considered for decreasing data processing costs in two different estimation tasks. In the first one, online algorithms were developed for large-scale linear regressions based on censoring to effect *data-driven* dimensionality reduction of streaming Big Data. First, a non-adaptive censoring setting was adopted for applications where observations are censored – possibly naturally – separately from 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 censored data log-likelihood. Performance was bounded analytically, while simulations demonstrated that the second-order method performs close to the CRLB.

Furthermore, online data reduction effected jointly with estimation was also explored. For this scenario, censoring was 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. Studied under the scope 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.

In the second task, we introduced RPs and censoring as dimensionality reduction and measurement selection methods for tracking dynamical processes with possibly time-varying parameters. Performance was not analytically performed, but simulations provide surprisingly strong evidence that the proposed AC-KF performs close to the greedy measurement selection method in terms of estimation error. Furthermore, censoring-based measurement selection enjoys much lower computational complexity than its greedy alternative, while also being capable of processing streaming observations of dynamically evolving processes (e.g, over large-scale networks [17]) online.

4.2 Future directions

Our future research agenda includes approaches to nonlinear (e.g., kernel-based) parametric and nonparametric large-scale regressions. Regarding estimation of dynamical (e.g., state-space) processes using adaptively censored measurements, future work will be focused on achieving the following goals:

1. Providing performance analysis of the AC-KF, and developing a deeper understanding on how censoring influences tracking performance.

2. Developing accurate threshold selection rules in order for the AC-KF to use a predetermined number of measurements.

3. Generalizing AC measurement selection for nonlinear filtering; e.g., Extended and Unscented KF, as well as particle filtering (PF).

4. Pursuing Big Data inference of dynamical processes evolving over large-scale networks, where AC-KF and/or RP-KF can play a key role in reducing data-related costs.

Bibliography

[1] 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.

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

[3] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation with Applications to Tracking and Navigation: Theory, Algorithms and Software.* John Wiley & Sons, 2004.

[4] G. Battistelli, A. Benavoli, and L. Chisci, "Data-driven strategies for selective data transmission in sensor networks," in *Proc. of 51st Annual Conference on Decision and Control*, Grand Wailea, Maui, 2012, pp. 800–805.

[5] D. Berberidis and G. B. Giannakis, "Budgeted kalman filtering and smoothing for economical tracking with big distributed data," in *Proc. of 49th Asilomar Conf. on Signals, Systems and Computers (to appear)*, Pacific Grove, CA, Nov. 2015.

[6] D. Berberidis, V. Kekatos, and G. B. Giannakis, "Online censoring for large-scale regressions with application to streaming big data," *arXiv preprint submit/1313747*, 2015.

[7] D. Berberidis, G. Wang, G. B. Giannakis, and V. Kekatos, "Online censoring for large-scale regressions," in *Proc. of 48th Asilomar Conf. on Signals, Systems and Computers*, Pacific Grove, CA, Nov. 2014, pp. 14–18.

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

[9] 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.

[10] 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.

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

[12] P. Drineas, M. W. Mahoney, and S. Muthukrishnan, "Sampling algorithms for l 2 regression and applications," in *Proc. of the 17-th Annual SIAM-ACM Symp. on Discrete Algorithms*, 2006, pp. 1127–1136.

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

[14] 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, May 1998.

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

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

[17] E. D. Kolaczyk, *Statistical analysis of network data*. Springer, 2009.

[18] Y. Li and L. Chen, "Big biological data: Challenges and opportunities," *Genomics, proteomics & bioinformatics*, vol. 12, no. 5, pp. 187–189, 2014.

[19] Q. Liu, Z. Wang, X. He, and D. Zhou, "A survey of event-based strategies on control and estimation," *Systems Science & Control Engineering Open Access Journal*, vol. 2, no. 1, pp. 90–97, 2014.

[20] P. Louka, G. Galanis, N. Siebert, G. Kariniotakis, P. Katsafados, I. Pytharoulis, and G. Kallos, "Improvements in wind speed forecasts for wind power prediction purposes using Kalman filtering," *J. of Wind Engin. and Indust. Aero.*, vol. 96, no. 12, pp. 2348–2362, Nov. 2008.

[21] H. Ma, Y.-H. Yang, Y. Chen, K. R. Liu, and Q. Wang, "Distributed state estimation with dimension reduction preprocessing," *IEEE Trans. Sig. Proc.*, vol. 62, no. 12, pp. 3098–3110, Dec. 2014.

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

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

[24] S. Maleki and G. Leus, "Censored truncated sequential spectrum sensing for cognitive radio networks," *IEEE J. on Selected Areas in Comm.*, vol. 31, no. 3, pp. 364–378, March 2013.

[25] 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.

[26] 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.

[27] 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, Jan. 2012.

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

[29] 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.

[30] F. Pukelsheim, *Optimal Design of Experiments*. SIAM, 1993, vol. 50.

[31] K. Rajawat, E. Dall'Anese, and G. Giannakis, "Dynamic network delay cartography," *IEEE Trans. Info. Theory*, vol. 60, no. 5, pp. 2910–2920, May 2014.

[32] H. E. Rauch, C. Striebel, and F. Tung, "Maximum likelihood estimates of linear dynamic systems," *AIAA Journal*, vol. 3, no. 8, pp. 1445–1450, 1965.

[33] 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.

[34] S. Shalev-Shwartz, "Online learning and online convex optimization," *Foundations and Trends in Machine Learning*, vol. 4, no. 2, pp. 107–194, 2011.

[35] M. Shamaiah, S. Banerjee, and H. Vikalo, "Greedy sensor selection: Leveraging submodularity," in *Proc. of 49th IEEE Conference on Decision and Control*, Atlanta, Dec. 2010, pp. 2572–2577.

[36] K. Slavakis, S.-J. Kim, G. Mateos, and G. 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, Nov. 2014.

[37] 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.

[38] 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.

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

[40] G. Wang, D. Berberidis, V. Kekatos, and G. B. Giannakis, "Online reconstruction from big data via compressive censoring," in *Proc. of IEEE Global Conf. on Signal and Inf. Process.*, Atlanta, GA, Dec. 2014.

[41] G. Wang, J. Chen, J. Sun, and Y. Cai, "Power scheduling of Kalman filtering in wireless sensor networks with data packet drops," *arXiv preprint arXiv:1312.3269v2*, 2013.

[42] Y. Wang, V. Krishnaswami, and G. Rizzoni, "Event-based estimation of indicated torque for ic engines using sliding-mode observers," *Control Engineering Practice*, vol. 5, no. 8, pp. 1123–1129, 1997.

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

[44] 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.

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

[46] Y. Zheng, R. Niu, and P. K. Varshney, "Sequential bayesian estimation with censored data for multi-sensor systems," *IEEE Trans. Sig. Proc.*, vol. 62, no. 10, pp. 2626–2641, Oct. 2014.

[47] H. Zhu, I. D. Schizas, and G. B. Giannakis, "Power-efficient dimensionality reduction for distributed channel-aware Kalman tracking using WSNs," *IEEE Trans. Sig. Proc.*, vol. 57, no. 8, pp. 3193–3207, Aug. 2009.

[48] L. Zuo, R. Niu, and P. K. Varshney, "Posterior CRLB based sensor selection for target tracking in sensor networks," in *Proc. of International Conference on Acoustics, Speech and Signal Processing*, vol. 2, Honolulu, Hawaii, 2007, pp. II–1041.

Appendices

Proof of Proposition 1

It can be verified that $\nabla^2 \ell_n(\boldsymbol{\theta}) \succeq \mathbf{0}$, which implies the convexity of $\ell_n(\boldsymbol{\theta})$ [27]. The regret of the SGD approach is then bounded as [34, Corollary 2.7]

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

where $\{\boldsymbol{\theta}_n\}_{n=1}^D$ is any sequence of estimates produced by the SA-MLE. By choosing

$$\mu = \|\boldsymbol{\theta}^* - \hat{\boldsymbol{\theta}}_K\|_2/(\sqrt{2D}\bar{\beta}\bar{x}),$$

the aforementioned bound leads to Proposition 1.

Proof of Proposition 2

For the SGD update in (2.22), the MSE $\mathbb{E}_{\mathbf{x},v}\left[\|\boldsymbol{\theta}_n - \boldsymbol{\theta}_o\|_2^2\right]$, with $\boldsymbol{\theta}_o = \arg\min_{\boldsymbol{\theta}} F(\boldsymbol{\theta})$ where $F(\boldsymbol{\theta}) := \mathbb{E}_{\mathbf{x},v}\left[f^{(\tau)}(\boldsymbol{\theta}; \mathbf{y})\right]$ is bounded as in [26]. For this result to hold, the stochastic gradient must be bounded at the optimum $\mathbb{E}_{\mathbf{x},v}\left[\|\nabla f^{(\tau)}(\boldsymbol{\theta}_o, \mathbf{y})\|_2^2\right] \leq \Delta$; it must be $L-$smooth for any other $\boldsymbol{\theta}$; and $F(\boldsymbol{\theta})$ has to be α-strongly convex [26]. Note that, by assuming both $\mathbf{x}$ and v to be generated randomly and independently across time, associated quantities do not depend on n. Furthermore, the points of discontinuity of $f^{(\tau)}(\cdot)$ are zero-measure in expectation and thus are neglected for simplicity.

Starting with the last one, the function $F(\boldsymbol{\theta})$ is $\alpha-$strongly convex if there exists a constant $\alpha > 0$ such that $\nabla^2 F(\boldsymbol{\theta}) \succeq \alpha \mathbf{I}$ for all $\boldsymbol{\theta}$. Upon interchanging differentiation and expectation

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

$$
\begin{aligned}
&+ Q\left(\tau - \frac{\mathbf{x}^T(\boldsymbol{\theta}_o - \boldsymbol{\theta})}{\sigma}\right)\Bigg] p_x(\mathbf{x}) d\mathbf{x} \\
&= \int_{\mathbf{x}} \mathbf{x}\mathbf{x}^T \Bigg[Q\left(\tau + \frac{\mathbf{x}^T(\boldsymbol{\theta}_o - \boldsymbol{\theta})}{\sigma}\right) \\
&+ Q\left(\tau - \frac{\mathbf{x}^T(\boldsymbol{\theta}_o - \boldsymbol{\theta})}{\sigma}\right)\Bigg] p_x(\mathbf{x}) d\mathbf{x}.
\end{aligned}
$$

It can be easily verified that the function $g(z) := Q(\tau + z) + Q(\tau - z)$ is minimized for $z = 0$ when $\tau > 0$. To see this, observe that its derivative $g'(z) = -\phi(\tau + z) + \phi(\tau - z)$ vanishes when $|\tau + z| = |\tau - z|$. Therefore, $g(z) \geq g(0) = 2Q(\tau)$ for all z; and hence,

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

for all $\mathbf{x}$ and $\boldsymbol{\theta}$. The latter implies

$$
\begin{aligned}
\nabla^2 F(\boldsymbol{\theta}) &\succeq \int_{\mathbf{x}} \mathbf{x}\mathbf{x}^T 2Q(\tau) f_x(\mathbf{x}) \partial\mathbf{x} = 2Q(\tau)\mathbf{R}_x \\
&\succeq 2Q(\tau)\lambda_{\min}(\mathbf{R}_x)\mathbf{I}\,.
\end{aligned}
$$

Thus, $F(\boldsymbol{\theta})$ 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 any pair $(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2)$. To that end,

$$
\begin{aligned}
&\mathbb{E}_{\mathbf{x},v}\left[\|\nabla f^{(\tau)}(\boldsymbol{\theta}_1) - \nabla f^{(\tau)}(\boldsymbol{\theta}_2)\|_2^2\right] \\
&= \mathbb{E}\left[\|\mathbf{x}e(\boldsymbol{\theta}_1)(1 - c_1) - \mathbf{x}e(\boldsymbol{\theta}_2)(1 - c_2)\|_2^2\right] \\
&= \mathbb{E}_{\mathbf{x},v}\Bigg[\|\mathbf{x}(\mathbf{x}^T\boldsymbol{\zeta}_1 + v)\mathbb{1}_{\{|\mathbf{x}^T\boldsymbol{\zeta}_1 + v| \geq \tau\sigma\}} \\
&- \mathbf{x}(\mathbf{x}^T\boldsymbol{\zeta}_2 + v)\mathbb{1}_{\{|\mathbf{x}^T\boldsymbol{\zeta}_2 + v| \geq \tau\sigma\}}\|_2^2\Bigg] \\
&= \mathbb{E}_{\mathbf{x},v}\Big[\|\mathbf{x}\mathbf{x}^T\boldsymbol{\zeta}_1\mathbb{1}_{\{|\mathbf{x}^T\boldsymbol{\zeta}_1 + v| \geq \tau\sigma\}} - \mathbf{x}\mathbf{x}^T\boldsymbol{\zeta}_2\mathbb{1}_{\{|\mathbf{x}^T\boldsymbol{\zeta}_2 + v| \geq \tau\sigma\}} \\
&+ \mathbf{x}v(\mathbb{1}_{\{|\mathbf{x}^T\boldsymbol{\zeta}_1 + v| \geq \tau\sigma\}} - \mathbb{1}_{\{|\mathbf{x}^T\boldsymbol{\zeta}_2 + v| \geq \tau\sigma\}})\|_2^2\Big] \\
&= \mathbb{E}_{\mathbf{x},v}\Bigg[\boldsymbol{\zeta}_1^T\left(\mathbf{x}\mathbf{x}^T\right)^2\mathbb{1}_{\{|\mathbf{x}^T\boldsymbol{\zeta}_1 + v| \geq \tau\sigma\}} + \boldsymbol{\zeta}_2^T\left(\mathbf{x}\mathbf{x}^T\right)^2\mathbb{1}_{\{|\mathbf{x}^T\boldsymbol{\zeta}_2 + v| \geq \tau\sigma\}}
\end{aligned}
$$

$$
\begin{aligned}
&- 2\boldsymbol{\zeta}_1^T \left(\mathbf{x}\mathbf{x}^T\right)^2 \boldsymbol{\zeta}_2 \mathbb{1}_{\{|\mathbf{x}^T\boldsymbol{\zeta}_1+v|\geq\tau\sigma\}} \mathbb{1}_{\{|\mathbf{x}^T\boldsymbol{\zeta}_2+v|\geq\tau\sigma\}} \\
&+ \mathbf{x}^T\mathbf{x}\mathbf{x}^T\boldsymbol{\zeta}_1 \mathbb{1}_{\{|\mathbf{x}^T\boldsymbol{\zeta}_1+v|\geq\tau\sigma\}} v \left(\mathbb{1}_{\{|\mathbf{x}^T\boldsymbol{\zeta}_1+v|\geq\tau\sigma\}} - \mathbb{1}_{\{|\mathbf{x}^T\boldsymbol{\zeta}_2+v|\geq\tau\sigma\}}\right) \\
&- \mathbf{x}^T\mathbf{x}\mathbf{x}^T\boldsymbol{\zeta}_2 \mathbb{1}_{\{|\mathbf{x}^T\boldsymbol{\zeta}_2+v|\geq\tau\sigma\}} v \left(\mathbb{1}_{\{|\mathbf{x}^T\boldsymbol{\zeta}_1+v|\geq\tau\sigma\}} - \mathbb{1}_{\{|\mathbf{x}^T\boldsymbol{\zeta}_2+v|\geq\tau\sigma\}}\right) \\
&+ \|\mathbf{x}\|_2^2 v^2 \left(\mathbb{1}_{\{|\mathbf{x}^T\boldsymbol{\zeta}_1+v|\geq\tau\sigma\}} - \mathbb{1}_{\{|\mathbf{x}^T\boldsymbol{\zeta}_2+v|\geq\tau\sigma\}}\right)^2 \Bigg]. \qquad \text{(A1)}
\end{aligned}
$$

It can be verified that since the cross-terms in (A1) can be bounded from below and above as

$$
\begin{aligned}
&\mathbb{E}_{\mathbf{x}}\left[\mathbf{x}^T\mathbf{x}\mathbf{x}^T\right]\boldsymbol{\zeta}_1 L(\boldsymbol{\zeta}_1,\boldsymbol{\zeta}_2) \leq \mathbb{E}_{\mathbf{x}}\Big[\mathbf{x}^T\mathbf{x}\mathbf{x}^T\boldsymbol{\zeta}_1 \\
&\times \mathbb{E}_v\left[\mathbb{1}_{\{|\mathbf{x}^T\boldsymbol{\zeta}_1+v|\geq\tau\sigma\}} v \left(\mathbb{1}_{\{|\mathbf{x}^T\boldsymbol{\zeta}_1+v|\geq\tau\sigma\}} - \mathbb{1}_{\{|\mathbf{x}^T\boldsymbol{\zeta}_2+v|\geq\tau\sigma\}}\right)\right]\Big] \\
&\leq \mathbb{E}_{\mathbf{x}}\left[\mathbf{x}^T\mathbf{x}\mathbf{x}^T\right]\boldsymbol{\zeta}_1 U(\boldsymbol{\zeta}_1,\boldsymbol{\zeta}_2),
\end{aligned}
$$

they are also equal to zero if the third-order moment $\mathbb{E}_{\mathbf{x}}\left[\mathbf{x}^T\mathbf{x}\mathbf{x}^T\right] = \mathbf{0}$. Furthermore, by simply bounding $\mathbb{E}_v\left[\mathbb{1}_{\{|\mathbf{x}^T\boldsymbol{\zeta}_i+v|\geq\tau\sigma\}}\right] \leq 1$ as probabilities, (A1) yields

$$
\begin{aligned}
&\mathbb{E}\left[\|\nabla f(\boldsymbol{\theta}_1) - \nabla f(\boldsymbol{\theta}_2)\|_2^2\right] \leq \mathbb{E}_{\mathbf{x}}\Bigg[(\boldsymbol{\zeta}_1-\boldsymbol{\zeta}_2)^T\left(\mathbf{x}\mathbf{x}^T\right)^2(\boldsymbol{\zeta}_1-\boldsymbol{\zeta}_2) \\
&+ \|\mathbf{x}\|_2^2 \mathbb{E}_v\left[v^2\left(\mathbb{1}_{\{|\mathbf{x}^T\boldsymbol{\zeta}_1+v|\geq\tau\sigma\}} - \mathbb{1}_{\{|\mathbf{x}^T\boldsymbol{\zeta}_2+v|\geq\tau\sigma\}}\right)^2\right]\Bigg] \\
&= (\boldsymbol{\zeta}_1-\boldsymbol{\zeta}_2)^T\mathbb{E}_{\mathbf{x}}\left[\left(\mathbf{x}\mathbf{x}^T\right)^2\right](\boldsymbol{\zeta}_1-\boldsymbol{\zeta}_2) \\
&+ \mathbb{E}_{\mathbf{x}}\left[\|\mathbf{x}\|_2^2\mathbb{E}_v\left[v^2\left(\mathbb{1}_{\{|\mathbf{x}^T\boldsymbol{\zeta}_1+v|\geq\tau\sigma\}} - \mathbb{1}_{\{|\mathbf{x}^T\boldsymbol{\zeta}_2+v|\geq\tau\sigma\}}\right)^2\right]\right] \\
&\leq \left(\lambda_{\max}\left(\mathbb{E}\left[\left(\mathbf{x}\mathbf{x}^T\right)^2\right]\right) + \lambda_\tau\right)\|\boldsymbol{\theta}_1-\boldsymbol{\theta}_2\|_2^2.
\end{aligned}
$$

However, the last expression reveals that the average distance between gradients can be decomposed into two parts. The first is a quadratic cost that can be bounded using the fourth-order moment. The second term appears due to data censoring and clearly depends on τ, while it is assumed bounded as

$$
\begin{aligned}
&\mathbb{E}_{\mathbf{x}}\left[\|\mathbf{x}\|_2^2\mathbb{E}_v\left[v^2\left(\mathbb{1}_{\{|\mathbf{x}^T\boldsymbol{\zeta}_1+v|\geq\tau\sigma\}} - \mathbb{1}_{\{|\mathbf{x}^T\boldsymbol{\zeta}_2+v|\geq\tau\sigma\}}\right)^2\right]\right] \\
&\leq \lambda_\tau\|\boldsymbol{\theta}_1-\boldsymbol{\theta}_2\|_2^2.
\end{aligned}
$$

Although we could not express λ_τ in closed form, for relatively small values of τ used in practice to censor more than 90% of the measurements, $\lambda_\tau \approx 0$; thus, the second term can be

ignored yielding $L^2 \approx \lambda_{\max}\left(\mathbb{E}\left[\left(\mathbf{x}\mathbf{x}^T\right)^2\right]\right)$. Furthermore, even for large τ some inaccuracy in the value of L can be tolerated, after considering that it does not affect the algorithm's stability or asymptotic performance when a vanishing step size is used.

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

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

which completes the proof.

Proof of Proposition 3

For the error $\boldsymbol{\zeta}_n := \boldsymbol{\theta}_n - \boldsymbol{\theta}_o$, AC-RLS satisfies $\boldsymbol{\zeta}_n = \mathbf{C}_n \sum_{i=1}^{n} \mathbf{x}_i v_i (1-c_i)$. 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

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

With $\mathbf{x}_n\mathbf{x}_n^T(1-c_n)$ assumed ergodic, we have for n large enough $\mathbf{C}_n^{-1} = \sum_{i=1}^{n} \mathbf{x}_i\mathbf{x}_i^T(1-c_i) \approx n\mathbb{E}_{\mathbf{x},v}\left[\mathbf{x}\mathbf{x}^T(1-c)\right] = n\mathbb{E}_{\mathbf{x}}\left[\mathbf{x}\mathbf{x}^T\mathbb{E}_v[1-c]\right] = n\mathbb{E}_{\mathbf{x}}\left[\mathbf{x}\mathbf{x}^T\Pr\{c=0|\mathbf{x}\}\right] = \mathbf{C}_\infty^{-1}$. However, since $2Q(\tau) \leq \Pr\{c=0|\mathbf{x}\} \leq 1\ \forall\mathbf{x}$, we have $2Q(\tau)n\mathbf{R}_x \preceq \mathbf{C}_\infty^{-1} \preceq n\mathbf{R}_x$. Consequently, if

$\mathbf{C}_n \succ \mathbf{0}\ \forall n$, it holds that $\mathbf{C}_n \to \mathbf{C}_\infty = [\mathbf{C}_\infty^{-1}]^{-1}$. Since $\mathbf{C}_n$ converges monotonically, 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 given that $\mathbb{E}\left[\|\boldsymbol{\theta}_n - \boldsymbol{\theta}_o\|_2^2\right] = \mathrm{tr}(\mathbf{K}_n) = \sigma^2 \mathrm{tr}(\mathbb{E}\left[\mathbf{C}_n\right])$.

Proof of Proposition 4

Suppose that randomness in the state and measurement equations (3.1) and (3.1) comes only from the noise variables. Thus, solving (3.6) in batch form via WLS allows expressing the KF based on censored data estimate estimate at time n, as

$$\begin{aligned}\hat{\boldsymbol{\theta}}_{n|n} &= \left(\check{\mathbf{X}}_n^T\check{\mathbf{R}}_n^{-1}\check{\mathbf{X}}_n + \mathbf{P}_{n|n-1}^{-1}\right)^{-1}\left(\check{\mathbf{X}}_n^T\check{\mathbf{R}}_n^{-1}\check{\mathbf{y}}_n + \mathbf{P}_{n|n-1}^{-1}\hat{\boldsymbol{\theta}}_{n|n-1}\right)\\ &= \left(\check{\mathbf{X}}_n^T\check{\mathbf{R}}_n^{-1}\check{\mathbf{X}}_n + \mathbf{P}_{n|n-1}^{-1}\right)^{-1}\left(\check{\mathbf{X}}_n^T\check{\mathbf{R}}_n^{-1}\check{\mathbf{X}}_n\boldsymbol{\theta}_n + \mathbf{P}_{n|n-1}^{-1}\hat{\boldsymbol{\theta}}_{n|n-1}\right)\\ &+ \left(\check{\mathbf{X}}_n^T\check{\mathbf{R}}_n^{-1}\check{\mathbf{X}}_n + \mathbf{P}_{n|n-1}^{-1}\right)^{-1}\check{\mathbf{X}}_n^T\check{\mathbf{R}}_n^{-1}\check{\mathbf{v}}_n. \end{aligned} \tag{A2}$$

We will prove that the corrector $\hat{\boldsymbol{\theta}}_{n|n}$ is biased, even when the predictor is unbiased, meaning $\mathbb{E}[\hat{\boldsymbol{\theta}}_{n|n-1}] = \boldsymbol{\theta}_{n|n}$. Using the latter along with (A2), we arrive at

$$\mathbb{E}[\hat{\boldsymbol{\theta}}_{n|n}] = \boldsymbol{\theta}_n + \mathbf{b}_n$$

where the bias term is

$$\mathbf{b}_n := \mathbb{E}[\left(\check{\mathbf{X}}_n^T\check{\mathbf{R}}_n^{-1}\check{\mathbf{X}}_n + \mathbf{P}_{n|n-1}^{-1}\right)^{-1}\check{\mathbf{X}}_n^T\check{\mathbf{R}}_n^{-1}\check{\mathbf{v}}_n].$$

To facilitate the analysis, assume that $\mathbf{P}_{n|n-1}^{-1}$ is large enough so that the influence of the term $\check{\mathbf{X}}_n^T\check{\mathbf{R}}_n^{-1}\check{\mathbf{X}}_n$ can be ignored. Then, it follows that

$$\mathbf{b}_n \approx \mathbf{P}_{n|n-1}^{-1}\mathbb{E}[\check{\mathbf{X}}_n^T\check{\mathbf{R}}_n^{-1}\check{\mathbf{v}}_n] = \mathbf{P}_{n|n-1}^{-1}\sum_{i=1}^{D}[\mathbf{X}_n^T\mathbf{R}_n^{-1}]_{:,i}g_n^{(i)} \tag{A3}$$

where

$$\begin{aligned} g_n^{(i)} &:= \mathbb{E}[(1-c_i)v_n^{(i)}] = \mathbb{E}_{\hat{\boldsymbol{\theta}}_{n|n-1}}[\mathbb{E}_{v_n^{(i)}}[(1-c_i)v_n^{(i)}]]\\ &= \mathbb{E}_{\hat{\boldsymbol{\theta}}_{n|n-1}}[\mathbb{E}_{v_n^{(i)}}[\mathbb{1}_{\{|\mathbf{x}_n^T(\boldsymbol{\theta}_n - \hat{\boldsymbol{\theta}}_{n|n-1}) + v_n^{(i)}| \geq \tau_n\}}v_n^{(i)}]\\ &\propto \mathbb{E}_{\hat{\boldsymbol{\theta}}_{n|n-1}}\left[-\exp -(-\tau_n - \mathbf{x}_n^T(\boldsymbol{\theta}_n - \hat{\boldsymbol{\theta}}_{n|n-1}))^2 + \exp -(\tau_n - \mathbf{x}_n^T(\boldsymbol{\theta}_n - \hat{\boldsymbol{\theta}}_{n|n-1}))^2\right]\\ &= \mathbb{E}_{\hat{\boldsymbol{\theta}}_{n|n-1}}\left[-\exp 2\tau_n\mathbf{x}_n^T(\boldsymbol{\theta}_n - \hat{\boldsymbol{\theta}}_{n|n-1}) + \exp -2\tau_n\mathbf{x}_n^T(\boldsymbol{\theta}_n - \hat{\boldsymbol{\theta}}_{n|n-1})\right] \end{aligned}$$

$$= \mathbb{E}_{\hat{\boldsymbol{\theta}}_{n|n-1}}\Big[- 4\tau_n \mathbf{x}_n^T(\boldsymbol{\theta}_n - \hat{\boldsymbol{\theta}}_{n|n-1}) - 2(\tau_n \mathbf{x}_n^T(\boldsymbol{\theta}_n - \hat{\boldsymbol{\theta}}_{n|n-1}))^3/3!$$
$$- 2(\tau_n \mathbf{x}_n^T(\boldsymbol{\theta}_n - \hat{\boldsymbol{\theta}}_{n|n-1}))^5/5! - \dots \Big]$$
$$= \mathbb{E}_{\hat{\boldsymbol{\theta}}_{n|n-1}}\Big[-2(\tau_n \mathbf{x}_n^T(\boldsymbol{\theta}_n - \hat{\boldsymbol{\theta}}_{n|n-1}))^3/3! - 2(\tau_n \mathbf{x}_n^T(\boldsymbol{\theta}_n - \hat{\boldsymbol{\theta}}_{n|n-1}))^5/5! - \dots\Big] \neq 0. \qquad \text{(A4)}$$

Upon inspecting (A4), it is clear that unbiasedness of $\hat{\boldsymbol{\theta}}_{n|n-1}$ is not enough to nullify the bias-inducing coefficients $\{g_n^{(i)}\}$ that depend on higher-order moments of the predictor. Since $g_n^{(i)} \neq 0, \forall i = 1, \dots, D$, it follows from (A3) that the bias $\mathbf{b}_n$ is also non-zero, in general.