

# Accelerated reconstruction of a compressively sampled data stream

Pantelis Sotasakis\*, Nikolaos Freris<sup>†</sup>, and Panagiotis Patrinos<sup>‡</sup>

\*IMT Institute for Advanced Studies Lucca, Piazza S. Ponziano 6, 55100 Lucca, Italy

<sup>†</sup>New York University Abu Dhabi, Division of Engineering, P.O. Box 129188, United Arab Emirates

<sup>‡</sup>KU Leuven, Dept. of Electrical Engineering (ESAT-STADIUS) & OPTEC, Kasteelpark Arenberg 10, 3001 Leuven, Belgium

**Abstract**—The traditional compressed sensing approach is naturally offline, in that it amounts to sparsely sampling and reconstructing a given dataset. Recently, an online algorithm for performing compressed sensing on streaming data was proposed: the scheme uses recursive sampling of the input stream and recursive decompression to accurately estimate stream entries from the acquired noisy measurements.

In this paper, we develop a novel Newton-type forward-backward proximal method to recursively solve the regularized Least-Squares problem (LASSO) online. We establish global convergence of our method as well as a local quadratic convergence rate. Our simulations show a substantial speed-up over the state of the art which may render the proposed method suitable for applications with stringent real-time constraints.

**Index Terms**—Compressed sensing, operator splitting methods, recursive algorithms, LASSO, Forward Backward Splitting.

## I. INTRODUCTION

In signal processing, continuous signals are typically sampled at discrete instances in order to store, process, and share. In doing so, signals of interest are assumed sparsely representable in an appropriately selected orthonormal basis, i.e., they can be reconstructed by storing few non-zero coefficients in the given basis. For instance, the Fourier basis is used for bandlimited signals—with prominent applications in communications—while wavelet bases are suitable for representing piecewise smooth signals, such as bitmap images. Traditionally, the celebrated Nyquist-Shannon sampling theorem suggests a sampling rate that is at least twice the signal bandwidth. This condition is sufficient but not necessary, and may suggest a rate that is unnecessarily high compared to the signal’s *innovation*, i.e., the minimum number of coefficients sufficient to accurately represent it in an appropriately selected basis [1], [2].

*Compressed Sensing* (CS) [3], [4] is a relatively new sampling paradigm that was introduced for sampling signals based entirely on their innovation, and has become ever since a major field of research in signal processing and information theory. The major contribution of this framework is a lower sampling rate compared to the classical sampling theory for signals that have sparse representation in some fixed basis [1], with notable applications in imaging [5], such as MRI.

The work of the first author was supported by the EU-funded H2020 research project DISIRE, grant agreement No. 636834. The work of the third author was supported by KU Leuven Research Council BOF/STG-15-043.

## A. Compressed Sensing

For a vector  $x \in \mathbb{R}^n$  we define the  $\ell_0$  pseudo-norm as the cardinality of its support  $\|x\|_0 := |\text{supp}(x)|$ , where the support is the set of non-zero entries  $\text{supp}(x) := \{i : x_i \neq 0\}$ . A vector  $x$  is called *s-sparse* if it has at most  $s$  non-zero entries,  $\|x\|_0 \leq s$ .

**Compression:** CS performs linear sampling  $y = Ax$ , where  $A \in \mathbb{R}^{m \times n}$ . Compression is performed by obtaining  $m \ll n$  measurements. The main result states that *any*  $s$ -sparse vector  $x \in \mathbb{R}^n$ , can be sampled by a *universal*<sup>1</sup> matrix  $A$  with  $m = \Theta(s \log(\frac{n}{s}))$ , and may then be reconstructed perfectly solely from (noiseless) measurements  $y$  [6]. For this purpose, matrix  $A$  needs to satisfy a certain *Restricted Isometry Property*: there exists  $\delta_s \in [0, 1)$  sufficiently small so that:

$$(1 - \delta_s)\|x\|_2^2 \leq \|Ax\|_2^2 \leq (1 + \delta_s)\|x\|_2^2$$

holds for all  $s$ -sparse vectors  $x$ . For many practical applications, taking  $m = 4s$  works well [1]. The success of CS lies in that the sampling matrix can be generated very efficiently as a random matrix, e.g., having i.i.d. Gaussian entries  $\mathcal{N}(0, 1/m)$ . Note that in practice measurements are typically noisy,  $y = Ax + w$ .

**Decompression:** In order to retrieve the original vector  $x$  from noisy measurements  $y = Ax + w$ , one needs to solve the  $\ell_1$ -regularized least squares problem:

$$\text{minimize } \frac{1}{2}\|Ax - y\|_2^2 + \lambda\|x\|_1, \quad (1)$$

where  $\lambda$  is the regularization parameter that controls the trade-off between sparsity and reconstruction error. This is best known as Least Absolute Selection and Shrinkage Operator (LASSO) in the statistics literature [7]. There are several results analyzing the reconstruction accuracy of LASSO; for example [8] states that if  $w \sim \mathcal{N}(0, \sigma^2 I)$ ,  $\min_{i \in \text{supp}(x)} |x_i| > 8\sigma\sqrt{2 \log n}$ , and we choose  $\lambda = 2\sigma\sqrt{2 \log n}$  then a solution  $x^*$  to (1) has the same support as  $x$ , and its non-zero entries have the same sign with their corresponding ones of  $x$ , with high probability. Additionally, the  $\ell_2$  reconstruction error is proportional to the standard deviation of the noise  $\sigma$  [1].

**Remark 1** (Algorithms for LASSO). *LASSO can easily be recast as a quadratic program which can be handled by interior point methods [9]. Additionally, iterative algorithms*

<sup>1</sup>The matrix  $A$  is universal in the sense it may be used to sample any vector with no more than  $s$  non-zeroes, regardless of their positions.

have been developed specifically for LASSO; all these are inspired by proximal methods [10] for non-smooth convex optimization: FISTA [11] and SpaRSA [12] are accelerated proximal gradient methods [10], SALSA [13] is an application of the alternative direction method of multipliers (ADMM). These methods are first-order methods, and feature sublinear convergence. In the current paper, we devise a proximal Newton-type method with substantial speedup exploiting the fact that its convergence rate is locally quadratic.

This paper extends and expands the online algorithm for performing compressed sensing on streaming data that was proposed in [6], [14]. We develop a novel scheme with favorable speed to recursively solve the LASSO problem needed for decompressing the compressively sampled data stream.

## II. RECURSIVE COMPRESSED SENSING

### A. Problem statement

The traditional CS framework is naturally *offline* and requires compressing and decompressing an entire given dataset at one shot. The Recursive Compressed Sensing (RCS) [6], [14] was developed as a new method for performing CS on an infinite data stream. The method consists in successively sampling the data stream via applying traditional CS to sliding overlapping windows in a recursive manner. Consider an infinite sequence,  $\{x_i\}_{i=0,1,\dots}$ . Let

$$S := \limsup_{j \rightarrow \infty} j^{-1} |\text{supp}(\{x_i\}_{i=0,\dots,j-1})| \in [0, 1]$$

be the *average* sparsity, where  $S$  is typically small. We define successive windows of length  $n$ , that is  $x^{(i)} := [x_i \ x_{i+1} \ \dots \ x_{i+n-1}]^\top$  and take  $s = Sn$  the average sparsity parameter for a window of length  $n$ . The sampling matrix  $A \in \mathbb{R}^{m \times n}$  (where we may take  $m = \Theta(s \log(\frac{n}{s}))$ ) is only generated once. Each window is compressively sampled given a matrix  $A^{(i)}$ :

$$y^{(i)} = A^{(i)}x^{(i)} + w^{(i)}.$$

The sampling matrix  $A^{(i)}$  is recursively computed by  $A^{(0)} = A$  and  $A^{(i)} = A^{(i-1)}\Pi$ , where  $\Pi$  is a permutation matrix which when right-multiplying a matrix cyclically rotates its columns to the left. This gives rise to an efficient recursive sampling mechanism for the input stream, where measurements for each window are taken via a rank-1 update [6].

For *decompression*, we need to solve the LASSO for each separate window:

$$\text{minimize } \frac{1}{2} \|A^{(i)}x^{(i)} - y^{(i)}\|_2^2 + \lambda \|x^{(i)}\|_1. \quad (2)$$

Note that the windows overlap, hence for a given stream entry, multiple estimates (one per each window that contains it) may be obtained. These estimates are then combined to boost estimation accuracy using non-linear support detection, least-squares debiasing and averaging with provable performance amelioration [6].

The overlap in sampling can further be exploited to speed-up the stream reconstruction: we use the estimate from a

previously decompressed window to warm-start the numerical solver for LASSO in the next one. This simple idea provides a mechanism for efficient *recursive estimation*.

Formally, let  $\hat{x}^{(i)}$  denote the optimal estimate obtained by solving LASSO in the  $i$ -th window. In order to obtain  $\hat{x}^{(i+1)}$  we may leverage overlapping to use a cyclic rotation of  $\hat{x}^{(i)}$  as the starting point  $\hat{x}_{[0]}^{(i+1)} = \Pi^\top \hat{x}^{(i)}$ , which is a suitable choice for the algorithm we propose.

### B. Our contribution

We devise a new numerical algorithm for solving LASSO based on the recently developed idea of proximal envelopes [15]–[18]. The new method demonstrates favorable convergence properties when compared to first order methods (FISTA [11], SpaRSA [12], SALSA [13], L1LS [9]); in particular it has local quadratic convergence. Furthermore, this scheme is very efficient as each iteration boils down to solving a linear system of low dimension. Using this solver in RCS along with warm-starting leads to substantive acceleration of stream decompression. We verify this via extensive experiments. Unfortunately, due to length constraints, we do not provide a complete convergence analysis, but rather some hints on global convergence and local quadratic rate.

## III. ALGORITHM

### A. Forward Backward Newton Algorithm

Let  $f(x) = \frac{1}{2} \|Ax - y\|_2^2$ ,  $g(x) = \lambda \|x\|_1$ . Then  $x^*$  is optimal for

$$\text{minimize } \varphi(x) = f(x) + g(x) \quad (3)$$

if and only if it satisfies

$$-\nabla f(x^*) \in \partial g(x^*) \quad (4)$$

where  $\partial g(x)$  is the subdifferential of  $g$  at  $x$ , defined by:

$$\partial g(x) = \{v \in \mathbb{R}^n \mid g(w) \geq g(x) + v^\top(w - x), \forall w \in \mathbb{R}^n\}, \quad (5)$$

which, in our case, is given by:  $v_i = \lambda \text{sign}(x_i)$ , for  $x_i \neq 0$  and  $|v_i| \leq \lambda$ , for  $x_i = 0$ . Therefore, the optimality conditions (4) for the LASSO problem (3) become

$$-\nabla_i f(x^*) = \lambda \text{sign}(x_i^*), \quad \text{if } x_i^* \neq 0, \quad (6a)$$

$$|\nabla_i f(x^*)| \leq \lambda, \quad \text{if } x_i^* = 0, \quad (6b)$$

Suppose for a moment that we knew the partition of indices  $\alpha$  and  $\beta$  corresponding to the nonzero and zero components of an optimal solution  $x^*$  respectively, as well as the signs of the nonzero components. Then we would be able to compute the nonzero components of  $x^*$  by solving the following linear system corresponding to (6a):

$$A_\alpha^\top A_\alpha x_\alpha^* = A_\alpha^\top y - \lambda \text{sign}(x_\alpha^*), \quad (7)$$

where  $x_\alpha^*$ ,  $A_\alpha$  respectively denote the vector/matrix obtained by selecting entries/columns of  $x^*$ ,  $A$  in  $\alpha$ . Notice that the support set  $\alpha$  of  $x^*$  is much smaller compared to its dimension  $n$ . Hence, provided that  $\alpha$  (as well as the signs of nonzero entries) has been identified, the problem becomes very

easy. Roughly speaking, the algorithm we will develop can be interpreted as a fast procedure for automatically identifying the partition  $\{\alpha, \beta\}$  corresponding to an optimal solution by solving a sequence of linear systems of the form (7). On the other hand, it can be seen as a Newton method for solving the following reformulation of the optimality conditions (4):

$$x^* = \text{prox}_{\gamma g}(x^* - \gamma \nabla f(x^*)), \quad (8)$$

where  $\text{prox}_{\gamma g}$  is the *proximal mapping* defined as

$$\text{prox}_{\gamma g}(z) = \underset{x \in \mathbb{R}^n}{\text{argmin}} \left\{ g(x) + \frac{1}{2\gamma} \|x - z\|_2^2 \right\}, \quad (9)$$

and  $\gamma$  is taken smaller than the Lipschitz constant of  $f$ , i.e.,  $\gamma < 1/\|A\|_2^2$ . In the case where  $g = \lambda \|\cdot\|_1$ ,  $\text{prox}_{\gamma g}$  is the soft-thresholding operator:

$$(\text{prox}_{\gamma g}(z))_i = \text{sign}(z_i)(|z_i| - \gamma\lambda)_+, \quad i = 1, \dots, n. \quad (10)$$

The iterative soft-thresholding algorithm (ISTA) is a fixed point iteration for solving (8): it is just an application of the well known forward-backward splitting technique for solving (4). On the other hand, FISTA is an accelerated version of ISTA where an extrapolation step between the current and the previous step precedes the forward-backward step [11], [19].

To motivate our algorithm, instead of a fixed point problem, we view (8) as a problem of finding a zero of the so-called *fixed point residual*:

$$R_\gamma(x) := x - \text{prox}_{\gamma g}(x - \gamma \nabla f(x)). \quad (11)$$

One then would be tempted to apply Newton's method for finding a root of (11). Unfortunately the fixed point residual is not everywhere differentiable, hence the classical Newton method is not well-defined. However, it is well known that  $R_\gamma$  is nonexpansive, hence globally Lipschitz continuous [16]. Therefore, the machinery of nonsmooth analysis can be employed to devise a generalized Newton method for  $R_\gamma(x) = 0$ , namely the *semismooth* Newton method. Due to a celebrated theorem by Rademacher, Lipschitz continuity of  $R_\gamma$  implies almost everywhere differentiability. Let  $\mathcal{F}$  stand for the set of points where  $R_\gamma$  is differentiable. The  $B$ -subdifferential of the nonsmooth mapping  $R_\gamma$  at  $x$  is defined by

$$\partial_B R_\gamma(x) := \left\{ B \in \mathbb{R}^{n \times n} \mid \begin{array}{l} \exists \{x_n\} \in \mathcal{F} : x_n \rightarrow x, \\ R'_\gamma(x_n) \rightarrow B \end{array} \right\}. \quad (12)$$

If  $R_\gamma$  is continuously differentiable at a point  $x \in \mathbb{R}^n$  then  $\partial_B R_\gamma(x) = \{R'_\gamma(x)\}$ . Otherwise  $\partial_B R_\gamma$  may contain more than one elements (matrices in  $\mathbb{R}^{n \times n}$ ). The semismooth Newton method for solving  $R_\gamma(x) = 0$  is simply

$$x^{k+1} = x^k - H_k^{-1} R_\gamma(x^k), \quad H_k \in \partial_B R_\gamma(x^k). \quad (13)$$

Since in the case of LASSO the fixed point residual is piecewise affine, it is also strongly semismooth. Provided that the solution  $x^*$  of (3) is unique (which is the case if for example the entries of  $A$  are drawn Gaussian i.i.d [20]) and that the initial iterate  $x^0$  is close enough to  $x^*$ , the sequence of iterates defined by (13) is well defined (any matrix in  $\partial_B R_\gamma(x^k)$  is nonsingular) and converges to  $x^*$  at a quadratic rate, i.e.,  $\limsup_{k \rightarrow \infty} \frac{\|x^{k+1} - x^*\|}{\|x^k - x^*\|^2} < \infty$  [16]. In fact, since

the fixed point residual is piecewise affine for the LASSO problem, it can be shown that (13) converges in a *finite number of iterations*, in exact arithmetic. Specializing iteration (13) to the LASSO problem, an element  $H_k$  of  $\partial_B R_\gamma(x^k)$  takes the form  $H_k = I - P_k(I - \gamma A^\top A)$ . Here  $P_k$  is a diagonal matrix with  $(P_k)_{ii} = 1$  for  $i \in \alpha_k$  and  $(P_k)_{ii} = 0$ , for  $i \in \beta_k$ , where

$$\alpha_k = \{i \mid |x_i^k - \gamma \nabla_i f(x^k)| > \gamma\lambda\}, \quad (14a)$$

$$\beta_k = \{i \mid |x_i^k - \gamma \nabla_i f(x^k)| \leq \gamma\lambda\}. \quad (14b)$$

Computing the Newton direction amounts to solving the so-called *Newton system*,  $H_k d^k = -R_\gamma(x^k)$ . Taking advantage of the special structure of  $P_k$  and applying some permutations, this simplifies to

$$d_{\beta_k}^k = -(R_\gamma(x^k))_{\beta_k}, \quad (15a)$$

$$\gamma A_{\alpha_k}^\top A_{\alpha_k} d_{\alpha_k}^k = -(R_\gamma(x^k))_{\alpha_k} - \gamma A_{\alpha_k}^\top A_{\beta_k} d_{\beta_k}^k. \quad (15b)$$

Taking into consideration (10), we obtain

$$(R_\gamma(x))_i = \begin{cases} \gamma(\nabla_i f(x) + s_i(x)\lambda), & i \in \alpha, \\ x_i, & i \in \beta, \end{cases} \quad (16)$$

where  $s_i(x) = \text{sign}(x_i - \gamma \nabla_i f(x))$ , and  $\alpha, \beta$  are defined using (14) (dropping index  $k$ ). Therefore, after further rearrangement and simplification, the Newton system becomes

$$d_{\beta_k}^k = -x_{\beta_k}^k, \quad (17a)$$

$$A_{\alpha_k}^\top A_{\alpha_k} (x_{\alpha_k}^k + d_{\alpha_k}^k) = A_{\alpha_k}^\top y - \lambda s_{\alpha_k}(x^k). \quad (17b)$$

Summing up, an iteration of the semismooth Newton method for solving the LASSO problem takes the form  $x^{k+1} = x^k + d^k$ , which becomes:

$$x_{\beta_k}^{k+1} = 0, \quad (18a)$$

$$x_{\alpha_k}^{k+1} = (A_{\alpha_k}^\top A_{\alpha_k})^{-1} (A_{\alpha_k}^\top y - \lambda s_{\alpha_k}(x^k)). \quad (18b)$$

Note a connection with (7): the index sets  $\alpha_k, \beta_k$  serve as estimates for the nonzero and zero components of  $x^*$ .

There are two obstacles that we need to overcome before we arrive at a sound, globally convergent algorithmic scheme for solving the LASSO problem. The first one is that the semismooth Newton method (18) converges only when started close to the solution. The second obstacle concerns the fact that (18) might not be well defined, in the sense that  $A_{\alpha_k}$  might not have full column rank, whence  $A_{\alpha_k}^\top A_{\alpha_k}$  will be singular. Indeed when  $\lambda$  is very small or when  $x^k$  is far from the solution, the index set  $\alpha_k$  defined in (14a) might have cardinality larger than  $m$  (which is the only case where a singularity may arise given our construction of  $A$ ), the number of rows of  $A$ . The next two subsections are devoted to proposing strategies to overcome these two issues.

Overall, the algorithm can be seen as an active set strategy where large changes on the active set are allowed in every iteration (instead of only one index) leading to faster convergence.

### B. Globalization strategy

To enforce global convergence of Newton-type methods for solving nonlinear systems of equations it is customary to use a merit function based on which a step  $\tau_k$  is selected which guarantees that

$$x^{k+1} = x^k + \tau_k d^k \quad (19)$$

decreases the merit function sufficiently.

Recently the following merit function, namely the *Forward-Backward Envelope* (FBE), was proposed for problems of the form (3):

$$\varphi_\gamma(x) = \inf_{z \in \mathbb{R}^n} \{f(x) + \nabla f(x)^\top (z - x) + g(z) + \frac{1}{2\gamma} \|z - x\|_2^2\}. \quad (20)$$

It is easy to see that function inside the infimum is strongly convex with respect to  $z$  and the infimum is uniquely achieved by the forward-backward step  $T_\gamma(x) = \text{prox}_{\gamma g}(x - \gamma \nabla f(x))$ . Therefore, in order to evaluate  $\varphi_\gamma$  at a point  $x$  one simply needs to be able to perform the same operations required by FISTA. Furthermore, function  $\varphi_\gamma$  is continuously differentiable with gradient given by

$$\nabla \varphi_\gamma(x) = \gamma^{-1} (I - \gamma \nabla^2 f(x)) R_\gamma(x). \quad (21)$$

If  $\gamma < 1/\|A\|_2^2$ , then solutions of (3) are exactly the stationary points of  $\varphi_\gamma$ , *i.e.*, the points for which  $\nabla \varphi_\gamma(x)$  becomes zero. In fact one can additionally show that minimizing the FBE (which is an unconstrained smooth optimization problem) is entirely equivalent to solving (3), in the sense that  $\inf \varphi = \inf \varphi_\gamma$  and  $\text{argmin} \varphi = \text{argmin} \varphi_\gamma$ . In the case of LASSO where  $f$  is quadratic the FBE is convex.

It is not hard to check that, provided it is well defined, the semismooth Newton direction  $d^k$  given by (15) is in fact a direction of descent for the FBE, *i.e.*,  $\nabla \varphi_\gamma(x^k)^\top d^k < 0$ . Therefore, one can perform a standard backtracking line-search to find a suitable step that guarantees the Armijo condition and hence global convergence: pick the first nonnegative integer  $i_k$  such that  $\tau_k = 2^{-i_k}$  satisfies

$$\varphi_\gamma(x^k + \tau_k d^k) \leq \varphi_\gamma(x^k) + \zeta \tau_k \nabla \varphi_\gamma(x^k)^\top d^k. \quad (22)$$

where  $\zeta \in (0, 1/2)$ , and then set  $x^{k+1} = x^k + \tau_k d^k$ . Furthermore, this step choice guarantees that as soon as  $x^k$  is close enough to the solution,  $\tau_k = 1$  will always satisfy (22) and the iterates will be given by the (pure) semismooth Newton method (18) inheriting all its convergence properties.

### C. Continuation strategy

Since the goal of solving the LASSO problem is to recover the sparsest solution of  $Ax = y$ , we know that such a solution will have a  $\|x^*\|_0$  much smaller than  $m$ , hence as soon as  $x^k$  is close to the solution,  $A_{\alpha_k}$  will have full column rank.

When  $\lambda$  is small or when  $x^0$  is far from the solution, the cardinality of (14a) can be larger than  $m$  rendering the left-hand side matrix in (17b) singular. This said, we know that iterates for which the set  $\alpha_k$  contains more than  $m$  indices, are known to be away from the optimal solution since we know for sure that at the solution the cardinality of  $\alpha(x^*) = \{i \mid$

$|x_i^* - \gamma \nabla f(x^*)| > \gamma \lambda\} = \{i \mid x_i^* \neq 0\}$  will be (much) smaller than  $m$ .

In order to enforce that  $\alpha_k$  contains few elements, a simple continuation strategy that gradually reduces the used regularization parameter  $\bar{\lambda}$  to the target value  $\lambda$  is employed. Following [21], we start with  $\bar{\lambda} = \max\{\lambda, \|A^\top \nabla f(x_0)\|_\infty\}$  and we decrease  $\bar{\lambda} \leftarrow \max\{\lambda, \eta \bar{\lambda}\}$ , for some  $\eta \in (0, 1)$ , whenever  $\|R_\gamma(x^k; \bar{\lambda})\| \leq \bar{\lambda} \epsilon_k$  with  $\{\epsilon_k\} \rightarrow 0$ , where  $R_\gamma(x; \bar{\lambda})$  is the fixed point residual introduced in (11) replacing  $g$  by  $\bar{g}(x) = \bar{\lambda} \|x\|_1$ . This allows to solve linear systems of small dimension to determine the Newton direction. A conceptual pseudo-algorithm summarizing the basic steps of the proposed algorithm is shown hereafter.

---

#### Algorithm 1 Forward-Backward Newton with continuation

---

**Input:**  $A, y, x^0 \in \mathbb{R}^n$  (initial guess),  $\gamma \in (0, 1/\|A\|_2^2)$ ,  $\lambda > 0$ ,  $\eta \in (0, 1)$ ,  $\epsilon$  (tolerance)  
 $\bar{\lambda} \leftarrow \max\{\lambda, \|A^\top \nabla f(x_0)\|_\infty\}$ ,  $\bar{\epsilon} \leftarrow \epsilon$   
**while**  $\bar{\lambda} > \lambda$  or  $\|R_\gamma(x^k; \bar{\lambda})\| > \epsilon$  **do**  
 $x^{k+1} = x^k + \tau_k d^k$ ,  
 where  $d^k$  solves (17) and  $\tau_k$  satisfies (22) (using  $\bar{\lambda}$ )  
**if**  $\|R_\gamma(x^k; \bar{\lambda})\| \leq \lambda \bar{\epsilon}$  **then**  
 $\bar{\lambda} \leftarrow \max\{\lambda, \eta \bar{\lambda}\}$ ,  $\bar{\epsilon} \leftarrow \eta \bar{\epsilon}$   
**end if**  
**end while**

---

## IV. SIMULATIONS

In this section we apply the proposed methodology to various data streams and we compare it to standard algorithms such as FISTA, ADMM [22] and the interior point method of Kim *et al.* [9], aka L1LS method. In our approach we used the continuation strategy described above with  $\eta = 0.5$  and an Armijo line search. The required tolerance for the termination of all algorithms was set to  $10^{-8}$ .

We observed that after decompressing the first window, the number of iterations required for convergence was remarkably low (in most cases, around 4 iterations for each window were sufficient). It should be highlighted that after the first decomposition, the computational cost of the algorithm decreases significantly. This is first because of the aforementioned warm-start and second because the value of the residual  $r = Ax - y$  is updated by using only vector-vector operations. Updating  $A^{(i+1)} \leftarrow A^{(i)} \Pi$  and  $x_0^{(i+1)} \leftarrow \Pi^\top \hat{x}^{(i)}$  (the starting point for solving LASSO in the  $(i+1)$ -th window) and using the fact that  $\Pi$  is orthogonal, we have that  $r^{(i+1)} = A^{(i+1)} x_0^{(i+1)} - y^{(i+1)} = r^{(i)} + y^{(i)} - y^{(i+1)}$ .

A stream of total length  $N = 10^6$  was generated as follows: its entries are drawn from  $\text{Ber}(S)$ , where  $S$  is average stream sparsity. Then, the non-zero entries are taken uniform in  $[-2, -1] \cup [1, 2]$  and multiplied by  $8\sigma\sqrt{2 \log N}$  (to satisfy the dynamic range assumption of [8]<sup>2</sup>, based on

<sup>2</sup>In practice, exact knowledge of noise variance is not crucial as this result essentially dictates the minimum required SNR for provable success of RCS, and the selection of  $\lambda$  can be made based on rough estimates of  $\sigma$ .

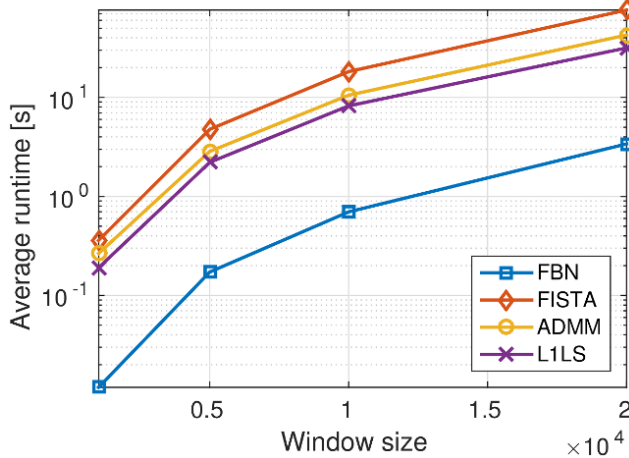


Figure 1: Average runtimes varying the window size. FBN: proposed forward-backward Newton method, L1LS [9], ADMM [13], FISTA [11].

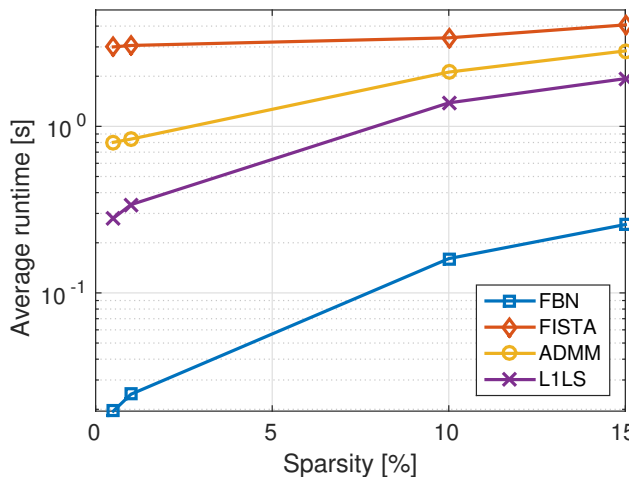


Figure 2: Average runtimes varying the sparsity of the data stream.

selected noise variance  $\sigma^2$ . We then select window size  $n$ , let  $s = nS$ ,  $m = 4s$ , and generate sampling matrix  $A$  with i.i.d. entries  $\mathcal{N}(0, 1/m)$ . For LASSO, we pick  $\lambda = 2\sigma\sqrt{2\log n}$ .

In Figures 1 and 2 we observe that the proposed algorithm outperforms all state-of-the-art methods by an order of magnitude (where the same warm-starting mechanism was used in all cases). The results presented in Figure 1 were obtained for a fixed sparsity 10% and  $w^{(i)}$  being a zero-mean normally distributed noise with variance 0.01. In Figure 2, for the same noise level and using a window of size  $n = 5000$  we show how the runtime is affected by the sparsity of the data stream.

## V. CONCLUSIONS

We have proposed an efficient method for successively decompressing the entries of a data stream sampled using Recursive Compressed Sensing [6], [14]. We have proposed

a second-order proximal method for solving LASSO with accelerated convergence. Our scheme is very efficient as each iteration boils down to solving a linear system of low dimension, which we may further avoid by a single Cholesky factorization at a pre-processing step. We have tested our algorithm against the state-of-art for various windows sizes and sparsity patterns; our experiments depict notable speed-up which may render RCS suitable for an online implementation under stringent time constraints.

## REFERENCES

- [1] E. Candès and M. Wakin, "An introduction to compressive sampling," *IEEE Signal Processing Magazine*, vol. 25, no. 2, pp. 21–30, 2008.
- [2] 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.
- [3] D. L. Donoho, "Compressed sensing," *IEEE Transactions on Information Theory*, vol. 52, pp. 1289–1306, 2006.
- [4] E. Candès and T. Tao, "Near-optimal signal recovery from random projections: Universal encoding strategies?" *IEEE Transactions on Information Theory*, vol. 52, no. 12, pp. 5406–5425, 2006.
- [5] S. Qaisar, R. Bilal, W. Iqbal, M. Naureen, and S. Lee, "Compressive sensing: From theory to applications, a survey," *Journal of Communications and Networks*, vol. 15, no. 5, pp. 443–456, 2013.
- [6] N. Freris, O. Öçal, and M. Vetterli, "Recursive Compressed Sensing," Tech. Rep., 2014. [Online]. Available: <http://arxiv.org/abs/1312.4895>
- [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] E. Candès and Y. Plan, "Near-ideal model selection by  $\ell_1$  minimization," *The Annals of Statistics*, vol. 37, pp. 2145–2177, 2009.
- [9] S.-J. Kim, K. Koh, M. Lustig, S. Boyd, and D. Gorinevsky, "An interior-point method for large-scale  $\ell_1$ -regularized least squares," *IEEE Journal of Selected Topics in Signal Processing*, vol. 1, no. 4, pp. 606–617, 2007.
- [10] N. Parikh and S. Boyd, "Proximal algorithms," *Foundations and Trends in Optimization*, vol. 1, no. 3, pp. 123–231, 2013.
- [11] 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.
- [12] S. J. Wright, R. D. Nowak, and M. A. T. Figueiredo, "Sparse reconstruction by separable approximation," *IEEE Transactions on Signal Processing*, vol. 57, no. 7, pp. 2479–2493, 2009.
- [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] N. Freris, O. Öçal, and M. Vetterli, "Compressed sensing of streaming data," in *Proceedings of the 51st Allerton Conference on Communication, Control and Computing*, Oct. 2013, pp. 1242–1249.
- [15] P. Patrinos and A. Bemporad, "Proximal Newton methods for convex composite optimization," in *IEEE Conference on Decision and Control*, Florence, Italy, 2013, pp. 2358–2363.
- [16] P. Patrinos, L. Stella, and A. Bemporad, "Forward-backward truncated Newton methods for convex composite optimization," Tech. Rep., 2014. [Online]. Available: <http://arxiv.org/abs/1402.6655>
- [17] L. Stella, A. Themelis, and P. Patrinos, "Forward-backward quasi-Newton methods for nonsmooth optimization problems," Tech. Rep., 2016. [Online]. Available: <http://arxiv.org/abs/1604.08096>
- [18] P. Patrinos, L. Stella, and A. Bemporad, "Douglas-Rachford splitting: Complexity estimates and accelerated variants," in *IEEE Conference on Decision and Control*, California, USA, 2014, pp. 4234–4239.
- [19] Y. Nesterov, "Gradient methods for minimizing composite functions," *Mathematical Programming*, vol. 140, no. 1, pp. 125–161, 2013.
- [20] R. J. Tibshirani, "The Lasso problem and uniqueness," Tech. Rep., 2012. [Online]. Available: <http://arxiv.org/abs/1206.0313>
- [21] L. Xiao and T. Zhang, "A proximal-gradient homotopy method for the sparse least-squares problem," *SIAM Journal on Optimization*, vol. 23, no. 2, pp. 1062–1091, 2013.
- [22] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2011.