

Compressed sensing of streaming data

Nikolaos M. Freris*, Orhan Öçal* and Martin Vetterli*

Abstract—We introduce a recursive scheme for performing Compressed Sensing (CS) on streaming data and analyze, both analytically and experimentally, the computational complexity and estimation error. The approach consists of sampling the input stream recursively via overlapping windowing and making use of the previous measurement in obtaining the next one. The signal estimate from the previous window is utilized in order to achieve faster convergence in an iterative optimization algorithm to decode the new window. To remove the bias of the estimator a two-step estimation procedure is proposed comprising support set detection and signal amplitude estimation. Estimation accuracy is enhanced by averaging estimates obtained from overlapping windows. The proposed method is shown to have asymptotic computational complexity $O(nm^{3/2})$, where n is the window length, and m is the number of samples. The variance of normalized estimation error is shown to asymptotically go to 0 if $\kappa = O(n^{1-\epsilon})$ as n increases. The simulation results show speed up of at least ten times with respect to applying traditional CS on a stream of data while obtaining significantly lower reconstruction error under mild conditions on the signal magnitudes and the noise level.

I. INTRODUCTION

In signal processing, it is often the case that signals of interest can be represented sparsely by using few coefficients in an appropriately selected orthonormal basis; take, for example, the Fourier basis for bandlimited signals or wavelet bases for piecewise continuous signals, e.g., images. While a small number of coefficients in the respective bases may be enough to represent such signals with high accuracy, the celebrated Nyquist/Shannon sampling theorem suggests a sampling rate that is at least twice the signal bandwidth, which, in many cases, is much higher than the sufficient number of coefficients [1], [2].

The Compressed Sensing (CS) framework was introduced aiming at sampling the signals not according to their bandwidth, but rather their *information* content, that is, the number of degrees of freedom. This sampling paradigm suggests a lower sampling rate compared to the classical sampling theory for signals that have sparse representation in some fixed basis [1], [2], e.g., biomedical imaging.

The foundations of the CS have been developed in [3], [4]. Although the field has been studied for nearly a decade, a recursive algorithm for performing CS on streaming data still remains by and large unaddressed. This paper studies the computational complexity and stability of signal estimation from noisy samples, when applying CS on an input stream through successive overlapping windowing.

*School of Computer and Communication Sciences, École Polytechnique Fédérale de Lausanne (EPFL), CH-1015 Lausanne, Switzerland. {nikolaos.freris, orhan.ocal, martin.vetterli}@epfl.ch

II. BACKGROUND

A. Notation

Throughout the paper, we use capital boldface letters to denote matrices (e.g., \mathbf{A}) and boldface lowercase letters to denote vectors (e.g., \mathbf{x}). We use x_i to denote the i^{th} entry of vector \mathbf{x} and \mathbf{a}_i to denote the i^{th} column of matrix \mathbf{A} . The i^{th} sampling instance (e.g., i^{th} window of the input stream, i^{th} sampling matrix, i^{th} sample) is denoted by superscript (e.g., $\mathbf{x}^{(i)}$, $\mathbf{A}^{(i)}$, $\mathbf{y}^{(i)}$). $|\mathcal{S}|$ denotes the cardinality of a set \mathcal{S} , and we use $\mathbb{E}_x[\cdot]$ to denote the conditional expectation $\mathbb{E}_x[\cdot] = \mathbb{E}[\cdot|x]$ and $\{x_i\}$ as the shorthand notation for the infinite sequence $\{x_i\}_{i=0,1,\dots}$.

B. Definitions and Properties

Definition 1 (κ -sparsity): For a vector $\mathbf{x} \in \mathbb{R}^n$ we define its support $\text{supp}(\mathbf{x}) := \{i : x_i \neq 0\}$ and the ℓ_0 pseudonorm $\|\mathbf{x}\|_0 := |\text{supp}(\mathbf{x})|$. We say that a vector \mathbf{x} is κ -sparse if and only if $\|\mathbf{x}\|_0 \leq \kappa$.

Definition 2 (Mutual Coherence): For a matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$, the mutual coherence is defined as the largest normalized inner product between any two different columns of \mathbf{A} [5]:

$$\mu(\mathbf{A}) := \max_{\substack{0 \leq i, j \leq n-1 \\ i \neq j}} \frac{|\mathbf{a}_i^\top \mathbf{a}_j|}{\|\mathbf{a}_i\|_2 \cdot \|\mathbf{a}_j\|_2}.$$

Definition 3 (Restricted Isometry Property): Let $\mathbf{A} \in \mathbb{R}^{m \times n}$. For given $0 < \kappa < n$, the matrix \mathbf{A} is said to satisfy the Restricted Isometry Property (RIP) if there exists $\delta_\kappa \in [0, 1]$ such that:

$$(1 - \delta_\kappa) \|\mathbf{x}\|_2^2 \leq \|\mathbf{A}\mathbf{x}\|_2^2 \leq (1 + \delta_\kappa) \|\mathbf{x}\|_2^2$$

holds for all $\mathbf{x} \in \mathbb{R}^n$ κ -sparse vectors, where δ_κ is sufficiently small [2].

Random matrices have been used as CS matrices in literature since they satisfy RIP with high probability [2]. Examples of matrices satisfying the RIP are a) n random vectors sampled from the m -dimensional unit sphere uniformly at random [2], b) random partial Fourier matrices obtained by selecting m rows from the n dimensional Fourier matrix uniformly at random, c) random Gaussian matrices having i.i.d. entries $A_{i,j} \sim \mathcal{N}(0, 1/m)$, d) random Bernoulli matrices where $A_{i,j} \in \{1/\sqrt{m}, -1/\sqrt{m}\}$ with equal probability. For the last two cases, \mathbf{A} satisfies a prescribed δ_κ for any $\kappa \leq c_1 m / \log(n/\kappa)$ with probability $\geq 1 - 2e^{-c_2 m}$, where constants c_1 and c_2 depend only on δ_κ [6].

Given measurements of vector $\mathbf{x} \in \mathbb{R}^n$:

$$\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{w}, \quad (1)$$

where $\mathbf{y} \in \mathbb{R}^m$ is the vector of obtained samples, $\mathbf{A} \in \mathbb{R}^{m \times n}$ is called the *sampling (sensing) matrix* and $\mathbf{w} \in \mathbb{R}^m$ represents additive measurement noise; our goal is to recover \mathbf{x} when $m \ll n$. In such setting, one can solve the ℓ_1 regularized least squares problem, best known as Least Absolute Selection and Shrinkage Operator (LASSO) in the statistics literature [7]:

$$\begin{aligned} & \text{minimize} && \|\mathbf{x}\|_1 \\ & \text{subject to} && \|\mathbf{A}\mathbf{x} - \mathbf{y}\|_2 \leq \tilde{\sigma}, \end{aligned} \quad (2)$$

where $\tilde{\sigma} \geq \|\mathbf{w}\|_2$ accounts for the additive noise in the measurements.

By duality, the problem can be posed equivalently [8] as an unconstrained optimization problem

$$\text{minimize} \quad \|\mathbf{A}\mathbf{x} - \mathbf{y}\|_2^2 + \lambda \|\mathbf{x}\|_1, \quad (3)$$

where λ is the regularization parameter that controls the trade-off between sparsity and reconstruction error.

Theorem 2.1 (Error of LASSO [2]): If \mathbf{A} satisfies RIP with $\delta_{2\kappa} < \sqrt{2} - 1$, the solution \mathbf{x}_* to (2) obeys:

$$\|\mathbf{x}_* - \mathbf{x}\|_2 \leq c_0 \cdot \|\mathbf{x} - \mathbf{x}_\kappa\|_1 / \sqrt{\kappa} + c_1 \cdot \tilde{\sigma}, \quad (4)$$

for constants c_0 and c_1 , where \mathbf{x}_κ is the vector \mathbf{x} with all but the largest κ components set to 0.

Theorem 2.1 states that the reconstruction error is upper bounded by the sum of two terms: the first is the error due to *model mismatch*, and the second is proportional to the *measurement noise variance*. In particular, if \mathbf{x} is κ -sparse and $\delta_{2\kappa} < \sqrt{2} - 1$ then $\|\mathbf{x}_* - \mathbf{x}\|_2 \leq c_1 \cdot \tilde{\sigma}$.

Definition 4 (Generic κ -sparse model): Let $\mathbf{x} \in \mathbb{R}^n$ denote a κ -sparse signal and $I_{\mathbf{x}} := \text{supp}(\mathbf{x})$ be its support set. Signal \mathbf{x} is said to be generated by *generic κ -sparse model* if:

- 1) Support $I_{\mathbf{x}} \subset \{1, 2, \dots, n\}$ of \mathbf{x} is selected uniformly at random, and $|I_{\mathbf{x}}| = \kappa$.
- 2) Conditioned on $I_{\mathbf{x}}$, signs of non zero elements are independent and equally likely to be -1 and 1 .

Theorem 2.2 (Support Detection [9]): Assume $\mu(\mathbf{A}) \leq c_1 / \log n$ for some constant $c_1 > 0$, \mathbf{x} is generated from *generic κ -sparse model* and $\kappa \leq c_2 n / (\|\mathbf{A}\|_2^2 \log n)$ for some constant c_2 . If $\min_{i \in I_{\mathbf{x}}} |x_i| > 8\sigma\sqrt{2 \log n}$, the LASSO estimate obtained by choosing $\lambda = 4\sigma\sqrt{2 \log n}$ for measurements (1) where $\mathbf{w} \sim \mathcal{N}(0, \sigma^2 \mathbf{I})$ satisfies:

$$\begin{aligned} \text{supp}(\hat{\mathbf{x}}) &= \text{supp}(\mathbf{x}) \\ \text{sgn}(\hat{x}_i) &= \text{sgn}(x_i) \text{ for } i \in I_{\mathbf{x}} \end{aligned}$$

with probability at least $1 - \frac{2}{n} \left(\frac{1}{\sqrt{2\pi \log n}} + \frac{|I_{\mathbf{x}}|}{n} \right) - O\left(\frac{1}{n^{2 \log 2}}\right)$ and

$$\|\mathbf{A}\mathbf{x} - \mathbf{A}\hat{\mathbf{x}}\|_2^2 \leq c_3 \kappa (\log n) \sigma^2,$$

with probability at least $1 - 6n^{-2 \log 2} - n^{-1} (2\pi \log n)^{-1/2}$, where c_3 is a constant.

C. Algorithms

There is a wealth of iterative algorithms developed for LASSO, inspired by proximal algorithms for non-smooth convex optimization: FISTA [10] and SpaRSA [11] are accelerated proximal gradient methods [12], SALSA [13] is an application of the alternative direction method of multipliers. For error defined as $G(\mathbf{x}_{[t]}) - G(\mathbf{x}_*)$ where $G(\mathbf{x})$ is the objective function of LASSO in (3), $\mathbf{x}_{[t]}$ is the estimate at iteration number t and $\mathbf{x}_* = \text{argmin} G(\mathbf{x})$, the error decays as $1/t^2$ for FISTA, SpaRSA and SALSA. In a recent paper by Patrinos and Bemporad [14] a Newton-type method was advised for non-smooth convex optimization, in which the convergence rate is no worse than $1/t^2$, but is locally *quadratic*.

III. RECURSIVE COMPRESSED SENSING

Let us consider the case that the signal of interest is an infinite sequence, $\{x_i\}_{i=0,1,\dots}$, as is the case when dealing with streaming data. We define

$$\mathbf{x}^{(i)} := [x_i \quad x_{i+1} \quad \dots \quad x_{i+n-1}]^\top \quad (5)$$

to be the i^{th} window taken from the streaming signal. If $\mathbf{x}^{(i)}$ is known to be sparse, one can apply the tools surveyed in Section II to recover the signal portion in each window, hence the data stream. However, the involved operations are costly, and an efficient online implementation is dubious.

In this section we propose the RCS method for efficiently sampling and recovering streaming data.

A. Problem Formulation

Our goal is to design a robust low-complexity sliding-window algorithm which provides estimates $\{\hat{x}_i\}$ using successive measurements $\mathbf{y}^{(i)}$ of the form

$$\mathbf{y}^{(i)} = \mathbf{A}^{(i)} \mathbf{x}^{(i)} + \mathbf{w}^{(i)}, \quad (6)$$

where $\{\mathbf{A}^{(i)}\}$ is a sequence of measurement matrices. This is possible if $\{x_i\}$ is sufficiently sparse in each window, namely if $\|\mathbf{x}^{(i)}\|_0 \leq \kappa$ for each i , where $\kappa \ll n$ (or if this holds with sufficiently high probability).

We propose an approach that leverages the signal overlap between successive windows, consisting of recursive sampling and recursive estimation.

B. Recursive Sampling

In the first iteration, there is no prior estimate, so we need to compute

$$\mathbf{y}^{(0)} = \mathbf{A}^{(0)} \mathbf{x}^{(0)} + \mathbf{w}^{(0)}.$$

We choose a sequence of sensing matrices $\mathbf{A}^{(i)}$ recursively as:

$$\mathbf{A}^{(i+1)} = \begin{bmatrix} \mathbf{a}_1^{(i)} & \mathbf{a}_2^{(i)} & \dots & \mathbf{a}_{n-1}^{(i)} & \mathbf{a}_0^{(i)} \end{bmatrix} = \mathbf{A}^{(i)} \mathbf{P} \quad (7)$$

where $\mathbf{a}_j^{(i)}$, $j \in \{0, 1, \dots, n-1\}$ is the j^{th} column of $\mathbf{A}^{(i)}$, and \mathbf{P} is a permutation matrix.

The following lemma ensures the success of this data encoding scheme.

Lemma 3.1: If $\mathbf{A}^{(0)}$ satisfies RIP for given κ with constant δ_κ , then $\mathbf{A}^{(i)}$ as defined in (7) satisfies RIP too, for same κ , δ_κ .

Proof: It follows from (7) that $\mathbf{A}^{(i)} = \mathbf{A}^{(0)} \mathbf{P}^i$, i.e., $\mathbf{A}^{(i)}$ is obtained by reordering the columns of $\mathbf{A}^{(0)}$. By definition, RIP depends only on $\|\mathbf{x}\|_0$, and is insensitive to permutations of the entries of \mathbf{x} , equivalently permutations of the columns of $\mathbf{A}^{(0)}$. ■

Given the particular recursive selection of $\mathbf{A}^{(i)}$ we can compute $\mathbf{y}^{(i+1)}$ recursively with *rank-1* update as:

$$\begin{aligned} \mathbf{y}^{(i+1)} &= \mathbf{A}^{(i+1)} \mathbf{x}^{(i+1)} + \mathbf{w}^{(i+1)} \\ &= \mathbf{A}^{(i)} \mathbf{P} \mathbf{x}^{(i+1)} + \mathbf{w}^{(i+1)} \\ &= \mathbf{A}^{(i)} \left(\mathbf{x}^{(i)} + \begin{bmatrix} 1 \\ \mathbf{0}_{n-1} \end{bmatrix} (x_{i+n} - x_i) \right) + \mathbf{w}^{(i+1)} \\ &= \mathbf{y}^{(i)} + (x_{i+n} - x_i) \mathbf{a}_1^{(i)} + \mathbf{w}^{(i+1)} - \mathbf{w}^{(i)}, \end{aligned} \quad (8)$$

where $\mathbf{0}_{n-1}$ is the all 0 vector of length $n-1$. Defining $\mathbf{z}^{(i)} := \mathbf{w}^{(i)} - \mathbf{w}^{(i-1)}$, we have $\mathbf{z}^{(i)}$ and $\mathbf{z}^{(i+1)}$ are independent if $\mathbf{w}^{(i)}$ is an independent increment process.

C. Recursive Estimation

In iterative schemes for convex optimization, convergence speed depends on the distance of the starting point to the optimal solution [15]. To attain accelerated convergence, we leverage the overlap between windows and use the starting point:

$$\hat{\mathbf{x}}_{[0]}^{(i)} = \begin{bmatrix} \hat{x}_1^{(i-1)} & \hat{x}_2^{(i-1)} & \dots & \hat{x}_{n-1}^{(i-1)} & \mathbb{E}_{\mathbf{x}^{(i-1)}} [x_{i+n-1}] \end{bmatrix}^\top,$$

where $\hat{x}_j^{(i-1)}$, for $j = 1, \dots, n-1$, is the portion of the optimal solution based on the previous window. This is casually referred to as ‘*warm start*’ in the optimization literature [8]. By choosing the starting point as such, the expected number of iterations for convergence is reduced (cf. Section V for quantitative results).

D. Sparse Signals in a given Orthonormal Basis

So far, we have implicitly assumed that for a given $n \in \mathbb{Z}^+$, windows $\mathbf{x}^{(i)}$ of length n obtained from the sequence $\{x_i\}$ satisfy the sparsity constraint $\|\mathbf{x}^{(i)}\|_0 \leq \kappa$, $\forall i$. In general, it might be the case that $\mathbf{x}^{(i)}$ is not sparse itself, but can be sparsely represented in a properly selected basis.

Let $\mathbf{x}^{(i)} \in \mathbb{R}^n$ be sparsely representable in a given orthonormal basis Φ as $\mathbf{x}^{(i)} = \Phi \boldsymbol{\alpha}^{(i)}$, where $\boldsymbol{\alpha}^{(i)}$ is sparse. Assuming a common basis for the entire sequence $\{x_i\}$ (over windows of size n) we have:

$$\mathbf{A}^{(i)} \mathbf{x}^{(i)} = \mathbf{A}^{(i)} \Phi \boldsymbol{\alpha}^{(i)}.$$

For the CS estimation to carry over, we need that $\mathbf{A}^{(i)} \Phi$ satisfy RIP. The key result here is that RIP is satisfied with high probability for the product of a random matrix $\mathbf{A}^{(i)}$ and any fixed matrix [6]. In this case the LASSO problem is expressed as:

$$\text{minimize} \quad \|\mathbf{A}^{(i)} \Phi \boldsymbol{\alpha}^{(i)} - \mathbf{y}^{(i)}\|_2^2 + \lambda \|\boldsymbol{\alpha}^{(i)}\|_1,$$

where the input signal is given by $\mathbf{x}^{(i)} = \Phi \boldsymbol{\alpha}^{(i)}$.

The problem now is to find a recursive update for $\boldsymbol{\alpha}^{(i+1)}$ based on $\boldsymbol{\alpha}^{(i)}$ so as to have a good initial estimate for accelerated convergence in recursive estimation as $\mathbb{E}[\boldsymbol{\alpha}^{(i+1)} | \boldsymbol{\alpha}^{(i)}]$.

Lemma 3.2 (Recursive Sampling in Orthonormal Basis): The representation of $\mathbf{x}^{(i+1)}$ and $\mathbf{x}^{(i)}$ in Φ satisfies:

$$\begin{aligned} \boldsymbol{\alpha}^{(i+1)} &= \Psi \Pi \Phi \boldsymbol{\alpha}^{(i)} \\ &\quad + \psi_{n-1} \left(\phi_{(n-1)} \boldsymbol{\alpha}^{(i+1)} - \phi_{(0)} \boldsymbol{\alpha}^{(i)} \right), \end{aligned} \quad (9)$$

where $\Psi := \Phi^{-1}$, $\Pi := \mathbf{P}^\top$, and $\phi_{(0)}$ and $\phi_{(n-1)}$ denote the first and the last rows of Φ respectively.

Proof: By the definition of $\mathbf{x}^{(i+1)}$ we have:

$$\mathbf{x}^{(i+1)} = \Pi \mathbf{x}^{(i)} + \begin{bmatrix} \mathbf{0}_{n-1} \\ 1 \end{bmatrix} (x_{i+n} - x_i).$$

The result is obtained by multiplying both sides by Ψ and using:

$$\begin{aligned} x_i &= x_0^{(i)} = [1 \quad \mathbf{0}_{n-1}] \Phi \boldsymbol{\alpha}^{(i+1)} \\ x_{i+n} &= x_{n-1}^{(i+1)} = [\mathbf{0}_{n-1} \quad 1] \Phi \boldsymbol{\alpha}^{(i+1)}, \end{aligned}$$

which both follow from $\mathbf{x}^{(i)} = \Phi \boldsymbol{\alpha}^{(i)}$. ■

Fourier basis is of particular interest for Φ since an efficient update rule can be derived for such basis:

Corollary 3.3 (Recursive Sampling in Fourier Basis):

Let Φ be $n \times n$ inverse Discrete Fourier Transform (IDFT) matrix with entries $\Phi_{i,j} = \omega^{ij} / \sqrt{n}$ where $i, j \in \{0, \dots, n-1\}$ and $\omega := e^{j \frac{2\pi}{n}}$. In such case:

$$\boldsymbol{\alpha}^{(i+1)} = \Omega_n \boldsymbol{\alpha}^{(i)} + \psi_{n-1} \left(\phi_{(n-1)} \boldsymbol{\alpha}^{(i+1)} - \phi_{(0)} \boldsymbol{\alpha}^{(i)} \right) \quad (10)$$

where Ω_n is the $n \times n$ diagonal matrix with $(\Omega_n)_{i,i} = \omega^{-i}$, and $\Psi = \Phi^{-1}$ is the orthonormal Fourier basis.

Proof: Circular shift in the time domain corresponds to multiplication by complex exponentials in the Fourier domain, $\Psi \Pi = \Omega_n \Psi$, and $\Psi \Phi = \mathbf{I}$. ■

From Corollary 3.3 it is seen that although the number of computations for calculating $\boldsymbol{\alpha}^{(i+1)}$ based on $\boldsymbol{\alpha}^{(i)}$ is $O(n^2)$ in general, for Fourier basis it is $O(n)$.

As before, in the presence of noise, we can use the estimate of $\mathbb{E}[\boldsymbol{\alpha}^{(i+1)} | \boldsymbol{\alpha}^{(i)}]$ as the starting point in the iterative LASSO solver for warm start, in order to attain accelerated convergence.

E. Averaging LASSO Estimates

One way to reduce error variance is by averaging the estimates obtained from successive windows. For the i^{th} entry of the streaming signal, x_i , we define the averaged estimate, \bar{x}_i , using the previous estimates $\{\hat{x}^{(j)}\}_{j=\max\{0, i-n+1\}, \dots, i}$ as:

$$\bar{x}_i := \frac{1}{\min\{i+1, n\}} \sum_{j=\max\{0, i-n+1\}}^i \hat{x}_{i-j}^{(j)}, \quad (11)$$

where we average n many estimates for $i \geq n-1$ and $i+1$ many estimates for $i < n-1$. Considering $i \geq n-1$ for

notational simplicity, and using Jensen's inequality we get:

$$\frac{1}{n} \sum_{j=i-n+1}^i \left(\hat{x}_{i-j}^{(j)} - x_i \right)^2 \geq \left(\frac{1}{n} \sum_{j=i-n+1}^i \left(\hat{x}_{i-j}^{(j)} - x_i \right) \right)^2 = (\bar{x}_i - x_i)^2,$$

which implies that we can only lower the reconstruction error by averaging the estimates.

In the following, we analyze the expected ℓ_2 -norm of the reconstruction error $(\bar{x}_i - x_i)^2$. Considering the case $i \geq n - 1$,

$$\begin{aligned} \mathbb{E}_x [(\bar{x}_i - x_i)^2] &= \mathbb{E}_x \left[\left(\frac{1}{n} \sum_{j=i-n+1}^i \hat{x}_{i-j}^{(j)} - x_i \right)^2 \right] \\ &= \left(\mathbb{E}_x [\hat{x}_0^{(i)}] - x_i \right)^2 + \frac{1}{n} \mathbb{E}_x \left[\left(\hat{x}_0^{(i)} - \mathbb{E}_x [\hat{x}_0^{(i)}] \right)^2 \right], \end{aligned}$$

where we have used $\text{Cov} [\hat{x}_{i-j}^{(j)}, \hat{x}_{i-k}^{(k)}] = 0$ for $j \neq k$, $j, k \in \{i - n + 1, \dots, i\}$ which follows from independence.

It is seen that as the window length is increased the second term goes to zero and the reconstruction error asymptotically converges to the square of the bias of LASSO.

F. The Proposed Algorithm

We propose a two-step estimation procedure for recovering the sampled signal to reduce the estimation error. First, we obtain the LASSO estimates $\{\hat{\mathbf{x}}^{(i)}\}$ which are subsequently used in a *de-biasing* algorithm. For de-biasing, we perform LSE on the support set of the estimate to obtain $\tilde{\mathbf{x}}^{(i)}$. The debiased estimates obtained over successive windows are subsequently averaged. The block diagram of the method and the pseudocode for the algorithm can be seen in Figure 1 and Algorithm 1 respectively.

Algorithm 1 Recursive Compressed Sensing

Input: $\mathbf{A}^{(0)} \in \mathbb{R}^{m \times n}$, $\{x_i\}_{i=0,1,\dots}$, $\lambda \geq 0$

Output: estimate $\{\bar{x}_i\}_{i=0,1,\dots}$.

- 1: initialize signal estimate: $\{\bar{x}\} \leftarrow \{0\}$
 - 2: **for** $i = 0, 1, 2, \dots$ **do**
 - 3: $\mathbf{x}^{(i)} \leftarrow [x_i \ x_{i+1} \ \dots \ x_{i+n-1}]^\top$
 - 4: $\mathbf{y}^{(i)} \leftarrow \mathbf{A}^{(i)} \mathbf{x}^{(i)} + \mathbf{w}^{(i)}$ ▷ encoding
 - 5: $\hat{\mathbf{x}}^{(i)} \leftarrow \underset{\mathbf{x} \in \mathbb{R}^n}{\text{argmin}} \|\mathbf{A}^{(i)} \mathbf{x} - \mathbf{y}^{(i)}\|_2^2 + \lambda \|\mathbf{x}\|_1$ ▷ LASSO
 - 6: $I \leftarrow \text{supp} \left(\hat{\mathbf{x}}^{(i)} \right)$ ▷ support estimation
 - 7: $\tilde{\mathbf{x}}^{(i)} \leftarrow \underset{\substack{\mathbf{x} \in \mathbb{R}^n \\ \mathbf{x}_{I^c} = \mathbf{0}}}{\text{argmin}} \|\mathbf{A}^{(i)} \mathbf{x} - \mathbf{y}^{(i)}\|_2^2$ ▷ LSE
 - 8: $\bar{x}_{i+j} \leftarrow \left((k_i(j) - 1) \bar{x}_{i+j} + \tilde{x}_j^{(i)} \right) / k_i(j)$ for $j = 0, \dots, n - 1$ where $k_i(j) = \min\{i + 1, n - j\}$ ▷ update average estimates
 - 9: $\mathbf{A}^{(i)} \leftarrow \mathbf{A}^{(i-1)} \mathbf{P}$ ▷ for recursive sampling
 - 10: **end for**
-

IV. EXTENSIONS

In this section we present extensions to the algorithm.

A. Sliding Window With Variable Overlap

Consider a generalization in which sensing is performed via recurring windowing with $0 < \tau \leq n$ overlaps, i.e., $\mathbf{x}^{(i)} := [x_{i\tau} \ x_{i\tau+1} \ \dots \ x_{i\tau+n-1}]^\top$.

We let η_i denote the sampling efficiency, that is the ratio of total samples taken until time $n + i$ to the number of entries sensed. For one window, sampling efficiency is m/n . By the end of i^{th} window, we have recovered $n + (i - 1)\tau$ elements while having sensed im many samples. The asymptotic sampling efficiency is:

$$\eta := \lim_{i \rightarrow \infty} \eta_i = \lim_{i \rightarrow \infty} \frac{im}{n + (i - 1)\tau} = \frac{m}{\tau}.$$

If instead we encode using a rank- τ update, i.e., by recursive sampling using the matrix obtained by circular shifting the sensing matrix τ times, $\mathbf{A}^{(i+1)} = \mathbf{A}^{(i)} \mathbf{P}^\tau$, the asymptotic sampling efficiency becomes:

$$\eta = \lim_{i \rightarrow \infty} \frac{m + (i - 1)\tau}{n + (i - 1)\tau} = 1.$$

In the latter case, recursive sampling approach is asymptotically equivalent to taking one sample for each time instance. Note, however, that the benefit of such approach lies in noise suppression. By taking overlapping windows each element is sensed at minimum $\lfloor n/\tau \rfloor$ many times, hence collaborative decoding using multiple estimates can be used to increase estimation accuracy.

When $\tau \geq m$, asymptotic sampling efficiency of rank- τ update is worse than the former sensing scheme. Hence storing samples by selecting the sampling strategy accordingly yields:

$$\eta = \min \left\{ 1, \frac{m}{\tau} \right\}.$$

B. Voting

When the signal magnitudes are not high enough to stand above the noise level, the aforementioned support detection mechanism may miss nonzero positions (false negative) or detect false support entries (false positive). As a remark to Theorem 2.2, it is stated in [9] that a smaller regularization constant, λ , would detect support for lower minimum nonzero values with a cost of reducing the probability given in Theorem 2.2. Assuming that the false positives are randomly distributed, by designating non zero positions as the entries that are detected sufficiently many times as the support of consecutive windows, we can decrease the false positives for low signal to noise ratio.

In detail, the two step algorithm with voting entails solving LASSO: $\hat{\mathbf{x}}^{(i)} = \underset{\mathbf{x} \in \mathbb{R}^n}{\text{argmin}} \left(\|\mathbf{A}^{(i)} \mathbf{x} - \mathbf{y}^{(i)}\|_2^2 + \lambda \|\mathbf{x}\|_1 \right)$. We find the indices having magnitude larger than $\xi_1 > 0$ to estimate the support of $\mathbf{x}^{(i)}$ as $Q_i := \left\{ j : |\hat{x}_j^{(i)}| \geq \xi_1 \right\}$.

We define the sequence containing the cumulative votes as $\{v_i\}$ and the number of times an index i is used in LSE as $\{l_i\}$. At the beginning of the algorithm $\{v_i\}$ and $\{l_i\}$

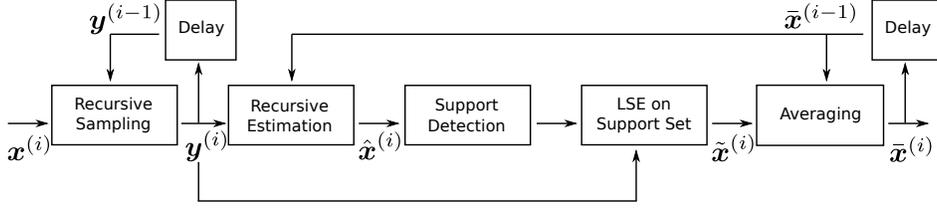


Fig. 1: Architecture of RCS.

are all set to zero. For each window, we add votes on the positions that are in the set Q_i as $v_{Q_i+i} \leftarrow v_{Q_i+i} + 1$, where the subscript $Q_i + i$ is used to translate the indices within the window to global indices on the streaming data. By applying threshold $\xi_2 \in \mathbb{Z}^+$ on the number of votes $\{v_i\}$, we get indices that have been voted sufficiently many times to be accepted as non-zeros and store them in $R_i = \{j : v_{j+i} \geq \xi_2, j = 0, \dots, n-1\}$. Note that threshold ξ_2 needs to be chosen such that $|R_i| < m$, hence yielding an *overdetermined* system for the LSE. Subsequently, we solve the overdetermined least squares problem based on these indices in R_i ,

$$\tilde{\mathbf{x}}^{(i)} = \underset{\mathbf{x} \in \mathbb{R}^n, \mathbf{x}_{R_i^c} = 0}{\operatorname{argmin}} \|\mathbf{A}^{(i)} \mathbf{x} - \mathbf{y}^{(i)}\|_2^2.$$

Note that this problem can be solved in closed form by noting $\tilde{\mathbf{x}}_{R_i}^{(i)} = \left(\mathbf{A}_{R_i}^{(i)\top} \mathbf{A}_{R_i}^{(i)}\right)^{-1} \mathbf{A}_{R_i}^{(i)\top} \mathbf{y}^{(i)}$, where $\tilde{\mathbf{x}}_{R_i}^{(i)}$ is the vector obtained by extracting elements indexed by R_i , and $\mathbf{A}_{R_i}^{(i)}$ is the matrix obtained by extracting columns of $\mathbf{A}^{(i)}$ indexed by R_i . Subsequently, we increment the number of recoveries for the entries used in LSE procedure as $l_{R_i+i} \leftarrow l_{R_i+i} + 1$, and the average estimates are updated as $\bar{x}_{i+j} \leftarrow \frac{l_{i+j}-1}{l_{i+j}} \bar{x}_{i+j} + \frac{1}{l_{i+j}} \tilde{x}_j$, for $j \in R_i$.

V. ANALYSIS

In this section we analyze the estimation error variance and computational complexity of the proposed method.

A. Estimation Error Variance

Given $\{x_i\}$ we give a bound on the normalized error variance of each window defined as:

$$\text{NE}(i) := \mathbb{E} \left[\frac{\|\bar{\mathbf{x}}^{(i)} - \mathbf{x}^{(i)}\|_2}{\|\mathbf{x}^{(i)}\|_2} \right].$$

Theorem 5.1 (Normalized Error): Under the assumptions of Theorem 2.2 and given $\mathbf{A}^{(0)}$ satisfying RIP with δ_κ , for $\{x_i\}_{i=0,1,\dots}$ satisfying $\|\mathbf{x}^{(i)}\|_0 \geq \Omega(\kappa)$, NE(i) satisfies:

$$\text{NE}(i) \leq P^n \cdot c_1 \frac{1}{\sqrt{n \log n}} + (1 - P^n) \left(c_2 + c_3 \frac{\sqrt{m}}{\sqrt{\kappa \log n}} \right),$$

where c_1, c_2 and c_3 are constants, and

$$P^n \geq \left(1 - \frac{2}{n\sqrt{2\pi \log n}} - \frac{2\kappa}{n^2} - O\left(\frac{1}{n^{2 \log 2}}\right) \right)^{2n-1}$$

Proof: Defining the event $S_{2n-1} := \{\text{support is detected correctly on } 2n-1 \text{ consecutive windows}\}$, we have the

following equality for NE given $\{x_i\}_{i=0,1,\dots}$:

$$\begin{aligned} \text{NE}(i) &= P(S_{2n-1}) \cdot \mathbb{E}_{x, S_{2n-1}} \left[\frac{\|\bar{\mathbf{x}}^{(i)} - \mathbf{x}^{(i)}\|_2}{\|\mathbf{x}^{(i)}\|_2} \right] \\ &\quad + (1 - P(S_{2n-1})) \cdot \mathbb{E}_{x, S_{2n-1}^c} \left[\frac{\|\bar{\mathbf{x}}^{(i)} - \mathbf{x}^{(i)}\|_2}{\|\mathbf{x}^{(i)}\|_2} \right], \end{aligned}$$

where, dropping the subscript $2n-1$, and using S as a shorthand notation for S_{2n-1} , we have:

$$P(S) \geq \left(1 - \frac{2}{n\sqrt{2\pi \log n}} - \frac{2\kappa}{n^2} - O\left(\frac{1}{n^{2 \log 2}}\right) \right)^{2n-1},$$

by Theorem 2.2.

In S , by LSE we get:

$$\begin{aligned} \mathbb{E}_{x, S} \left[\frac{\|\bar{\mathbf{x}}^{(i)} - \mathbf{x}^{(i)}\|_2}{\|\mathbf{x}^{(i)}\|_2} \right] &\leq \frac{1}{\|\mathbf{x}^{(i)}\|_2} \sqrt{\mathbb{E}_{x, S} [\|\bar{\mathbf{x}}^{(i)} - \mathbf{x}^{(i)}\|_2^2]} \\ &\stackrel{(a)}{\leq} \frac{\sigma \sqrt{\kappa}}{\|\mathbf{x}^{(i)}\|_2 \sqrt{n(1 - \delta_\kappa)}}, \end{aligned}$$

where (a) follows from

$$\begin{aligned} \mathbb{E}_{x, S} [\|\bar{\mathbf{x}}^{(i)} - \mathbf{x}^{(i)}\|_2^2] &= \mathbb{E}_{x, S} \left[\sum_{j \in I} (\bar{x}_{i+j} - x_{i+j})^2 \right] \\ &= \mathbb{E}_{x, S} \left[\sum_{j \in I} \left(\frac{1}{n} \sum_{t=0}^{n-1} \tilde{x}_{j+t}^{(i+j-t)} - x_{i+j} \right)^2 \right] \\ &= \sum_{j \in I} \sum_{t,r=0}^{n-1} \mathbb{E}_{x, S} \left[\frac{(\tilde{x}_{j+t}^{(i+j-t)} - x_{i+j})(\tilde{x}_{j+r}^{(i+j-r)} - x_{i+j})}{n^2} \right] \\ &= \frac{1}{n^2} \sum_{j \in I} \sum_{t=0}^{n-1} \mathbb{E}_{x, S} \left[(\tilde{x}_{j+t}^{(i+j-t)} - x_{i+j})^2 \right] \\ &\stackrel{(b)}{\leq} \frac{1}{n^2} \sum_{j \in I} \sum_{t=0}^{n-1} \frac{\sigma^2}{1 - \delta_\kappa} \leq \frac{\kappa \sigma^2}{n(1 - \delta_\kappa)} \end{aligned}$$

where $I = \text{supp}(\hat{\mathbf{x}}^{(i)})$, also equal to $\text{supp}(\mathbf{x}^{(i)})$ given S , and (b) follows since covariance matrix of LSE is $\sigma^2(\mathbf{A}_I^T \mathbf{A}_I)^{-1}$ and by RIP we have all of the eigenvalues of $\mathbf{A}_I^T \mathbf{A}_I$ greater than $(1 - \delta_\kappa)$ since $(1 - \delta_\kappa) \|\mathbf{x}\|_2^2 \leq \|\mathbf{A}\mathbf{x}\|_2^2$ for all \mathbf{x} κ -sparse.

To bound the estimation error in S^c , note that independent of the selected support, by triangle inequality, we have:

$$\begin{aligned} \|\mathbf{A}^{(i)} \tilde{\mathbf{x}}^{(i)} - \mathbf{y}^{(i)}\|_2 &\stackrel{(a)}{\leq} \|\mathbf{y}^{(i)}\|_2 \\ &\leq \|\mathbf{A}^{(i)} \mathbf{x}^{(i)}\|_2 + \|\mathbf{w}^{(i)}\|_2 \\ &\leq (1 + \delta_\kappa) \|\mathbf{x}^{(i)}\|_2 + \|\mathbf{w}^{(i)}\|_2, \end{aligned}$$

and

$$\begin{aligned}\|\mathbf{y}^{(i)}\|_2 &\geq \|\mathbf{A}^{(i)}\tilde{\mathbf{x}}^{(i)} - \mathbf{y}^{(i)}\|_2 \geq \|\mathbf{A}^{(i)}\tilde{\mathbf{x}}^{(i)}\|_2 - \|\mathbf{y}^{(i)}\|_2 \\ &\geq (1 - \delta_\kappa)\|\tilde{\mathbf{x}}^{(i)}\|_2 - \|\mathbf{y}^{(i)}\|_2,\end{aligned}$$

where (a) follows since

$$\tilde{\mathbf{x}}^{(i)} = \underset{\mathbf{x} \in \mathbb{R}^n, \mathbf{x}_I^c = 0}{\operatorname{argmin}} \|\mathbf{A}^{(i)}\mathbf{x}^{(i)} - \mathbf{y}^{(i)}\|_2.$$

From these two inequalities we have:

$$\begin{aligned}\|\tilde{\mathbf{x}}^{(i)}\|_2 &\leq \frac{2}{1 - \delta_\kappa} \|\mathbf{y}^{(i)}\|_2 \\ &\leq \frac{2}{1 - \delta_\kappa} \left((1 + \delta_\kappa) \|\mathbf{x}^{(i)}\|_2 + \|\mathbf{w}^{(i)}\|_2 \right).\end{aligned}$$

By applying triangle inequality once more we get:

$$\begin{aligned}\|\tilde{\mathbf{x}}^{(i)} - \mathbf{x}^{(i)}\|_2 &\leq \|\tilde{\mathbf{x}}^{(i)}\|_2 + \|\mathbf{x}^{(i)}\|_2 \\ &\leq \|\mathbf{x}^{(i)}\|_2 \left(1 + \frac{2(1 + \delta_\kappa)}{1 - \delta_\kappa} \right) + \frac{2\|\mathbf{w}^{(i)}\|_2}{1 - \delta_\kappa}.\end{aligned}$$

Thus in \mathcal{S}^c we have:

$$\begin{aligned}\mathbb{E}_{\mathcal{S}^c} \left[\frac{\|\tilde{\mathbf{x}}^{(i)} - \mathbf{x}^{(i)}\|_2}{\|\mathbf{x}^{(i)}\|_2} \right] \\ &= \frac{1}{\|\mathbf{x}^{(i)}\|_2} \mathbb{E}_{\mathcal{S}^c} \left[\|\tilde{\mathbf{x}}^{(i)} - \mathbf{x}^{(i)}\|_2 \right] \\ &\stackrel{(b)}{\leq} \frac{1}{\|\mathbf{x}^{(i)}\|_2} \mathbb{E}_{\mathcal{S}^c} \left[\|\tilde{\mathbf{x}}^{(i)} - \mathbf{x}^{(i)}\|_2 \right] \\ &\leq \left(1 + \frac{2(1 + \delta_\kappa)}{1 - \delta_\kappa} \right) + \frac{2}{1 - \delta_\kappa} \frac{\mathbb{E}[\|\mathbf{w}^{(i)}\|_2]}{\|\mathbf{x}^{(i)}\|_2},\end{aligned}$$

where (b) follows from Jensen's inequality.

We get the result by taking the expectation over $\{x_i\}_{i=0,1,\dots}$ and noting by the assumptions of Theorem 2.2 we have $\mathbb{E}[\|\mathbf{w}^{(i)}\|_2] \leq \sigma\sqrt{m}$, $|x_{i+j}| \geq 8\sigma\sqrt{2\log n}$ where $j \in \operatorname{supp}(\mathbf{x}^{(i)})$ and $\|\mathbf{x}^{(i)}\|_0 \geq \Omega(\kappa)$. ■

Corollary 5.2: For window sparsity $\kappa = O(n^{1-\epsilon})$ and $m = O(\kappa \log n)$, NE goes to 0 as $n \rightarrow \infty$.

Proof: For $\kappa = O(n^{1-\epsilon})$ we have $P^n \geq \left(1 - O\left(\frac{1}{n\sqrt{\log n}}\right) \right)^{2n-1}$, and from the assumptions we have $c_3 \frac{\sqrt{m}}{\sqrt{k \log n}}$ constant. We get the result by noting P^n goes to 1 as the window length, n , increases. ■

B. Computational Complexity Analysis

In this section, we analyze the computational complexity of RCS. Let i be the window index, $\mathbf{A}^{(i)} \in \mathbb{R}^{m \times n}$ be the sampling matrix, and recall the extension on τ , the number of sliding slots between successive windows. By the end of i^{th} window, we have recovered $n + (i - 1)\tau$ many entries. As discussed in Section III, the first window is sampled by $\mathbf{A}^{(0)}\mathbf{x}^{(0)}$; this requires $O(mn)$ basic operations (additions and multiplications). After the initial window, sampling of the window $\mathbf{x}^{(i)} = [x_{i\tau} \ x_{i\tau+1} \ \dots \ x_{i\tau+n-1}]^\top$ is achieved by recursive sampling having rank- τ update with complexity $O(m\tau)$. Thus, by the end of i^{th} window, total complexity of sampling is $O(mn) + O(m\tau)i$. Thus the average complexity scales as $O(m\tau)$ for recursive sampling.

The other contribution to computational complexity is due to iterative solver, where the expected complexity can be calculated as number of operations in single iteration times the expected number of iterations for convergence. The latter is a function of the distance of the starting point to the optimal solution [15], which we bound in the case of using *recursive estimation*, as follows:

Lemma 5.3: Using $\hat{\mathbf{x}}_{[0]}^{(i)} = \begin{bmatrix} x_{*\tau}^{(i-1)} & \dots & x_{*n-1}^{(i-1)} & \mathbf{0}_\tau^\top \end{bmatrix}^\top$ as the starting point we have:

$$\begin{aligned}\|\hat{\mathbf{x}}_{[0]}^{(i)} - \mathbf{x}_*^{(i)}\|_2 &\leq c_0 \|\mathbf{x}^{(i-1)} - \mathbf{x}_\kappa^{(i-1)}\|_1 / \sqrt{\kappa} \\ &\quad + c_0 \|\mathbf{x}^{(i)} - \mathbf{x}_\kappa^{(i)}\|_1 / \sqrt{\kappa} \\ &\quad + c_1 \tilde{\sigma} + \left\| \begin{bmatrix} x_{n-\tau}^{(i)} & \dots & x_{n-1}^{(i)} \end{bmatrix} \right\|_2,\end{aligned}$$

where c_0 and c_1 are constants.

Proof: Defining:

$$\begin{aligned}\mathbf{e}'^{(i)} &:= \begin{bmatrix} x_{*\tau}^{(i-1)} & \dots & x_{*n-1}^{(i-1)} & \mathbf{0}_\tau^\top \end{bmatrix}^\top - \begin{bmatrix} x_{*0}^{(i)} & \dots & x_{*n-1}^{(i)} \end{bmatrix}^\top \\ \mathbf{e}^{(i)} &:= \mathbf{x}_*^{(i)} - \mathbf{x}^{(i)},\end{aligned}$$

we have

$$\mathbf{e}'^{(i)} = \begin{bmatrix} x_{*\tau}^{(i-1)} & \dots & x_{*n-1}^{(i-1)} & \mathbf{0}_\tau^\top \end{bmatrix}^\top - \mathbf{x}^{(i)} + \mathbf{x}^{(i)} - \mathbf{x}_*^{(i)}.$$

Taking the norm and using triangle inequality yields:

$$\|\mathbf{e}'^{(i)}\|_2 \leq \|\mathbf{e}^{(i-1)}\|_2 + \|\mathbf{e}^{(i)}\|_2 + \left\| \begin{bmatrix} x_{n-\tau}^{(i)} & \dots & x_{n-1}^{(i)} \end{bmatrix} \right\|_2.$$

Using Theorem 2.1 we get:

$$\begin{aligned}\|\mathbf{e}'^{(i)}\|_2 &\leq c_0 \|\mathbf{x}^{(i-1)} - \mathbf{x}_\kappa^{(i-1)}\|_1 / \sqrt{\kappa} \\ &\quad + c_0 \|\mathbf{x}^{(i)} - \mathbf{x}_\kappa^{(i)}\|_1 / \sqrt{\kappa} \\ &\quad + c_1 \tilde{\sigma} + \left\| \begin{bmatrix} x_{n-\tau}^{(i)} & \dots & x_{n-1}^{(i)} \end{bmatrix} \right\|_2.\end{aligned}\quad (12)$$

Exact computational complexity of each iteration depends on the algorithm. Minimally, iterative solver for LASSO requires multiplication of sampling matrix and the estimate at each iteration which requires $O(mn)$ operations. In an algorithm where cost function decays sublinearly (e.g., $1/t^2$), as in FISTA, the number of iterations, t , required for obtaining $\hat{\mathbf{x}}_{[t]}$ such that $G(\hat{\mathbf{x}}_{[t]}) - G(\mathbf{x}_*) \leq \epsilon$, where \mathbf{x}_* is the optimal solution, is proportional to $\|\mathbf{x}_{[0]} - \mathbf{x}_*\|_2$ (e.g., $\|\mathbf{x}_{[0]} - \mathbf{x}_*\|_2 / \sqrt{\epsilon}$) where $\mathbf{x}_{[0]}$ is the starting point of the algorithm [10]. From this bound, it is seen that average number of iterations is proportional to the Euclidean distance of the starting point of the algorithm from the optimal point.

Lemma 5.4 (Expected number of iterations): For the sequence $\{x_i\}_{i=0,1,\dots}$ where $\|\mathbf{x}^{(i)}\|_0 \leq \kappa$ with the positions of non-zeros chosen uniformly at random and $\max_{j=0,\dots,n-1} |x_j^{(i)}| = O(\sqrt{\log n})$ for all i , the expected number of iterations for convergence of algorithms where cost function decays as $1/t^2$ is $O(\sqrt{(\kappa\tau \log n)/n})$ for noiseless measurements and $O(\sqrt{m})$ for i.i.d. measurement noise.

Proof: Since $\mathbf{x}^{(i)}$ is κ -sparse, the terms $\|\mathbf{x}^{(i-1)} - \mathbf{x}_\kappa^{(i-1)}\|_1$ and $\|\mathbf{x}^{(i)} - \mathbf{x}_\kappa^{(i)}\|_1$ vanish in (12). By $|x_i| = O(\sqrt{\log n})$ and uniform distribution of non-zero elements

we have $\mathbb{E} \left[\left\| \begin{bmatrix} x_{n-\tau}^{(i)} \\ \dots \\ x_{n-1}^{(i)} \end{bmatrix} \right\|_2 \right] \leq \sqrt{(\kappa\tau \log n)/n}$.

With noisy measurements, the term $c_1\tilde{\sigma}$ is related to the noise level. Since noise has distribution $\mathbf{w}^{(i)} \sim \mathcal{N}(0, \sigma^2 \mathbf{I})$, the squared norm of the noise $\|\mathbf{w}^{(i)}\|_2^2$ has chi-squared distribution with mean $\sigma^2 m$ and standard deviation $\sigma^2 \sqrt{2m}$; probability of the squared norm exceeding its mean plus 2 standard deviations is small, hence we can pick $\tilde{\sigma}^2 = \sigma^2 (m + 2\sqrt{2m})$ [3] to satisfy the conditions of Theorem 2.1. Using this result in (12), we get $O(\sqrt{(\kappa \log n \tau)/n}) + O(\sqrt{m})$, where the second term dominates since $\tau \leq n$ not to leave out any element of the signal and $m \sim O(\kappa \log n)$. Hence it is found that the expected number of iterations is $O(\sqrt{m})$ in the noisy case. ■

The other source of complexity is the LSE in each iteration, which requires solving a linear $\kappa \times \kappa$ system, requiring $O(\kappa^3)$ operations.

It follows that recursive sampling needs $O(m\tau)$ and optimization algorithm needs $O(nm^{3/2})$, in the presence of noise and LSE requires $O(\kappa^3)$ operations.

VI. SIMULATION RESULTS

The data used in the simulations are generated from the random model:

$$f_X(x) = \begin{cases} (1-p)\delta(x) + \frac{1}{2p} & \text{if } x \in [-1, 1] \\ 0 & \text{else} \end{cases} \quad (13)$$

with $p = 0.05$. The measurement model is $\mathbf{y}^{(i)} = \mathbf{A}^{(i)} \mathbf{x}^{(i)} + \mathbf{w}^{(i)}$ with $\mathbf{w}^{(i)} \sim \mathcal{N}(0, \sigma^2 \mathbf{I})$ where $\sigma \in \mathbb{R}^+$, and the sampling matrix is $\mathbf{A}^{(0)} \in \mathbb{R}^{m \times n}$ where $m = 6pn$ and n is equal to the window length.

A. Runtime

We experimentally test the speed gain achieved by RCS by comparing the average time required to estimate a given window while using FISTA for solving LASSO. RCS is compared against so called ‘naïve approach’, where the sampling is done by matrix multiplication in each window and FISTA is started from all zero vector. The average time required to recover one window in each case is shown in Figure 2.

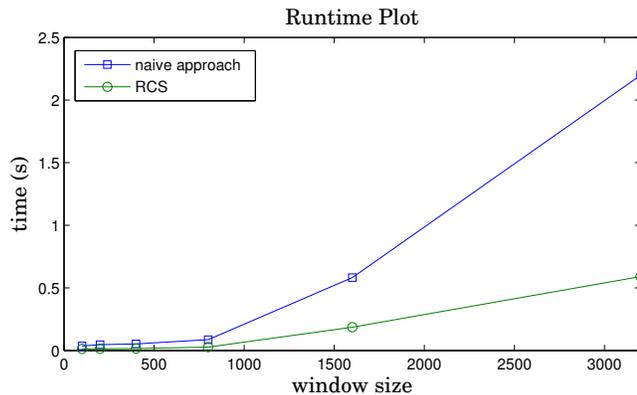


Fig. 2: Average processing time of RCS vs. ‘naïve approach’ over a single time window.

B. Support Estimation

We present the results of experiments on the support estimation using LASSO. In the measurements $\mathbf{x} \in \mathbb{R}^{6000}$, $\|\mathbf{x}\|_0 = 60$, $\mathbf{A} \in \mathbb{R}^{m \times 6000}$ is generated by i.i.d. Gaussian distribution with $A_{i,j} \sim \mathcal{N}(0, 1/m)$, and \mathbf{w} has $\sigma = 0.1$. As suggested in Theorem 2.2 for these parameters, LASSO is solved with $\lambda = 4\sigma\sqrt{2\log n}$, and the nonzero entries of \mathbf{x} are chosen so that $\min_{i=1,2,\dots,n} |x_i| \geq 3.34$ by sampling from $\mathcal{U}([-4.34, -3.34] \cup [3.34, 4.34])$. In simulations, we vary the number of samples taken from the signal, m , and study the accuracy of support estimation by using

$$\begin{aligned} \text{true positive rate} &= \frac{|\text{detected support} \cap \text{true support}|}{|\text{true support}|} \\ \text{false positive rate} &= \frac{|\text{detected support} \setminus \text{true support}|}{n - |\text{true support}|}, \end{aligned}$$

where $|\cdot|$ denotes the cardinality of a set and \setminus is the set difference operator.

The support is detected by taking the positions where the magnitude of the LASSO estimate is greater than threshold ξ_1 for values 0.01, 0.1, 1. Figure 3 shows the resulting curves, obtained by randomly generating the input signal 20 times for each m and averaging the results. It can be seen that the false positive rate can be reduced significantly by properly adjusting the threshold on the resulting LASSO estimates.

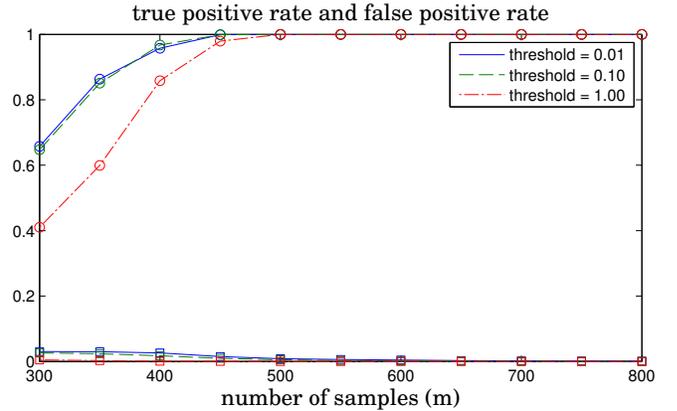


Fig. 3: Support set estimation using LASSO: for $n = 6000$, $\sigma = 0.1$, $\min |x_i| \geq 3.34$, threshold $\xi_1 = 0.01, 0.10$ and 1.00 . Circle markers depict *true positive rate*, and square markers depict *false positive rate*.

C. Reconstruction Error

As was discussed in Section IV-B, LASSO can be used together with a voting strategy and least squares estimation to reduce error variance. Figure 4 shows the comparison of performance of a) averaged LASSO estimates, b) debiasing and averaging with voting strategy, and c) debiasing and averaging without voting. The figure is obtained by using fixed \mathbf{x} and taking measurements with i.i.d. noise. It can be seen that the error does not decrease to zero for averaged estimate, which is due to LASSO being a biased estimator, cf. Section III.

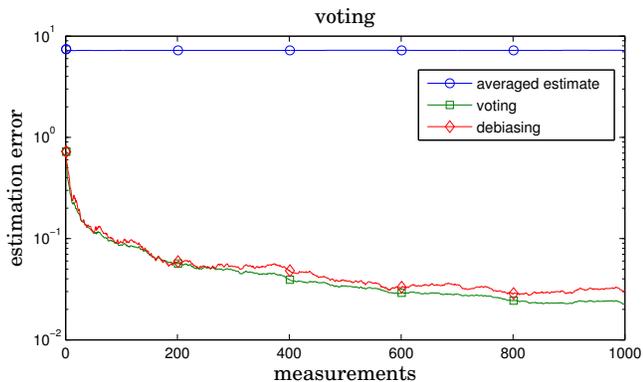


Fig. 4: Error plots for a) averaged estimates, b) debiased and averaged estimates with voting, and c) debiased and averaged estimates without voting.

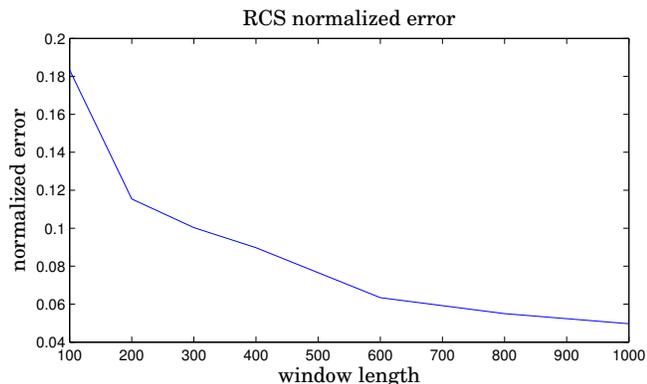


Fig. 5: Normalized error variance vs. window length for RCS on streaming data.

Figure 5 shows the behavior of normalized error variance

$$\lim_{T \rightarrow \infty} \frac{\sum_{i=1}^T (\bar{x}_i - x_i)^2}{\sum_{i=1}^T (x_i)^2}$$

as the window length, n , increases. The signals are generated to be 5% sparse, m is chosen to be 5 times the expected window sparsity, and the measurement noise is $w^{(i)} \sim \mathcal{N}(0, \sigma^2 \mathbf{I})$ where $\sigma = 0.1$. The non-zero amplitudes of the signal are drawn from uniform distribution $\mathcal{U}([- \chi_n - 1, - \chi_n] \cup [\chi_n, \chi_n + 1])$ where $\chi_n = 8\sigma\sqrt{2} \log n$ to satisfy the conditions of Theorem 2.2. The figure shows that normalized error variance decreases as the window length increases which agrees with our theoretical results.

VII. CONCLUSIONS AND FUTURE WORK

We have proposed an efficient method for recursive sampling and iterative recovery pertinent to CS on streaming data. The method leverages signal overlaps between successive processing windows in obtaining faster decoding of the signal while achieving estimation variance reduction in the presence of noise. We have proposed a two-step estimation

procedure to approximate an unbiased estimator of the signal based on LASSO where a) support detection is performed by solving LASSO, and b) signal estimation is obtained by solving ordinary least squares on the estimated support set. Furthermore, we have introduced a voting strategy for robust support detection in low signal to noise ratio. The computational complexity of the algorithm is $O(nm^3/2)$ where m is the number of samples taken and n is the window length, and the normalized estimation error is shown to asymptotically go to zero when window sparsity $\kappa = O(n^{1-\epsilon})$ as the window length is increased.

Future work consists of studying accuracy vs. complexity vs. delay tradeoffs of the algorithm, accelerating decoding by using faster solvers (e.g., [14]), automatic selection of thresholds for the voting strategy and applications of the method to real world problems.

VIII. ACKNOWLEDGEMENT

The authors would like to thank Qualcomm Inc., San Diego, for supporting this work.

REFERENCES

- [1] M. Vetterli, P. Marziliano, and T. Blu, "Sampling signals with finite rate of innovation," *IEEE Transactions on Signal Processing*, vol. 50, no. 6, pp. 1417–1428, 2002.
- [2] E. Candès and M. Wakin, "An introduction to compressive sampling," *IEEE Signal Processing Magazine*, vol. 25, no. 2, pp. 21–30, March 2008.
- [3] E. J. Candès, J. K. Romberg, and T. Tao, "Stable signal recovery from incomplete and inaccurate measurements," *Communications on Pure and Applied Mathematics*, vol. 59, pp. 1207–1223, Aug. 2006.
- [4] D. L. Donoho, "Compressed sensing," *IEEE Transactions on Information Theory*, vol. 52, pp. 1289–1306, 2006.
- [5] A. M. Bruckstein, D. L. Donoho, and M. Elad, "From sparse solutions of systems of equations to sparse modeling of signals and images," *SIAM Review*, vol. 51, no. 1, pp. 34–81, Mar. 2009.
- [6] R. Baraniuk, M. Davenport, R. DeVore, and M. Wakin, "A simple proof of the restricted isometry property for random matrices," *Constructive Approximation*, vol. 28, no. 3, pp. 253–263, 2008.
- [7] R. Tibshirani, "Regression Shrinkage and Selection via the LASSO," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 58, pp. 267–288, 1996.
- [8] D. P. Bertsekas, *Convex Optimization Theory*. Athena Scientific, 2009.
- [9] E. Candès and Y. Plan, "Near-ideal model selection by ℓ_1 minimization," *The Annals of Statistics*, vol. 37, pp. 2145–2177, 2009.
- [10] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM Journal on Imaging Sciences*, vol. 2, no. 1, pp. 183–202, 2009.
- [11] S. J. Wright, R. D. Nowak, and M. A. T. Figueiredo, "Sparse Reconstruction by Separable Approximation," *Signal Processing, IEEE Transactions on*, vol. 57, no. 7, pp. 2479–2493, 2009.
- [12] N. Parikh and S. Boyd, "Proximal algorithms," *Foundations and Trends in Optimization*, 2013.
- [13] M. Afonso, J. Bioucas-Dias, and M. A. T. Figueiredo, "Fast image recovery using variable splitting and constrained optimization," *IEEE Transactions on Image Processing*, vol. 19, no. 9, pp. 2345–2356, 2010.
- [14] P. Patrinos and A. Bemporad, "Proximal Newton methods for convex composite optimization," *IEEE Conference on Decision and Control*, 2013.
- [15] D. P. Bertsekas, *Nonlinear programming*. Athena Scientific, 1995.