

# Recursive Dynamic CS: Recursive Recovery of Sparse Signal Sequences from Compressive Measurements: A Review

Namrata Vaswani and Jinchun Zhan

**Abstract**—In this article, we review the literature on recursive algorithms for reconstructing a time sequence of sparse signals from a greatly reduced number of linear projection measurements. The signals are sparse in some transform domain referred to as the sparsity basis, and their sparsity pattern (support set of the sparsity basis coefficients’ vector) can change with time. We also summarize the theoretical results (guarantees for exact recovery and accurate recovery at a given time and for stable recovery over time) that exist for some of the proposed algorithms. An important application where this problem occurs is dynamic magnetic resonance imaging (MRI) for real-time medical applications such as interventional radiology and MRI-guided surgery, or in functional MRI to track brain activation changes.

## I. INTRODUCTION

In this article, we review the literature on the design and analysis of *recursive* algorithms for causally reconstructing a time sequence of sparse (or approximately sparse) signals from a greatly reduced number of linear projection measurements. The signals are sparse in some transform domain referred to as the sparsity basis, and their sparsity pattern (support set of the sparsity basis coefficients’ vector) can change with time. By “recursive”, we mean use only signal estimate from the previous time and the current measurements’ vector to get the current signal’s estimate. A key application where this problem occurs is dynamic magnetic resonance imaging (MRI) for real-time medical applications such as interventional radiology and MRI-guided surgery [1], or in functional MRI to track brain activation changes. We show an example of a vocal tract (larynx) MR image sequence in Fig. 1. Notice that the images are piecewise smooth and hence wavelet sparse. As shown in Fig 2, their sparsity pattern in the wavelet transform domain changes with time, but the changes are slow.

Other applications where real-time imaging is needed, and hence recursive recovery approaches would be useful, include real-time single-pixel video imaging [2], [3], real-time video compression/decompression, real-time sensor network based sensing of time-varying fields [4], or real-time extraction of the foreground image sequence (sparse image) from a slow changing background image sequence (well modeled as lying in a slow-changing low-dimensional subspace of the full space [5], [6]) using recursive projected compressive sensing

(ReProCS) [7], [8]. For other potential applications, see [9], [10]. We explain these applications in Section III-C.

The sparse recovery problem has been studied for a long time. In the signal processing literature, the works of Mallat and Zhang [11] (matching pursuit), Chen and Donoho (basis pursuit) [12], Feng and Bresler [13], [14] (spectrum blind recovery of multi-band signals), Gorodnitsky and Rao [15], [16] (a reweighted minimum 2-norm algorithm for sparse recovery) and Wipf and Rao [17] (sparse Bayesian learning for sparse recovery) were among the first works on this topic. The papers by Candes, Romberg, Tao and by Donoho [18], [19], [20] introduced the compressed sensing or compressive sensing (CS) problem. The idea of CS is to compressively sense signals and images that are sparse in some known domain and then use sparse recovery techniques to recover them. The most important contribution of [18], [19], [20] was that they provided practically meaningful conditions for exact sparse recovery using basis pursuit. In the last decade since these papers appeared, this problem has received a lot of attention. Very often the terms “sparse recovery” and “CS” are used interchangeably and we also do this in this article.

Consider the dynamic CS problem, i.e. the problem of recovering a time sequence of sparse signals. Most of the initial solutions for this problem consisted of batch algorithms. These can be split into two categories depending on what assumption they use on the time sequence. The first category is batch algorithms that solve the multiple measurements’ vectors (MMV) problem and these use the assumption that the support set of the sparse signals *does not* change with time [21], [22], [23], [24]. The second category is batch algorithms that treat the entire time sequence as a single sparse spatiotemporal signal by assuming Fourier sparsity along the time axis [3], [25], [26]. However, in many situations neither assumption is valid: the MMV assumption of constant support over time is not valid in dynamic MRI (see Fig. 2); the Fourier sparsity along time assumption is not valid for brain functional MRI data when studying brain activations in response to stimuli [27]. Moreover even when these are valid assumptions, batch algorithms are offline, slower and their memory requirement increases linearly with the sequence length.

The alternative – solving the CS problem at each time separately (henceforth referred to as *simple-CS*) – is online, fast and low on memory, but it needs many more measurements for accurate recovery. For dynamic MRI or any of the other projection imaging applications, this means a proportionally higher scan time. On the other hand, the computational and

N. Vaswani and J. Zhan are with the Dept. of Electrical and Computer Engineering, Iowa State University, Ames, IA, USA. Email: namrata@iastate.edu. This work was supported by NSF grant CCF-0917015 and CCF-1117125.

Original sequence



Fig. 1. We show a dynamic MRI sequence of the vocal tract (larynx) that was acquired when the person was speaking a vowel.

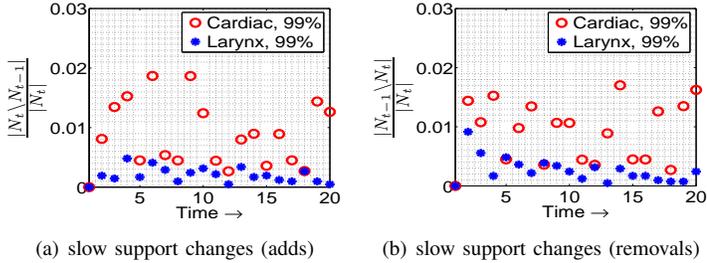


Fig. 2. In these figures,  $\mathcal{N}_t$  refers to the 99%-energy support of the 2D discrete wavelet transform (two-level Daubechies-4 2D DWT) of the larynx sequence shown in Fig. 1 and of a cardiac sequence. The 99%-energy support size,  $|\mathcal{N}_t|$ , varied between 6-7% of the image size in both cases. We plot the number of additions (top) and the number of removals (bottom) as a fraction of the support size. Notice that all support change sizes are less than 2% of the support size.

storage complexity of most of the recursive algorithms that we will discuss below is only as much as that of simple-CS solutions, but their reconstruction performance is much better.

### A. Paper Organization

The rest of this article is organized as follows. We summarize the notation used in the entire paper and provide a short overview of some of the approaches for solving the static sparse recovery or compressive sensing (CS) problem in Section II. Next, in Section III, we define the recursive recovery problem formulation, discuss its applications and explain why new approaches are needed to solve it. We split the discussion of the proposed solutions into two sections. In Section IV, we describe algorithms that only use slow support change. In Section V, we discuss algorithms that also use slow signal value change. Sections VI and VII describe the theoretical results. Section VI summarizes the key results for exact reconstruction in the noise-free case. Section VII gives error bounds in the noisy case as well as key results for error stability over time (obtaining time-invariant and small bounds on the error). In Section VIII, we either provide links for the code to implement an algorithm, or we give the stepwise algorithm and explain how to set its parameters in practice. Numerical experiments comparing the various proposed approaches are shown in Section IX.

In Section X, we provide a detailed discussion of related problems and their solutions and how some of those can be used in conjunction with the work described here. Many open questions for future work are also mentioned here. We conclude the paper in Section XI.

## II. NOTATION AND BACKGROUND

### A. Notation

For a set  $\mathcal{T}$ , we use  $\mathcal{T}^c$  to denote the complement of  $\mathcal{T}$  w.r.t.  $[1, m] := [1, 2, \dots, m]$ , i.e.  $\mathcal{T}^c := \{i \in [1, m] : i \notin \mathcal{T}\}$ . The notation  $|\mathcal{T}|$  denotes the size (cardinality) of the set  $\mathcal{T}$ . The set operation  $\setminus$  denotes set difference, i.e. for two sets  $\mathcal{T}_1, \mathcal{T}_2$ ,  $\mathcal{T}_1 \setminus \mathcal{T}_2 := \mathcal{T}_1 \cap \mathcal{T}_2^c$ . We use  $\emptyset$  to denote the empty set.

For a vector,  $v$ , and a set,  $\mathcal{T}$ ,  $v_{\mathcal{T}}$  denotes the  $|\mathcal{T}|$  length sub-vector containing the elements of  $v$  corresponding to the indices in the set  $\mathcal{T}$ . Also,  $\|v\|_k$  denotes the  $\ell_k$  norm of a vector  $v$ . When  $k = 0$ ,  $\|v\|_0$  counts the number of nonzero elements in the vector  $v$ . If just  $\|v\|$  is used, it refers to  $\|v\|_2$ .

For a matrix  $M$ ,  $\|M\|_k$  denotes its induced  $k$ -norm, while just  $\|M\|$  refers to  $\|M\|_2$ .  $M'$  denotes the transpose of  $M$  and  $M^\dagger$  denotes its Moore-Penrose pseudo-inverse. For a tall matrix,  $M$ ,  $M^\dagger := (M'M)^{-1}M'$ . For a fat matrix (a matrix with more columns than rows),  $A$ ,  $A_{\mathcal{T}}$  denotes the sub-matrix obtained by extracting the columns of  $A$  corresponding to the indices in  $\mathcal{T}$ . We use  $I$  to denote the identity matrix.

### B. Sparse Recovery or Compressive Sensing (CS)

The goal of sparse recovery or ‘‘CS’’ is to reconstruct an  $m$ -length sparse signal,  $x$ , with support  $\mathcal{N}$ , from an  $n$ -length measurement vector,  $y := Ax$  or from  $y := Ax + w$ , with  $\|w\|_2 \leq \epsilon$  (noisy case) when  $A$  has more columns than rows (underdetermined system), i.e.  $n < m$ . Consider the noise-free case. Let  $s = |\mathcal{N}|$ . It is easy to show that this problem is solved if we can find the sparsest vector satisfying  $y = A\beta$ , i.e. if we can solve

$$\min_{\beta} \|\beta\|_0 \text{ subject to } y = A\beta \quad (1)$$

and if any set of  $2s$  columns of  $A$  are linearly independent [19]. However doing this is impractical since it requires a combinatorial search. The complexity of solving (1) is  $O(m^s)$ , i.e. it is exponential in the support size. In the last two decades, many practical (polynomial complexity) approaches have been developed. Most of the classical approaches can be split as follows (a) convex relaxation approaches, (b) greedy algorithms, (c) iterative thresholding methods, and (d) sparse Bayesian learning (SBL) based methods. We explain the key ideas of these approaches below. We should mention that, besides these, there are many more solution approaches which are not reviewed here.

The convex relaxation approaches are also referred to as  $\ell_1$  minimization and these replace the  $\ell_0$  norm in (1) by the  $\ell_1$  norm which is the closest norm to  $\ell_0$  that is convex. Thus, in the noise-free case, one solves

$$\min_{\beta} \|\beta\|_1 \text{ subject to } y = A\beta \quad (2)$$

The above program is referred to as *basis pursuit (BP)* [12]. Since this was the program analyzed in the first two CS papers, some later works just use the term ‘‘CS’’ when referring to it. In the noisy case, the constraint is replaced by  $\|y - A\beta\|_2 \leq \epsilon$  where  $\epsilon$  is the bound on the  $\ell_2$  norm of the noise, i.e.,

$$\min_{\beta} \|\beta\|_1 \text{ subject to } \|y - A\beta\|_2 \leq \epsilon \quad (3)$$

This is referred to as *BP-noisy*. In practical problems where the noise bound may not be known, one can solve an unconstrained version of this problem by including the data term as what is often called a “soft constraint” (the Lagrangian version); the resulting program is also faster to solve:

$$\min_{\beta} \gamma \|\beta\|_1 + 0.5 \|y - A\beta\|_2^2 \quad (4)$$

The above is referred to as *BP denoising (BPDN)* [12].

Another class of solutions for the CS problem consists of greedy algorithms. These get an estimate of the support of  $x$ , and of  $x$ , in a greedy fashion. This is done by finding one, or a set of, indices “contributing the maximum energy” to the measurement residual from the previous iteration. We summarize below the simplest greedy algorithm - orthogonal matching pursuit (OMP) to explain this idea. The first known greedy algorithm is Matching Pursuit [11] and this was a precursor to OMP [28]. OMP assumes that the columns of  $A$  have unit norm. As we explain below in Remark 2.1, when this is not true, there is a simple fix. OMP proceeds as follows. Let  $\hat{x}^k$  denote the estimate of  $x$  and let  $\hat{\mathcal{N}}^k$  denote the estimate of its support  $\mathcal{N}$  at the  $k^{\text{th}}$  iteration. Also let  $r^k$  denote the measurement residual at iteration  $k$ . Initialize with  $\hat{\mathcal{N}}^0 = \emptyset$  and  $r^0 = y$ , i.e. initialize the measurement residual with the measurement itself. For each  $k \geq 1$  do

- 1) Compute the index  $i$  for which  $A_i$  has maximum correlation with the previous residual and add it to the support estimate, i.e. compute  $\hat{i} = \arg \max_i |A_i' r^{k-1}|$  and update  $\hat{\mathcal{N}}^k \leftarrow \hat{\mathcal{N}}^{k-1} \cup \{\hat{i}\}$ .
- 2) Compute  $\hat{x}^k$  as the LS estimate of  $x$  on  $\hat{\mathcal{N}}^k$  and use to compute the new measurement residual, i.e.  $\hat{x}^k = I_{\hat{\mathcal{N}}^k} A_{\hat{\mathcal{N}}^k}^\dagger y$  and  $r^k = y - A\hat{x}^k$ .
- 3) Stop when  $k$  equals the support size of  $x$  or when  $\|r^k\|_2$  is small enough. Output  $\hat{\mathcal{N}}^k, \hat{x}^k$ .

*Remark 2.1 (unit norm columns of  $A$ ):* Any matrix can be converted into a matrix with unit  $\ell_2$ -norm columns by right multiplying it with a diagonal matrix  $D$  that contains  $\|a_i\|_2^{-1}$  as its entries. Here  $a_i$  is the  $i$ -th column of  $A$ . Thus, for any matrix  $A$ ,  $A_{\text{normalized}} = AD$ . Whenever normalized columns of  $A$  are needed, one can rewrite  $y = Ax$  as  $y = ADD^{-1}x = A_{\text{normalized}}\tilde{x}$  and first recover  $\tilde{x}$  from  $y$  and then obtain  $x = D\tilde{x}$ .

A third solution approach is called Iterative Hard Thresholding or IHT [29], [30]. This is an iterative algorithm that proceeds by hard thresholding the current “estimate” of  $x$  to  $s$  largest elements. Let  $H_s(a)$  denote the hard thresholding operator which zeroes out all but the  $s$  largest elements of the vector  $a$ . Let  $\hat{x}^k$  denote the estimate of  $x$  at the  $k^{\text{th}}$  iteration. Then IHT proceeds as follows.

$$\begin{aligned} \hat{x}^0 &= 0 \\ \hat{x}^{k+1} &= H_s(\hat{x}^k + A'(y - A\hat{x}^k)) \end{aligned} \quad (5)$$

Another commonly used approach to solving the sparse recovery problem is sparse Bayesian learning (SBL) [31], [17]. SBL was first developed for sparse recovery in Wipf and Rao [17]. In SBL, one models the sparse vector  $x$  as consisting of independent Gaussian components with zero mean and variances  $\gamma_i$  for the  $i$ -th component. The observation noise

is assumed to be i.i.d. Gaussian with variance  $\sigma^2$ . It then develops an expectation maximization (EM) algorithm to estimate the hyper-parameters  $\{\sigma^2, \gamma_1, \gamma_2, \dots, \gamma_m\}$  from the observation vector  $y$  using evidence maximization (type-II maximum likelihood). Since  $x$ 's are sparse, it is shown that the estimates of a lot of the  $\gamma_i$ 's will be zero or nearly zero (and can be zeroed out). Once the hyper-parameters are estimated, SBL computes the MAP estimate of  $x$  (which is a simple closed form expression under the assumed joint Gaussian model).

### C. Restricted Isometry Property and Null Space Property

In this section we describe properties introduced in recent work that are either sufficient or necessary and sufficient to ensure exact sparse recovery in the noise-free case.

The Restricted Isometry Property (RIP) which was introduced in [19] is defined as follows.

*Definition 2.2:* A matrix  $A$  satisfies the *RIP of order  $s$*  if its restricted isometry constant (RIC)  $\delta_s(A) < 1$  [19]. The *restricted isometry constant (RIC)*,  $\delta_s(A)$ , for a matrix  $A$ , is the smallest real number satisfying

$$(1 - \delta_s)\|c\|^2 \leq \|A_{\mathcal{T}}c\|^2 \leq (1 + \delta_s)\|c\|^2 \quad (6)$$

for all subsets  $\mathcal{T} \subseteq [1, m]$  of cardinality  $|\mathcal{T}| \leq s$  and all real vectors  $c$  of length  $|\mathcal{T}|$  [19].

It is easy to see that  $(1 - \delta_s) \leq \|A_{\mathcal{T}'}A_{\mathcal{T}}\| \leq (1 + \delta_s)$ ,  $\|(A_{\mathcal{T}'}A_{\mathcal{T}})^{-1}\| \leq 1/(1 - \delta_s)$  and  $\|A_{\mathcal{T}}^\dagger\| \leq 1/\sqrt{(1 - \delta_s)}$ .

*Definition 2.3:* The *restricted orthogonality constant (ROC)*,  $\theta_{s,\bar{s}}$ , for a matrix  $A$ , is the smallest real number satisfying

$$|c_1' A_{\mathcal{T}_1}' A_{\mathcal{T}_2} c_2| \leq \theta_{s,\bar{s}} \|c_1\| \|c_2\| \quad (7)$$

for all disjoint sets  $\mathcal{T}_1, \mathcal{T}_2 \subseteq [1, m]$  with  $|\mathcal{T}_1| \leq s$ ,  $|\mathcal{T}_2| \leq \bar{s}$ ,  $s + \bar{s} \leq m$ , and for all vectors  $c_1, c_2$  of length  $|\mathcal{T}_1|, |\mathcal{T}_2|$  [19].

It is not hard to show that  $\|A_{\mathcal{T}_1}' A_{\mathcal{T}_2}\| \leq \theta_{s,\bar{s}}$  [32] and that  $\theta_{s,\bar{s}} \leq \delta_{s+\bar{s}}$  [19].

The following result was proved in [33].

*Theorem 2.4 (Exact recovery and error bound for BP and BP-noisy):* In the noise-free case, i.e. when  $y := Ax$ , if  $\delta_s(A) < \sqrt{2} - 1$ , the solution of BP, (2), achieves exact recovery.

Consider the noisy case, i.e.,  $y := Ax + w$  with  $\|w\|_2 \leq \epsilon$ . Denote the solution of BP-noisy, (3), by  $\hat{x}$ . If  $\delta_s(A) < 0.207$ , then

$$\|x - \hat{x}\|_2 \leq C_1(s)\epsilon \leq 7.50\epsilon \text{ where } C_1(k) := \frac{4\sqrt{1 + \delta_k}}{1 - 2\delta_k}$$

With high probability (whp), random Gaussian matrices and various other random matrix ensembles satisfy the RIP of order  $s$  whenever the number of measurements  $n$  is of the order of  $s \log m$  and  $m$  is large enough [19].

Null space property (NSP) is another property used to prove results for exact sparse recovery [34], [35]. NSP ensures that every vector  $v$  in the null space of  $A$  is not too sparse.

*Definition 2.5:* A matrix  $A$  is said to satisfy the *null space property (NSP)* of order  $s$  if for any vector  $v$  in the null space of  $A$ ,

$$\|v_S\|_1 < 0.5\|v\|_1, \text{ for all sets } S \text{ with } |S| \leq s$$

NSP is known to be a necessary and sufficient condition for exact recovery of  $s$ -sparse vectors [34], [35].

### III. THE PROBLEM, APPLICATIONS AND MOTIVATION

We describe the problem setting in Sec III-A. In Sec III-B we explain why new techniques are needed to solve this problem. Applications are described in Sec III-C.

#### A. Problem Definition

The goal of the work reviewed in this article is to design recursive algorithms for causally reconstructing a time sequence of sparse signals from a greatly reduced number of measurements at each time. To be more specific, we would like to develop approaches that provably require *fewer* measurements for exact or accurate recovery compared to simple-CS solutions. This problem was first introduced in [36], [32]. In this paper, we use “*simple-CS*” to refer to the problem of recovering each sparse signal in a time sequence separately from its measurements’ vector.

Let  $t$  denote the discrete time index. We would like to recover the sparse vector sequence  $\{x_t\}$  from undersampled and possibly noisy measurements  $\{y_t\}$  satisfying

$$y_t := A_t x_t + w_t, \quad \|w_t\|_2 \leq \epsilon \quad (8)$$

where  $A_t := H_t \Phi$  is an  $n \times m$  matrix with  $n < m$  (fat matrix). Here  $H_t$  is the measurement matrix and  $\Phi$  is an orthonormal matrix for the sparsity basis or it can be a dictionary matrix. In the above formulation,  $z_t := \Phi x_t$  is actually the signal (or image) whereas  $x_t$  is its representation in its sparsity basis.

We use  $\mathcal{N}_t$  to denote the support set of  $x_t$ , i.e.

$$\mathcal{N}_t := \{i : (x_t)_i \neq 0\}.$$

When we say  $x_t$  is sparse, it means that  $|\mathcal{N}_t| \ll m$ .

The goal is to “recursively” reconstruct  $x_t$  from  $y_0, y_1, \dots, y_t$ , i.e. use only  $\hat{x}_{t-1}$  and  $y_t$  for reconstructing  $x_t$ . In the rest of this article, we often refer to this problem as the “*recursive recovery*” problem.

In order to solve this problem using fewer measurements than those needed for simple-CS techniques, one can use the practically valid assumption of slow support (sparsity pattern) change [36], [32], [37], i.e.,

$$|\mathcal{N}_t \setminus \mathcal{N}_{t-1}| \approx |\mathcal{N}_{t-1} \setminus \mathcal{N}_t| \ll |\mathcal{N}_t|. \quad (9)$$

Notice from Fig. 2 that this is valid for dynamic MRI sequences.

A second assumption that can also be exploited in cases where the signal value estimates are reliable enough (or it is at least known how reliable the signal values are), is that of slow signal value change, i.e.

$$\|(x_t - x_{t-1})_{\mathcal{N}_{t-1} \cup \mathcal{N}_t}\|_2 \ll \|(x_t)_{\mathcal{N}_{t-1} \cup \mathcal{N}_t}\|_2. \quad (10)$$

This is, of course, is a commonly used assumption in almost all past work on tracking algorithms as well as in work on adaptive filtering algorithms.

#### B. Motivation: why are new techniques needed?

One question that comes to mind when thinking about how to solve the recursive recovery problem is why is a new set of approaches needed and why can we not use adaptive filtering ideas applied to simple-CS solutions? This was also a question raised by an anonymous reviewer.

The reason is as follows. Adaptive filtering relies on slow signal value change which is the only thing one can use for dense signal sequences. This can definitely be done for sparse and approximately sparse (compressible) signal sequences as well, and does often result in good experimental results. However sparse signal sequences have more structure that can often be exploited to (i) get better algorithms and (ii) prove stronger results about them. For example, as we explain next, adaptive filtering based recursive recovery techniques do not allow for exact recovery using fewer measurements than what simple-CS solutions need. Also, even when adaptive filtering based techniques work well enough, one cannot obtain recovery error bounds for them under weak enough assumptions without exploiting an assumption specific to the current problem, that of sparsity and slow sparsity pattern change.

Adaptive filtering ideas can be used to adapt simple-CS solutions in one of the following fashions. BP and BP-noisy can be replaced by the following

$$\begin{aligned} \hat{\beta} &= \arg \min_{\beta} [\|\beta\|_1 \text{ s.t. } \|y_t - A\hat{x}_{t-1} - A\beta\|_2 \leq \epsilon] \\ \hat{x}_t &= \hat{x}_{t-1} + \hat{\beta} \end{aligned} \quad (11)$$

with setting  $\epsilon = 0$  in the noise-free case. The above was referred as *CS-residual* in [38] (where the authors first explained why this could be significantly improved by using the slow support change assumption). However, this is a misnomer (that is used because BP is often referred to as “CS”), and one should actually refer to the above as *BP-residual*. Henceforth, we refer to it as *CS-residual(BP-residual)*.

Next consider IHT. It can be adapted in one of two possible ways. The first is similar to the above: replace  $y$  by  $y_t - A\hat{x}_{t-1}$  and  $x$  by  $\beta_t := (x_t - \hat{x}_{t-1})$  in (5). Once  $\beta_t$  is recovered, recover  $x_t$  by adding  $\hat{x}_{t-1}$  to it. An alternative and simpler approach is to adapt IHT as follows.

$$\begin{aligned} \hat{x}_t^0 &= \hat{x}_{t-1} \\ \hat{x}_t^k &= H_s(\hat{x}_t^{k-1} + A'(y_t - A\hat{x}_t^{k-1})). \end{aligned} \quad (12)$$

We refer to the above as *IHT-residual*.

Both the above solutions are using the assumption that the difference  $x_t - x_{t-1}$  is small. More generally, in some applications,  $x_t - x_{t-1}$  can be replaced by the prediction error vector,  $(x_t - f(x_{t-1}))$  where  $f(\cdot)$  is a known prediction function. However, notice that if  $x_t$  and  $x_{t-1}$  (or  $f(x_{t-1})$ ) are  $k$ -sparse, then the prediction error vector will also be at least  $k$ -sparse unless the prediction error is exactly zero along one or more coordinates. There are very few practical situations, e.g., quantized signal sequences [39] and a very good prediction scheme, where one can hope to get perfect prediction along a few dimensions. In most other cases, the prediction error vector will have support size  $k$  or larger. In all the results known so far, the sufficient conditions for exact recovery (or,

equivalently, the number of measurements needed to guarantee exact recovery) in a sparse recovery problem depend only on the support size of the sparse vector, e.g., see [19], [33], [29], [30]. Thus, when using either CS-residual(BP-residual) or IHT-residual, this number will be as much or more than what simple-CS solutions (e.g., BP or IHT) need. A similar thing is also observed in Monte Carlo based computations of the probability of exact recovery, e.g., see Fig. 3(a). Notice that both BP and CS-residual(BP-residual) need the same  $n$  for exact recovery with (Monte Carlo) probability one. On the other hand, solutions that exploit slow support change such as modified-CS(modified-BP) and weighted- $\ell_1$  need a much smaller  $n$ .

### C. Applications

As explained earlier an important application where recursive recovery with a reduced number of measurements is desirable is in real-time dynamic MR image reconstruction from a highly reduced set of measurements. Since MR data is acquired one Fourier projection at a time, the ability to accurately reconstruct using fewer measurements directly translates into reduced scan times. Shorter scan times along with online (causal) and fast (recursive) reconstruction can enable real-time imaging of fast changing physiological phenomena, thus making many interventional MRI applications such as MRI-guided surgery practically feasible [1]. Cross-sectional images of the brain, heart, larynx or other human organ images are piecewise smooth, and thus approximately sparse in the wavelet domain. In a time sequence, their sparsity pattern changes with time, but quite slowly [36].

In undersampled dynamic MRI,  $H_t$  is the partial Fourier matrix (consists of a randomly selected set of rows of the 2D discrete Fourier transform (DFT) matrix). If all images are rearranged as 1D vectors (for ease of understanding and for ease of using standard convex optimization toolboxes such as CVX), the measurement matrix

$$H_t = M_t(F_{m_1} \otimes F_{m_2})$$

where  $F_m$  is the  $m$ -point discrete Fourier transform (DFT) matrix,  $\otimes$  denotes Kronecker product and  $M_t$  is a random row selection matrix (if  $\mathcal{O}_t$  contains the set of indices of the observed discrete frequencies at time  $t$ , then  $M_t = I_{\mathcal{O}_t}$ ). Let  $z$  be an  $m = m_1 m_2$  length vector that contains an  $m_1 \times m_2$  image arranged as a vector (column-wise). Then  $(F_{m_1} \otimes F_{m_2})z$  returns a vector that contains the 2D DFT of this image. Medical images are often well modeled as being wavelet sparse and hence  $\Phi$  is the inverse 2D discrete wavelet transform (DWT) matrix. If  $W_m$  is the inverse DWT matrix corresponding to a chosen 1D wavelet, e.g. Haar, then

$$\Phi = W_{m_1} \otimes W_{m_2}.$$

Thus  $\Phi^{-1}z = \Phi'z$  returns the 2D-DWT of the image. In undersampled dynamic MRI, the measurement vector  $y_t$  is the observed Fourier coefficients of the vectorized image of interest,  $z_t$ , and the vector  $x_t$  is the 2D DWT of  $z_t$ . Thus  $y_t = A_t x_t + w_t$  where  $A_t = H_t \Phi$  with  $H_t$  and  $\Phi$  as defined

above. Slow support change of the wavelet coefficients vector,  $x_t$ , of medical image sequences is verified in Fig. 2.

In practice for large-sized images the matrix  $H_t$  as expressed above becomes too big to store in memory. Hence computing the 2D-DFT as  $F_{m_1} X F_{m_2}'$  where  $X$  is the image (matrix) is much more efficient. The same is true for DWT and its inverse. Moreover, in most cases, for large sized images, one needs algorithms that implement DFT or DWT directly without having to store the measurement matrix in memory. The only thing stored in memory is the set of indices of the observed entries,  $\mathcal{O}_t$ . The algorithms do not compute  $F_m x$  and  $F_m' z$  as matrix multiplications, but instead do this using a fast DFT (FFT) and inverse FFT algorithm. The same is done for the DWT and its inverse operations.

In single-pixel camera based video imaging,  $z_t$  is again the vectorized image of interest which can be modeled as being wavelet sparse, i.e.  $\Phi$  is as above. In this case, the measurements are random-Gaussian or Rademacher, i.e. each element of  $H_t$  is either an independent and identically distributed (i.i.d.) Gaussian with zero mean and unit variance or is i.i.d.  $\pm 1$  with equal probability of 1 or  $-1$ .

In real-time extraction of the foreground image sequence (sparse image) from a slow changing background image sequence (well modeled as lying in a low-dimensional space [5]) using recursive projected compressive sensing (ReProCS) [7], [40], [6],  $z_t = x_t$  is the foreground image sequence. The foreground images are sparse since they usually contains one or a few moving objects. Slow support change is often valid for them (assuming objects are not moving too fast compared to the camera frame rate), e.g., see [40, Fig 1c]. For this problem,  $\Phi = I$  (identity matrix) and  $A_t = H_t = I - \hat{P}_{t-1} \hat{P}_{t-1}'$  where  $\hat{P}_{t-1}$  is a tall matrix with orthonormal columns that span the estimated principal subspace of the background images. We describe this algorithm in detail in Section X-B1.

In the first two applications above, the image is only compressible (approximately sparse). Whenever we say ‘‘slow support change’’, we are referring to the changes in the  $b\%$ -energy-support (the largest set containing at most  $b\%$  of the total signal energy). In the third application, the foreground image sequence to be recovered is actually an exactly sparse image. Due to correlated motion over time, very often the support sets do indeed change slowly over time.

## IV. EXPLOITING SLOW SUPPORT CHANGE: SPARSE RECOVERY WITH PARTIAL SUPPORT KNOWLEDGE

Recall that when defining the recursive recovery problem, we introduced two assumptions that are often valid in practice - slow support change and slow nonzero signal value change. In problems where the signal value estimate from the previous time instant is not reliable (or, more importantly, if it is not known how reliable it is), it should not be used.

In this section, we describe solutions to the recursive recovery problem that only exploit the slow support change assumption, i.e. (9). Under this assumption this problem can be reformulated as one of *sparse recovery using partial support knowledge*. We can use the support estimate obtained from the previous time instant,  $\hat{\mathcal{N}}_{t-1}$ , as the ‘‘partial support

knowledge". If the support does not change slowly enough, but the change is still highly correlated, one can predict the support by using the correlation model information applied to the previous support estimate [8] (as explained in Sec IV-G). We give the reformulated problem in Sec IV-A followed by the proposed solutions for it. Support estimation is discussed in Sec IV-F. Finally, in Sec IV-G, we summarize the resulting algorithms for the original recursive recovery problem.

#### A. Reformulated problem: Sparse recovery using partial support knowledge

The goal is to recover a sparse vector,  $x$ , with support set  $\mathcal{N}$ , either from noise-free undersampled measurements,  $y := Ax$ , or from noisy measurements,  $y := Ax + w$ , when partial and possibly erroneous support knowledge,  $\mathcal{T}$ , is available. This problem was introduced in [41], [37].

The true support  $\mathcal{N}$  can be rewritten as

$$\mathcal{N} = \mathcal{T} \cup \Delta \setminus \Delta_e \quad \text{where} \quad \Delta := \mathcal{N} \setminus \mathcal{T}, \quad \Delta_e := \mathcal{T} \setminus \mathcal{N}$$

Let

$$s := |\mathcal{N}|, k := |\mathcal{T}|, u := |\Delta|, e := |\Delta_e|$$

It is easy to see that

$$s = k + u - e$$

The set  $\Delta$  contains the *misses* in the support knowledge and the set  $\Delta_e$  is the *extras* in it. We say the *support knowledge is accurate* if  $u \ll s$  and  $e \ll s$ .

This problem is also of independent interest, since in many static sparse recovery applications, partial support knowledge is often available. For example, when using wavelet sparsity for an image with very little black background (most of its pixel are nonzero), most of its wavelet scaling coefficients will also be nonzero. Thus, the set of indices of the wavelet scaling coefficients could serve as accurate partial support knowledge.

#### B. Least Squares CS-residual (LS-CS) or LS-BP

The Least Squares CS-residual (LS-CS), or more precisely the LS BP-residual (LS-BP), algorithm [42], [32] can be interpreted as the first solution for the above problem. It starts with computing an initial LS estimate of  $x$ ,  $\hat{x}_{\text{init}}$ , by assuming that its support set is equal to  $\mathcal{T}$ . Using this, it solves the CS-residual(BP-residual) problem followed by adding its solution to  $\hat{x}_{\text{init}}$ :

$$\hat{x}_{\text{init}} = I_{\mathcal{T}}(A_{\mathcal{T}}'A_{\mathcal{T}})^{-1}A_{\mathcal{T}}'y_t \quad (13)$$

$$\hat{x} = \hat{x}_{\text{init}} + [\arg \min_b \|b\|_1 \text{ s.t. } \|y - A\hat{x}_{\text{init}} - Ab\|_2 \leq \epsilon] \quad (14)$$

This is followed by support estimation and computing a final LS estimate on the estimated support as described in Sec IV-F. The dynamic LS-CS(LS-BP) algorithm is summarized in Algorithm 1.

Notice that the signal residual,  $\beta := x - \hat{x}_{\text{init}}$  satisfies

$$\beta = I_{\mathcal{T}}(A_{\mathcal{T}}'A_{\mathcal{T}})^{-1}A_{\mathcal{T}}'(A_{\Delta}x_{\Delta} + w) + I_{\Delta}x_{\Delta}$$

When  $A$  satisfies the RIP or order at least  $|\mathcal{T}| + |\Delta|$ ,  $\|A_{\mathcal{T}}'A_{\Delta}\|_2 \leq \theta_{|\mathcal{T}|,|\Delta|}$  is small. If the noise  $w$  is also small,

---

#### Algorithm 1 Dynamic LS-CS

---

*Parameters:*  $\epsilon, \alpha_{\text{add}}, \alpha_{\text{del}}$

*BP-noisy.* At  $t = 0$ , compute  $\hat{x}_0$  as the solution of  $\min_b \|b\|_1$  s.t.  $\|y - Ab\|_2 \leq \epsilon$  and compute its support by thresholding:  $\hat{\mathcal{N}}_0 = \{i : |(\hat{x}_0)_i| > \alpha\}$ .

For  $t > 0$  do

- 1) Set  $\mathcal{T} = \hat{\mathcal{N}}_{t-1}$
- 2)  $\ell_1$ -min with partial support knowledge. Solve (14) with  $\hat{x}_{\text{init}}$  given by (13).
- 3) Support Estimation via Add-LS-Del.

$$\begin{aligned} T_{\text{add}} &= \mathcal{T} \cup \{i \in \mathcal{T}^c : |(\hat{x}_t)_i| > \alpha_{\text{add}}\} \\ (\hat{x}_{\text{add}})_{\mathcal{T}_{\text{add}}} &= A_{\mathcal{T}_{\text{add}}}^{\dagger} y_t, \quad (\hat{x}_{\text{add}})_{\mathcal{T}_{\text{add}}^c} = 0 \\ \hat{\mathcal{N}}_t &= T_{\text{add}} \setminus \{i \in T : |(\hat{x}_{\text{add}})_i| \leq \alpha_{\text{del}}\} \end{aligned} \quad (15)$$

- 4) Final LS Estimate.

$$\hat{x}_{t,\text{final}} = I_{\hat{\mathcal{N}}_t} A_{\hat{\mathcal{N}}_t}^{\dagger} y_t \quad (16)$$


---

clearly  $\beta_{\mathcal{T}} := I_{\mathcal{T}}'\beta$  will be small and hence  $\beta$  will be approximately supported on  $\Delta$ . Under slow support change,  $|\Delta| \ll |\mathcal{N}|$  and this is why one expects LS-CS to have smaller reconstruction error than BP-noisy when fewer measurements are available [32].

However, notice that the support size of  $\beta$  is  $|\mathcal{T}| + |\Delta| \geq |\mathcal{N}|$ . Since the number of measurements required for exact recovery is governed by the exact support size, LS-CS is not able to achieve exact recovery using fewer noiseless measurements than those needed by BP-noisy.

#### C. Modified-CS(modified-BP)

The search for a solution that achieves exact reconstruction using fewer measurements led to the modified-CS idea [41], [37]. To understand the approach, suppose first that  $\Delta_e$  is empty, i.e.  $\mathcal{N} = \mathcal{T} \cup \Delta$ . Then the sparse recovery problem becomes one of trying to find the vector  $b$  that is sparsest outside the set  $\mathcal{T}$  among all vectors that satisfy the data constraint. In the noise-free case, this can be written as

$$\min_b \|b_{\mathcal{T}^c}\|_0 \text{ s.t. } y = Ab$$

The above is referred to as the *modified- $\ell_0$*  problem [41], [37]. The above also works if  $\Delta_e$  is not empty. It is easy to show that it can exactly recover  $x$  in the noise-free case if every set of  $|\mathcal{T}| + 2|\Delta| = s + 2u = s + u + e = |\mathcal{N}| + |\Delta_e| + |\Delta|$  columns of  $A$  are linearly independent [37, Proposition 1]. In comparison, the original  $\ell_0$  program, (1), requires every set of  $2s$  columns of  $A$  to be linearly independent [19]. This is a much stronger requirement when  $u \approx e \ll s$ .

Like simple  $\ell_0$ , the modified- $\ell_0$  program also has exponential complexity, and hence we can again replace it by the  $\ell_1$  program, i.e. solve

$$\arg \min_b \|b_{\mathcal{T}^c}\|_1 \text{ s.t. } y = Ab \quad (17)$$

The above was referred to as *modified-CS* in [41], [37] where it was first introduced. However, to keep a uniform nomenclature, it should be called *modified-BP*. In the rest of

this paper, we call it *modified-CS(modified-BP)*. Once again, the above program works, and can provably achieve exact recovery, even when  $\Delta_e$  is not empty. In early 2010, we learnt about an idea similar to modified-CS(modified-BP) that was briefly mentioned in the work of von Borries et al [43], [44].

It has been shown [41], [37] that if  $\delta_{|\mathcal{T}|+2|\Delta_e|} \leq 1/5$  then modified-CS(modified-BP) achieves exact recovery in the noise-free case. We summarize this and other results for exact recovery in Sec VI. For noisy measurements, one can relax the data constraint as

$$\min_b \|b_{\mathcal{T}^c}\|_1 \text{ s.t. } \|y - Ab\|_2 \leq \epsilon \quad (18)$$

This is referred to as *modified-CS-noisy(modified-BP-noisy)*. Alternatively, as in BPDN, one can add the data term as a soft constraint to get an unconstrained problem (which is less expensive to solve):

$$\min_b \gamma \|b_{\mathcal{T}^c}\|_1 + 0.5 \|y - Ab\|_2^2 \quad (19)$$

We refer to this as *modified-BPDN* [45], [38]. The complete stepwise algorithm for modified-CS-noisy(modified-BP-noisy) is given in Algorithm 2, while that for mod-BPDN is given in Algorithm 5.

1) *Reinterpreting Modified-CS(modified-BP)* [46]: The following interesting interpretation of modified-CS(modified-BP) is given by Bandeira et al [46]. Assume that  $A_{\mathcal{T}}$  is full rank (this is a necessary condition in any case). Let  $\mathcal{P}_{\mathcal{T},\perp}$  denote a projection onto the space perpendicular to  $A_{\mathcal{T}}$ , i.e. let

$$\mathcal{P}_{\mathcal{T},\perp} := (I - A_{\mathcal{T}}(A_{\mathcal{T}}'A_{\mathcal{T}})^{-1}A_{\mathcal{T}}').$$

Let  $\tilde{y} := \mathcal{P}_{\mathcal{T},\perp}y$ ,  $\tilde{A} := \mathcal{P}_{\mathcal{T},\perp}A_{\mathcal{T}^c}$  and  $\tilde{x} := x_{\mathcal{T}^c}$ . Then Modified-CS(modified-BP) can be interpreted as finding a  $|\Delta|$ -sparse vector  $\tilde{x} := x_{\mathcal{T}^c}$  of length  $m - |\mathcal{T}|$  from  $\tilde{y} := \tilde{A}\tilde{x}$ . One can then recover  $x_{\mathcal{T}}$  as the (unique) solution of  $A_{\mathcal{T}}x_{\mathcal{T}} = y - A_{\mathcal{T}^c}x_{\mathcal{T}^c}$ . More precisely let  $\hat{x}_{\text{modcs}}$  denote the solution of modified-CS(modified-BP), i.e. (17). Then,

$$\begin{aligned} (\hat{x}_{\text{modcs}})_{\mathcal{T}^c} &= \arg \min_b \|b\|_1 \text{ s.t. } (\mathcal{P}_{\mathcal{T},\perp}y) = (\mathcal{P}_{\mathcal{T},\perp}A_{\mathcal{T}^c})b, \\ (\hat{x}_{\text{modcs}})_{\mathcal{T}} &= (A_{\mathcal{T}})^\dagger (y - A_{\mathcal{T}^c}(\hat{x}_{\text{modcs}})_{\mathcal{T}^c}) \end{aligned}$$

This interpretation can then be used to define a partial NSP or a partial RIC, e.g., the partial RIC is defined as follows.

*Definition 4.1:* We refer to  $\delta_u^k$  as the *partial RIC* for a matrix  $A$  if, for any  $\mathcal{T}$  with  $|\mathcal{T}| \leq k$ ,  $A_{\mathcal{T}}$  is full column rank and  $\delta_u^k$  is the order  $u$  RIC of the matrix  $(\mathcal{P}_{\mathcal{T},\perp}A_{\mathcal{T}^c})$ .

With this, any of the results for BP or BP-noisy can be directly applied to get a result for modified-CS(modified-BP). While the above is a very nice interpretation of modified-CS(modified-BP), the exact recovery conditions obtained this way are not too different from those obtained directly.

2) *Truncated basis pursuit:* The modified-CS(modified-BP) program has been used in the parallel work of Wang and Yin [47] for a different purpose. They call it *truncated basis pursuit* and use it iteratively to improve the recovery error for regular sparse recovery. In the zeroth iteration, they solve the BP program and estimate the estimated signal's support by thresholding. This is then used to solve modified-CS(modified-BP) in the second iteration and the process is repeated with a specific support threshold setting scheme.

#### D. Weighted- $\ell_1$

The weighted- $\ell_1$  program studied in the work of Khajehnejad et al [48], [49] (that appeared in parallel with modified-CS [41], [37]) and the later work of Friedlander et al [50] can be interpreted as a generalization of the modified-CS(modified-BP) idea. Their idea is to partition  $\{1, 2, \dots, m\}$  into sets  $\mathcal{T}_1, \mathcal{T}_2, \dots, \mathcal{T}_q$  and to assume that the percentage of nonzero entries in each set is known. They then use this knowledge to weight the  $\ell_1$  norm along each of these sets differently. The performance guarantees are obtained for the two set partition case, i.e.  $q = 2$ . Using our notation, the two set partition can be labeled  $\mathcal{T}, \mathcal{T}^c$ . In this case, weighted- $\ell_1$  solves

$$\min_b \|b_{\mathcal{T}^c}\|_1 + \tau \|b_{\mathcal{T}}\|_1 \text{ s.t. } y = Ab \quad (20)$$

In situations where it is known that the support of the true signal  $x$  contains the known part  $\mathcal{T}$ , modified-CS(modified-BP), i.e.  $\tau = 0$  in the above, is the best thing to solve. However, in general, the known part  $\mathcal{T}$  also contains some extra entries,  $\Delta_e$ . As long as  $|\Delta_e|$  is small, modified-CS(modified-BP) still yields significant advantage over BP and cannot be improved much further by weighted- $\ell_1$ . Weighted- $\ell_1$  is most useful when the set  $\Delta_e$  is large. We discuss the exact recovery conditions for weighted- $\ell_1$  in Sec VI.

In the noisy case, one either solves *weighted- $\ell_1$ -noisy*:

$$\min_b \|b_{\mathcal{T}^c}\|_1 + \tau \|b_{\mathcal{T}}\|_1 \text{ s.t. } \|y - Ab\|_2 \leq \epsilon \quad (21)$$

or the unconstrained version, *weighted- $\ell_1$ -BPDN*:

$$\min_b \gamma \|b_{\mathcal{T}^c}\|_1 + \gamma\tau \|b_{\mathcal{T}}\|_1 + 0.5 \|y - Ab\|_2^2 \quad (22)$$

The complete stepwise algorithm for weighted- $\ell_1$ -noisy is given in Algorithm 3, while that for weighted- $\ell_1$ -BPDN is given in Algorithm 6.

#### E. Modified Greedy Algorithms and Modified-IHT

The modified-CS(modified-BP) idea can also be used to modify other approaches for sparse recovery. This has been done in recent work by Stankovic et al and Carillo et al [51], [52] with encouraging results. They have developed and evaluated OMP with partially known support (OMP-PKS) [51], Compressive Sampling Matching Pursuit (CoSaMP)-PKS and IHT-PKS [52]. The greedy algorithms (OMP and CoSaMP) are modified as follows. Instead of starting with an initial empty support set, one starts with  $\mathcal{T}$  as being the initial support set. For OMP this can be a problem unless  $\mathcal{T} \subseteq \mathcal{N}$  and hence OMP only solves this special case. For CoSaMP, this is not a problem because there is a step where support entries are deleted too.

IHT is modified as follows. Let  $k = |\mathcal{T}|$ . Then we iterate as

$$\begin{aligned} \hat{x}^0 &= 0 \\ \hat{x}^{k+1} &= (\hat{x}^k)_{\mathcal{T}} + H_{s-k}((\hat{x}^k + A'(y - A\hat{x}^k))_{\mathcal{T}^c}) \end{aligned}$$

The above is referred to as IHT-PKS. The authors also bound its error by modifying the result for IHT.

### F. Support Estimation: thresholding and add-LS-del

In order to use any of the approaches described above for recursive recovery, we need to use the support estimate from the previous time as the set  $\mathcal{T}$ . Thus, we need to estimate the support of the sparse vector at each time. The simplest way to do this is by thresholding, i.e. we compute

$$\hat{\mathcal{N}} = \{i : |(\hat{x})_i| > \alpha\}$$

where  $\alpha \geq 0$  is the zeroing threshold. In case of exact reconstruction, i.e. if  $\hat{x} = x$ , we can use  $\alpha = 0$ . In other situations, we need a nonzero value. In case of very accurate reconstruction, we can set  $\alpha$  to be a little smaller than the magnitude of the smallest nonzero element of  $x$  (assuming its rough estimate is available) [37]. This will ensure close to zero misses and few false additions. In general,  $\alpha$  should depend on both the noise level and the magnitude of the smallest nonzero element of  $x$ .

For compressible signals, one should do the above but with “support” replaced by the  $b\%$ -energy support. For a given number of measurements,  $b$  can be chosen to be the largest value so that all elements of the  $b\%$ -energy support can be exactly reconstructed [37].

In all of LS-CS(LS-BP), modified-CS(modified-BP) and weighted- $\ell_1$ , it can be argued that the estimate  $\hat{x}$  is a biased estimate of  $x$ : it is biased towards zero along  $\Delta$  and away from zero along  $\mathcal{T}$  [32], [53] and as a result the threshold  $\alpha$  is either too low and does not delete all extras (subset of  $\mathcal{T}$ ) or is too high and does not detect all misses. A partial solution to this issue is provided by the Add-LS-Del approach:

$$\mathcal{T}_{\text{add}} = \mathcal{T} \cup \{i : |(\hat{x})_i| > \alpha_{\text{add}}\} \quad (23)$$

$$\hat{x}_{\text{add}} = I_{\mathcal{T}_{\text{add}}} A_{\mathcal{T}_{\text{add}}}^\dagger y \quad (24)$$

$$\hat{\mathcal{N}} = \mathcal{T}_{\text{add}} \setminus \{i : |(\hat{x}_{\text{add}})_i| \leq \alpha_{\text{del}}\} \quad (25)$$

The addition step threshold,  $\alpha_{\text{add}}$ , needs to be just large enough to ensure that the matrix used for LS estimation,  $A_{\mathcal{T}_{\text{add}}}$  is well-conditioned. If  $\alpha_{\text{add}}$  is chosen properly and if the number of measurements,  $n$ , is large enough, the LS estimate on  $\mathcal{T}_{\text{add}}$  will have smaller error, and will be less biased, than  $\hat{x}$ . As a result, deletion will be more accurate when done using this estimate. This also means that one can use a larger deletion threshold,  $\alpha_{\text{del}}$ , which will ensure deletion of more extras. A similar issue for noisy CS, and a possible solution (Gauss-Dantzig selector), was first discussed in [54].

Support estimation is usually followed by LS estimation on the final support estimate, in order to get a solution with reduced bias (Gauss-Dantzig selector idea) [54].

### G. Dynamic modified-CS(modified-BP), weighted- $\ell_1$ , LS-CS

For recursive recovery, the simplest way to use the above algorithms is to use them with  $\mathcal{T} = \hat{\mathcal{N}}_{t-1}$  where  $\hat{\mathcal{N}}_{t-1}$  is the estimated support of  $\hat{x}_{t-1}$ . We summarize the complete dynamic LS-CS algorithm in Algorithm 1, the dynamic modified-CS(modified-BP) algorithm in Algorithm 2 and the dynamic weighted  $\ell_1$  algorithm in Algorithm 3. All of these are stated for the noisy case problem. By setting  $\epsilon = 0$  and the support threshold  $\alpha = 0$ , we get the corresponding algorithm for the noise-free case.

---

### Algorithm 2 Dynamic Modified-CS(modified-BP)-noisy

---

*Parameters:*  $\epsilon, \alpha$

*BP-noisy.* At  $t = 0$ , compute  $\hat{x}_0$  as the solution of  $\min_b \|b\|_1$  s.t.  $\|y - Ab\|_2 \leq \epsilon$  and compute its support by thresholding:  $\hat{\mathcal{N}}_0 = \{i : |(\hat{x}_0)_i| > \alpha\}$ .

For  $t > 0$  do

- 1) Set  $\mathcal{T} = \hat{\mathcal{N}}_{t-1}$
- 2) *Modified-CS(modified-BP)-noisy.* Compute  $\hat{x}_t$  as the solution of

$$\min_b \|b_{\mathcal{T}^c}\|_1 \text{ s.t. } \|y - Ab\|_2 \leq \epsilon$$

- 3) *Support Estimation - Simple Thresholding.*

$$\hat{\mathcal{N}}_t = \{i : |(\hat{x}_t)_i| > \alpha\} \quad (26)$$

*Parameter setting in practice:* Set  $\alpha = 0.25x_{\min}$  (or some appropriate fraction) where  $x_{\min}$  is an estimate of the smallest nonzero entry of  $x_t$ . One can get this estimate from training data or one can use the minimum nonzero entry of  $\hat{x}_{t-1}$  to set  $\alpha$  at time  $t$ . Set  $\epsilon$  using a short initial noise-only training sequence or approximate it by  $\|y_{t-1} - A_{t-1}\hat{x}_{t-1}\|_2$ . Also see Section VIII.

Note: For compressible signal sequences, the above algorithm is the best. For exactly sparse signal sequences, it is better to replace the support estimation step by the Add-LS-Del procedure and the final LS step from Algorithm 1. See discussion in Section VIII on setting its parameters.

---



---

### Algorithm 3 Dynamic Weighted- $\ell_1$ -noisy

---

*Parameters:*  $\tau, \epsilon, \alpha$

*BP-noisy.* At  $t = 0$ , compute  $\hat{x}_0$  as the solution of  $\min_b \|b\|_1$  s.t.  $\|y - Ab\|_2 \leq \epsilon$  and compute its support by thresholding:  $\hat{\mathcal{N}}_0 = \{i : |(\hat{x}_0)_i| > \alpha\}$ .

For  $t > 0$  do

- 1) Set  $\mathcal{T} = \hat{\mathcal{N}}_{t-1}$
- 2) *Weighted  $\ell_1$ .* Compute  $\hat{x}_t$  as the solution of

$$\min_b \|b_{\mathcal{T}^c}\|_1 + \tau \|b_{\mathcal{T}}\|_1 \text{ s.t. } \|y - Ab\|_2 \leq \epsilon$$

- 3) *Support Estimation - Simple Thresholding.*

$$\hat{\mathcal{N}}_t = \{i : |(\hat{x}_t)_i| > \alpha\}$$

*Parameter setting in practice:* Set  $\alpha$  and  $\epsilon$  as in Algorithm 2. Set  $\tau$  as the ratio of the fraction of the number of extras to the size of  $\mathcal{T}$  (estimate this number from training data or from the previous two estimates of  $x_t$ ). Also see Section VIII.

---

Recent work [8] has introduced solutions for the more general case where the support change may not be slow, but is still highly correlated over time. Assume that the form of the correlation model is known and linear. Then one can obtain the support prediction,  $\mathcal{T}$ , by “applying the correlation model” to  $\hat{\mathcal{N}}_{t-1}$ . For example, in case of video, if the sparse foreground image consists of a single moving object which moves according to a constant velocity model, one can obtain  $\mathcal{T}$  by “moving”  $\hat{\mathcal{N}}_{t-1}$  by an amount equal to the estimate of the object’s predicted velocity at the current time. Using

this in dynamic modified-CS(modified-BP) (Algorithm 2), one gets  $\hat{\mathcal{N}}_t$ . The centroid of the indices in  $\hat{\mathcal{N}}_t$  can serve as an ‘‘observation’’ of the object’s current location and this can be fed into a simple Kalman filter, or any adaptive filter, to track the object’s location and velocity over time.

## V. EXPLOITING SLOW SUPPORT AND SLOW SIGNAL VALUE CHANGE

So far we talked about the problem in which only reliable support knowledge is available. In many applications, reliable partial signal value knowledge is also available. In many recursive recovery problems, often the signal values change very slowly over time and in these cases, using the signal value knowledge should significantly improve recovery performance. We state the reformulated static problem in Section V-A. Next we describe the regularized modified-CS(modified-BP) and the modified-CS-residual (modified-BP-residual) solutions in Section V-B. After this we explain other algorithms that were directly designed for the problem of recursive recovery of sparse signal sequences, when both slow support change and slow signal value change are used.

### A. Reformulated problem: Sparse recovery with partial support and signal value knowledge

The goal is to recover a sparse vector  $x$ , with support set  $\mathcal{N}$ , either from noise-free undersampled measurements,  $y := Ax$ , or from noisy measurements,  $y := Ax + w$ , when partial erroneous support knowledge,  $\mathcal{T}$ , is available and partial erroneous signal value knowledge on  $\mathcal{T}$ ,  $\hat{\mu}_{\mathcal{T}}$ , is available. The true support  $\mathcal{N}$  can be written as

$$\mathcal{N} = \mathcal{T} \cup \Delta \setminus \Delta_e \quad \text{where} \quad \Delta := \mathcal{N} \setminus \mathcal{T}, \quad \Delta_e := \mathcal{T} \setminus \mathcal{N}$$

and the true signal  $x$  can be written as

$$\begin{aligned} (x)_{\mathcal{N} \cup \mathcal{T}} &= (\hat{\mu})_{\mathcal{N} \cup \mathcal{T}} + e \\ (x)_{\mathcal{N}^c} &= 0, \quad (\hat{\mu})_{\mathcal{T}^c} = 0 \end{aligned} \quad (27)$$

The error  $e$  in the prior signal estimate is assumed to be small, i.e.  $\|e\| \ll \|x\|$ .

### B. Regularized modified-CS(modified-BP) and modified-CS-residual(modified-BP-residual)

Regularized modified-CS(modified-BP) adds the slow signal value change constraint to modified-CS(modified-BP) and solves the following [39]:

$$\min_b \|b_{\mathcal{T}^c}\|_1 \quad \text{s.t.} \quad \|y - Ab\|_2 \leq \epsilon, \quad \text{and} \quad \|b_{\mathcal{T}} - \hat{\mu}_{\mathcal{T}}\|_{\infty} \leq \rho \quad (28)$$

We obtained exact recovery conditions for the above in [39]. The slow signal value change can be imposed either as a bound on the max norm (as above) or as a bound on the 2-norm. In practice, the following Lagrangian version (constraints added as weighted costs to get an unconstrained problem) is fastest and most commonly used:

$$\min_b \gamma \|b_{\mathcal{T}^c}\|_1 + 0.5 \|y - Ab\|_2^2 + 0.5 \lambda \|b_{\mathcal{T}} - \hat{\mu}_{\mathcal{T}}\|_2^2 \quad (29)$$

We refer to the above as *regularized-modified-BPDN*. This was analyzed in detail in [38] where we obtained computable bounds on its recovery error.

A second approach to using slow signal value knowledge is to use an approach similar to CS-residual(BP-residual), but with BP-noisy replaced by modified-CS-noisy(modified-BP-noisy). Once again the following unconstrained version is most useful:

$$\hat{x} = \hat{\mu} + [\arg \min_b \gamma \|b_{\mathcal{T}^c}\|_1 + 0.5 \|y - A\hat{\mu} - Ab\|_2^2] \quad (30)$$

We refer to the above as *modified-CS-residual(modified-BP-residual)* [55].

For recursive reconstruction, one uses  $\mathcal{T} = \hat{\mathcal{N}}_{t-1}$ . For  $\hat{\mu}$ , one can either use  $\hat{\mu} = \hat{x}_{t-1}$ , or, in certain applications, e.g., functional MRI reconstruction [27], where the signal values do not change much w.r.t. the first frame, using  $\hat{\mu} = \hat{x}_0$  is a better idea. Alternatively, as we explain next, especially in situations where the support does not change at each time, but only every so often, one could obtain  $\hat{\mu}$  by a Kalman filter on the sub-vector of  $x_t$  that is supported on  $\mathcal{T} = \hat{\mathcal{N}}_{t-1}$ .

### C. Kalman filtered CS (KF-CS) and Kalman filtered Modified-CS(modified-BP) or KMoCS

Kalman Filtered CS-residual (KF-CS) was introduced in the context of recursive reconstruction in [36] and in fact this was the first work that studied the recursive recovery problem. With the modified-CS approach and results now known, a much better idea than KF-CS is *Kalman Filtered Modified-CS-residual (KMoCS)*. This can be understood as modified-CS-residual(modified-BP-residual) but with  $\hat{\mu}$  obtained as a Kalman filtered estimate on the previous support  $\mathcal{T} = \hat{\mathcal{N}}_{t-1}$ . For the KF step, one needs to assume a model on signal value change. In the absence of specific information, a Gaussian random walk model with equal change variance in all directions is the best option [36]:

$$\begin{aligned} (x_0)_{\mathcal{N}_0} &\sim \mathcal{N}(0, \sigma_{sys,0}^2), \\ (x_t)_{\mathcal{N}_t} &= (x_{t-1})_{\mathcal{N}_t} + \nu_t, \quad \nu_t \sim \mathcal{N}(0, \sigma_{sys}^2 I) \\ (x_t)_{\mathcal{N}_t^c} &= 0 \end{aligned} \quad (31)$$

Here  $\mathcal{N}(a, \Sigma)$  denotes a Gaussian distribution with mean  $a$  and covariance matrix  $\Sigma$ . Assume for a moment that the support does not change with time, i.e.  $\mathcal{N}_t = \mathcal{N}_0$ , and  $\mathcal{N}_0$  is known or perfectly estimated using BP-noisy followed by support estimation. With the above model on  $x_t$  and the observation model given in (8), if the observation noise  $w_t$  is Gaussian, then the Kalman filter provides the causal minimum mean squared error (MMSE) solution, i.e. it returns  $\hat{x}_{t|t}$  which solves

$$\arg \min_{\hat{x}_{t|t}: (\hat{x}_{t|t})_{\mathcal{N}_0^c} = 0} \mathbb{E}_{x_t|y_1, y_2, \dots, y_t} [\|x_t - \hat{x}_{t|t}(y_1, y_2, \dots, y_t)\|_2^2]$$

(notice that the above is solving for  $\hat{x}_{t|t}$  supported on  $\mathcal{N}_0$ ). Here  $\mathbb{E}_{x_t|y}[q]$  denotes the expected value of  $q$  conditioned on  $y$ . However, our problem is significantly more difficult because the support set  $\mathcal{N}_t$  changes with time and is unknown. To solve this problem, KMoCS is a practical heuristic that combines the modified-CS-residual(modified-BP-residual) idea for tracking

the support with an adaption of the regular KF algorithm to the case where the set of entries of  $x_t$  that form the state vector for the KF change with time: the KF state vector at time  $t$  is  $(x_t)_{\mathcal{N}_t}$  at time  $t$ . Unlike the regular KF for a fixed dimensional linear Gaussian state space model, KF-CS or KMoCS do not enjoy any optimality properties. However, one expects KMoCS to outperform modified-CS(modified-BP) when accurate prior knowledge of the signal values is available and we see this in simulations. We summarize KMoCS in Algorithm 7.

An open question is how to analyze KF-CS or KMoCS and get a meaningful performance guarantee? This is a hard problem because the KF state vector is  $(x_t)_{\mathcal{N}_t}$  at time  $t$  and  $\mathcal{N}_t$  changes with time. In fact, even if we suppose that the sets  $\mathcal{N}_t$  are given, there are no known results for the resulting “genie-aided KF”.

#### D. Pseudo-measurement based CS-KF (PM-CS-KF)

In work that appeared soon after the KF-CS paper [36], Carmi, Gurfil and Kanevski [56] introduced the pseudo-measurement based CS-KF (PM-CS-KF) algorithm. It uses an indirect method called the pseudo-measurement (PM) technique [57] to include the sparsity constraint while trying to minimize the estimation error in the KF update step. To be precise, it uses PM to approximately solve the following

$$\min_{\hat{x}_{k|k}} \mathbb{E}_{x_k|y_1, y_2, \dots, y_k} [\|x_k - \hat{x}_{k|k}\|_2^2] \text{ s.t. } \|\hat{x}_{k|k}\|_1 \leq \epsilon$$

The idea of PM is to replace the constraint  $\|\hat{x}_{k|k}\|_1 \leq \epsilon$  by a linear equation of the form

$$\tilde{H}x_k - \epsilon = 0$$

where  $\tilde{H} = \text{diag}(\text{sgn}((x_k)_1), \text{sgn}((x_k)_2), \dots, \text{sgn}((x_k)_m))$  and  $\epsilon$  serves as measurement noise. It then uses an extended KF approach iterated multiple times to enforce the sparsity constraint. The covariance of the pseudo noise  $\epsilon$ ,  $R_\epsilon$  is a tuning parameter.

The complete algorithm is summarized in [56, Algorithm 1]. We do not copy this here because it is not clear if the journal allows copying an algorithm from another author’s work. Its code is available at <http://www2.ece.ohio-state.edu/~schniter/DCS/index.html>.

#### E. Dynamic CS via approximate message passing (DCS-AMP)

Another approximate Bayesian approach was developed in very interesting recent work by Ziniel and Schniter [58], [59]. They introduced the dynamic CS via approximate message passing (DCS-AMP) algorithm by developing the recently introduced AMP approach of Donoho et al [60] for the dynamic CS problem. The authors model the dynamics of the sparse vectors over time using a stationary Bernoulli Gaussian prior as follows: for all  $i = 1, 2, \dots, m$ ,

$$(x_t)_i = (s_t)_i(\theta_t)_i$$

where  $(s_t)_i$  is a binary random variable that forms a stationary Markov chain over time and  $(\theta_t)_i$  follows a stationary first order autoregressive model with nonzero mean. Independence is assumed across the various indices  $i$ . Suppose, at any time

$t$ ,  $\Pr((s_t)_i = 1) = \lambda$  and  $\Pr((s_t)_i = 1 | (s_{t-1})_i = 0) = p_{10}$ . Using stationarity, this tells us that  $\Pr((s_t)_i = 0 | (s_{t-1})_i = 1) := p_{01} = \frac{\lambda p_{10}}{1 - \lambda}$ . Also,

$$(\theta_t)_i = (1 - \alpha)((\theta_{t-1})_i - \zeta) + \alpha(v_t)_i$$

with  $0 \leq \alpha \leq 1$ . The choice of  $\alpha$  controls how slowly or quickly the signal values change over time. The choice of  $p_{10}$  controls the likelihood of new support addition(s).

Exact computation of the minimum mean squared error (MMSE) estimate of  $x_t$  cannot be done under the above model. On one end, one can try to use sequential Monte Carlo techniques (particle filtering) to approximate the MMSE estimate as in [61], [62]. But these can get computationally expensive for high dimensional problems and it is never clear what number of particles is sufficient to get an accurate enough estimate. The AMP approach developed in [59] is also approximate but is extremely fast and hence is useful.

The complete DCS-AMP algorithm is available in [59, Table II]. We do not copy this here because it is not clear if the journal allows copying an algorithm from another author’s work. Its code is available at <http://www2.ece.ohio-state.edu/~schniter/DCS/index.html>.

#### F. Hierarchical Bayesian KF

In recent work by Dai et al [63], the Hierarchical Bayesian KF (hKF) algorithm was introduced that developed a recursive KF-based algorithm to solve the dynamic CS problem in the sparse Bayesian learning (SBL) [31], [17] framework. In it, the authors assume a random walk state transition model on the  $x_t$ ’s similar to (31) but with the difference that the vector  $\nu_t$  is zero mean independent Gaussian with variance  $\gamma_i$  along index  $i$ . Then they used the SBL approach (developed an EM algorithm) to obtain a type-II maximum likelihood estimate of the hyper-parameters  $\gamma_i$ ’s from the observations. The final estimate of  $x_t$  was computed as the KF estimate by using the estimated hyper-parameters (simple closed form estimate because of the joint-Gaussian assumption). It is not very clear how support change is handled in this algorithm. Their code is not available online; a draft version of the code provided by the authors did not work in many of our experiments and hence we do not report results using it.

#### G. Dynamic Homotopy for Streaming signals

In recent work [64], Asif and Romberg have designed a fast homotopy to solve a large class of dynamic CS problems. In particular they design fast homotopies to solve the following two problems:

$$\min_x \|W_t x\|_1 + 0.5 \|A_t x - y_t\|_2^2$$

and

$$\min_x \|W_t x\|_1 + 0.5 \|A_t x - y_t\|_2^2 + 0.5 \|F_t x_{t-1} - x\|_2^2$$

for arbitrary matrices  $W_t$  and  $F_t$ . They show experiments with letting  $W_t$  be a reweighting matrix to solve the reweighted- $\ell_1$  problem. Their code is available at <http://users.ece.gatech.edu/~sasif/homotopy>.

## VI. THEORETICAL RESULTS: EXACT RECOVERY

In this section, we summarize the exact recovery conditions for modified-CS(modified-BP) and weighted- $\ell_1$ . We first give two RIP based results in Section VI-A below. Next, in Section VI-B, we give results that compute the “weak thresholds” on the number of measurements needed for high probability exact recovery similar to those obtained by Donoho [65] for BP.

### A. RIP based results for modified-CS(modified-BP) and weighted- $\ell_1$

Recall that, in the noise-free case, the problem is to recover a sparse vector,  $x$ , with support,  $\mathcal{N} = \mathcal{T} \cup \Delta \setminus \Delta_e$ , where  $\Delta = \mathcal{N} \setminus \mathcal{T}$  and  $\Delta_e = \mathcal{T} \setminus \mathcal{N}$ , from  $y := Ax$  using partial support knowledge  $\mathcal{T}$ . Let

$$s := |\mathcal{N}|, \quad k := |\mathcal{T}|, \quad u := |\Delta|, \quad e := |\Delta_e|$$

Clearly,

$$s = k + u - e$$

The first result for modified-CS(modified-BP) proved in [41], [37] is as follows.

*Theorem 6.1 (RIP-based Modified-CS(modified-BP) exact recovery):* [37, Theorem 1] Consider recovering  $x$  with support  $\mathcal{N}$  from  $y := Ax$  by solving Modified-CS(modified-BP), i.e. (17).  $x$  is the unique minimizer of (17) if

- 1)  $\delta_{k+u} < 1$  and  $\delta_{2u} + \delta_k + \theta_{k,2u}^2 < 1$  and
- 2)  $a_k(2u, u) + a_k(u, u) < 1$  where  $a_k(i, \tilde{i}) := \frac{\theta_{i,i} + \frac{\theta_{i,k} \theta_{i,k}}{1 - \delta_k}}{1 - \delta_i - \frac{\theta_{i,k}^2}{1 - \delta_k}}$

Both the above conditions hold if

$$2\delta_{2u} + \delta_{3u} + \delta_k + \delta_{k+u}^2 + 2\delta_{k+2u}^2 < 1.$$

This, in turn holds if

$$\delta_{k+2u} \leq 0.2$$

(recall that  $k = |\mathcal{T}|$  and  $u = |\Delta|$ ). The conditions can also be rewritten in terms of  $s, e, u$  by substituting  $k = s + e - u$ .

Compare this result with that for BP which requires [33], [66], [54]

$$\delta_{2s} < \sqrt{2} - 1 \quad \text{or} \quad \delta_{2s} + \delta_{3s} < 1.$$

To compare the conditions numerically, we can use  $u = e = 0.02s$  which is typical for time series applications (see Fig. 2). Using  $\delta_{cr} \leq c\delta_{2r}$  [67, Corollary 3.4], it can be show that modified-CS(modified-BP) only requires  $\delta_{2u} < 0.004$ . On the other hand, BP requires  $\delta_{2u} < 0.008$  which is clearly stronger.

Later work by Friedlander et al [50] proved an improved result for weighted- $\ell_1$ , and hence also for modified-CS(modified-BP) which is a special case of weighted- $\ell_1$ .

*Theorem 6.2 (RIP-based weighted- $\ell_1$  exact recovery [50]):* Consider recovering  $x$  with support  $\mathcal{N}$  from  $y := Ax$  by solving weighted- $\ell_1$ , i.e. (20). Let  $\alpha = \frac{|\mathcal{T} \cap \mathcal{N}|}{|\mathcal{T}|} = \frac{(s-u)}{(s+e-u)}$  and  $\rho = \frac{|\mathcal{T}|}{|\mathcal{N}|} = \frac{(s+e-u)}{s}$ . Let  $\mathbb{Z}$  denote the set of integers. Pick an  $a \in \frac{1}{s}\mathbb{Z}$  that is such that  $a > \max(1, (1 - \alpha)\rho)$ . Weighted- $\ell_1$  achieves exact recovery if

$$\delta_{as} + \frac{a}{\gamma^2} \delta_{(a+1)s} < \frac{a}{\gamma^2} - 1 \quad \text{for} \quad \gamma = \tau + (1 - \tau)\sqrt{1 + \rho - 2\alpha\rho}$$

Modified-CS(modified-BP) achieves exact recovery if the above holds with  $\tau = 0$ .

Exact recovery conditions for regularized modified-CS(modified-BP) for noise-free measurements, i.e. for (28) with  $\epsilon = 0$  were obtained in [39, Theorem 1]. These are weaker than those for modified-CS(modified-BP) if  $x_i - \hat{\mu}_i = \pm\rho$  for some  $i \in \mathcal{T}$  (some of the constraints  $\|b_{\mathcal{T}} - \hat{\mu}_{\mathcal{T}}\|_{\infty} \leq \rho$  are active for the true signal,  $x$ ) and some elements of this active set satisfy the condition given in [39, Theorem 1]. One set of practical applications where  $x_i - \hat{\mu}_i = \pm\rho$  with nonzero probability is when dealing with quantized signals and their estimates.

### B. Weak thresholds for high probability exact recovery for weighted- $\ell_1$ and modified-CS(modified-BP)

In very interesting work, Khajehnejad et al. [49] obtained “weak thresholds” on the minimum number of measurements,  $n$ , (as a fraction of  $m$ ) that are sufficient for exact recovery with overwhelming probability (the probability of not getting exact recovery decays to zero as the signal length  $m$  increases). The weak threshold was first defined by Donoho in [65]. Khajehnejad et al. [49] proved the following result.

*Theorem 6.3 (weighted- $\ell_1$  weak threshold [49]):* Consider recovering  $x$  with support  $\mathcal{N}$  from  $y := Ax$  by solving weighted- $\ell_1$ , i.e. (20). Let  $\omega := 1/\tau$ ,  $\gamma_1 := \frac{|\mathcal{T}|}{m}$  and  $\gamma_2 := \frac{|\mathcal{T}^c|}{m} = 1 - \gamma_1$ . Also let  $p_1, p_2$  be the sparsity fractions on the sets  $\mathcal{T}$  and  $\mathcal{T}^c$ , i.e. let  $p_1 := \frac{|\mathcal{T}| - |\Delta_e|}{|\mathcal{T}|}$  and  $p_2 := \frac{|\Delta|}{|\mathcal{T}^c|}$ . Then there exists a critical threshold

$$\delta_c = \delta_c(\gamma_1, \gamma_2, p_1, p_2, \omega)$$

such that for all  $\frac{n}{m} > \delta_c$ , the probability that a sparse vector  $x$  is not recovered decays to zero exponentially with  $m$ . The expression for  $\delta_c$  is complicated and is available at the bottom of page 189 of [49]. It is such that it can be numerically calculated.

The above result provides an approach for picking the best  $\tau$  out of a discrete set of possible values. To do this, for each  $\tau$  from the set, one can compute the weak threshold  $\delta_c$  and then pick the  $\tau$  that needs the smallest weak threshold. The weak threshold for this  $\tau$  also specifies the required number of measurements,  $n$ , needed.

When no prior knowledge is available,  $\gamma_1 = 0$ ,  $\gamma_2 = 1$ ,  $p_1 = 0$  and  $p_2 = s/m$ . In this case, if BP, (2), is solved,  $\omega = 1$  and so the required weak threshold is given by  $\delta_c(0, 1, 0, \frac{s}{m}, 1)$ .

The modified-CS(modified-BP) program is a special case of weighted- $\ell_1$  with  $\tau = 0$ . Thus one can get a simple corollary for it.

*Corollary 6.4 (Modified-CS(modified-BP) weak threshold [49]):* Consider recovering  $x$  from  $y := Ax$  by solving modified-CS(modified-BP), i.e. (17). Assume all notation from Theorem 6.3. The weak threshold for modified-CS(modified-BP) is given by  $\delta_c(\gamma_1, \gamma_2, p_1, p_2, \infty)$ . In the special case when  $\mathcal{T} \subseteq \mathcal{N}$ ,  $p_1 = 1$ . In this case, the weak threshold satisfies

$$\delta_c(\gamma_1, \gamma_2, 1, p_2, \infty) = \gamma_1 + \gamma_2 \delta_c(0, 1, 0, p_2, 1).$$

As explained in [49], when  $\mathcal{T} \subseteq \mathcal{N}$ , the above has a nice physical interpretation. In this case, the number of measurements  $n$  needed by modified-CS(modified-BP) is equal to  $|\mathcal{T}|$

plus the number of measurements needed for recovering the remaining  $|\Delta|$  entries from  $\mathcal{T}^c$  using BP.

### C. Recursive reconstruction: error stability over time

In the noise-free case, once an exact recovery result is obtained, its extension to the recursive recovery problem is trivial. For example a simple corollary of Theorem 6.1 is the following.

*Corollary 6.5 (dynamic modified-CS(modified-BP) exact recovery):* Consider recovering  $x_t$  with support  $\mathcal{N}_t$  from  $y_t := A_t x_t$  using dynamic modified-CS(modified-BP), i.e. Algorithm 2 with  $\epsilon = 0$  and  $\alpha = 0$ . Let  $u_t := |\mathcal{N}_t \setminus \mathcal{N}_{t-1}|$  and let  $e_t = |\mathcal{N}_{t-1} \setminus \mathcal{N}_t|$  denote the number of additions to and the number of removals from the support set at time  $t$ . Also let  $s_t = |\mathcal{N}_t|$  denote the support size. If  $\delta_{2s_0}(A_0) \leq 0.2$  and if, for all times  $t > 0$ ,  $\delta_{s_t+u_t+e_t}(A_t) \leq 0.2$ , then  $\hat{x}_t = x_t$  (exact recovery is achieved) at all times  $t$ .

## VII. THEORETICAL RESULTS: NOISY CASE ERROR BOUNDS AND STABILITY OVER TIME

When the measurements are noisy, one can only bound the recovery error. These results can be obtained by adapting of the existing tools used to get error bounds for BP, BP-noisy or BPDN. We summarize these in Sections VII-A and VII-B. The result given in Section VII-B is particularly useful since it is a computable bound and it holds always. As we will explain, it can be used to design a useful heuristic to choose the parameters for the mod-BPDN and reg-mod-BPDN convex programs.

Next, in Section VII-C, we provide results for error stability over time in the recursive recovery problem. In the noise-free case, such a result followed as a direct extension of the exact recovery result at a given time. In the noisy case, obtaining such a result is significantly harder because the recovery error is directly proportional to the size of the misses,  $\Delta_t := \mathcal{N}_t \setminus \hat{\mathcal{N}}_{t-1}$ , and extras,  $\Delta_{e,t} := \hat{\mathcal{N}}_{t-1} \setminus \mathcal{N}_t$ , with respect to the previous support estimate. We need to find sufficient conditions to ensure that these sets' sizes are bounded by a time-invariant value in order to get a stability result.

### A. Error bounds for modified-CS(modified-BP)-noisy and weighted- $\ell_1$

When measurements are noisy, one cannot get exact recovery, but can only bound the reconstruction error. The first such result was proved for LS-CS in [32, Lemma 1].

By adapting the approach of [33], the error of modified-CS(modified-BP) can be bounded as a function of  $|\mathcal{T}| = |\mathcal{N}| + |\Delta_e| - |\Delta|$  and  $|\Delta|$  [68], [69], [53].

*Theorem 7.1 (modified-CS(modified-BP) error bound):* [53, Lemma 2.7] Let  $x$  be a sparse vector with support  $\mathcal{N}$  and let  $y := Ax + w$  with  $\|w\|_2 \leq \epsilon$ . Let  $\hat{x}$  be the solution of modified-CS(modified-BP)-noisy given in (18). If  $\delta_{|\mathcal{T}|+3|\Delta|} = \delta_{|\mathcal{N}|+|\Delta_e|+2|\Delta|} < (\sqrt{2} - 1)/2$ , then

$$\|x - \hat{x}\| \leq C_1(|\mathcal{T}|+3|\Delta|)\epsilon \leq 7.50\epsilon \text{ where } C_1(k) := \frac{4\sqrt{1+\delta_k}}{1-2\delta_k}.$$

A similar result for a compressible (approximately sparse) signal and weighted- $\ell_1$  was proved in [50].

*Theorem 7.2 (weighted- $\ell_1$  error bound):* [50] Consider recovering a compressible vector  $x$  from  $y := Ax + w$  with  $\|w\| \leq \epsilon$ . Let  $\hat{x}$  be the solution of weighted- $\ell_1$ -noisy, (21). Let  $x_s$  denote the best  $s$  term approximation for  $x$  and let  $\mathcal{N} = \text{support}(x_s)$ . Also, let  $\alpha = \frac{|\mathcal{T} \cap \mathcal{N}|}{|\mathcal{T}|} = \frac{(s-u)}{(s+e-u)}$  and  $\rho = \frac{|\mathcal{T}|}{|\mathcal{N}|} = \frac{(s+e-u)}{s}$ . Let  $\mathbb{Z}$  be the set of integers and pick an  $a \in \frac{1}{s}\mathbb{Z}$  that is such that  $a > \max(1, (1-\alpha)\rho)$ . If

$$\delta_{as} + \frac{a}{\gamma^2} \delta_{(a+1)s} < \frac{a}{\gamma^2} - 1 \text{ for } \gamma = \tau + (1-\tau)\sqrt{1+\rho-2\alpha\rho}$$

then

$$\|\hat{x} - x\|_2 \leq C'_0 \epsilon + C'_1 s^{-1/2} (\tau \|x - x_s\|_1 + (1-\tau) \|x_{(\mathcal{N} \cup \mathcal{T})^c}\|_1)$$

where  $C'_0 = C'_0(\tau, s, a, \delta_{(a+1)s}, \delta_{as})$  and  $C'_1 = C'_1(\tau, s, a, \delta_{(a+1)s}, \delta_{as})$  are constants specified in Remark 3.2 of [50].

When  $x$  is  $s$ -sparse, the above simplifies to  $\|\hat{x} - x\|_2 \leq C'_0 \epsilon$ . The result for modified-CS(modified-BP) follows by setting  $\tau = 0$  in the above expressions.

### B. Computable error bounds for reg-mod-BPDN and mod-BPDN

In [70], Tropp introduced the Exact Recovery Coefficient (ERC) and used it to obtain a computable error bound for BPDN that holds under a sufficient condition that is also computable. By modifying this approach, one can get a similar result for reg-mod-BPDN and hence also for mod-BPDN (which is a special case). With some extra work, one can obtain a computable bound that holds always (does not require any sufficient conditions) [38]. A direct corollary of it then gives a similar result for mod-BPDN. We state this result below. Since the bound is computable and holds without sufficient conditions and, from simulations, is also fairly tight [see [38, Fig 4]], it provides a good heuristic for setting  $\gamma$  for mod-BPDN and  $\gamma$  and  $\lambda$  for reg-mod-BPDN.

Let  $I_{\mathcal{T}, \mathcal{T}}$  denote the identity matrix on the row, column indices  $\mathcal{T}, \mathcal{T}$  and let  $\mathbf{0}_{\mathcal{T}, \mathcal{S}}$  be a zero matrix on the row, column indices  $\mathcal{T}, \mathcal{S}$ .

*Theorem 7.3:* Let  $x$  be a sparse vector with support  $\mathcal{N}$  and let  $y := Ax + w$  with  $\|w\| \leq \epsilon$ . Let  $\hat{x}$  be the solution of reg-mod-BPDN, i.e. (29). Assume that the columns of  $A$  have unit 2-norm (when they do not have unit norm, we can use Remark 2.1). If  $\gamma = \gamma_{\mathcal{T}, \lambda}^*(\tilde{\Delta}^*(k_{\min}))$ , then it has a unique minimizer,  $\hat{x}$  which satisfies

$$\|x - \hat{x}\|_2 \leq g(\tilde{\Delta}^*(k_{\min})).$$

Here

$$\gamma_{\mathcal{T}, \lambda}^*(\tilde{\Delta}) := \frac{\text{maxcor}(\tilde{\Delta})}{\text{ERC}_{\mathcal{T}, \lambda}(\tilde{\Delta})} \left[ \lambda f_2(\tilde{\Delta}) \|x_{\mathcal{T}} - \hat{\mu}_{\mathcal{T}}\|_2 + f_3(\tilde{\Delta}) \|w\|_2 + f_4(\tilde{\Delta}) \|x_{\Delta \setminus \tilde{\Delta}}\|_2 \right] + \frac{\|w\|_{\infty}}{\text{ERC}_{\mathcal{T}, \lambda}(\tilde{\Delta})}, \quad (32)$$

$$g(\tilde{\Delta}) := g_1(\tilde{\Delta}) \|x_{\mathcal{T}} - \hat{\mu}_{\mathcal{T}}\|_2 + g_2(\tilde{\Delta}) \|w\|_2 + g_3(\tilde{\Delta}) \|x_{\Delta \setminus \tilde{\Delta}}\|_2 + g_4(\tilde{\Delta}), \text{ and} \quad (33)$$

$$\tilde{\Delta}^*(k) := \arg \min_{\tilde{\Delta} \subseteq \Delta, |\tilde{\Delta}|=k} \|x_{\Delta \setminus \tilde{\Delta}}\|_2 \quad (34)$$

is the subset of  $\Delta$  that contains the  $k$  largest magnitude entries of  $x$  and

$$k_{\min} := \arg \min_k B_k \text{ where}$$

$$B_k := \begin{cases} g(\tilde{\Delta}^*(k)) & \text{if } \text{ERC}_{\mathcal{T},\lambda}(\tilde{\Delta}^*(k)) > 0 \text{ and} \\ & Q_{\mathcal{T},\lambda}(\tilde{\Delta}^*(k)) \text{ is invertible} \\ \infty & \text{otherwise} \end{cases}$$

In the above expressions,

$$\text{maxcor}(\tilde{\Delta}) := \max_{i \notin (T \cup \tilde{\Delta})^c} \|A_i' A_{T \cup \tilde{\Delta}}\|_2,$$

$$Q_{\mathcal{T},\lambda}(S) := A_{T \cup S}' A_{T \cup S} + \lambda \begin{bmatrix} I_{\mathcal{T},\mathcal{T}} & \mathbf{0}_{\mathcal{T},S} \\ \mathbf{0}_{S,\mathcal{T}} & \mathbf{0}_{S,S} \end{bmatrix}$$

$$\text{ERC}_{\mathcal{T},\lambda}(S) := 1 - \max_{\omega \notin T \cup S} \|P_{\mathcal{T},\lambda}(S) A_S' M_{\mathcal{T},\lambda} A_\omega\|_1,$$

$$P_{\mathcal{T},\lambda}(S) := (A_S' M_{\mathcal{T},\lambda} A_S)^{-1}$$

$$M_{\mathcal{T},\lambda} := I - A_{\mathcal{T}} (A_{\mathcal{T}}' A_{\mathcal{T}} + \lambda I_{\mathcal{T},\mathcal{T}})^{-1} A_{\mathcal{T}}'$$

and

$$g_1(\tilde{\Delta}) := \lambda f_2(\tilde{\Delta}) \left( \frac{\sqrt{|\tilde{\Delta}|} f_1(\tilde{\Delta}) \text{maxcor}(\tilde{\Delta})}{\text{ERC}_{\mathcal{T},\lambda}(\tilde{\Delta})} + 1 \right),$$

$$g_2(\tilde{\Delta}) := \frac{\sqrt{|\tilde{\Delta}|} f_1(\tilde{\Delta}) f_3(\tilde{\Delta}) \text{maxcor}(\tilde{\Delta})}{\text{ERC}_{\mathcal{T},\lambda}(\tilde{\Delta})} + f_3(\tilde{\Delta}),$$

$$g_3(\tilde{\Delta}) := \frac{\sqrt{|\tilde{\Delta}|} f_1(\tilde{\Delta}) f_4(\tilde{\Delta}) \text{maxcor}(\tilde{\Delta})}{\text{ERC}_{\mathcal{T},\lambda}(\tilde{\Delta})} + f_4(\tilde{\Delta}),$$

$$g_4(\tilde{\Delta}) := \frac{\sqrt{|\tilde{\Delta}|} \|A_{(T \cup \tilde{\Delta})^c}\|_\infty \|w\|_\infty f_1(\tilde{\Delta})}{\text{ERC}_{\mathcal{T},\lambda}(\tilde{\Delta})}$$

$$f_1(\tilde{\Delta}) := \sqrt{\|(A_{\mathcal{T}}' A_{\mathcal{T}} + \lambda I_{\mathcal{T}})^{-1} A_{\mathcal{T}}' A_{\tilde{\Delta}} P_{\mathcal{T},\lambda}(\tilde{\Delta})\|_2^2 + \|P_{\mathcal{T},\lambda}(\tilde{\Delta})\|_2^2}$$

$$f_2(\tilde{\Delta}) := \|Q_{\mathcal{T},\lambda}(\tilde{\Delta})^{-1}\|_2$$

$$f_3(\tilde{\Delta}) := \|Q_{\mathcal{T},\lambda}(\tilde{\Delta})^{-1} A_{T \cup \tilde{\Delta}}'\|_2,$$

$$f_4(\tilde{\Delta}) := \sqrt{\|Q_{\mathcal{T},\lambda}(\tilde{\Delta})^{-1} A_{T \cup \tilde{\Delta}}' A_{\tilde{\Delta} \setminus \tilde{\Delta}}\|_2^2 + 1}.$$

The result for modified-BPDN follows by setting  $\lambda = 0$  in the above expressions.

*Remark 7.4 (Choosing  $\lambda$ ,  $\gamma$  using Theorem 7.3):* Notice that  $\arg \min_{\tilde{\Delta} \subset \Delta, |\tilde{\Delta}|=k} \|x_{\Delta \setminus \tilde{\Delta}}\|_2$  in (34) is computable in polynomial time by just sorting the entries of  $x_\Delta$  in decreasing order of magnitude and retaining the indices of its largest  $k$  entries. Hence everything in the above result is computable in polynomial time if  $x$ ,  $\mathcal{T}$  and a bound on  $\|w\|_\infty$  are available. Thus if training data is available, the above theorem can be used to compute good choices of  $\gamma$  and  $\lambda$ . In a time series problem, all that one needs is a bound on  $\|w\|_\infty$  and two consecutive  $x_t$ 's; call them  $x_1$  and  $x_2$ . If these are not exactly sparse, we first sparsify them. We let  $\mathcal{T}$  be the support set of sparsified  $x_1$  and we let  $\mathcal{N}$  be the support set of sparsified  $x_2$ . We can then compute  $\Delta$ ,  $\Delta_e$  and all the quantities defined above. These can be used to get an expression for the reconstruction error bound  $g(\tilde{\Delta}^*(k_{\min}))$  as a function of  $\lambda$ . Notice that  $k_{\min}$  itself is also a function of  $\lambda$ . A good value of  $\lambda$  can be obtained by picking the one that maximizes this upper bound from a discrete set of

choices. This value can be substituted into the expression for  $\gamma_{\mathcal{T},\lambda}^*(\tilde{\Delta}^*(k_{\min}))$  given in (32) to get a good value of  $\gamma$ .

If true values of  $x_t$  for  $t = 1, 2$  are not available, one can use enough measurements and recover  $x_t$  for  $t = 1, 2$  using BPDN, (4), with  $\gamma$  selected using the heuristic suggested in [64]:  $\gamma = \max\{10^{-2} \|A_1' [y_1 \ y_2]\|_\infty, \sigma_{\text{obs}} \sqrt{m}\}$  (here  $\sigma_{\text{obs}}^2$  is the variance of any entry of  $w_t$ ).

### C. Error stability over time

In this section, we summarize the error stability results. We first obtained such a result for LS-CS in [32, Theorem 2]. We state here two results from [53] that improve upon this result in various ways: (i) they analyze modified-CS(modified-BP) and hence hold under weaker RIP conditions; (ii) they allow support change at each time (not just every so often as in [32]); and (iii) the first result below does not assume anything about how signal values change while the second one assumes a realistic signal change model.

Both these results are obtained by finding conditions to ensure that (i) the number of small (and hence undetectable) entries is not too large at any time, this ensures small number of misses; and (ii) the same is true for the number of extras. In the second result below, to ensure the former, we assume a signal model that ensures that all new additions to the support set are either detected immediately or are detected within a finite delay of getting added. To ensure the latter, we set the support threshold large enough so that  $|\hat{\mathcal{N}}_t \setminus \mathcal{N}_t| = 0$ .

*Theorem 7.5 (Modified-CS(modified-BP) error stability: no signal model):* [53, Theorem 3.2] Consider recovering  $x_t$ 's from  $y_t$  satisfies (8) using Algorithm 2. Assume that the support size of  $x_t$  is bounded by  $s$  and that there are at most

- 1) additions and at most  $s_a$  removals at all times. If
- 1) (*support estimation threshold*)  $\alpha = 7.50\epsilon$ ,
- 2) (*number of measurements*)  $\delta_{s+6s_a}(A_t) \leq 0.207$ ,
- 3) (*number of small magnitude entries*)  $|\{i \in \mathcal{N}_t : |(x_t)_i| \leq \alpha + 7.50\epsilon\}| \leq s_a$
- 4) (*initial time*) at  $t = 0$ ,  $n_0$  is large enough to ensure that  $|\tilde{\Delta}_t| = 0$ ,  $|\tilde{\Delta}_{e,t}| = 0$ .

then for all  $t$ ,

- $|\Delta_t| \leq 2s_a$ ,  $|\Delta_{e,t}| \leq s_a$ ,  $|\mathcal{T}_t| \leq s$ ,
- $\|x_t - \hat{x}_t\| \leq 7.50\epsilon$ ,
- $|\mathcal{N}_t \setminus \hat{\mathcal{N}}_t| \leq s_a$ ,  $|\hat{\mathcal{N}}_t \setminus \mathcal{N}_t| = 0$ ,  $|\tilde{\mathcal{T}}_t| \leq s$

The above result is the most general, but it does not give us practical models on signal change that would ensure the required upper bound on the number of small magnitude entries. Next we give one realistic model on signal change that would ensure this followed by a stability result for it.

*Model 7.6 (Model on signal change over time (parameter  $\ell$ ):* Assume the following model on signal change

- 1) At  $t = 0$ ,  $|\mathcal{N}_0| = s_0$ .
- 2) At time  $t$ ,  $s_{a,t}$  elements are added to the support set. Denote this set by  $\mathcal{A}_t$ . A new element  $j$  gets added to the support at an initial magnitude  $a_{j,t}$  and its magnitude increases for at least the next  $d_{\min}$  time instants. At time  $\tau$  (for  $t < \tau \leq t + d_{\min}$ ), the magnitude of element  $j$  increases by  $r_{j,\tau}$ .

Note:  $a_{j,t}$  is nonzero only if element  $j$  got added at time  $t$ , for all other times, we set it to zero.

- 3) For a given scalar  $\ell$ , define the “large set” as

$$\mathcal{L}_t(\ell) := \{j \notin \cup_{\tau=t-d_{\min}+1}^t \mathcal{A}_\tau : |(x_t)_j| \geq \ell\}.$$

Elements in  $\mathcal{L}_{t-1}(\ell)$  either remain in  $\mathcal{L}_t(\ell)$  (while increasing or decreasing or remaining constant) or decrease enough to leave  $\mathcal{L}_t(\ell)$ . At time  $t$ , we assume that  $s_{d,t}$  elements out of  $\mathcal{L}_{t-1}(\ell)$  decrease enough to leave it. All these elements continue to keep decreasing and become zero (removed from support) within at most  $b$  time units.

- 4) At all times  $t$ ,  $0 \leq s_{a,t} \leq s_a$ ,  $0 \leq s_{d,t} \leq \min\{s_a, |\mathcal{L}_{t-1}(\ell)|\}$ , and the support size,  $s_t := |\mathcal{N}_t| \leq s$  for constants  $s$  and  $s_a$  such that  $s + s_a \leq m$ .

Notice that an element  $j$  could get added, then removed and added again later. Let

$$\text{addtimes}_j := \{t : a_{j,t} \neq 0\}$$

denote the set of time instants at which the index  $j$  got added to the support. Clearly,  $\text{addtimes}_j = \emptyset$  if  $j$  never got added. Let

$$a_{\min} := \min_{j:\text{addtimes}_j \neq \emptyset} \min_{t \in \text{addtimes}_j, t > 0} a_{j,t}$$

denote the minimum of  $a_{j,t}$  over all elements  $j$  that got added at  $t > 0$ . We are excluding coefficients that never got added and those that got added at  $t = 0$ . Let

$$r_{\min}(d) := \min_{j:\text{addtimes}_j \neq \emptyset} \min_{t \in \text{addtimes}_j, t > 0} \min_{\tau \in [t+1, t+d]} r_{j,\tau}$$

denote the minimum, over all elements  $j$  that got added at  $t > 0$ , of the minimum of  $r_{j,\tau}$  over the first  $d$  time instants after  $j$  got added.

*Theorem 7.7 (Modified-CS(modified-BP) error stability):* [53] Consider recovering  $x_t$ 's from  $y_t$  satisfies (8) using Algorithm 2. Assume that Model 7.6 on  $x_t$  holds with

$$\ell = a_{\min} + d_{\min} r_{\min}(d_{\min})$$

where  $a_{\min}$ ,  $r_{\min}$  and the set  $\text{addtimes}_j$  are defined above. If there exists a  $d_0 \leq d_{\min}$  such that the following hold:

- 1) (algorithm parameters)  $\alpha = 7.50\epsilon$ ,
- 2) (number of measurements)  $\delta_{s+3(b+d_0+1)s_a}(A_t) \leq 0.207$ ,
- 3) (initial magnitude and magnitude increase rate)

$$\min\{\ell, \min_{j:\text{addtimes}_j \neq \emptyset} \min_{t \in \text{addtimes}_j} (a_{j,t} + \sum_{\tau=t+1}^{t+d_0} r_{j,\tau})\} > \alpha + 7.50\epsilon,$$

- 4) at  $t = 0$ ,  $n_0$  is large enough to ensure that  $|\tilde{\Delta}_0| \leq bs_a + d_0s_a$ ,  $|\tilde{\Delta}_{e,0}| = 0$ ,

then, for all  $t$ ,

- $|\Delta_t| \leq bs_a + d_0s_a + s_a$ ,  $|\Delta_{e,t}| \leq s_a$ ,  $|\mathcal{T}_t| \leq s$ ,
- $\|x_t - \hat{x}_t\| \leq 7.50\epsilon$ ,
- $|\mathcal{N}_t \setminus \hat{\mathcal{N}}_t| \leq bs_a + d_0s_a$ ,  $|\hat{\mathcal{N}}_t \setminus \mathcal{N}_t| = 0$ ,  $|\tilde{\mathcal{T}}_t| \leq s$

As long as the number of new additions or removals,  $s_a \ll s$  (slow support change), the above result shows that the worst case number of misses or extras is also small compared to the support size. This makes it a meaningful result. The reconstruction error bound is also small compared to the signal

energy as long as the signal-to-noise ratio is high enough ( $\epsilon^2$  is small compared to  $\|x_t\|^2$ ). Notice that both the above results need a bound on the RIC of  $A_t$  of order  $s + ks_a$  where  $k$  is a constant. On the other hand, BP-noisy needs the same bound on the RIC of  $A_t$  of order  $2s$  (see Theorem 2.4). This is stronger when  $s_a \ll s$  (slow support change).

## VIII. PARAMETER SETTING AND CODE LINKS

We describe approaches for setting the parameters for algorithms for which code is not available online as yet. The PM-CS-KF code is at <http://www2.ece.ohio-state.edu/~schniter/DCS/index.html>. The DCS-AMP code is at <http://www2.ece.ohio-state.edu/~schniter/DCS/index.html>. The streaming 11-homotopy code is at <http://users.ece.gatech.edu/~sasif/homotopy>. The CS-MUSIC [71] code is at <http://bispl.weebly.com/compressive-music.html> and the Temporal SBL (T-SBL) [72] code is at <http://dsp.ucsd.edu/~zhilin/TMSBL.html>.

We will post code or link to the code for all algorithms that are compared here at <http://www.ece.iastate.edu/~namrata/RecReconReview.html>.

Consider dynamic modified-CS(modified-BP) (Algorithm 2). It has two parameters  $\alpha$  and  $\epsilon$ . When setting these parameters automatically, they can change with time. Define the minimum nonzero value at time  $t$ ,  $x_{\min,t} = \min_{j \in \mathcal{N}_t} |(x_t)_j|$ . This can be estimated either as the minimum nonzero entry of the sparsified training data (if it is available) or as  $\hat{x}_{\min,t} = \min_{j \in \tilde{\mathcal{T}}_{t-1}} |(\hat{x}_{t-1})_j|$  or by taking an average over past few time instants of this quantity. In cases where this quantity varies a lot and cannot be estimated very reliably, it can occasionally result in a large support size. If it is under-estimated, it may result in the estimated support size becoming too large and this will cause the matrix  $A_{\mathcal{T}}$  to become ill-conditioned. To prevent this, one should also include a maximum support size constraint while estimating the support. The noise bound parameter,  $\epsilon$  is often either assumed known or it can be estimated using a short initial noise-only training sequence. In cases where it is not known, one can approximate it by  $\|y_{t-1} - A_{t-1}\hat{x}_{t-1}\|_2$  (assuming accurate recovery at  $t-1$ ).

Consider dynamic reg-mod-BPDN (Algorithm 4) and its special case, dynamic mod-BPDN (Algorithm 5). Algorithm 4 has three parameters  $\alpha$ ,  $\gamma$  and  $\lambda$ . We set  $\gamma$  and  $\lambda$  using Theorem 7.3. To do this one needs a short training sequence of  $x_t$ 's; at least two  $x_t$ 's are needed. This is obtained in one of two ways - either an actual training sequence is available; or one uses more measurements for the first two frames and solves BPDN for these two frames. The  $\gamma$  needed for BPDN for the first two frames is computed using the heuristic from [64]: set  $\gamma = \max\{10^{-2} \|A_1^T [y_1 \ y_2]\|_\infty, \sigma_{\text{obs}} \sqrt{m}\}$ . The estimated  $x_t$ 's are then used as training sequence. We then sparsify the training sequence to 99.9% (or some appropriate percent) energy; let  $\mathcal{T}$  equal the support set of sparsified  $x_1$  and let  $\mathcal{N}$  denote the support set of sparsified  $x_2$ ; and proceed as explained in Remark 7.4, i.e. select the  $\lambda$  that maximizes  $g(\tilde{\Delta}^*(k_{\min}))$  defined in Theorem 7.3 out of a discrete set of values, in our experiments this set was  $\{0.5, 0.2, 0.1, 0.05, 0.01, 0.005, 0.001, 0.0001\}$ ; use the selected  $\lambda$  in (32) given in Theorem 7.3 to get the value

---

**Algorithm 4 Dynamic Regularized Modified-BPDN (reg-mod-BPDN) [38]**


---

*Parameters:*  $\alpha, \gamma, \lambda$

At  $t = 0$ : Solve BPDN with sufficient measurements, i.e. compute  $\hat{x}_0$  as the solution of  $\min_b \gamma \|b\|_1 + 0.5 \|y_0 - A_0 b\|_2^2$ . For each  $t > 0$  do

- 1) Set  $\mathcal{T} = \hat{\mathcal{N}}_{t-1}$
- 2) *Reg-Mod-BPDN* Compute  $\hat{x}_t$  as the solution of

$$\min_b \gamma \|b_{\mathcal{T}^c}\|_1 + 0.5 \|y - Ab\|_2^2 + 0.5 \lambda \|b_{\mathcal{T}} - \hat{\mu}_{\mathcal{T}}\|_2^2$$

- 3) *Support Estimation - Simple Thresholding:*

$$\hat{\mathcal{N}}_t = \{i : |(\hat{x}_t)_i| > \alpha\} \quad (35)$$

*Parameter setting:* Set  $\gamma$  and  $\lambda$  using Theorem 7.3. To do this one needs a short training sequence of  $x_t$ 's; at least two  $x_t$ 's are needed. This is obtained in one of two ways - either an actual training sequence is available; or one uses more measurements for the first two frames and solves BPDN, (4), for these two frames with  $\gamma = \max\{10^{-2} \|A'_1 [y_1 \ y_2]\|_\infty, \sigma_{\text{obs}} \sqrt{m}\}$  [64]. (a) Sparsify the training sequence to 99.9% energy. (b) Let  $\mathcal{T}$  equal the support set of sparsified  $x_1$  and let  $\mathcal{N}$  denote the support set of sparsified  $x_2$ . (c) Select the  $\lambda$  that maximizes  $g(\hat{\Delta}^*(k_{\min}))$  given in Theorem 7.3 by choosing values from a discrete set. In our experiments, we select it out of the set  $\{0.5, 0.2, 0.1, 0.05, 0.01, 0.005, 0.001, 0.0001\}$ . (d) Use this  $\lambda$  and use  $\gamma$  given by (32) with this choice of  $\lambda$ . For more details, see Remark 7.4. (e) Set  $\alpha = 0.25x_{\min}$ . Here  $x_{\min}$  is the minimum nonzero entry of the sparsified training sequence. Also see Sec. VIII.

---



---

**Algorithm 5 Dynamic Modified-BPDN [38]**


---

*Parameters:*  $\alpha, \gamma$

Implement Algorithm 4 with  $\lambda = 0$ .

*Parameter setting:* Obtain the training sequence as explained in Algorithm 4 and sparsify it to 99.9% energy. Set  $\alpha$  as explained there. Set  $\gamma$  using (32) given in Theorem 7.3 computed with  $\lambda = 0.0001$ . For the reasoning, see Sec. VIII.

---

of  $\gamma$ . We set  $\alpha = 0.25x_{\min}$  as explained above. Consider dynamic mod-BPDN (Algorithm 5). Mod-BPDN is reg-mod-BPDN with  $\lambda = 0$ . So ideally this is what should be used when computing  $\gamma$  using the expression given in (32) of Theorem 7.3. However, notice that this expression requires inverting certain matrices which can sometimes be ill-conditioned if we set  $\lambda = 0$ . Hence, we instead use  $\lambda = 0.0001$  when computing  $\gamma$  using (32). The sparsified training sequence needed is obtained as explained earlier. We again set  $\alpha = 0.25x_{\min}$ .

Consider dynamic weighted- $\ell_1$  (Algorithm 6). It has three parameters  $\alpha, \gamma$  and  $\tau$ . We can set  $\alpha$  and  $\gamma$  as explained above. As explained in [50], we set  $\tau$  equal to the ratio of the number of extras to the number of entries in the support knowledge  $\mathcal{T}$ , i.e. we set  $\tau = |\Delta_e|/|\mathcal{T}|$ . Either this ratio is computed using the training data or it can be estimated from the support estimates for the previous two time instants.

Finally, consider KMoCS. We set  $\gamma$  and  $\alpha$  as explained above for mod-BPDN. We set  $\sigma_{\text{sys}}^2$  by maximum likelihood

---

**Algorithm 6 Dynamic Weighted- $\ell_1$  [49]**


---

*Parameters:*  $\alpha, \gamma, \tau$

Implement Algorithm 4 with the reg-mod-BPDN convex program replaced by

$$\min_b \gamma \|b_{\mathcal{T}^c}\|_1 + \gamma \tau \|b_{\mathcal{T}}\|_1 + 0.5 \|y - Ab\|_2^2$$

*Parameter setting:* Set  $\gamma$  and  $\alpha$  as explained in Algorithm 5. Set  $\tau$  equal to the ratio of the number of extras to the number of entries in the support knowledge  $\mathcal{T}$ , i.e.  $\tau = |\Delta_e|/|\mathcal{T}|$ . Compute this ratio from the sparsified training data. Also see Sec. VIII.

---



---

**Algorithm 7 KMoCS or Kalman Filtered Modified-CS-residual(modified-BP-residual): use of Modified-CS-residual(modified-BP-residual) in the KF-CS algorithm of [36]**


---

*Parameters:*  $\sigma_{\text{sys}}^2, \sigma_{\text{obs}}^2, \alpha, \gamma$

At  $t = 0$ : Solve BPDN with sufficient measurements, i.e. compute  $\hat{x}_0$  as the solution of  $\min_b \gamma \|b\|_1 + 0.5 \|y_0 - A_0 b\|_2^2$ . Denote the solution by  $\hat{x}_0$ . Estimate its support,  $\mathcal{T} = \hat{\mathcal{N}}_0$  by thresholding. Initialize  $P_0 = \sigma_{\text{sys}}^2 I_{\mathcal{T}} I_{\mathcal{T}}'$ .

For each  $t > 0$  do

- 1) Set  $\mathcal{T} = \hat{\mathcal{N}}_{t-1}$
- 2) *Modified-CS-residual(modified-BP-residual):*

$$\hat{x}_{t,\text{mod}} = \hat{x}_{t-1} + [\arg \min_b \gamma \|b_{\mathcal{T}^c}\|_1 + \|y_t - A\hat{x}_{t-1} - Ab\|_2^2]$$

- 3) *Support Estimation - Simple Thresholding:*

$$\hat{\mathcal{N}}_t = \{i : |(\hat{x}_{t,\text{mod}})_i| > \alpha\} \quad (36)$$

- 4) *Kalman Filter:*

$$\begin{aligned} \hat{Q}_t &= \sigma_{\text{sys}}^2 I_{\hat{\mathcal{N}}_t} I_{\hat{\mathcal{N}}_t}' \\ K_t &= (P_{t-1} + \hat{Q}_t) \left( A(P_{t-1} + \hat{Q}_t)A' + \sigma_{\text{obs}}^2 I \right)^{-1} \\ P_t &= (I - K_t A)(P_{t-1} + \hat{Q}_t) \\ \hat{x}_t &= (I - K_t A)\hat{x}_{t-1} + K_t y_t \end{aligned} \quad (37)$$

*Parameter setting:* Either a training sequence is available or obtain it as explained in Algorithm 4. Sparsify it to 99.9% energy. Set  $\gamma$  and  $\alpha$  as explained in Algorithm 5. Set  $\sigma_{\text{sys}}^2$  by maximum likelihood estimation on the sparsified training data. Either assume  $\sigma_{\text{obs}}^2$  is known or set it by computing the average of  $\|y_{t-1} - A_{t-1}\hat{x}_{t-1}\|_2^2$  over the last few time instants and dividing by  $n$ . In our experiments, we assumed the former. Also see Sec. VIII.

---

estimation on the sparsified training data. The observation noise variance  $\sigma_{\text{obs}}^2$  is usually assumed known. In cases where it is not, one can approximate it by computing the average of  $\|y_{t-1} - A_{t-1}\hat{x}_{t-1}\|_2^2$  over the last few time instants and dividing by  $n$ .

Lastly consider the add-LS-del support estimation approach. This is most useful for exactly sparse signal sequences. This needs two thresholds  $\alpha_{\text{add}}, \alpha_{\text{del}}$ . We can set  $\alpha_{\text{add},t}$  using the following heuristic. It is not hard to see (see [53, Lemma 2.8]) that  $(x_t - \hat{x}_{t,\text{add}})_{\mathcal{T}_{\text{add},t}} = (A_{\mathcal{T}_{\text{add},t}}' A_{\mathcal{T}_{\text{add},t}})^{-1} [A_{\mathcal{T}_{\text{add},t}}' w_t +$

$A_{\mathcal{T}_{\text{add},t}}' A_{\Delta_{\text{add},t}}(x_t)_{\Delta_{\text{add},t}}$ . To ensure that the RHS is bounded, we need  $A_{\mathcal{T}_{\text{add},t}}$  to be well conditioned. To ensure this we pick  $\alpha_{\text{add},t}$  as the smallest number such that  $\sigma_{\min}(A_{\mathcal{T}_{\text{add},t}}) \geq 0.4$ . If one could set  $\alpha_{\text{del},t}$  equal to the lower bound on  $x_{\min,t} - \|(x_t - \hat{x}_{t,\text{add}})_{\mathcal{T}_{\text{add},t}}\|_{\infty}$ , there will be zero misses. Using this idea, we let  $\alpha_{\text{del},t}$  be an estimate of the lower bound of this quantity. As explained in [53, Section VII-A], this leads to  $\alpha_{\text{del},t} = 0.7\hat{x}_{\min,t} - \|A_{\mathcal{T}_{\text{add},t}}^{\dagger}(y_t - A\hat{x}_{t,\text{modcs}})\|_{\infty}$ . An alternative approach that has been successful for a large class of video sequences is explained in [40, Algorithm 1].

## IX. NUMERICAL EXPERIMENTS

We report three experiments. The first studies the noise-free case and compares algorithms that only exploit slow support change of the sparse signal sequence. As explained earlier, this problem can be reformulated as one of sparse recovery with partial support knowledge. In the noise-free case, once the static problem is solved, so is the dynamic one. So we only simulate the static problem. In the second experiment, we study the noisy case and the dynamic (recursive recovery) problem for a simulated sparse signal sequence and random Gaussian measurements. In the third experiment, we study the same problem but for a real image sequence (a sub-image sequence of the larynx sequence shown in Fig. 1) and simulated MRI measurements. In the second and third experiment, we compare all types of algorithms - those that exploit none, one or both of slow support change and slow signal value change assumptions.

### A. Sparse recovery with partial support knowledge: phase transition plots for comparing $n$ needed for exact recovery

We compare BPDN and CS-residual(BP-residual) with all the approaches that only use partial support knowledge - LS-CS, modified-CS(modified-BP) and weighted- $\ell_1$  - using phase transition plots. CS-residual(BP-residual) is an approach that uses signal value knowledge but not support knowledge. The reason for including it in the comparisons here is to demonstrate what we explained in Section III-B. The number of measurements needed for exact recovery will not reduce if we use signal value knowledge, without explicitly using support knowledge.

For a given choice of the support size,  $s$ , the number of misses in the support knowledge,  $u$ , and the number of extras,  $e$ , we vary the number of measurements,  $n$ , and use Monte Carlo to estimate the probability of exact recovery of the various approaches. We generated the true support  $\mathcal{N}$  of size  $s$  uniformly at random from  $[1, 2, \dots, m]$ . The nonzero entries of  $x$  were generated from a Gaussian  $\mathcal{N}(0, \sigma_x^2)$  distribution. The support knowledge  $\mathcal{T}$  was generated as  $\mathcal{T} = \mathcal{N} \cup \Delta_e \setminus \Delta$  where  $\Delta_e$  is generated as a set of size  $e$  uniformly at random from  $\mathcal{N}^c$  and  $\Delta$  was generated as a set of size  $u$  uniformly at random from  $\mathcal{N}$ . We generated the observation vector  $y = Ax$  where  $A$  is an  $n \times m$  random Gaussian matrix. Since CS-residual(BP-residual) uses signal value knowledge, for it, we generate a ‘‘signal value knowledge’’  $\hat{\mu}$  as follows. Generate  $\hat{\mu}_{\mathcal{N} \cup \mathcal{T}} = x_{\mathcal{N} \cup \mathcal{T}} + e_{\mathcal{N} \cup \mathcal{T}}$  with  $e \sim \mathcal{N}(0, \sigma_e^2 I)$ , and set  $\hat{\mu}_{(\mathcal{N} \cup \mathcal{T})^c} = 0$ .

Modified-CS(modified-BP) solved (17) which has no parameters. For weighted- $\ell_1$ , we solved (20) with  $\tau = e/s$  as suggested in [50]. For CS-residual(BP-residual), we computed  $\hat{x} = \hat{\mu} + [\arg \min_b \|b\|_1 \text{ s.t. } y - A\hat{\mu} = Ab]$ , where  $\hat{\mu}$  was generated as above. For LS-CS, we did the same thing but  $\hat{\mu}$  was now  $I_{\mathcal{T}} A_{\mathcal{T}}^{\dagger} y$  (LS estimate on  $\mathcal{T}$ ). All convex programs were solved using CVX.

In our experiment, we used  $m = 200$ ,  $s = 0.1m$ ,  $u = 0.1s$ ,  $\sigma_x^2 = 5$  and three values of  $e$ :  $e = 0$  (Fig. 3(a)) and  $e = u$  (Fig. 3(b)) and  $e = 4u$  (Fig. 3(c)). The phase transition plot varies  $n$  and plots the Monte Carlo estimate of the probability of exact recovery. For CS-residual(BP-residual), we generated two types of prior signal knowledge, the good prior case with  $\sigma_e^2 = 0.0001\sigma_x^2$  and the bad prior case with  $\sigma_e^2 = \sigma_x^2$ . We label them as CSres(BPres)-good-prior and CSres(BPres)-bad-prior. The probability of exact recovery was computed by generating 100 realizations of the data and counting the number of times  $x$  was exactly recovered by a given approach. We say that  $x$  is exactly recovered if  $\frac{\|x - \hat{x}\|}{\|x\|} < 10^{-6}$  (precision used by CVX). We show three cases,  $e = 0$ ,  $e = u$  and  $e = 4u$  in Fig. 3. In all cases we observe the following. (1) BP and CS-residual(BP-residual) need the same number of measurements to achieve exact recovery with (Monte Carlo) probability one. This is true even when very good prior knowledge is provided to CS-residual(BP-residual). (2) LS-CS needs more measurements for exact recovery than either of BP or CS-residual(BP-residual). This is true even in the  $e = 0$  case (and this is something we are unable to explain). (3) Weighted- $\ell_1$  and modified-CS(modified-BP) significantly outperform all other approaches - they need a significantly smaller  $n$  for exact recovery with (Monte Carlo) probability one. (4) Finally, from this set of simulations, it is hard to differentiate modified-CS(modified-BP) and weighted- $\ell_1$ . However, in other works [50], [38], in the noisy measurements’ case, it has been observed in other works that when  $e$  is larger, weighted- $\ell_1$  has a smaller recovery error than modified-CS(modified-BP). We also notice this in Fig. 5.

### B. Recursive recovery of a simulated sparse signal sequence from noisy random-Gaussian measurements

We compare BPDN, CS-residual(BPDN-residual), PM-CS-KF [56], modified-BPDN, Algorithm 5, weighted- $\ell_1$ , Algorithm 6, streaming  $\ell_1$ -homotopy [64], reg-mod-BPDN, Algorithm 4, KMoCS, Algorithm 7, DCS-AMP [58], [59], CS-MUSIC [71] and Temporal SBL [72]. Among these, BPDN uses no prior knowledge; CS-residual(BPDN-residual) and PM-CS-KF use only slow signal value change and sparsity; modified-BPDN and weighted- $\ell_1$  use only slow support change; and reg-mod-BPDN, KMoCS, DCS-AMP [58], [59] use both slow support and slow signal value change. CS-MUSIC [71] and Temporal SBL [72] are two batch algorithms that solve the MMV problem (assumes the support of  $x_t$  does not change with time). Since  $A$  is fixed, we are able to apply these as well. Temporal SBL [72] additionally also uses temporal correlation among the nonzero entries while solving the MMV problem. The MMV problem is discussed in detail in Sec. X-A2 (part of related work and future directions).

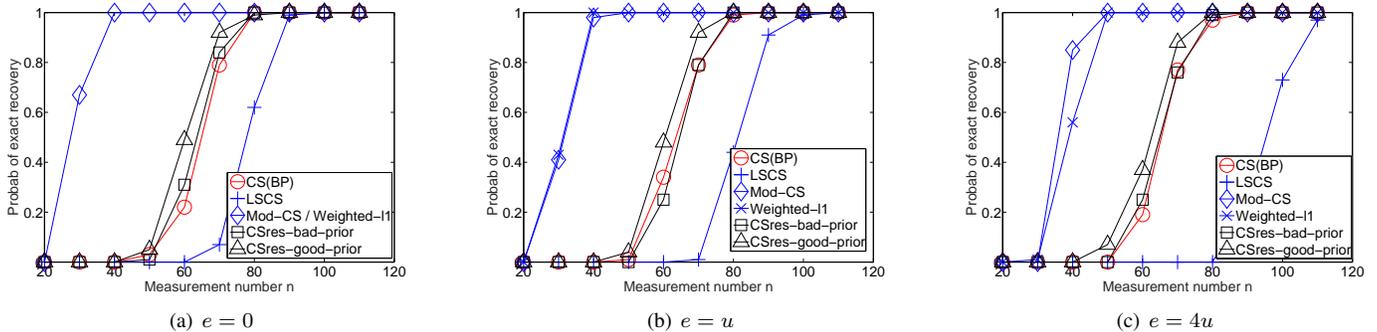


Fig. 3. Phase transition plots. We plot the Monte Carlo estimate of the probability of exact recovery against  $n$  (number of measurements). Notice that in the  $e = 0$  case, weighted- $\ell_1$  uses  $\tau = e/s = 0$  and hence it is the same as modified-CS(modified-BP). All figures used  $m = 200$ ,  $s = 0.1m$ ,  $u = 0.1s$ . CS-residual(BP-residual) was compared for two cases: good prior and bad prior.

Notice also that modified-CS(modified-BP), reg-mod-BPDN and KMoCS are algorithms from our group. The rest are either baseline (BPDN and CS-residual(BPDN-residual)) or algorithms from other groups (DCS-AMP, PM-CS-KF, weighted- $\ell_1$ , streaming  $\ell_1$ -homotopy, CS-MUSIC, Temporal SBL).

We used the authors' code for DCS-AMP, PM-CS-KF, l1-homotopy, Temporal-SBL and CS-MUSIC. The code links are as follows: l1-homotopy: <http://users.ece.gatech.edu/~sasif/homotopy>, Temporal-SBL: <http://dsp.ucsd.edu/~zhilin/TMSBL.html>, CS-MUSIC: <http://bispl.weebly.com/compressive-music.html>, DCS-AMP: <http://www2.ece.ohio-state.edu/~schniter/DCS/index.html>.

Code for PM-CS-KF was obtained by email from the authors. For the others, we wrote our own MATLAB code and used CVX to solve the convex programs. BPDN solved (4) with  $y = y_t$ ,  $A = A_t$  at time  $t$  and  $\gamma = \max\{10^{-2}\|A'[y_1 \ y_2 \ \dots \ y_{t_{max}}]\|_\infty, \sigma_{\text{obs}}\sqrt{m}\}$  [64]. CS-residual(BPDN-residual) solved (4) with  $y = y_t - A_t\hat{x}_{t-1}$ ,  $A = A_t$  at time  $t$  and  $\gamma = \max\{10^{-2}\|A'[y_1 \ y_2 \ \dots \ y_{t_{max}}]\|_\infty, \sigma_{\text{obs}}\sqrt{m}\}$ . For reg-mod-BPDN, mod-BPDN, weighted- $\ell_1$  and KMoCS the complete stepwise algorithms are specified. We will take permission from all authors and post code for all the algorithms on a single webpage, <http://www.ece.iastate.edu/~namrata/RecReconReview.html> soon.

We generated a sparse signal sequence using a generative model that satisfies the assumptions given in Model 7.6. The specific generative model used is the one specified in [53, Appendix I] with one change: the magnitude of *all* entries in the support (not just the newly added ones) changed over time. We generated  $y_t = Ax_t + w_t$  where  $A$  was a random Gaussian matrix of size  $n_1 \times m$  for  $t = 1, 2$  and of size  $n_3 \times m$  for  $t \geq 3$  and  $w_t$  was Gaussian noise with zero mean and variance  $\sigma_{\text{obs},t}^2$ . The parameters used for Model 7.6 were  $m = 200$ ,  $s_t = s = 0.1m = 20$ ,  $s_{r,t} = s_{d,t} = s_{a,t} = s_a = 0.04s = 1$ ,  $d_{\min} = 2$ ,  $b = 4$ ,  $a_{\min} = 2$ ,  $r_{\min} = 5$ ,  $a_{j,t}$  varied between  $[a_{\min}, 3a_{\min}]$ ,  $r_{j,t}$  varied between  $[r_{\min}, 3r_{\min}]$ . At the initial time, all entries in the initial support,  $\mathcal{N}_1$ , had magnitude that varied between  $[a_{\min} + d_{\min}r_{\min}, a_{\min} + 3d_{\min}r_{\min}]$ . For the measurements, we used  $n_1 = 180$  for  $t = 1, 2$  in order to

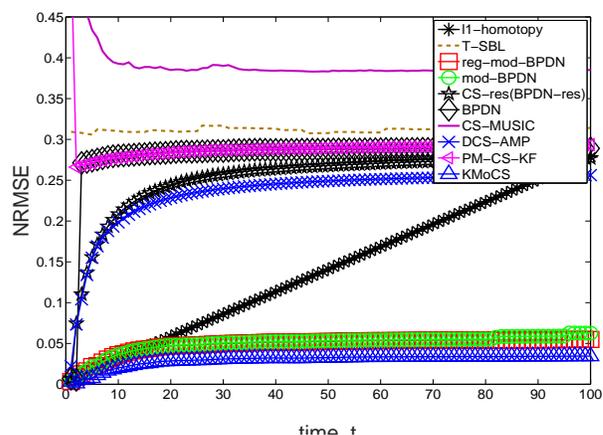


Fig. 4. NMSE plot for a simulated sparse signal sequence generated as explained in Section IX-B.

obtain a very accurate recovery of  $x_1$  and  $x_2$  (these are used as the “training data” for learning the algorithm parameters for future time instants). This training data was provided to all the algorithms. For  $t > 2$ , we used  $n_3 = 0.13m = 33$  and  $\sigma_{\text{obs},t}^2 = 0.0004$ . Also for  $t = 1, 2$ ,  $\sigma_{\text{obs},1}^2 = 0.00001$ .

We plot the results in Fig. 4. As can be seen, KMoCS and reg-mod-BPDN have the smallest and stable normalized root mean squared error (NRMSE). Mod-BPDN and DCS-AMP have the next smallest errors. The CS-residual(BP-residual) error starts out small but becomes unstable pretty quickly. This is because signal value changes over time are not too slow in this example. Support changes are slow but then CS-residual(BP-residual) does not use slow support change.

### C. Recursive recovery of a real vocal tract dynamic MRI sequence (approximately sparse) from simulated partial Fourier measurements

We took the 256 x 256 larynx (vocal tract) dynamic MRI sequence (shown in Fig. 1) and selected a 32 x 32 block of it that contains most of the significant motion. The indices of the selected block were (60 : 91, 60 : 91). Thus  $m = 1024$ . This was done to allow us to use CVX to solve all our convex

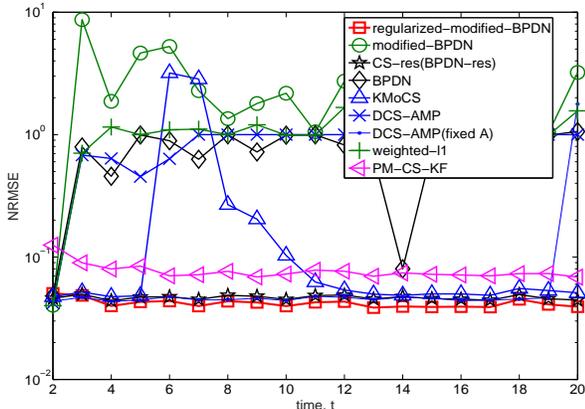


Fig. 5. NMSE plot for recovering a  $32 \times 32$  sub-image sequence of the larynx MRI sequence from simulated partial Fourier measurements corrupted by Gaussian noise. In this case,  $m = 1024$ . We used  $n_1 = n_2 = 0.18m$  and  $n_t = 0.06m$  for  $t > 2$ . DCS-AMP (fixed A) is a plot of DCS-AMP with using a time-invariant measurement matrix  $A$  with  $n = 0.12m$  rows.

programs. CVX requires that the measurement matrix be small enough so that it can be saved in the memory. This is why we needed to pick a smaller sized image sequence. Let  $z_t$  denote the image at time  $t$  arranged as vector. We simulated MRI measurements as  $y_t = H_t z_t + w_t$  where  $H_t = M_t F_{1024}$  where  $F_m$  is an  $m$ -point DFT matrix,  $M_t$  consisted of  $n_t$  rows selected using a modification of the low-frequency random undersampling scheme of Lustig et al [73] and  $w_t$  was zero mean i.i.d. Gaussian noise with variance  $\sigma_{obs}^2 = 10$ . We used  $n_1 = n_2 = 0.18m$  and  $n_t = 0.06m = 62$  for  $t > 2$ . To get 6% measurements, we generated three mask matrices with 50%, 40% and 30% measurements each using the low-frequency random undersampling scheme, multiplied them, and selected 62 rows out of the resulting matrix uniformly at random.

For all algorithms, the sparsity basis we used was a two-level Daubechies-4 wavelet. Thus  $\Phi$  was the inverse wavelet transform corresponding to this wavelet written as a matrix.

We compare BPDN,  $\ell_1$ -homotopy, CS-residual(BPDN-residual), PM-CS-KF [56], modified-BPDN, Algorithm 5, weighted- $\ell_1$ , Algorithm 6; and reg-mod-BPDN, Algorithm 4, KMoCS, Algorithm 7, DCS-AMP [58], [59]. The algorithms were implemented exactly as explained in the previous subsection. The training data (first two images recovered using simple BPDN) was provided to all algorithms.

The NMSE of the recovered sequence obtained using the various algorithms is plotted in Fig. 5. As can be seen reg-mod-BPDN has the smallest NMSE. The NMSE of CS-residual(BP-residual) and of PM-CS-KF is only a little larger since, in this case, the signal value change is very gradual. KMoCS error is large for sometime but becomes small again. DCS-AMP does not work well in this experiment most likely because the number of measurements used is too small. We show another plot for DCS-AMP with 12% measurements and with  $A$  kept fixed (labeled DCS-AMP (fixed-A)). In this plot it has much smaller error. Mod-BPDN and weighted- $\ell_1$  fail because the available number of measurements is too few in

this case ( $n_t = 0.06m$  where as the 99.9% support set size is around  $0.07m$  for this sequence).

## X. RELATED WORK AND FUTURE DIRECTIONS

We split the discussion in this section into four parts. We first describe work that uses related signal models and its implications for the problem discussed here. Next we discuss more general measurement models beyond the linear measurements and small noise case studied here, and how some of the approaches described here can also be used for these problems. Next, we describe some algorithms that may seem to be solving the problem studied here but are actually quite different. Finally, we discuss the problem of compressive sequential detection, filtering and tracking.

### A. Signal Models

1) *Structured Sparsity*: There has been a lot of recent work on structured sparsity for a single signal. Dynamic extensions of these ideas should prove to be useful in applications where the structure is present and changes slowly. Two common examples of structured sparsity are block sparsity [74], [75] and tree structured sparsity (for wavelet coefficients) [76]. A length  $m$  vector is block sparse if it can be partitioned into length  $k$  blocks such that a lot of the blocks are entirely zero. One way to recover block sparse signals is by solving the  $\ell_2$ - $\ell_1$  minimization problem [74], [75]. Block sparsity is valid for many applications, e.g., for the foreground image sequence of a video consisting of one or a few moving objects, or for the activation regions in brain fMRI. In both of these cases, it is also true that the blocks do not change arbitrarily and hence the block support from the previous time instant should be a good estimate of the current block support. In this case one, can again use a modified-CS(modified-BP) type idea applied to the blocks. Denote the set of known nonzero blocks by  $\mathcal{T}$ . Then modified- $\ell_2$ - $\ell_1$ , developed by Stojnic [77], solves

$$\min_b \sum_{j=1, j \notin \mathcal{T}}^{m/k} \sqrt{\sum_{i=1}^{k-1} b_{j^{k+i}}^2} \quad \text{s.t.} \quad y = Ab$$

The above idea can be easily used for recursive recovery problems by letting the set  $\mathcal{T}$  be the block-support estimate from the previous time instant. Similarly, it should be possible to develop a dynamic extension of tree-structured CoSaMP that was developed and analyzed in [76].

2) *MMV and dynamic MMV*: In the MMV problem [21], [22], [23], [24], [71], [78], [79], the goal is to recover a set of sparse signals with a common support but different nonzero signal values from a set of their measurements (all obtained using the same measurement matrix). In our notation, we have  $y_t = Ax_t$  for  $t = 0, 1, 2, \dots, t_0$ , and the support set of  $x_t$ ,  $\mathcal{N}_t = \mathcal{N}_0$  (constant support) [21]. This problem is similar to the block-sparse signal case and hence one commonly used solution is the  $\ell_2$ - $\ell_1$  program. Another more recent set of solutions is inspired by the MUSIC algorithm and are called CS-MUSIC [71] or iterative-MUSIC [78]. For signals with time-varying support (the case studied in this article), one can still use the MMV approaches as long as the size of their

joint support (union of their supports),  $\mathcal{N} := \cup_t \mathcal{N}_t$ , is small enough. This may not be true for the entire sequence, but will usually be true for short durations. One could design a modified-MMV algorithm that utilizes the joint support of the previous duration to solve the current MMV problem better. A related idea was explored in the recent dynamic MMV work of Kim et al [80]. Instead of short batches of time signals, they considered the case where a set of sparse signals have a common support at a given time and over time this support changes slowly.

Another related work is that of Zhang and Rao [72]. In it, the authors develop what they call the temporal SBL (T-SBL) algorithm. This is a batch (but fast) algorithm that solves the MMV problem with temporal correlations. It assumes that the support of  $x_t$  does not change over time and the signal values are correlated over time. The various indices of  $x_t$  are assumed to be independent. For a given index,  $i$ , it is assumed that  $[(x_1)_i, (x_2)_i, \dots, (x_{t_{\max}})_i]^T \sim \mathcal{N}(0, \gamma_i B)$ . All the  $x_t$ 's are strung together to get a long  $t_{\max}m$  length vector  $x$  that is Gaussian with block diagonal covariance matrix  $\text{diag}(\gamma_1 B, \gamma_2 B, \dots, \gamma_m B)$ . T-SBL develops the SBL approach to estimate the hyper-parameters  $\{\sigma^2, \gamma_1, \gamma_2, \dots, \gamma_m, B\}$  and then compute an MAP estimate of the sparse vector sequence. It is further speeded up by certain approximations introduced in [72]. The hierarchical KF algorithm explained earlier in Sec V-F can be interpreted as a recursive solution for a problem that is very similar to the one solved by T-SBL.

Another related work [81] develops and studies a causal but batch algorithm for CS for time-varying signals.

### 3) Sparse Transform Learning and Dictionary Learning:

In certain applications involving natural images, the wavelet transform provides a good enough sparsifying basis. However, for many other applications, while the wavelet transform is one possible sparsifying basis, it can be significantly improved. There has been a large amount of recent work both on dictionary learning, e.g., [82], and more recently on sparsifying transform learning from a given dataset of sparse signals [83]. The advantage of the latter work is that it is much faster than existing dictionary learning approaches. For dynamic sparse signals, an open question of interest is how to learn a sparsifying transform that is optimized for signal sequences with slow support change?

## B. Measurement Models

### 1) Recursive Recovery in Large but Structured Noise:

The work discussed in this article solves the sparse recovery problem either in the noise-free case or in the small noise case. Only in this case, one can show that the reconstruction error is small compared to the signal energy. In fact, this is true for almost all work on sparse recovery; one can get reasonable error bounds only for the small noise case.

However, in some applications, the noise energy can be much larger than that of the sparse signal. If the noise is large but has no structure then nothing can be done. But if it does have structure, that can be exploited. This was first done for outliers (modeled as sparse vectors) in the work of Wright and Ma [84]. Their work, and then many later works, showed

exact sparse recovery from large but sparse noise (outliers) as long as the sparsity bases for the signal and the noise/outlier are “different” or “incoherent” enough. More recent work on robust principal components’ analysis (PCA) by Candès et al [5] and Chandrasekharan et al [85] studied the problem of separating a low-rank matrix and a sparse matrix from their sum. A key application where this problem occurs is in separating video sequences into foreground and background image sequences. In more recent work [7], [6], [40], [86], the recursive or online robust PCA problem was solved. This can be interpreted as a problem of recursive recovery of sparse signal sequences in large but structured noise (noise that is dense and lies in a fixed or “slowly changing” low-dimensional subspace of the full space). As we explain next, one step in the proposed solution, Recursive Projected CS (ReProCS), is an example of the problem studied in this article.

ReProCS proceeds as follows. Suppose that  $y_t = x_t + \ell_t$  where  $x_t$  is the sparse vector and  $\ell_t$  is the large but structured noise. Given an initial estimate of the subspace of the noise, it projects  $y_t$  into the space orthogonal to this subspace. Because of the slow subspace change assumption, this approximately nullifies  $\ell_t$  and gives projected measurements of  $x_t$ . The resulting problem now becomes one of sparse recovery in small noise (due to  $\ell_t$  not being completely nullified). The denseness assumption on the basis vectors that span the noise subspace ensures that RIP holds for this problem and  $x_t$  can be accurately recovered using BP-noisy. One then recovers  $\ell_t$  by subtraction from  $y_t$  and updates the subspace estimate every few frames. The sparse recovery step in ReProCS is an example of the problem of recursive recovery of sparse signal sequences from compressive measurements and small noise. For solving it, BP-noisy can be replaced by any of the approaches described here. For example, in the video experiments shown in [40], weighted- $\ell_1$  was used. An open question is how to obtain performance guarantees for the resulting approach?

2) *Recursive Recovery from Nonlinear Measurements and Dynamic Phase Retrieval:* The work described in this article focuses on sparse recovery from linear measurements. However, in many applications such as computer vision, the measurement (e.g., image) is a nonlinear function of the sparse signal of interest (e.g., object’s boundary which is often modeled as being Fourier sparse). Some recent work that has studied the static version of this “nonlinear CS” problem includes [87], [88], [89], [90], [91]. These approaches use an iterative linearization procedure along with adapting standard iterative sparse recovery techniques such as IHT. An extension of these techniques for the dynamic case can potentially be developed using an extended Kalman filter and adapting modified-IHT. There has been other work on solving the dynamic CS problem from nonlinear measurements by using particle filtering based approaches [62], [92], [93], [61].

An important special case of the nonlinear CS problem is sparse phase retrieval [94], [95], i.e. recover a sparse  $x$  from  $y := |Ax|$ . Here  $|\cdot|$  takes the element-wise magnitude of the vector  $(Ax)$ . The sparse phase retrieval problem from Fourier magnitude measurements (recover  $x$  from  $y := |Fx|$ ) occurs in applications such as astronomical imaging, optical imaging

and X-ray crystallography where one can only measure the magnitude of the Fourier coefficients of the unknown quantity. One solution approach for this problem (that also comes with performance guarantees) involves first using a combinatorial algorithm to estimate the signal's support, followed by using a lifting technique to get a convex optimization program for positive semi-definite matrices [96], [97], [98], [99]. As explained in [97], the lifting based convex program cannot be used directly (without the support information) because of the location ambiguity introduced by the Fourier transform - the magnitude of the FT of a signal and of its shifted version is the same. Consider the dynamic sparse phase retrieval problem. An open question is whether the support and the signal value estimates from the previous time instant can help regularize this problem enough to ensure a unique solution when directly solving the resulting lifting-based convex program? If the combinatorial support recovery algorithm can be eliminated, it would make the solution approach a lot faster. A similar question can also be asked for the other more recent sparse Fourier phase retrieval solutions such as GESPAR [100].

### C. Algorithms

1) *“Optimal” algorithm(s)*: The work done on this topic so far consists of good algorithms that improve significantly over simple-CS solutions, and some of them come with provable guarantees. However, it is not clear if any of these are “optimal” in any sense. An open question is, can we develop an “optimal” algorithm or can we show that an algorithm is close enough to an “optimal” one? The original KF-CS algorithm [36] was developed with this question in mind; however so far there has not been any reasonable performance bound for it or for its improved version, KMoCS. An open question is, can we show that KMoCS comes within a bounded distance of the genie-aided causal MMSE solution for this problem (the causal MMSE solution assuming that the support sets at each time are known)?

2) *Homotopy methods to speed up the reconstruction algorithm*: In this article, we reviewed work on algorithms for recursively recovering a time sequence of sparse signals from a greatly reduced number of linear projection measurements (the goal is to be able to use fewer measurements than what simple-CS algorithms need). In recent work by Asif and Romberg [101], [102], the authors have studied the problem of using the recovered signal from the previous time instant and the fact that signals change slowly over time to speed up the solving of the BPDN program. Since their algorithm is a homotopy to solve the BPDN program, it needs the same number of measurements as BPDN. For the problem studied in this article, a question of interest is how to design homotopies for the modified-CS(modified-BP) or the weighted- $\ell_1$  program? This question is answered, in part, in [64].

### D. Compressive Sequential Signal Detection, Classification, Estimation and Filtering

In many signal processing applications, the final goal is to use the recovered signal for detection, classification, estimation or to filter out a certain component of the signal (e.g. a

band of frequencies or some other subspace). The question is can we do this directly with compressive measurements without having to first recover the sparse signal? This has been studied in some recent works such as the work of Davenport et al [103]. In fact the approach suggested in this work for filtering solves the same convex program as modified-CS(modified-BP) - there the set  $\mathcal{T}$  is the known support of the interference. For a time sequence of sparse signals, a question of interest is how to do the same thing for compressive sequential detection/classification or sequential estimation (tracking)? The question to answer would be how to use the previously reconstructed/detected signal to improve compressive detection at the current time and how to analyze the performance of the resulting approach?

### E. Other very recent related work

We briefly mention here some very recent related work that appeared since we started writing this review article. There has been very recent work by Hegde et al on approximation tolerant model-based CS [104] which develops faster but approximate algorithms for model-based CS. An interesting question is whether something similar can be done for approximate modified-CS(modified-BP)?

Another related work [105] assumes that the earth-mover's distance between the support sets of consecutive signals is small and uses this to design a recursive dynamic CS algorithm (instead of just using slow support change - which can be interpreted as the  $\ell_0$  distance between the support sets).

There is also very recent work on analyzing weighted- $\ell_1$  and obtaining necessary and sufficient conditions for it to work [106].

## XI. CONCLUSIONS

This article reviewed the literature on recursive recovery of sparse signal sequences or what can be called “recursive dynamic CS”. Most of the literature on this topic exploits one or both of the practically valid assumptions of slow support change and slow signal value change. While slow signal value change is commonly used in a lot of previous tracking and adaptive filtering literature, the slow support change is a new assumption introduced for solving this problem. As shown in Fig. 2, this is indeed valid in application such as dynamic MRI. We summarized both theoretical and experimental results that demonstrate the advantage of using these assumptions. In the section above, we also discussed related problems and open questions for future work.

A key limitation of almost all the work reviewed here is that the algorithms assume more measurements are available at the first time instant. This is needed in order to get an accurate initial signal recovery using simple-CS solutions. In some applications such as dynamic MRI or functional MRI, it is possible to use more measurements at the initial time (the scanner can be configured to allow this). In cases where this is not possible, a possible option is to solve the MMV problem described above for a small initial batch of time instants with the common support given by the union of the supports for all the time instants in this batch.

## REFERENCES

- [1] A. J. Martin, O. M. Weber, D. Saloner, R. Higashida, M. Wilson, M. Saeed, and C.B. Higgins, "Application of MR Technology to Endovascular Interventions in an XMR Suite," *Medica Mundi*, vol. 46, December 2002.
- [2] M. Wakin, J. Laska, M. Duarte, D. Baron, S. Sarvotham, D. Takhar, K. Kelly, and R. Baraniuk, "An architecture for compressive imaging," in *IEEE Intl. Conf. Image Proc. (ICIP)*, 2006.
- [3] M. Wakin, N. Vaswani, B. Loïs, and L. Hogben, "Recursive robust pca or recursive sparse recovery in large but structured noise," *IEEE Trans. Info. Th.*, August 2014, early versions in ICASSP 2013 and ISIT 2013.
- [7] C. Qiu and N. Vaswani, "Real-time robust principal components' pursuit," in *Allerton Conf. on Communications, Control and Computing*, 2010.
- [8] C. Qiu and N. Vaswani, "Support predicted modified-cs for recursive robust principal components' pursuit," in *IEEE Intl. Symp. Info. Th. (ISIT)*, 2011.
- [9] I. Carron, "Nuit blanche," in <http://nuit-blanche.blogspot.com/>.
- [10] "Rice compressive sensing resources," in <http://www-dsp.rice.edu/cs>.
- [11] S.G. Mallat and Z. Zhang, "Matching pursuits with time-frequency dictionaries," *IEEE Trans. Sig. Proc.*, vol. 41(12), pp. 3397 – 3415, Dec 1993.
- [12] S. Chen, D. Donoho, and M. Saunders, "Atomic decomposition by basis pursuit," *SIAM Journal of Scientific Computing*, vol. 20, pp. 33–61, 1998.
- [13] P. Feng and Y. Bresler, "Spectrum-blind minimum-rate sampling and reconstruction of multiband signals," in *IEEE Intl. Conf. Acoustics, Speech, Sig. Proc. (ICASSP)*, 1996, vol. 3, pp. 1688–1691.
- [14] Y. Bresler and P. Feng, "Spectrum-blind minimum-rate sampling and reconstruction of 2d multiband signals," in *IEEE Intl. Conf. Image Proc. (ICIP)*, 1996.
- [15] I. F. Gorodnitsky and B. D. Rao, "Sparse signal reconstruction from limited data using focuss: A re-weighted norm minimization algorithm," *IEEE Trans. Sig. Proc.*, pp. 600 – 616, March 1997.
- [16] I.F. Gorodnitsky and B.D. Rao, "Sparse signal reconstruction from limited data using focuss: A re-weighted norm minimization algorithm," *IEEE Trans. Sig. Proc.*, vol. 45, pp. 600–616, March 1997.
- [17] D.P. Wipf and B.D. Rao, "Sparse bayesian learning for basis selection," *IEEE Trans. Sig. Proc.*, vol. 52, pp. 2153–2164, Aug 2004.
- [18] D. Donoho, "Compressed sensing," *IEEE Trans. Info. Th.*, vol. 52(4), pp. 1289–1306, April 2006.
- [19] E. Candes and T. Tao, "Decoding by linear programming," *IEEE Trans. Info. Th.*, vol. 51(12), pp. 4203 – 4215, Dec. 2005.
- [20] E. Candes, J. Romberg, and T. Tao, "Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information," *IEEE Trans. Info. Th.*, vol. 52(2), pp. 489–509, February 2006.
- [21] David P Wipf and Bhaskar D Rao, "An empirical bayesian strategy for solving the simultaneous sparse approximation problem," *Signal Processing, IEEE Transactions on*, vol. 55, no. 7, pp. 3704–3716, 2007.
- [22] Joel A Tropp, "Algorithms for simultaneous sparse approximation. part ii: Convex relaxation," *Signal Processing*, vol. 86, no. 3, pp. 589–602, 2006.
- [23] Jie Chen and Xiaoming Huo, "Theoretical results on sparse representations of multiple-measurement vectors," *Signal Processing, IEEE Transactions on*, vol. 54, no. 12, pp. 4634–4643, 2006.
- [24] Joel A Tropp, Anna C Gilbert, and Martin J Strauss, "Algorithms for simultaneous sparse approximation. part i: Greedy pursuit," *Signal Processing*, vol. 86, no. 3, pp. 572–588, 2006.
- [25] U. Gamper, P. Boesiger, and S. Kozerke, "Compressed sensing in dynamic mri," *Magnetic Resonance in Medicine*, vol. 59(2), pp. 365–373, January 2008.
- [26] H. Jung, K. H. Sung, K. S. Nayak, E. Y. Kim, and J. C. Ye, "k-t focuss: a general compressed sensing framework for high resolution dynamic mri," *Magnetic Resonance in Medicine*, 2009.
- [27] W. Lu, T. Li, I. Atkinson, and N. Vaswani, "Modified-cs-residual for recursive reconstruction of highly undersampled functional mri sequences," in *IEEE Intl. Conf. Image Proc. (ICIP)*, 2011.
- [28] J. Tropp and A. Gilbert, "Signal recovery from random measurements via orthogonal matching pursuit," *IEEE Trans. on Information Theory*, vol. 53(12), pp. 4655–4666, December 2007.
- [29] T. Blumensath and M. E. Davies, "Iterative hard thresholding for compressed sensing," *Applied and Computational Harmonic Analysis*, pp. 265–274, 2009.
- [30] T. Blumensath and M.E. Davies, "Normalised iterative hard thresholding; guaranteed stability and performance," *IEEE Journal of Selected Topics in Signal Processing*, 2010.
- [31] Michael E Tipping, "Sparse bayesian learning and the relevance vector machine," *The journal of machine learning research*, vol. 1, pp. 211–244, 2001.
- [32] N. Vaswani, "LS-CS-residual (LS-CS): Compressive Sensing on Least Squares residual," *IEEE Trans. Sig. Proc.*, vol. 58(8), pp. 4108–4120, August 2010.
- [33] E. Candes, "The restricted isometry property and its implications for compressed sensing," *Compte Rendus de l'Academie des Sciences, Paris, Serie I*, pp. 589–592, 2008.
- [34] I. Daubechies, R. DeVore, M. Fornasier, and S. Gunturk, "Iteratively re-weighted least squares minimization: Proof of faster than linear rate for sparse recovery," in *Proceedings of the 42nd IEEE Annual Conference on Information Sciences and Systems (CISS 2008)*, March 2008.
- [35] Albert Cohen, Wolfgang Dahmen, and Ronald DeVore, "Compressed sensing and best k-term approximation," *Journal of the American mathematical society*, vol. 22, no. 1, pp. 211–231, 2009.
- [36] N. Vaswani, "Kalman filtered compressed sensing," in *IEEE Intl. Conf. Image Proc. (ICIP)*, 2008.
- [37] N. Vaswani and W. Lu, "Modified-cs: Modifying compressive sensing for problems with partially known support," *IEEE Trans. Sig. Proc.*, vol. 58(9), pp. 4595–4607, September 2010.
- [38] W. Lu and N. Vaswani, "Regularized modified bpdn for noisy sparse reconstruction with partial erroneous support and signal value knowledge," *IEEE Trans. Sig. Proc.*, January 2012.
- [39] W. Lu and N. Vaswani, "Exact reconstruction conditions for regularized modified basis pursuit," *IEEE Trans. Sig. Proc.*, May 2012.
- [40] H. Guo, C. Qiu, and N. Vaswani, "An online algorithm for separating sparse and low-dimensional signal sequences from their sum," *IEEE Trans. Sig. Proc.*, August 2014.
- [41] N. Vaswani and W. Lu, "Modified-cs: Modifying compressive sensing for problems with partially known support," in *IEEE Intl. Symp. Info. Th. (ISIT)*, 2009.
- [42] N. Vaswani, "Analyzing least squares and kalman filtered compressed sensing," in *IEEE Intl. Conf. Acoustics, Speech, Sig. Proc. (ICASSP)*, 2009.
- [43] R. von Borries, C. J. Miosso, and C. Potes, "Compressed sensing using prior information," in *IEEE Intl. Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, 2007.
- [44] C. J. Miosso, R. von Borries, M. Argez, L. Valazquez, C. Quintero, and C. Potes, "Compressive sensing reconstruction with prior information by iteratively reweighted least-squares," *IEEE Trans. Sig. Proc.*, vol. 57 (6), pp. 2424–2431, June 2009.
- [45] W. Lu and N. Vaswani, "Modified bpdn for noisy compressive sensing with partially known support," in *IEEE Intl. Conf. Acoustics, Speech, Sig. Proc. (ICASSP)*, 2010.
- [46] A. S. Bandeira, K. Scheinberg, and L. N. Vicente, "On partial sparse recovery," [arXiv:1304.2809](https://arxiv.org/abs/1304.2809).
- [47] Y. Wang and W. Yin, "Sparse signal reconstruction via iterative support detection," *SIAM Journal on Imaging Sciences*, pp. 462–491, 2010.
- [48] A. Khajehnejad, W. Xu, A. Avestimehr, and B. Hassibi, "Weighted  $\ell_1$  minimization for sparse recovery with prior information," in *IEEE Intl. Symp. Info. Th. (ISIT)*, 2009.
- [49] A. Khajehnejad, W. Xu, A. Avestimehr, and B. Hassibi, "Weighted  $\ell_1$  minimization for sparse recovery with prior information," *IEEE Trans. Sig. Proc.*, 2011.
- [50] M.P. Friedlander, H. Mansour, R. Saab, and O. Yilmaz, "Recovering compressively sampled signals using partial support information," *IEEE Trans. Info. Th.*, vol. 58, no. 2, pp. 1122–1134, 2012.
- [51] V. Stankovic, L. Stankovic, and S. Cheng, "Compressive image sampling with side information," in *ICIP*, 2009.
- [52] R. Carrillo, L.F. Polania, and K. Barner, "Iterative algorithms for compressed sensing with partially known support," in *ICASSP*, 2010.
- [53] J. Zhan and N. Vaswani, "Time invariant error bounds for modified-cs based sparse signal sequence recovery," *IEEE Trans. Info. Th.*, March 2015.
- [54] E. Candes and T. Tao, "The dantzig selector: statistical estimation when p is much larger than n," *Annals of Statistics*, vol. 35 (6), pp. 2313–2351, 2007.

- [55] W. Lu and N. Vaswani, "Modified Compressive Sensing for Real-time Dynamic MR Imaging," in *IEEE Intl. Conf. Image Proc. (ICIP)*, 2009.
- [56] A. Carmi, P. Gurfil, and D. Kanevsky, "Methods for sparse signal recovery using kalman filtering with embedded pseudo-measurement norms and quasi-norms," *IEEE Trans. Sig. Proc.*, pp. 2405–2409, April 2010.
- [57] S. J. Julier and J. J. LaViola, "On kalman filtering with nonlinear equality constraints," *IEEE Trans. Sig. Proc.*, 2007.
- [58] J. Ziniel, L. C. Potter, and P. Schniter, "Tracking and smoothing of time-varying sparse signals via approximate belief propagation," in *Asilomar Conf. on Sig. Sys. Comp.*, 2010.
- [59] J. Ziniel and P. Schniter, "Dynamic compressive sensing of time-varying signals via approximate message passing," *IEEE Trans. Sig. Proc.*, Nov 2013.
- [60] D. L. Donoho, A. Maleko, and A. Montanari, "Message-passing algorithms for compressed sensing," *Proceedings of National Academy of Sciences (PNAS)*, 2009.
- [61] R. Sarkar, S. Das, and N. Vaswani, "Tracking sparse signal sequences from nonlinear/non-gaussian measurements and applications in illumination-motion tracking," in *IEEE Intl. Conf. Acoustics, Speech, Sig. Proc. (ICASSP)*, 2013.
- [62] S. Das and N. Vaswani, "Particle filtered modified compressive sensing (pafimocs) for tracking signal sequences," in *Asilomar*, 2010.
- [63] J. Filos, E. Karseras, S. Yan, and W. Dai, "Tracking dynamic sparse signals with hierarchical kalman filters: A case study," 2013.
- [64] M. Asif and Justin Romberg, "Sparse recovery of streaming signals using l1 homotopy," *IEEE Trans. Sig. Proc.*, 2014.
- [65] D. Donoho, "For most large underdetermined systems of linear equations, the minimal ell-1 norm solution is also the sparsest solution," *Comm. Pure and App. Math.*, vol. 59(6), pp. 797–829, June 2006.
- [66] S. Foucart and M. J. Lai, "Sparsest solutions of underdetermined linear systems via ell-q-minimization for  $0 \leq q \leq 1$ ," *Applied and Computational Harmonic Analysis*, vol. 26, pp. 395–407, 2009.
- [67] D. Needell and J.A. Tropp, "Cosamp: Iterative signal recovery from incomplete and inaccurate samples," *Appl. Comp. Harmonic Anal.*, vol. 26(3), pp. 301–321, May 2009.
- [68] L. Jacques, "A short note on compressed sensing with partially known signal support," *Signal Processing*, 2010.
- [69] N. Vaswani, "Stability (over time) of Modified-CS for Recursive Causal Sparse Reconstruction," in *Allerton Conf. Communication, Control, and Computing*, 2010.
- [70] J. A. Tropp, "Just relax: Convex programming methods for identifying sparse signals," *IEEE Trans. Info. Th.*, pp. 1030–1051, March 2006.
- [71] Jong Min Kim, Ok Kyun Lee, and Jong Chul Ye, "Compressive music: revisiting the link between compressive sensing and array signal processing," *Information Theory, IEEE Transactions on*, vol. 58, no. 1, pp. 278–301, 2012.
- [72] Z. Zhang and B. D. Rao, "Sparse signal recovery with temporally correlated source vectors using sparse bayesian learning," *IEEE J. Sel. Topics Sig. Proc., Special Issue on Adaptive Sparse Representation of Data and Applications in Signal and Image Processing*, vol. 5, no. 5, pp. 912–926, Sept 2011.
- [73] M. Lustig, D. Donoho, and J. M. Pauly, "Sparse mri: The application of compressed sensing for rapid mr imaging," *Magnetic Resonance in Medicine*, vol. 58(6), pp. 1182–1195, December 2007.
- [74] M. Stojnic, F. Parvaresh, and B. Hassibi, "On the reconstruction of block-sparse signals with an optimal number of measurements," in *arXiv:0804.0041*, 2008.
- [75] M. Stojnic, "Strong thresholds for l2/l1-optimization in block-sparse compressed sensing," in *IEEE Intl. Conf. Acoustics, Speech, Sig. Proc. (ICASSP)*, 2009.
- [76] R. Baraniuk, V. Cevher, M. Duarte, and C. Hegde, "Model-based compressive sensing," *IEEE Trans. Info. Th.*, March 2010.
- [77] M. Stojnic, "Block-length dependent thresholds for  $l_2/l_1$ -optimization in block-sparse compressed sensing," in *ICASSP*, 2010.
- [78] Kiryung Lee, Yoram Bresler, and Marius Junge, "Subspace methods for joint sparse recovery," *IEEE Trans. Info. Th.*, June 2012.
- [79] Yonina C Eldar, Patrick Kuppinger, and Helmut Bölcskei, "Compressed sensing of block-sparse signals: Uncertainty relations and efficient recovery," *arXiv preprint arXiv:0906.3173*, 2009.
- [80] J. Kim, O. K. Lee, and J. C. Ye, "Dynamic sparse support tracking with multiple measurement vectors using compressive music," in *IEEE Intl. Conf. Acoustics, Speech, Sig. Proc. (ICASSP)*, 2012, pp. 2717–2720.
- [81] D. Angelosante, G.B. Giannakis, and E. Grossi, "Compressed sensing of time-varying signals," in *Dig. Sig. Proc. Workshop*, 2009.
- [82] M. Aharon, M. Elad, and A. Bruckstein, "The k-svd: An algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Trans. Sig. Proc.*, 2016.
- [83] S. Ravishankar and Y. Bresler, "Learning sparsifying transforms," *IEEE Trans. Sig. Proc.*, 2013.
- [84] J. Wright and Y. Ma, "Dense error correction via l1-minimization," *IEEE Trans. Info. Th.*, vol. 56, no. 7, pp. 3540–3560, July 2010.
- [85] V. Chandrasekaran, S. Sanghavi, P. A. Parrilo, and A. S. Willsky, "Rank-sparsity incoherence for matrix decomposition," *SIAM Journal on Optimization*, vol. 21, 2011.
- [86] B. Lois and N. Vaswani, "Online matrix completion and online robust pca," in *IEEE Intl. Symp. Info. Th. (ISIT)*, 2015, longer version at arXiv:1503.03525 [cs.IT].
- [87] T. Blumensath, "Compressed sensing with nonlinear observations and related nonlinear optimisation problems," *arXiv:1205.1650*, 2012.
- [88] Weiyu Xu, Meng Wang, and Ao Tang, "Sparse recovery from nonlinear measurements with applications in bad data detection for power networks," *arXiv:1112.6234 [cs.IT]*, 2011.
- [89] L. Li and B. Jafarpour, "An iteratively reweighted algorithm for sparse reconstruction of subsurface flow properties from nonlinear dynamic data," *arXiv:0911.2270*, 2009.
- [90] T. Blumensath, "Compressed sensing with nonlinear observations and related nonlinear optimization problems," in *Tech. Rep. arXiv:1205.1650*, 2012.
- [91] A. Beck and Y. C. Eldar, "Sparsity constrained nonlinear optimization: Optimality conditions and algorithms," in *Tech. Rep. arXiv:1203.4580*, 2012.
- [92] Henrik Ohlsson, Michel Verhaegen, and S. Shankar Sastry, "Nonlinear compressive particle filtering," in *IEEE Conf. Decision and Control (CDC)*, 2013.
- [93] D. Sejdicinovic, C. Andrieu, and R. Piechocki, "Bayesian sequential compressed sensing in sparse dynamical systems," in *Allerton Conf. Communication, Control, and Computing*, 2010.
- [94] Emmanuel J. Candes, Thomas Strohmer, and Vladislav Voroninski, "Phaselift: Exact and stable signal recovery from magnitude measurements via convex programming," *Communications on Pure and Applied Mathematics*, 2011.
- [95] Henrik Ohlsson, Allen Y. Yang, Roy Dong, Michel Verhaegen, and S. Shankar Sastry, "Quadratic basis pursuit," in *SPARS*, 2012.
- [96] M. L. Moravec, J. K. Romberg, and R. G. Baraniuk, "Compressive phase retrieval," in *Wavelets XII in SPIE International Symposium on Optical Science and Technology*, 2007.
- [97] Kishore Jaganathan, Samet Oymak, and Babak Hassibi, "Recovery of sparse 1-d signals from the magnitudes of their fourier transform," in *Information Theory Proceedings (ISIT), 2012 IEEE International Symposium on*, IEEE, 2012, pp. 1473–1477.
- [98] Kishore Jaganathan, Samet Oymak, and Babak Hassibi, "Sparse phase retrieval: Convex algorithms and limitations," *arXiv preprint arXiv:1303.4128*, 2013.
- [99] Emmanuel J Candes, Yonina C Eldar, Thomas Strohmer, and Vladislav Voroninski, "Phase retrieval via matrix completion," *SIAM Journal on Imaging Sciences*, vol. 6, no. 1, pp. 199–225, 2013.
- [100] Yoav Shechtman, Amir Beck, and Yonina C Eldar, "Gespars: Efficient phase retrieval of sparse signals," *arXiv preprint arXiv:1301.1018*, 2013.
- [101] M. S. Asif and J. Romberg, "Dynamic updating for sparse time varying signals," in *Conf. Info. Sciences and Systems*, 2009.
- [102] M. S. Asif and J. Romberg, "Dynamic Updating for  $l_1$  minimization," *IEEE J. Selected Topics in Signal Processing*, vol. 4(2), April 2010.
- [103] Mark A Davenport, Petros T Boufounos, Michael B Wakin, and Richard G Baraniuk, "Signal processing with compressive measurements," *IEEE Journal of Selected Topics in Signal Processing*, pp. 445–460, April 2010.
- [104] C. Hegde, P. Indyk, and L. Schmidt, "Approximation-tolerant model-based compressive sensing," in *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2014.
- [105] L. Schmidt, C. Hegde, and P. Indyk, "The constrained earth mover distance model, with applications to compressive sensing," in *CAMSAP*, 2013.
- [106] H. Mansour and R. Saab, "Weighted one-norm minimization with inaccurate support estimates: Sharp analysis via the null-space property," in *IEEE Intl. Conf. Acoustics, Speech, Sig. Proc. (ICASSP)*, 2015.