Non-Iterative Computation of Sparsifiable Solutions to Underdetermined Kronecker Product Linear Systems of Equations

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

Abstract—**The problem of computing sparse (mostly zero) or sparsifiable (by linear transformation) solutions to underdetermined linear systems of equations has applications in compressed sensing and minimum-exposure medical imaging. We present a simple, non-iterative, low-computational-cost algorithm for computing a sparse solution to an underdetermined linear system of equations. The system matrix is the Kronecker (tensor) product of two matrices, as in separable 2D deconvolution and reconstruction from partial 2D Fourier data, where the image is sparsifiable by a separable 2D wavelet or other transform. Numerical examples and program illustrate the new algorithm.**

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 goal is to solve underdetermined linear system

$$y = \tilde{H}x = \tilde{H}W'Wx = Hz; H = \tilde{H}W'; z = Wx \quad (1)$$

- H is the Kronecker product $H = H_1 \bigotimes H_2$;
- H_1 and H_2 are both M×N matrices; N>>M;
- $\tilde{H}$ and H=$\tilde{H}W'$ are both $M^2 \times N^2$ matrices;
- W is $N^2 \times N^2$ unitary transformation matrix that sparsifies the unknown column N^2-vector x to z;
- z has only $M-1$ out of N^2 non-zero elements;
- $y, \tilde{H}, W, H, M, N$ are known; x, z are unknown.

The sparsifying transformation W is any 2D separable orthonormal transform, such as the 2D wavelet transform. Of course, if the unknown x is itself sparse, as in X-ray crystallography, then $W = I$.

Applications with system matrix $\tilde{H}$ and H=$\tilde{H}W'$ are Kronecker products of two matrices are:

- 2D deconvolution with separable point-spread-function (PSF), e.g., a 2D Gaussian PSF;
- 2D reconstruction from partial Fourier data, since the 2D DFT is the Kronecker product of 1D DFTs;
- The image (unwrapped to x) is sparsifiable by a separable 2D transform, e.g., 2D wavelet transform.

We note that if H is substantially underdetermined $N >> M$, but each row is known to be bandlimited to $\pi\frac{M}{N}$, then our non-iterative algorithm [1] can be used. We have also used related but different approaches to X-ray crystallography [2] and to the limited-angle tomography formulation of SAR [3].

B. Relevant Previous Approaches

Use of sparseness as side information in signal reconstruction goes back at least as far as 1979 [4]. Most early work considered the deconvolution of sparse 1D spike trains arising in reflections from layered media in seismic exploration, although [5] considered 1D reconstruction from bandlimited data. The first use of sparseness in 2D (image) reconstruction of which we are aware is [6]. All of this early work minimized the ℓ_1 norm (sum of absolute values) of the signal, using linear programming. The idea was that the ℓ_1 norm solution lies on a vertex of the simplex and so is sparse. This idea has recently been put on a firmer theoretical ground in [7] and other recent papers. More recent work using this approach is [8]-[10]. Using the ℓ_1 norm for the signal and ℓ_2 for the error is called LASSO. The problem with this approach is the amount of computation required by linear programming for image reconstruction.

Another way of formulating the linear sparse reconstruction problem is as a matrix "subset selection" problem [11]-[14]. The forward greedy algorithm successively selects the matrix column closest (in the mean-square sense) to the residual error resulting from the previous matrix column selections. The backward greedy algorithm starts with a general solution and successively removes the matrix column that increases the mean-square residual error the least. The latter algorithm has been shown to give the correct answer if the noise level is sufficiently small [13]. However, again the problem is the amount of computation required here by subset selection.

Still another recent approach is to include a thresholding constraint in a Landweber-like iterative algorithm [15]-[18], often arising from statistical image priors that implicitly (but not explicitly) maximize sparsity. This is a straightforward approach, and unlike the above methods requires reasonable computation for 2-D problems. However, even in the absence of noise, convergence to an optimally-sparse solution is not in general guaranteed for these algorithms.

C. New Approach of This Paper

All of the above approaches are iterative, requiring many iterations, are computationally intensive, or both. The approach of this paper requires only:

- Computation of singular value decompositions (SVDs) of M×N matrices H_1 and H_2 (H=$H_1 \bigotimes H_2$);
- Computation of left and right null vectors of the M×M matrix formed from the M^2 elements of y;
- Solution of an $M^2 \times (M-1)^2$ linear system;
- Computation of $x = W'z$ (if x is not sparse).

Note that all of these operations are straightforward linear algebra and can be implemented in a non-iterative way. Of course, iterative methods such as conjugate gradient may be used in some steps, but the overall algorithm is non-iterative, and can even be considered to be a closed-form computation.

II. Derivation of New Algorithm

A. Review of Kronecker Product

The Kronecker (tensor) product $A \bigotimes B$ of two $M \times N$ matrices A and B is the $M^2 \times N^2$ matrix

$$[A \bigotimes B]_{\substack{iM+m \\ jM+n}} = A_{i,j}B_{m,n}, \begin{array}{c} 0 \le i,j \le M-1 \\ 0 \le m,n \le N-1 \end{array} \quad (2)$$

A simple numerical example:

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \bigotimes \begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix} = \begin{bmatrix} 5 & 6 & 10 & 12 \\ 7 & 8 & 14 & 16 \\ 15 & 18 & 20 & 24 \\ 21 & 24 & 28 & 32 \end{bmatrix} \quad (3)$$

Relevant properties of the Kronecker product are:

$$(A \bigotimes B)(C \bigotimes D) = (AC) \bigotimes (BD) \quad (4)$$

If C and D are inverses of (sparsifying) 1D wavelet transforms, and A and B are 1D partial discrete Fourier transform (DFT) matrices, or 1D Toeplitz (convolutional) matrices, then $C \bigotimes D$ is the inverse 2D wavelet transform, and $A \bigotimes B$ is the partial 2D DFT matrix or 2D convolution for separable PSF.

$$vec(AXB) = (B' \bigotimes A)vec(X). \quad (5)$$

$vec(X)$ unwraps the matrix X by columns (`X(:)` in Matlab). Applying this to the current problem,

$$vec(Y) = (H_1 \bigotimes H_2)vec(Z) \leftrightarrow Y = H_2 Z H_1' \quad (6)$$

B. Reformulation of Problem

Recall that the original problem is

$$y = Hz = (H_1 \bigotimes H_2)z \quad (7)$$

where z has at most $M-1$ nonzero elements. Wrapping vectors y and z into matrices Y and Z:

- M^2-vector $y \rightarrow (M \times M)$ matrix Y;
- N^2-vector $z \rightarrow (N \times N)$ matrix Z,

and defining the SVDs of H_1 and H_2 as

$$H_1 = U_1 S_1 V_1; \qquad H_2 = U_2 S_2 V_2 \quad (8)$$

the original problem is equivalent to

$$Y = H_2 Z H_1' = (U_2 S_2 V_2) Z (V_1' S_1 U_1') \quad (9)$$

which in turn becomes

$$R = S_2^{-1} U_2' Y U_1 S_1^{-1} = V_2 Z V_1' \quad (10)$$

where the M×M matrix R is quickly computable from the M×M matrix Y composed of values of data y.

Since only $M-1$ elements of Z are nonzero, Z and hence Y both have rank $M-1$. Assuming that no two nonzero elements of Z are in the same row or column, rows and columns containing nonzero elements of Z are known. Let $\vec{n}$ be the right null vector of R:

$$0 = R\vec{n} = (V_2 Z V_1')\vec{n} \rightarrow 0 = Z V_1' \vec{n} \quad (11)$$

If the i^{th} row of Z is all zero, then the i^{th} element of $Z V_1' \vec{n}$ is zero. If the $(i,j)^{th}$ element of Z is nonzero, the j^{th} row of V_1' is orthogonal to $\vec{n}$. Hence zeros of $V_1'\vec{n}$ indicate columns of Z with a nonzero element.

Repeating this with R' identifies rows of Z with nonzero elements. Hence rows and columns of Z with nonzero elements can be identified. Each nonzero element is at the intersection of one of these rows and columns, for a total of $(M-1)^2$ possible locations of nonzero elements. So M^2 values of y are sufficient to determine which of these elements is nonzero.

III. Numerical Examples

A. Example Illustrating the Algorithm

Matlab's `rand` was used to create the following:

- A 256×256 sparse image Z, with 21 nonzero pixels;
- A 484×65536 system matrix HH, which is the Kronecker product of a 22×256 matrix H with itself;
- The length=484 vector of data y=HH*Z(:).

The goal is to reconstruct the 65536-vector Z(:) from the 484-vector y, using the side information that only 21 elements of Z(:) (locations unknown) are nonzero.

The above algorithm was applied to this problem:

- SVD of 22×256 matrix H (HH=H$\otimes$H);
- Left and right null vectors of the 22×22 matrix Y;
- Two 256×22 matrix-vector multiplications;
- Solution of a 484×421 linear system of equations to determine values of the nonzero elements of Z.

Results are shown in three figures:
1. Original sparse image Z (21 nonzero pixels);
2. 22×22 matrix of data Y from data vector y;
3. Reconstructed rows and columns containing nonzero pixels (actual nonzero pixels are also shown).

The Matlab code for this example is given below.

```
clear;rand('state',0);X=ceil(rand(256,256)-0.99965);
H=rand(22,256);[U S V]=svd(H);Y=H*(H*X)';Q=V(:,1:22);
%Y(:)=kron(H,H)*X(:).484X65536 matrixXvector too big.
%GOAL: Get sparse X(length=65536) from Y(length=484).
W=inv(S(:,1:22))*U';YY=reshape(kron(W,W)*Y(:),22,22);
I=find(abs(Q*null(YY))<0.000001);%=find(sum(X')>0)'
J=find(abs(Q*null(YY'))<0.000001);%=find(sum(X)>0)'
Z(256,256)=0;Z(I,:)=1;Z(:,J)=1;%Lines with nonzero X
figure,imagesc(X),colormap(gray),title('SPARSE IMAGE')
figure,imagesc(Y),colormap(gray),title('DATA IMAGE')
figure,imagesc(Z+3*X),colormap(gray),title('INDICATOR')
```

B. Larger Numerical Example

This larger and more complete problem features:

- A 499-sparse vector of length 1,000,000;
- 250,000 linear combinations of this as data;
- Valid 2-D deconvolution of a separable PSF.

The problem is to reconstruct a sparse 1000×1000 image from its valid 2-D convolution with a 501×501 separable random PSF. Since only 1/9 of the 2-D convolution is known, the FFT cannot be used here.

The algorithm was run using the Matlab code below. The 250,000-point data vector was rearranged into a 500×500 matrix Y, whose singular values were

$$\sigma_{499} = 2.5 \times 10^{-8}; \quad \sigma_{500} = 5.6 \times 10^{-15}. \qquad (12)$$

so its null vector was clearly defined. This was then used to determine the rows and columns of the image containing nonzero pixels. Reconstruction of the image is then straightforward (see the program).

Results are shown in three figures:

4. Original 1000×100 499-sparse image X;
5. 500×500 valid 2-D convolution data Y;
6. Reconstructed image (matches original image).

Despite the size of the problem, solution requires:
- A 500×1000 SVD (performed off-line);
- Solving 2 500×500 linear systems;
- A 499×499 matrix inversion.

The Matlab code for this example is given below.

```
clear;K=0;rand('seed',0);H=rand(1,501);
%GOAL:499-sparse X from Y=[250,000X1,000,000 H]X
%499-sparse 1000 X 1000 image such that no two pixels
%are in same row or same column, as required by the algorithm:
X(1000,1000)=0;while(K<499);I=ceil(1000*rand);J=ceil(1000*rand);
if(sum(abs(X(I,:)))+sum(abs(X(:,J)))==0);X(I,J)=1;K=K+1;end;end;
%Matrix to implement valid convolution with random H:
T=toeplitz([H(1) zeros(1,499)],[H zeros(1,499)]);[U S V]=svd(T);
Q=V(:,1:500);Y=T*X'*T';S=diag(1./diag(S));YY=S*U'*Y*U*S;
%GOAL:Recover sparse X from its valid 2-D convolution Y.
[U1 S1 V1]=svd(YY); II=find(abs(Q*V1(:,500))<0.00000001);
[U2 S2 V2]=svd(YY');JJ=find(abs(Q*V2(:,500))<0.00000001);
%II and JJ are rows and columns with nonzero pixels
G1=T(:,II);G2=T(:,JJ);W=G2\Y;W=G1\W';Z(II,JJ)=W;
figure,imagesc(X),colormap(gray),title('ORIGINAL IMAGE')
figure,imagesc(Y),colormap(gray),title('CONVOLVED IMAGE')
figure,imagesc(Z),colormap(gray),title('RECONSTRUCTED')
```

References

[1] A.E. Yagle, “A Non-Iterative Procedure for Sparse Solutions to Linear Equations with Bandlimited Rows,” 2008.

[2] A.E. Yagle, “New Atomicity-Exploiting Algorithms for Super-Resolution X-Ray Crystallography,” 2008.

[3] A.E. Yagle, “Limited Angle Tomography of Sparse Images from Noisy Data using TLS MUSIC,” 2008.

[4] H. Taylor, S. Banks, F. McCoy, “Deconvolution with the L_1 norm,” *Geophysics* 44, 39-52, 1979.

[5] F. Santosa and W.W. Symes, “Linear inversion of bandlimited reflection seismograms,” *SIAM J. Sci. Stat. Comp.* 7, 1307-1330, 1986.

[6] D.C. Dobson and F. Santosa, “Recovery of blocky images from noisy and blurred data,” *SIAM J. Appl. Math* 56, 1181-1198, 1996.

[7] D. Donoho and M. Elad, “Optimally sparse representation in general (nonorthogonal) dictionaries via ℓ^1 minimization,” *Proc. Nat. Acad. Sci.* 100(5), 2197-2202, 2003.

[8] K. Tsuda and G. Ratsch, “Image reconstruction by linear programming,” *IEEE Trans. IP* 14, 737-744, 2005.

[9] E. Candes and J. Romberg, *Inverse Problems* June 2007.

[10] J. Haupt and R. Nowak, “Signal reconstruction from noisy random projections,” *IEEE Trans. Info. Th.* 52, 4036-4048, 2006.

[11] G. Harikumar and Y. Bresler, “A new algorithm for computing sparse solutions to linear inverse problems,” Proc. ICASSP 1996, 1331-1334.

[12] G. Harikumar, C. Couvreur, Y. Bresler, “Fast optimal and suboptimal algorithms for sparse solutions to linear inverse problems,” Proc. ICASSP 1998, 1877-1880.

[13] C. Couvreur and Y. Bresler, “On the optimality of the backward greedy algorithm for the subset selection problem,” *SIAM J. Matrix Anal. and Appl.* 21, 797-808, 2000.

[14] F.R. de Hoog and R.M.M. Mattheij, “Subset selection for matrices,” *Linear Algebra and its Applications* 422, 349-359, 2007.

[15] B.K. Natarajan, “Sparse approximate solutions to linear systems,” *SIAM J. Computers* 24, 227-234, 1995.

[16] M.A. Figueiredo and R.D. Nowak, “An EM algorithm for wavelet based image restoration,” *IEEE Trans. Image Proc.* 12, 906-916, 2003.

[17] I. Daubechies, M. Defrise, C. de Mol, “An iterative thresholding algorithm for linear inverse problems with a sparsity constraint,” *Comm. Pure Appl. Math* 57, 1413-1457, 2004.

[18] K.K. Herrity, A.C. Gilbert, J.A. Tropp, “Sparse approximation via iterative thresholding,” Proc. ICASSP 2006.

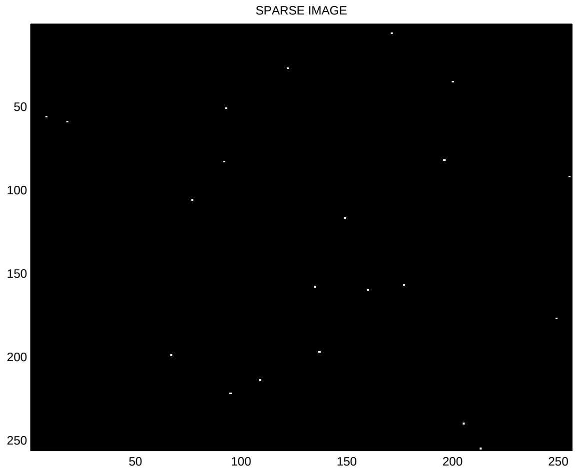

Fig. 1. Original 256×256 21-sparse image

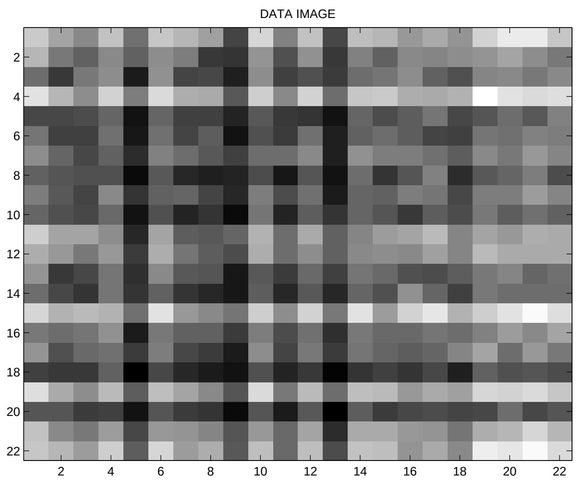

Fig. 2. 22×22 matrix of observations

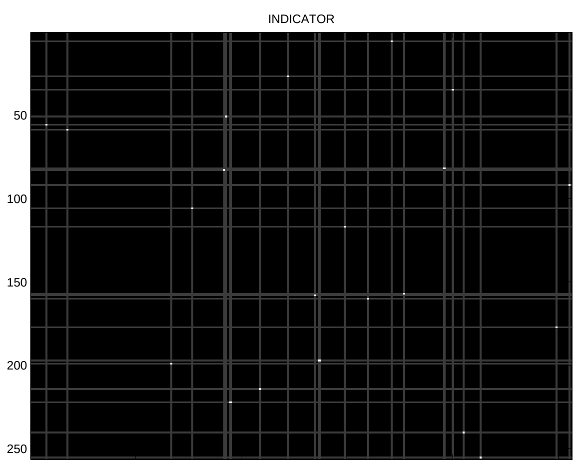

Fig. 3. Rows and columns containing nonzero pixels

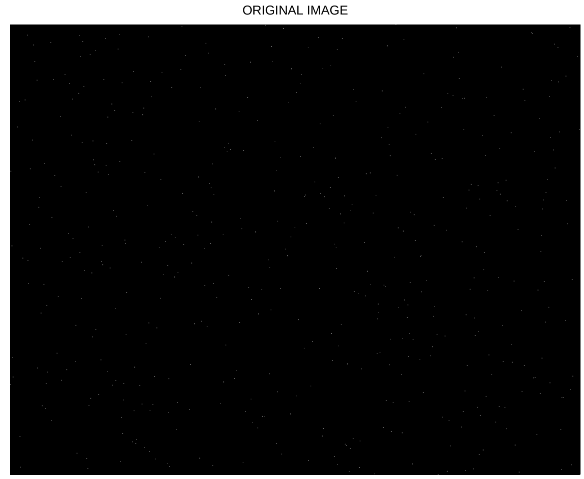

Fig. 4. Original 1000×1000 499-sparse image

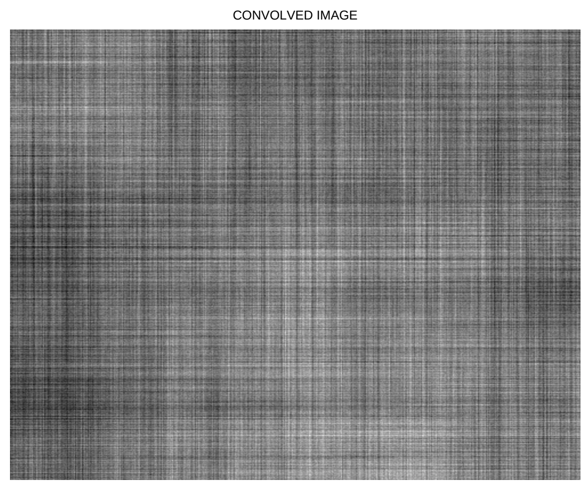

Fig. 5. 500×500 valid 2-D convolution data

Fig. 6. Reconstructed 1000×1000 499-sparse image