Single-system multiple-output (SSMO) sparse solution mapped to single-output sparse solution of systems of equations

Andrew E. Yagle
Department of EECS, The University of Michigan, Ann Arbor, MI 48109-2122

Abstract—**The single-system multiple-output (SSMO) sparse reconstruction problem is to solve the underdetermined linear systems of equations Y=HX where the number of rows of Y exceeds the number of rows of X that are not all zero. We show that this problem can be mapped to a dual problem of solving an underdetermined linear system of equations w=Gz where the size of vector w exceeds the number of nonzero elements of vector z by the number of columns of X.**

Keywords— **Sparse reconstruction**
Phone: 734-763-9810. Fax: 734-763-1503.
Email: aey@eecs.umich.edu. **EDICS: 2-REST.**

I. INTRODUCTION

A. Problem Statement

The single-system multiple-output (SSMO) sparse reconstruction problem is to solve the multichannel underdetermined linear systems of equations

$$Y = HX \tag{1}$$

- Y is an $M \times L$ matrix having full rank.
- H is an $M \times N$ matrix having full rank.
- X is an $N \times L$ matrix having full rank.
- X has at most $(M–1)$ rows that are not all zero. But the location of these rows is unknown.
- $L < M << N$.

The SSMO problem is a multichannel version of the usual sparse reconstruction problem of solving the underdetermined linear system of equations $y = Hx$ where y is an M-vector, at most M–1 elements of x are nonzero, but the location of the nonzero elements of x is unknown. This is the single-system single-output (SSSO) sparse reconstruction problem.

Sparse reconstruction is currently of great interest in compressed sensing, since many real-world signals and images have sparse (mostly zero) representations in an appropriate basis, such as a set of wavelet or curvelet basis functions. The number of observations necessary to reconstruct the signal is therefore greatly reduced from the size of the signal. The SSMO problem arises in several applications in which multiple snapshots over time of a signal whose characteristics are varying over time are available.

B. Problem Background

Many approaches to solving the SSSO problem are known. The most common approach is basis pursuit, in which linear programming is used to find the solution to $y = Hx$ which has the minimum ℓ_1 norm (sum of absolute values). Another approach is matching pursuit, in which columns of H most highly correlated with the residual $y–H\hat{x}_i$ are successively chosen to minimize the residual $y–H\hat{x}_{i+1}$. Iterative thresholding, in which the Landweber iteration is applied to $y = Hx$ and some elements of $\hat{x}_i$ are thresholded to zero at each iteration. We will not attempt to summarize the many variations on these themes.

Extending these approaches to the SSMO problem has proven to be difficult. The most common approach is to find the solution to $Y = HX$ that minimizes a mixed-norm criterion like

$$\underset{x_{ij}}{\text{MIN}} \sum_{i=1}^{N} \sqrt{\sum_{j=1}^{L} x_{ij}^2}. \tag{2}$$

This is the ℓ_1 norm over columns i of the ℓ_2 norm of each row. The ℓ_2 norm of a row is zero if and only if all elements in that row are zero, and positive otherwise. The ℓ_1 norm is then just the sum of these, and so can be expected to maximize the number of all-zero rows of X, as desired.

C. Contribution of This Paper

This paper maps the SSMO problem $Y = HX$ to a so-called dual SSMO problem $w = Gz$ where

- G is an $(N–M+L) \times (N–1)$ matrix.
- w is a known $(N–M+L)$-length column vector.
- z is an unknown $(N–1)$-length column vector.
- z has only $(N–M)$ nonzero elements out of $(N–1)$. So z has $(N–1)–(N–M)=(M–1)$ zero elements. But the location of these elements is unknown.
- The location of the zero elements of z indicate the location of the $(M–1)$ nonzero rows of the matrix X.

We call this the dual SSSO problem to the original SSMO problem. Solving this SSSO problem yields the location of the rows of X that are not all zero; z is an indicator function for these rows. Using these locations the SSMO problem can be solved. The mapping between the SSMO and its dual SSSO problem is invertible, so the two problems are equivalent.

II. SSMO TO SSSO MAPPING

A. Left and Right Nullspaces of Submatrix of H

We make the following definitions:

- $\{i_1, i_2 \dots i_{M-1}\}$=rows of X that are not all zero.
- $H_{i_1 \dots i_{M-1}}$=$M \times (M-1)$ submatrix of H.
- $X'_{i_1 \dots i_{M-1}}$=$L \times (M-1)$ submatrix of X'.
- $M\{\underbrace{\boxed{\text{H}}}_{N}$ indicates H is an $M \times N$ matrix.

Then the original SSMO problem, repeated as

$$M\{\underbrace{\boxed{\text{Y}}}_{L} = M\{\underbrace{\boxed{\text{H}}}_{N}\, N\{\underbrace{\boxed{\text{X}}}_{L}. \tag{3}$$

can be rewritten as

$$M\{\underbrace{\boxed{\text{H}}}_{N}\underbrace{\boxed{\text{Y}}}_{L}\begin{bmatrix} N\{\boxed{\text{X}} \\ L\{\underbrace{\boxed{\text{-I}}}_{L} \end{bmatrix} = M\{\underbrace{\boxed{0}}_{L}. \tag{4}$$

In particular, we have

$$M\{\underbrace{\boxed{H_{i_1 \dots i_{M-1}}}}_{M-1}\underbrace{\boxed{\text{Y}}}_{L}\begin{bmatrix} \boxed{X_{i_1}} \\ \vdots \\ \boxed{X_{i_{M-1}}} \\ \underbrace{\boxed{\text{-I}}}_{L} \end{bmatrix} = M\{\underbrace{\boxed{0}}_{L}. \tag{5}$$

Nominally, the right nullspace of an $M \times (M-1+L)$ matrix has dimension $(M-1+L)-M=L-1$. But the right nullspace of the left matrix in (5) has dimension L. So the left matrix in (5) is rank-deficient. Hence, it has a left null row M-vector d', where

$$\underbrace{\boxed{d'}}_{M}\, M\{\underbrace{\boxed{H_{i_1 \dots i_{M-1}}}}_{M-1}\underbrace{\boxed{\text{Y}}}_{L} = \underbrace{\boxed{0 \dots 0}}_{M-1+L}. \tag{6}$$

Now define the column N-vector z by

$$\begin{bmatrix} N\{\boxed{\text{z}} \\ L\{\boxed{0} \end{bmatrix} = \underbrace{\begin{bmatrix} N\{\boxed{H'} \\ L\{\boxed{Y'} \end{bmatrix}}_{M} M\{\boxed{\text{d}}. \tag{7}$$

Note $Y'd$=0 since $d'Y$=$0'$. z_{i_1}=z_{i_2}=$\dots$=$z_{i_{M-1}}$=0 and all other entries of z are generically nonzero.

Now, suppose that the original H was augmented by a single, randomly-chosen column H_N, and the original X was augmented by a single row X'_N of zeros. Since the original problem generically has no solution, but here has a solution by assumption, this augmentation will not introduce another solution. Then $z_N \neq 0$, since the last row of X is all zero.

B. Formulation of Dual SSSO Problem

Since the $M \times (N+L)$ matrix $\boxed{\text{H Y}}$ has full rank, its right null space has dimension $(N+L-M)$. Hence it has a $(N+L) \times (N+L-M)$ right null matrix G'. Partitioning G as shown and taking a transpose gives

$$N+L-M\{\underbrace{\boxed{G_1}}_{N}\underbrace{\boxed{G_2}}_{L}\underbrace{\begin{bmatrix} N\{\boxed{H'} \\ L\{\boxed{Y'} \end{bmatrix}}_{M} = \underbrace{\boxed{0 \dots 0}}_{M}. \tag{8}$$

Since $[z'\ 0]'$ lies in the right null space of G,

$$N+L-M\{\underbrace{\boxed{G_1}}_{N}\underbrace{\boxed{G_2}}_{L}\begin{bmatrix} N\{\boxed{\text{z}} \\ L\{\boxed{0} \end{bmatrix} = N+L-M\{\boxed{0} \tag{9}$$

which becomes

$$N+L-M\{\underbrace{\boxed{G_1}}_{N} N\{\boxed{\text{z}} = N+L-M\{\boxed{0}. \tag{10}$$

Now partition G_1 into its first $(N-1)$ columns $\tilde{G}_1$ and its final column $(G_1)_N$, and partition z similarly:

$$N+L-M\{\underbrace{\boxed{G_1}}_{N} = N+L-M\{\underbrace{\boxed{\tilde{G}_1}}_{N-1}\underbrace{\boxed{(G_1)_N}}_{1}. \tag{11}$$

After partitioning, (10) can be rewritten as

$$N+L-M\{\underbrace{\boxed{\tilde{G}_1}}_{N-1}\underbrace{\boxed{(G_1)_N}}_{1}\begin{bmatrix} \boxed{\tilde{z}} \\ \boxed{z_N} \end{bmatrix} = \boxed{0}. \tag{12}$$

Defining $w = -(G_1)_N$ gives the dual SSSO system

$$N+L-M\{\underbrace{\boxed{\tilde{G}_1}}_{N-1}\boxed{\tilde{z}/z_N} = -(G_1)_N = w. \tag{13}$$

C. Properties of Dual SSSO Problem

- Submatrix $\tilde{G}_1$ is an $(N-M+L) \times (N-1)$ matrix.
- w=$-(G_1)_N$ is an $(N-M+L)$ column vector.
- $\tilde{z}$ is an unknown $(N-1)$-length column vector.
- $\tilde{z}$ has only $(N-M)$ nonzero elements out of $(N-1)$. So $\tilde{z}$ has $(N-1)-(N-M)$=$(M-1)$ zero elements.
- The location of the zero elements of $\tilde{z}$ indicate the location of the $(M-1)$ not-all-zero rows of matrix X.

Increasing the number L of channels has two effects:

- The difference between #rows $(N-M+L)$ and #columns $(N-1)$ of $\tilde{G}_1$ becomes smaller.
- The difference between #rows $(N-M+L)$ and the sparsity $(N-M)$ of $\tilde{z}$ becomes larger.

Both effects make the SSSO problem easier to solve, as the following tables show for various problem sizes.

For the original SSMO problem:

example#	variable	#1	#2	#3
#equations	M	100	100	50
#unknowns	N	900	9000	9000
#nonzero	M–1	99	99	49
#channels	L	50	50	25

For the dual SSSO problem, these give:

example#	variable	#1	#2	#3
#equations	N–M+L	850	8950	8975
#unknowns	N–1	899	8999	8999
#nonzero	N-M	800	8900	8950

Note that for each of these SSSO problems the system matrix aspect ratio $\frac{N-M+L}{N-1}$ and the sparsity to #equations ratio $\frac{N-M}{N-M+L}$ are roughly equal, and close to unity. It is well known that in this case use of ℓ_1-norm minimization will almost always find the sparsest solution, for a random system matrix. The requirement for the SSMO problem is that the number channles be half the number of equations.

D. Complete Procedure

1. **Given:** The SSMO problem $Y = HX$.
2. Compute the right null matrix G' of $[H\ Y]$.
Or, G is the left null matrix of $[H\ Y]'$.
3. Partition G into $[\underbrace{\tilde{G}_1}_{N-1}\ \underbrace{(G_1)_N}_{1}\ \underbrace{G_2}_{L}]$.
4. Solve the SSSO $(G_1)_N = \tilde{G}_1\tilde{z}$ for sparse vector $\tilde{z}$.
The solution of this SSSO is actually $-\tilde{z}/z_N$, not $\tilde{z}$.
5. The location of the zero elements of $\tilde{z}$ indicate the location of the not-all-zero rows of matrix X.
6. Solve Y=HX after eliminating all-zero rows of X.

The following MATLAB program demonstrates the procedure. Given the SSMO problem $Y = HX$, it computes all of the variables used in the derivation, and obtains the SSSO problem $w = Gz$. The location of zero values of z indicate the location of not-all-zero rows of X. The SSSO problem itself is not solved; any of the panoply of methods for sparse solution of underdetermined systems of equations may be used.

```
clear;N=30;M=10;L=5;
H=randn(M,N);X=randn(N,L);
%Input the M-1 rows to be nonzero.
K=[2 5 8 14 16 19 22 25 28];
X1=X;X1(K,:)=0;X=X-X1;Y=H*X;
%GOAL: Compute K from Y.
D=null([H(:,K) -Y]');
Z=H'*D;G=(null([H -Y]))';
G1=G(:,[1:N-1]);W=-G(:,N);
%Solve SSSO W=G1*Z1.
Z1=Z(1:N-1)/Z(N);[Z X]
```

III. Special Case: Number of Not-All-Zero Rows Equals the Number of Channels

We now consider the special case when M–1=L. Then the number of equations N–M+L=N–1 equals the number of unknowns N–1. So the SSSO is no longer underdetermined; it can simply be solved directly for $\tilde{z}$, which indicates the locations of not-all-zero rows of X. SSMO has a closed-form solution!

A. Simpler Algorithm

A simpler algorithm for computing $\tilde{z}$, that avoids computation of matrix G, can be derived as follows.

We are given the SSMO problem $Y = HX$, where now only K rows of X are not all zero. The number of channels L is assumed to equal or exceed the number of not-all-zero rows of X, so that $L \geq K$.

Let D be the left null matrix of Y, so DY=0. Then

$$[0] = DY = D(HX) = D[H_{i_1 \ldots i_K}][X'_{i_1 \ldots i_K}]. \quad (14)$$

But if $L \geq K$, then the $K \times L$ matrix $[X'_{i_1 \ldots i_K}]$ has full row rank. Then the left null spaces of both Y and $[H_{i_1 \ldots i_K}]$ are identical. Hence columns $i_1 \ldots i_K$ of $Z = DH$ will be columns of all zeros, so that

$$[0] = DH_{i_1 \ldots i_K}. \quad (15)$$

Hence Z=DH is again an indicator function for the not-all-zero rows of X.

B. Complete Procedure

1. **Given:** The SSMO problem $Y = HX$ where now only K rows of X are not all zero
and the number of SSMO channels $L \geq K$.
2. Compute the left null matrix D of Y.
3. Compute Z=DH. The all-zero columns of Z indicate the location of not-all-zero rows of X.

The previous algorithm solves this problem for the special case K=M–1=L. The following MATLAB program demonstrates the procedure. Given SSMO problem $Y = HX$, it computes indicator function Z.

```
clear;N=30;M=10;L=8;
H=randn(M,N);X=randn(N,L);
%Input the L rows to be nonzero.
K=[2 5 8 14 16 19 22 25];
X1=X;X1(K,:)=0;X=X-X1;Y=H*X;
%GOAL: Compute K from Y.
D=(null(Y'))';Z=D*H;[Z' X]
```

We note in passing that if only K rows of X are not all zero and $L < K < M$–1, the derivation of the original algorithm still holds, except that d' is now a $(M–K) \times M$ matrix instead of a row M-vector. The dual SSSO problem is now a dual SSMO problem.