AKRON: An Algorithm for Approximating Sparse Kernel Reconstruction

Gregory Ditzler

Dept. of Electrical and Computer Engineering
The University of Arizona
Tucson, AZ 85721 USA
ditzler@email.arizona.edu

Nidhal Carla Bouaynaya

Dept. of Electrical and Computer Engineering
Rowan University
Glassboro, NJ 08028 USA
bouaynaya@rowan.edu

Roman Shterenberg

Dept. of Mathematics
University of Alabama at Birmingham
Birmingham, AL 35294 USA
shterenb@uab.edu

Abstract

Exact reconstruction of a sparse signal for an under-determined linear system using the ℓ_0-measure is, in general, an NP-hard problem. The most popular approach is to relax the ℓ_0-optimization problem to an ℓ_1-approximation. However, the strength of this convex approximation relies upon rigid properties on the system, which are not verifiable in practice. Greedy algorithms have been proposed in the past to speed up the optimization of the ℓ_1 problem, but their computational efficiency comes at the expense of a larger error. In an effort to control error and complexity, this paper goes beyond the ℓ_1-approximation by growing neighborhoods of the ℓ_1-solution that moves towards the optimal solution. The size of the neighborhood is tunable depending on the computational resources. The proposed algorithm, termed Approximate Kernel RecONstruction (AKRON), yields significantly smaller errors than current greedy methods with a controllable computational cost. By construction, the error of AKRON is smaller than or to equal the ℓ_1-solution. AKRON enjoys all the error bounds of ℓ_1 under the restricted isometry property condition. We benchmarked AKRON on simulated data from several under-determined systems, and the results show that AKRON can significantly improve the reconstruction error with slightly more computational cost than solving the ℓ_1 problem directly.

1. Introduction

Many engineering problems are formulated as inverse problems, which is where the number of parameters (p) greatly exceeds the number of measurements (n) available. Examples include:

Preprint submitted to Elsevier *September 14, 2017*

source estimation of electroencephalographic (EEG) and magnetoencephalographic (MEG) data [1, 2], reverse-engineering of genetic regulatory networks from high-throughput gene expression data [3, 4], magnetic resonance imaging [5], information theory and communication engineering [6], and electromagnetics and antenna design [7]. These inverse problems, known as "large p small n", pose a challenge, because of the non-identifiability of a solution. Additional constraints or prior knowledge are needed to solve such under-determined systems. In many cases, such as inference of genetic regulatory networks [3, 4], we are interested in the sparsest solution. The objective is then to recover the sparsest signal, $\mathbf{x} \in \mathbb{C}^p$, from a measurement matrix, $\mathbf{A} \in \mathbb{C}^{n \times p}$, and observed vector $\mathbf{y} \in \mathbb{C}^n$ such that $\mathbf{y} = \mathbf{A}\mathbf{x}$, where $n \ll p$. In a noisy setting, the problem is formulated as $\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{e}$, where $\mathbf{e}$ is a vector of measurement noise with a bounded variance, i.e., $\|\mathbf{e}\|_2 \leq \epsilon$. Without loss of generality, it is assumed that $\mathbf{A}$ is full-rank; otherwise, the observations would be redundant.

Finding the sparsest solution amounts to solving the following optimization problem:

$$\mathbf{x}^* = \text{argmin} \left\{ \|\mathbf{x}\|_0 : \mathbf{A}\mathbf{x} = \mathbf{y} \right\}, \tag{1}$$

where $\|\mathbf{x}\|_0$ denotes the ℓ_0-measure of vector $\mathbf{x}$, i.e., the number of non-zero elements of $\mathbf{x}$. Observe that ℓ_0 is not a proper norm and that is why we refer to it as a "measure" although, by abuse of notation, we may also write ℓ_0-norm. Unfortunately, (1) is in general an NP-hard combinatorial problem since it involves finding the number *and* positions of the zeros in a p-dimensional space [8]. The field of compressive sensing (CS) addresses this problem by solving the under-determined system with a unique sparsest solution under specific conditions on the system. The ℓ_0-norm objective in (1) can be relaxed to the ℓ_1-norm, solving the following convex optimization problem:

$$\widehat{\mathbf{x}}_1 = \text{argmin} \left\{ \|\mathbf{x}\|_1 : \mathbf{A}\mathbf{x} = \mathbf{y} \right\}. \tag{2}$$

This convex relaxation makes the problem more tractable; however, in general, the solutions of (1) and (2) are not equivalent. CS theory has shown that, if $\mathbf{A}$ satisfies the null space property (NSP) or the restricted isometry property (RIP), then the ℓ_1 problem yields the optimal ℓ_0 solution [8]. Unfortunately, these conditions are not verifiable in practice. In particular, one cannot check if the obtained ℓ_1-solution is the sparsest solution or not! Through examples and simulations, we show that, in general, the ℓ_1-solution may be far from the ℓ_0-optimal solution. Hence, it is crucial to develop greedy algorithms that achieve a balance between computational complexity and reconstruction error.

2. Related Work

Recent efforts focused on greedy algorithms to infer a sparse solution. In particular, a family of *Hard Thresholding* (HT) algorithms have been suggested in [9], which makes an initial guess for the support of $\mathbf{x}$ and then projects the measurements $\mathbf{y}$ onto this support. An iterative version called *Iterative Hard Thresholding* (IHT) updates the residual and estimates a new $\mathbf{x}$ at every iteration until the stopping criterion is satisfied. Another version of lower complexity per iteration, called *Matching Pursuit* (MP), has been suggested. The *Orthogonal Matching Pursuit* (OMP) [10] is an iterative greedy algorithm that selects at each step the column which is most correlated with the current residuals, and estimates the nonzero entries in the vector $\mathbf{x}$ with a computational complexity $\mathrm{O}(k \log p)$. OMP's computational improvements, however, come at the cost of increased reconstruction error. *Compressive Sampling Matching Pursuit* (CoSaMP) [10] combines the approaches of OMP and HT in a two-stage greedy algorithm that aims to improve the reconstruction error of OMP. Unfortunately, these methods sacrifice accuracy of the reconstruction for the runtime as they approximate the ℓ_0-norm by other cost functions. Recently, SL0, or *smoothed* ℓ_0, has been proposed as a fast algorithm to directly approximate the ℓ_0 solution [11]. Candés *et al.* proposed an iterative re-weighted ℓ_1 minimization algorithm that has theoretical guarantees that is can improve the ℓ_1 solution [12].

In our previous work, we presented *Kernel RecONstruction* (KRON), a greedy algorithm, that achieves an exact solution to (1), without exhaustively searching $\mathbb{C}^p$ [13]. In KRON, finding the sparsest solution amounts to solving $\binom{p}{s=p-n}$ linear equations. All $\binom{p}{s}$ potential solutions have at least s zeros. The sparsest solution is guaranteed to be one of them. The computational complexity of KRON is $\mathcal{O}(p^s)$. KRON yields the optimal sparsest solution (zero reconstruction error) at a high computational cost.

Against this background, we seek to develop an approach for approximating (1) yielding reconstruction errors lower than ℓ_1-norm, and other approaches such as OMP and CoSaMP, and at the same time having comparable, or at least controllable, computational cost.

3. Approximate Kernel Reconstruction

3.1. Central Idea behind AKRON

In this section, we motivate the central idea behind AKRON, given general linear algebra knowledge about the under-determined system. First, we know that the system $\mathbf{Ax} = \mathbf{y}$ always admits solutions with $s = (p - n)$ zeros because s is the dimension of the Kernel of $\mathbf{A}$; hence the name Kernel RecONstruction (KRON) in [13]. KRON distributes s zeros among the p entries then searches for all the solutions with exactly s zeros. The sparsest solution is guaranteed to be among these $\binom{p}{s}$ solutions. However, we do not know in advance which one it will be. KRON tries all possible $\binom{p}{s}$ solutions then chooses the sparsest. Notice that no conditions are imposed on the matrix $\mathbf{A}$; that is, KRON recovers the optimal sparsest solution whether the RIP condition is satisfied or not. The central issue with KRON is that it becomes computationally prohibitive when p is large *and* s in the order of $\frac{p}{2}$. Therefore, we propose AKRON to reduce the number of enumerations that need to be performed in KRON. To achieve this, the central idea behind AKRON is to use the standard ℓ_1-approximation to "guess" the locations of the s zeros that will result in the sparsest solution. Finding s correct zero locations is *sufficient* to find the optimal sparsest solution. This idea can also be viewed as *a "perturbation" of the ℓ_1-approximation to make it closer to the ℓ_0-norm*. Formally, we define a δ-neighborhood of the ℓ_1-approximation that allows AKRON to find sparser solutions and reduce the reconstruction error. The size of the neighborhood is tunable depending on the computational power available, and vary from 0 (ℓ_1-approximation) to n (KRON, i.e., perfect reconstruction). In particular, when the ℓ_1-approximation is optimal (RIP conditions satisfied), AKRON is also optimal, but when the ℓ_1-solution is suboptimal, AKRON results in a better (i.e., sparser) solution with smaller recovery error.

3.2. The Noiseless Case

AKRON begins by solving the ℓ_1 convex optimization problem in (2). Denote the solution by $\widehat{\mathbf{x}}_1$. In general, $\widehat{\mathbf{x}}_1$ is different from the desired ℓ_0-solution. However, since ℓ_1 is the closest convex norm to ℓ_0, we can use $\widehat{\mathbf{x}}_1$ to find the locations of s zeros, which would correspond to the s-smallest magnitudes in $\widehat{\mathbf{x}}_1$. The central idea behind AKRON's 0-neighborhood solution is as follows: (1) find the indices ($\mathcal{Q}$) with the s-smallest magnitudes of the ℓ_1 solution, (2) set these indices to zero, then (3) re-solve the system $\mathbf{A}\mathbf{x} = \mathbf{y}$. Call this solution $\widehat{\mathbf{x}}^*_{\delta=0}$. The following proposition bounds the error between the ℓ_1-solution and the $(\delta = 0)$-neighborhood solution $\widehat{\mathbf{x}}^*_{\delta=0}$.

Proposition 1. Let $\widehat{\mathbf{x}}_1$ denote the ℓ_1-solution of the under-determined problem in (2). Without loss of generality, we assume that $\mathbf{A} \in \mathbb{C}^{n\times p}$ is full-rank, and call $s = p-n$. Let $\{|\widehat{\mathbf{x}}_1(i_1)|, \ldots, |\widehat{\mathbf{x}}_1(i_s)|\}$ be the set of the s-smallest magnitudes of $\widehat{\mathbf{x}}_1$. Then, we have

$$\|\widehat{\mathbf{x}}_1 - \widehat{\mathbf{x}}^*_{\delta=0}\|_2 \leq \sqrt{s}\, C_A \max\left\{|\widehat{\mathbf{x}}_1(i_1)|, \cdots, |\widehat{\mathbf{x}}_1(i_s)|\right\}, \tag{3}$$

where C_A is a constant that depends only on the matrix $\mathbf{A}$: $C_A = (1 + \|\mathbf{A}_Q^{-1}\|_2 \|\mathbf{A}_{Q^\perp}\|_2)$, where $\mathbf{A}_Q$ is the $(n \times n)$ sub-matrix of $\mathbf{A}$ obtained by removing the s columns indexed by $\{i_1, \cdots, i_s\}$, and $\mathbf{A}_{Q^\perp} \in \mathbb{C}^{n\times s}$ is the complement matrix, i.e., the matrix that contains only the columns corresponding to these s-smallest elements.

Proof. Denote by $\mathbf{A}_Q \in \mathbb{C}^{n\times n}$ the reduced matrix, where the columns corresponding to the indices of the s-smallest elements in $\widehat{\mathbf{x}}_1$ were removed. Notice that $\mathbf{A}_Q$ is invertible because $\mathbf{A}$ is full-rank. Let $\mathbf{A}_{Q^\perp} \in \mathbb{C}^{n\times s}$ be the complement matrix, i.e., the matrix that contains only the columns corresponding to the s-smallest elements $\{i_1, \cdots, i_s\}$. We adopt similar notations for $\widehat{\mathbf{x}}_{1_Q} \in \mathbb{C}^{n\times 1}$ and $\widehat{\mathbf{x}}_{1_{Q^\perp}} \in \mathbb{C}^{s\times 1}$. We have

$$\begin{aligned} \mathbf{A}\widehat{\mathbf{x}}_1 \quad &= \quad \mathbf{A}\widehat{\mathbf{x}}^*_{\delta=0} = \mathbf{y} && (4)\\ &\Longleftrightarrow \\ \mathbf{A}_Q\widehat{\mathbf{x}}_{1_Q} + \mathbf{A}_{Q^\perp}\widehat{\mathbf{x}}_{1_{Q^\perp}} \quad &= \quad \mathbf{A}\widehat{\mathbf{x}}^*_{\delta=0}. && (5) \end{aligned}$$

Observe that since, by construction, $\widehat{\mathbf{x}}^*_{\delta=0_{Q^\perp}} = \mathbf{0}$, we have that

$$\mathbf{A}_Q\widehat{\mathbf{x}}^*_{\delta=0_Q} = \mathbf{A}\widehat{\mathbf{x}}^*_{\delta=0}. \tag{6}$$

From Eqs. (5) and (6), we have

$$\mathbf{A}_Q(\widehat{\mathbf{x}}^*_{\delta=0_Q} - \widehat{\mathbf{x}}_{1_Q}) = \mathbf{A}_{Q^\perp}\, \widehat{\mathbf{x}}_{1_{Q^\perp}}.$$

Therefore,

$$\widehat{\mathbf{x}}^*_{\delta=0_Q} - \widehat{\mathbf{x}}_{1_Q} = \mathbf{A}_Q^{-1}\mathbf{A}_{Q^\perp}\, \widehat{\mathbf{x}}_{1_{Q^\perp}}.$$

Using norm inequalities, we obtain

$$\|\widehat{\mathbf{x}}^*_{\delta=0_Q} - \widehat{\mathbf{x}}_{1_Q}\|_2 \leq \|\mathbf{A}_Q^{-1}\|_2\|\mathbf{A}_{Q^\perp}\|_2\, \|\widehat{\mathbf{x}}_{1_{Q^\perp}}\|_2. \tag{7}$$

On the other hand, we have by triangle inequality

$$\|\widehat{\mathbf{x}}_1 - \widehat{\mathbf{x}}^*_{\delta=0}\|_2 \leq \|\widehat{\mathbf{x}}_{1_Q} - \widehat{\mathbf{x}}^*_{\delta=0_Q}\|_2 + \|\widehat{\mathbf{x}}_{1_{Q^\perp}} - \widehat{\mathbf{x}}^*_{\delta=0_{Q^\perp}}\|_2.$$

But, by construction, $\widehat{\mathbf{x}}^*_{\delta=0_{Q^\perp}} = \mathbf{0}$, hence

$$\begin{aligned} \|\widehat{\mathbf{x}}_1 - \widehat{\mathbf{x}}^*_{\delta=0}\|_2 &\leq \|\widehat{\mathbf{x}}_{1_Q} - \widehat{\mathbf{x}}^*_{\delta=0_Q}\|_2 + \|\widehat{\mathbf{x}}_{1_{Q^\perp}}\|_2 \\ &\leq \|\widehat{\mathbf{x}}_{1_{Q^\perp}}\|_2 \, \left(1 + \|\mathbf{A}_Q^{-1}\|_2 \|\mathbf{A}_{Q^\perp}\|_2\right) \qquad (8) \end{aligned}$$

where the inequality in (8) follows from (7). By using the fact that

$$\|\widehat{\mathbf{x}}_{1_{Q^\perp}}\|_2 \leq \sqrt{s} \max\left\{|\widehat{\mathbf{x}}_1(i_1)|, \cdots, |\widehat{\mathbf{x}}_1(i_s)|\right\},$$

we obtain the desired result with $C_A = (1 + \|\mathbf{A}_Q^{-1}\| \|\mathbf{A}_{Q^\perp}\|)$. □

The interpretation of the bound in Proposition 1 is quite intuitive: if the s-smallest elements of the ℓ_1-approximation are all "small", then $\widehat{\mathbf{x}}^*_{\delta=0}$ will be close to the ℓ_1-solution. However, if at least one of these s-smallest elements is "large", then the obtained $\widehat{\mathbf{x}}^*_{\delta=0}$ will be "far" from the ℓ_1-approximation. In other words, the s-smallest elements of the ℓ_1-approximation are all "small" if ℓ_1 is a "good" approximation of ℓ_0; In that case, the AKRON solution $\widehat{\mathbf{x}}^*_{\delta=0}$ will be close to this "good" ℓ_1 solution. On the other hand, having a large element among the s-smallest indices indicates that ℓ_1-solution is not a "good" estimate of the optimal sparsest solution, and hence AKRON solution will be sparser and further from ℓ_1 solution.

The following theorem derives an error bound of the AKRON recovery. We first need to recall the restricted isometry property (RIP). The restricted isometry constant α_k of a matrix $\mathbf{A} \in \mathbb{C}^{n \times p}$ is the smallest number such that

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

for all k-sparse $\mathbf{x}$. A matrix $\mathbf{A}$ is said to satisfy the RIP of order k with constant α_k if $\alpha_k \in (0, 1)$.

Theorem 1. Assume that $\mathbf{A} \in \mathbb{C}^{n \times p}$ satisfies the RIP of order $3k$ with $\alpha_{3k} < \frac{1}{3}$, and let $s = p - n$. Let $|\widehat{\mathbf{x}}_1(i_1)|, \cdots, |\widehat{\mathbf{x}}_1(i_s)|$ be the s-smallest magnitude elements of the ℓ_1-solution $\widehat{\mathbf{x}}_1$. Then, letting $\mathbf{x}^*$ be the optimal ℓ_0-solution, we have

$$\|\mathbf{x}^* - \widehat{\mathbf{x}}^*_{\delta=0}\|_2 \leq \sqrt{s}\, C_A \max\left\{|\widehat{\mathbf{x}}_1(i_1)|, \cdots, |\widehat{\mathbf{x}}_1(i_s)|\right\} + C\frac{\sigma_k(\mathbf{x}^*)_1}{\sqrt{k}}, \qquad (9)$$

where $\sigma_k(\mathbf{x})_1 = \inf_{\mathbf{z} \in \{\mathbf{z} \in \mathbb{C}^p : \|\mathbf{z}\|_0 \leq k\}} \|\mathbf{z} - \mathbf{x}\|_1$ is the best k-term approximation error of the vector $\mathbf{x}$ in ℓ_1, C is a constant that depends on α_{3k} [8], and C_A is a constant that depends on the matrix $\mathbf{A}$ as defined in Proposition 1.

Proof. From [8], we have, under the RIP condition, for all $\mathbf{x}$,

$$\|\mathbf{x} - \widehat{\mathbf{x}}_1\|_2 \leq C\frac{\sigma_k(\mathbf{x})_1}{\sqrt{k}}, \qquad (10)$$

with $C = \frac{2}{1-\gamma}\left(\frac{\gamma+1}{\sqrt{2}}+\gamma\right), \gamma = \sqrt{\frac{1+\alpha_{3k}}{2(1-\alpha_{3k})}}$ [8]. From the triangle inequality, we have

$$\begin{aligned} \|\mathbf{x}^* - \widehat{\mathbf{x}}^*_{\delta=0}\|_2 &\leq \|\widehat{\mathbf{x}}^*_{\delta=0} - \widehat{\mathbf{x}}_1\|_2 + \|\mathbf{x}^* - \widehat{\mathbf{x}}_1\|_2 \\ &\leq \sqrt{s}\, C_A \max\left\{|\widehat{\mathbf{x}}_1(i_1)|, \cdots, |\widehat{\mathbf{x}}_1(i_s)|\right\} + C\frac{\sigma_k(\mathbf{x}^*)_1}{\sqrt{k}}, \end{aligned} \tag{11}$$

where, in (11), the first inequality follows from Proposition 1 and the second inequality follows from (10) in [8]. □

δ-neighborhoods of ℓ_1 in quest of sparser solutions: The solution given by $\widehat{\mathbf{x}}^*_{\delta=0}$ is sparser than the ℓ_1-approximation; however, $\widehat{\mathbf{x}}^*_{\delta=0}$ may still be far from the optimal ℓ_0 solution in (1). We can improve upon $\widehat{\mathbf{x}}^*_{\delta=0}$ by finding sparser solutions that increase "the neighborhood of the ℓ_1-approximation" as follows: consider the $(s+\delta)$-smallest elements of $\widehat{\mathbf{x}}_1$, where the "true" s zeros, that lead to the optimal sparsest solution, may be located. Then, consider all possible $\binom{s+\delta}{s}$ combinations of setting s elements to zero. For each of these s zero locations, AKRON re-solves the system. The sparsity of the solution is recorded for each combination and the sparsest solution, $\widehat{\mathbf{x}}^*_\delta$, is returned by AKRON. It is important to note that AKRON is highly parallelizable once the $\binom{s+\delta}{s}$ combinations are known.

We state that $\widehat{\boldsymbol{x}}^*_\delta$ *is the optimal solution within the δ-neighborhood of the ℓ_1-approximation*. If the ℓ_1-approximation is close to the ℓ_0-solution, then we can find the optimal solution within a small δ-neighborhood. To observe this result, let $\mathbf{x}^*$ be the optimal ℓ_0-solution, and assume it is unique and has $K > s$ zeros and that $\widehat{\mathbf{x}}_1$ is close to $\mathbf{x}^*$. These assumptions imply that the corresponding K entries of $\widehat{\mathbf{x}}_1$ are close to zero. By choosing $\delta = K - s$, AKRON will surely find the optimal solution. It may appear that if $K \gg s$, then $\delta \gg 1$ and significant computational resources are needed. However, if $\widehat{\mathbf{x}}_1$ is close to $\mathbf{x}^*$, then even putting the smallest s elements to zero will lead to the exact solution by uniqueness. Therefore, we suggest a step-wise approach. First, consider $\delta = 0$. We may find the optimal sparsest solution (i.e., number of zeros $\gg s$) right away by uniqueness of the sparsest solution and the fact that ℓ_1-approximation may be close (for the specific system at hand) to the optimal ℓ_0-solution. Otherwise, we increase $\delta \in [1, n]$ depending on the available computational power. The entire process can be repeated for growing values of $\delta = 1, \ldots, \delta_{\max} \leq n$, where $\delta_{\max}$ is set depending on the computational power available or until we reach a sufficiently sparse solution. Note that computational complexity of AKRON is of the order of $\mathcal{O}(s^\delta)$, and when $\delta = n$, AKRON reduces to the perfect reconstruction KRON algorithm in [13].

State-of-the-art approaches, such as OMP and CoSaMP, require that the sparsity level be specified in advance, which is, in general, impossible to correctly guess. AKRON, on the other hand, starts with an educated initial estimate of the level of sparsity, given by the s smallest elements in the ℓ_1 solution. This initial estimate has well-known theoretical properties in the compressive sensing literature [8]. We derived a theoretical bound between AKRON (with $\delta = 0$) and the optimal solution. We then improve upon this estimate by exploring higher δ-neighborhoods in quest of sparser solutions. In particular, the parameter δ controls the tradeoff between sparsity and computational complexity. The sensitivity of this free parameter δ is evaluated in Section 4. The pseudo code for AKRON is detailed in Algorithm 1.

Example 1: To understand the AKRON algorithm and illustrate the importance of the δ-neighborhoods, we present a simple numerical example. Consider the following randomly generated noiseless system

$$\mathbf{A} = \begin{pmatrix} -0.4588 & 1.5977 & -0.8724 & -0.1121 & -1.3068 \\ 0.2942 & 3.0954 & -1.0530 & 0.3454 & 1.5257 \\ -0.1948 & -0.7558 & -0.9756 & 0.1549 & 0.9586 \end{pmatrix};$$

$$\mathbf{y} = \begin{pmatrix} -1.2316 & 1.1739 & 0.8135 \end{pmatrix}^T$$

The optimal ℓ_0-norm sparsest solution is given by

$$\mathbf{x}^* = \begin{pmatrix} 0 & 0 & 0 & -1.2372 & 1.04858 \end{pmatrix}^T \tag{12}$$

The ℓ_1 solution, which solves (2), is given by

$$\widehat{\mathbf{x}}_1 = \begin{pmatrix} 0.0 & -0.034 & 0.047 & 0.0 & 0.870 \end{pmatrix}^T$$

Clearly, the ℓ_1-solution is not as sparse as the optimal solution and has incorrect zero locations. We have $n = 3, p = 5$ and thus $s = 2$. If we choose $\delta = 1$ the AKRON considers the $s+\delta = 3$-smallest magnitudes of $\widehat{\mathbf{x}}_1$, which are located at indices 1, 2 and 4. We set $s = 2$ locations to zero among these 3 indices. We consider all $\binom{s+\delta}{s} = \binom{3}{2} = 3$ combinations of two zeros in indices 1, 2 and 4 of $\widehat{\mathbf{x}}_1$. The combination of indices 1 and 2 set to zero leads to the sparsest optimal solution $\mathbf{x}^*$ in (12). Thus, in this case, the ℓ_1-norm solution is sub-optimal; but by considering a $\delta = 1$-neighborhood of this approximation, AKRON is able to exactly recover the sparsest optimal ℓ_0-solution.

3.3. The Noisy Case

AKRON can be reformulated to cope with noise (AKRONoi) in a system given by $\mathbf{y} = \mathbf{A}\mathbf{x}+\mathbf{e}$, where $\mathbf{e}$ is the noise vector. The noisy case is formulated as the following optimization problem

$$\mathbf{x}^*_\epsilon = \text{argmin}\{\|\mathbf{x}\|_0 : \|\mathbf{A}\mathbf{x} - \mathbf{y}\|_2 \leq \epsilon\}, \tag{13}$$

where ϵ is the fixed error bound. In the noisy case, the sparsest solution can be sparser than the solution of the homologous noiseless system. Intuitively, the noise allows a “margin of error” where more entries can be set to zero as long as the constraint in (13) is satisfied.

Following this logic of ‘noise enhances sparsity’, one may start by solving the noiseless case using the algorithm in Section 3.2, and then further set more elements to zero while ensuring the constraint in (13) is satisfied. This approach is promising if the corresponding noiseless system $\mathbf{A}\mathbf{x} = \mathbf{y}$ admits a sparse solution, i.e., with a number of zeros $K \gg s$. However, in general, the pair $(\mathbf{A}, \mathbf{y})$, corresponding to the noisy system in (13), may not admit sparse solutions to the noiseless homologue $\mathbf{A}\mathbf{x} = \mathbf{y}$. In this case, the ℓ_1-approximation of the noiseless system will be very far from the optimal ℓ_0 solution, and small perturbations (small δ) of the ℓ_1-solution may not be sufficient to approximate the optimal solution with desired accuracy.

Instead, consider the corresponding ℓ_1 approximation of the noisy system $\widehat{\mathbf{x}}_\epsilon = \text{argmin}\{\|\mathbf{x}\|_1 : \|\mathbf{A}\mathbf{x}-\mathbf{y}\|_2 \leq \epsilon\}$. AKRONoi starts from $\widehat{\mathbf{x}}_\epsilon$ and considers a δ-neighborhood to obtain the locations of the $s + \delta$ smallest magnitude elements of $\widehat{\mathbf{x}}_\epsilon$. For all possible combinations of s zeros among these $s + \delta$ locations, we obtain the corresponding solutions to the noisy system with constraint $\|\mathbf{A}\mathbf{x} - \mathbf{y}\|_2 \leq \epsilon$. Call these solutions $\widehat{\mathbf{x}}^q_\delta$, where $q = 1, \cdots, \binom{s+\delta}{s}$. We can further make these exact solutions sparser by putting “small” elements to zero until the constraint $\|\mathbf{A}\mathbf{x} - \mathbf{y}\|_2 \leq \epsilon$ is satisfied. This can be done, for instance, by putting to zero all elements in $\widehat{\mathbf{x}}_\delta^q$ such that $|\widehat{\mathbf{x}}_\delta^q(i)| \leq \frac{\epsilon}{\sqrt{p}\|\mathbf{A}\|_F}$, where $\|\mathbf{A}\|_F$ is the Frobenius norm of $\mathbf{A}$. A more direct approach would be to put the smallest elements to zero one at a time then check if the constraint in (13) is satisfied. On the other hand, AKRONoi is not guaranteed to lead to the exact optimal ℓ_0 solution for large neighborhoods because the exact value of $\mathbf{y}$ for which one attains the sparsest solution is not known. The following numerical example illustrates the premise behind AKRONoi.

Example 2: Let $\mathbf{A}$ and $\mathbf{x}^*$ be as provided in Example 1, and consider the noisy system $\mathbf{y} = \mathbf{A}\mathbf{x}^* + \mathbf{e}$, where $\mathbf{e} = (-0.011, 0.017, 0.028)^\mathsf{T}$ is a sample noise vector that was randomly generated from a Gaussian distribution. Observe that the power of the noise is about 21% the power of the signal. The starting point for AKRONoi is the ℓ_1 solution of the noisy system, which is given by $\widehat{\mathbf{x}}_\epsilon = (0, -0.046, 0, 0, 0.752)^\mathsf{T}$, where ϵ is the norm of the noise vector $\mathbf{e}$. Clearly this solution is far from the optimal solution in terms of zero locations and norm. AKRONoi with $\delta = 2$ yields the solution $(0, 0, 0, -1.0034, 1.023)^\mathsf{T}$. Although this solution is not exact, AKRONoi is able to correctly recover the locations of the non-zero entries in $\mathbf{x}^*$. Furthermore, the reconstruction error of AKRONoi is significantly lower than that of noisy ℓ_1. The following section demonstrates the effectiveness of AKRON and AKRONoi on a variety of different under-determined linear systems.

4. Experiments

AKRON was benchmarked against OMP [14], CoSaMP [10], SL0 [11], Iterative Re-weighted ℓ_1 Minimization (IRLM) [12], and the ℓ_1-norm using CVX [15]. The data sets are designed as follows: $\mathbf{A}$'s entries are randomly sampled and $\mathbf{x}$ has k indices sampled from $\mathcal{N}(0, 1)$, but all other entries are set to zero. The matrix $\mathbf{A}$ is full rank. We assess the performance of the different algorithms using two criteria: The reconstruction error defined as $\mathrm{err}(\mathbf{x}) = \|\mathbf{x}^* - \mathbf{x}\|_2 / \|\mathbf{x}^*\|_2$, where $\mathbf{x}$ is the solution from an algorithm, and $\mathbf{x}^*$ is the optimal solution. However, this error does not always capture the correct location of zeros. For instance, assume that $\mathbf{x}^* = [0, 0, 10]^t$ and the solutions to two different algorithms are given by $\mathbf{x}_1 = [5, 5, 5]^t$ and $\mathbf{x}_2 = [0, 0, 20]^t$. Then, we have $\mathrm{err}(\mathbf{x}_1) < \mathrm{err}(\mathbf{x}_2)$ although, intuitively, $\mathbf{x}_2$ is "closer" to the desired solution. To capture the similarity of the zero locations, we use the Jaccard index computed as $\frac{|Z^* \cap Z|}{|Z^* \cup Z|}$, where Z^* and Z are the set of indices of the non-zero entries in the true solution and those returned by the algorithm, respectively, and $|\cdot|$ denotes the cardinality of a set. The sparsity level parameter for OMP and CoSaMP is set to k, which should give them an advantage at error evaluation. All results presented were averaged over 150 simulations of linear systems. Finally, reproducibles for these experiments are publicly available[1].

4.1. *Experiment #1: Fix p Sweep n*

The first simulation examines the impact of the number of measurements n with fixed dimension $p = 20$ and fixed sparsity $k = 8$ (here k denotes the number of nonzero elements). Intuitively, the reconstruction errors should decrease as n increases. Figures 1(a)-1(d) show the reconstruction errors and Jaccard stability indices for the noiseless and noisy cases. First, we observe that the reconstruction error of KRON drops to zero once $n = 10$, which is exactly what is expected, as KRON leads perfect reconstruction for $n > k + 1$ [13]. Among all the other approaches, AKRON,

[1] http://github.com/gditzler/AKRON/

with $\delta = 3$ in these experiments, provides the smallest error and largest Jaccard index. AKRON is even able to outperform SL0 and IRLM. Furthermore, AKRON outperforms the other algorithms in the noisy scenarios as shown in Figures 1(c) and 1(d).

4.2. Experiment #2: Sweep p fix n

The second simulation shows the impact, on the reconstruction error, when the dimension p varies from $50, \ldots, 250$ and $k = 0.05p$, corresponding to a fixed 95% sparsity level. The reconstruction errors (Figure 2), and Jaccard indices (Figure 3) are computed for each of the different values of n. AKRON used a neighborhood of $\delta = 3$. The sparsity threshold for OMP and CoSaMP was set to 9. When $n \ll p$ (i.e., $n = 0.1p, 0.2p$), the ℓ_1 error is large and AKRON, although with a small neighborhood, is able to improve upon the ℓ_1 approximation and provide nearly zero reconstruction error with high zero-location stability (for $n \geq 0.2p$). Furthermore, not only does AKRON lead to a low reconstruction error, but it is also able to identify the locations of the zeros as shown in the Jaccard stability figures. When n increases (here $n = 0.3p, 0.4p$), the ℓ_1-approximation is quite good in the sense that the ℓ_1 solution is very close to the optimal sparsest solution (both in terms of reconstruction error and stability). AKRON has a similar performance as ℓ_1, but with an even smaller reconstruction error. It is also worth noting that the performance of CoSaMP also improved as n increases.

4.3. Runtime Analysis

The computational run-times for all algorithms in the different scenario cases are shown in Figure 4. In this simulation, p is fixed at 200 and the value of n is swept from 10 to 70. As expected, the computational complexity of AKRON decreases because s becomes smaller, which decreases the number of combinations that need to be evaluated. Finally, the run-times can be improved further by distributing the `for` loop in Figure 1 across more cores. In our ongoing work, we are looking into GPU implementation of the combinations involved in AKRON to further reduce the computational time.

5. Conclusions

In this paper, we presented AKRON and AKRONoi to address the issue of obtaining a trade-off between reconstruction error and computational complexity for recovering sparse signals from under-determined linear systems. The ℓ_1-norm solution may have a large or small error depending on the algebraic properties of the system at hand. Greedy algorithms, such as OMP and CoSaMP, run quickly but lead to a high error. AKRON starts with the ℓ_1-approximation and builds δ-neighborhoods, where a sparser solution can be found. These δ-neighborhoods can grow until reaching the optimal sparsest ℓ_0-solution. δ is tunable depending on the available computational resources. Our simulation results showed that, for $n \ll p$, the ℓ_1-approximation can have a high error and AKRON can bring this error down substantially with a comparable run-time (in seconds). The user can control the parameter δ depending on a tradeoff between the desired sparsity/optimality and the computational power available.

Acknowledgements

This work was supported by the National Science Foundation under Award Numbers CCF-1527822 and ACI-1429467.

6. References

[1] P. Georgieva, N. Bouaynaya, F. Silva, L. Mihaylova, L. Jain, A beamformer-particle filter framework for localization of correlated eeg sources, IEEE Journal of Biomedical and Health Informatics (2015).

[2] G. Ditzler, J. C. Morrison, Y. Lan, G. Rosen, Fizzy: Feature selection for metagenomics, BMC Bioinformatics 16 (2015).

[3] J. Khan, N. Bouaynaya, H. Fathallah-Shaykh, Tracking of time-varying genomic regulatory networks with a LASSO-kalman smoother, EURASIP Journal on Bioinformatics and Systems Biology 2014 (2014).

[4] B. Bayar, N. Bouaynaya, R. Shterenberg, SMURC: High-dimension small-sample multivariate regression with covariance estimation, IEEE Journal of Biomedical and Health Informatics (2016). To appear.

[5] E. J. Candès, J. K. Romberg, T. Tao, Stable signal recovery from incomplete and inaccurate measurements, Communications on Pure and Applied Mathematics 59 (2006) 1207–1223.

[6] J. Meng, W. Yin, Y. Li, N. Nguyen, Z. Han, Compressive sensing based high-resolution channel estimation for ofdm system, IEEE Journal on Selected Topics in Signal Processing 6 (2012) 15–25.

[7] A. Massa, P. Rocca, G. Oliveri, Compressive sensing in electromagnetics - a review, IEEE Antennas and Propagation Magazine 57 (2015) 224–238.

[8] M. Fornasier, H. Rauhut, Compressive Sensing, Handbook of Mathematical Methods in Imaging, 2011.

[9] T. Blumensath, M. E. Davis, Iterative hard thresholding for compressed sensing, Applied and Computational Harmonic Analysis 27 (2009) 265–274.

[10] D. Needell, J. A. Tropp, Cosamp: iterative signal recovery from incomplete and inaccurate samples, Applied and Computational Harmonic Analysis 26 (2009) 301–321.

[11] H. Mohimani, M. Babaie-Zadeh, C. Jutten, A fast approach for overcomplete sparse decomposition based on smoothed l0 norm, IEEE Transactions on Signal Processing 57 (2009) 289–301.

[12] E. Candés, M. B. Wakin, S. Boyd, Enhancing sparsity by reweighted ℓ_1 minimization, Journal of Fourier Analysis and Applications 14 (2008) 877–905.

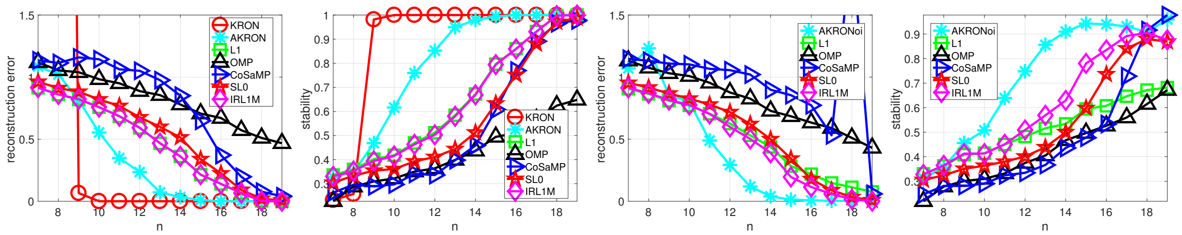
[13] B. Bayar, N. Bouaynaya, R. Shterenberg, Kernel reconstruction: an exact greedy algorithm for compressive sensing, in: International Workshop on Genomic Signal Processing and Statistics.

[14] J. A. Tropp, A. C. Gilbert, Signal recovery from random measurements via orthogonal matching pursuit, IEEE Transactions on Information Theory 53 (2007) 4655–4666.

[15] M. Grant, S. Boyd, CVX: Matlab software for disciplined convex programming, version 2.1, http://cvxr.com/cvx/, 2014.

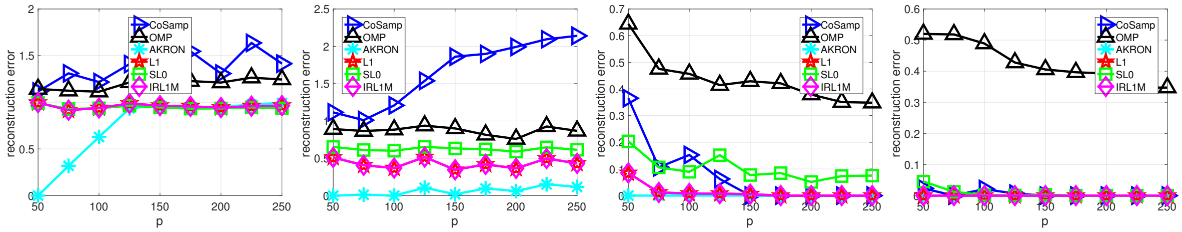
Algorithm 1 AKRON pseudo-code

```
Input: $\mathbf{A}$, $\mathbf{y}$, $\delta \in [n]$
Output: $\mathbf{x}^*$
$\widehat{\mathbf{x}}_1 = \text{argmin}\,\{\|\mathbf{x}\|_1 : \mathbf{A}\mathbf{x} = \mathbf{y}\}$
$s = |\text{Ker}(\mathbf{A})|$
Choose $B_s^{s+\delta}$ to be the combinations of the smallest $s+\delta$ magnitudes in $\widehat{\mathbf{x}}_1$
// Parallel Loop
for $q = 1, \ldots, |B_s^{s+\delta}|$ do
    $\mathcal{Q} = [p] \backslash B_s^{s+\delta}(q)$
    $\widehat{\mathbf{x}}_\delta^* = 0$
    $\widehat{\mathbf{x}}_\delta^* = \mathbf{A}_{\mathcal{Q}}^{-1}\mathbf{y}$ // Update only the indices in $\mathcal{Q}$
    $\text{spar}_q = \|\widehat{\widehat{\mathbf{x}}_\delta^*}\|_0$
end for
$q_{\min} = \min\{\text{spar}_q : q \in |B_s^{s+\delta}|\}$
$\mathcal{Q}^* = [p] \backslash B_s^{s+\delta}(q_{\min})$
$\mathbf{x}^* = 0$
$\mathbf{x}_{\mathcal{Q}^*}^* = \mathbf{A}_{\mathcal{Q}^*}^{-1}\mathbf{y}$
```



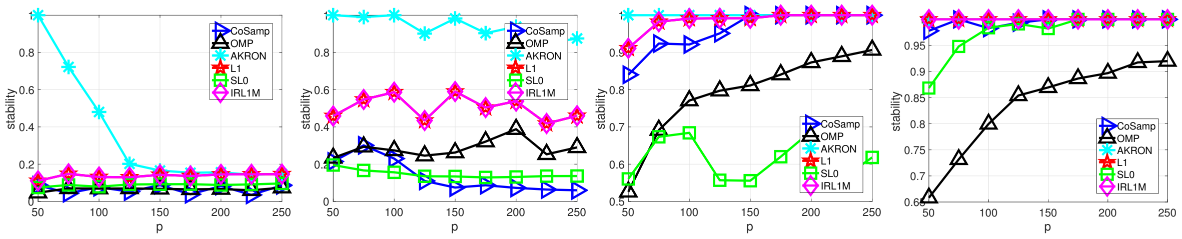
(a) Error of noiseless system (b) Stability of noiseless system (c) Error of noisy system (d) Stability of noisy system

Figure 1: Performance evaluation (using reconstruction error and Jaccard stability index) of AKRON with $\delta = 3$, ℓ_1-approximation, CoSaMP, SL0, IRLM and OMP on synthetic data sets with fixed dimension $p = 20$ and increasing number of measurements n. In the noisy systems, the error variance $\epsilon = 0.005$. CoSaMP and OMP were given the correct number of non-zero elements.



(a) $n = 0.1p$ (b) $n = 0.2p$ (c) $n = 0.3p$ (d) $n = 0.4p$

Figure 2: Reconstruction error of AKRON with $\delta = 3$, ℓ_1-approximation, CoSaMP, OMP, IRLM and SL0 on synthetic data sets of increasing dimensionality p. $k = 0.05p$ was fixed in all experiments, corresponding to 95% sparsity level. KRON has been omitted due the the computational complexity mentioned in Section 1.



(a) $n = 0.1p$ (b) $n = 0.2p$ (c) $n = 0.3p$ (d) $n = 0.4p$

Figure 3: Jaccard stability index for AKRON with $\delta = 3$, ℓ_1-approximation, CoSaMP, OMP, IRLM and SL0 on synthetic data sets of increasing dimensionality p and fixed $k = 0.05p$.

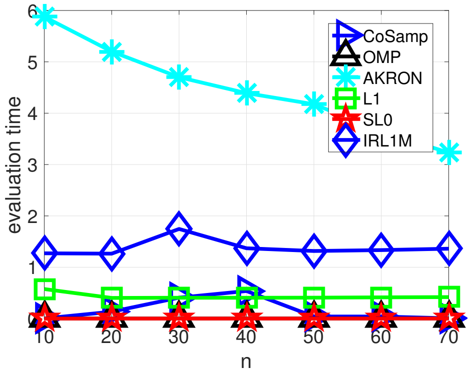


Figure 4: Run-time in seconds for CoSaMP, OMP, AKRON with $\delta = 3$, and ℓ_1-approximation on synthetic data sets for a fixed $p = 200$ and an increasing n.