Deconvolution of Sparse Signals and Images from Noisy Partial Data using MUSIC

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

Abstract— **Deconvolution has three major difficulties: (1) the partial data issue–only the "valid" part of the convolution (which uses no image edge information) is known; (2) the lowpass nature of the impulse response (1D) or point-spread (2D) function; and (3) noise in the data. This paper shows that the second issue can be used to solve the first, and that sparsity side information can be used to obtain an exact reconstruction in the absence of noise, and an improved reconstruction in its presence. The partial data is extrapolated to the full cyclic convolution using its lowpass nature, and the sparse signal or image computed from its lowpass spectrum using a variation of TLS MUSIC. Examples illustrate the proposed algorithms.**

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

I. INTRODUCTION

A. Problem Statement

The basic 1D problem is as follows. We observe

$$y_n = \sum_{i=1}^{L} h_i x_{n+1-i} + v_n, \quad L \leq n \leq M \quad (1)$$

where the variables are defined as follows:

- $\{x_n, 1 \leq n \leq M\}$ is a length=M segment of a larger signal known to have more than half its values (3/4 in 2D) equal to zero (sparsity side information);
- $\{h_n, 1 \leq n \leq L\}$ is a length=L impulse response function known to be lowpass (defined below);
- $\{y_n, L \leq n \leq M\}$ is the length=M-L+1 segment of the convolution $h_n * x_n$ that does not use the edges of x_n, so that x_n may be only part of a larger signal. This "partial data" is called the "valid" convolution;
- $\{v_n, L \leq n \leq M\}$ is M-L+1 independent 0-mean Gaussian random variables (white Gaussian noise).

We start our indexing at $n = 1$ instead of $n = 0$ for convenience in matrix indexing. The M-point DFT

$$H_k = \sum_{n=1}^{L} h_n e^{-j2\pi(n-1)(k-1)/M}, \quad 1 \leq k \leq M \quad (2)$$

is assumed to be lowpass in that

$$H_k = 0 \text{ for } (M-L)/2 < (k-1) < (M+L)/2. \quad (3)$$

B. Deconvolution Issues

The problem can be rewritten in matrix form as

$$\begin{bmatrix} y_L \\ \vdots \\ y_M \end{bmatrix} = \begin{bmatrix} h_L & \cdots & h_1 & 0 & 0 \\ 0 & \ddots & \cdots & \ddots & 0 \\ 0 & 0 & h_L & \cdots & h_1 \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_M \end{bmatrix} + \begin{bmatrix} v_L \\ \vdots \\ v_M \end{bmatrix} \quad (4)$$

This shows that the partial data problem is always underdetermined: Even in the absence of noise we have (M-L+1) equations in M unknowns. This proves we need some prior information about the signal x_n. This information may be either statistical (e.g., x_n is Poisson) or deterministic (e.g., $x_n = 0$ for more than half the values of n). Statistical priors for the signal and noise can be used to obtain an MAP estimator of x_n, but this will not give the correct answer even in the absence of noise. A deterministic prior (side information) for the signal should lead to the correct answer in the absence of noise, although this is not always the case for many algorithms (see below).

Another issue is that h_n is often lowpass. Even if full data are available, some frequencies of x_n will be zeroed out by h_n. In the 1D case we can use z-transforms to separate zeros of $H(z)$ from those of $Y(z)$, leaving those of $X(z)$, but this cannot be done in 2D. Regularization techniques such as Tikhonov-regularized least-squares choose the solution that minimizes some criterion (this can also be interpreted as a Gaussian prior on x_n), but this does not lead to the correct answer, even in the absence of noise.

Of course, in practice there is always noise in the data. But an algorithm that gives the correct answer in the noiseless case should give an answer close to the correct answer if the noise level is low. And the algorithm should make explicit use of side information about x_n, such as a sparsity assumption.

C. Relevant Previous Work

We will not attempt to survey the vast library of deconvolution algorithms; instead we will restrict our attention to those using sparse priors for the signal.

The idea of exploiting signal sparsity in deconvolution goes back at least as far as 1979 [1]. The common approach of [1]-[9] was to minimize 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 thus is sparse [10]. In [4] an assumption of bounded noise led to a series of inequality constraints in (1) again solved by linear programming. The problem with this approach is the enormous amount of computation required for linear programming, especially in deconvolution of images.

The mid-1990's saw a resurgence of interest in both sparse deconvolution [8]-[10] and the more general problem of subset selection [11]-[13] in which the Toeplitz matrix in (4) above is replaced by a general matrix. 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 for subset selection algorithms.

A statistical approach used in the 1980's was maximum likelihood [14],[15]. The mid-2000's saw the use of statistical priors on x_n that implicitly (but not explicitly) maximize sparsity. These tend to lead to iterative algorithms with thresholding [16]-[18]. However, even in the absence of noise, convergence to an optimally-sparse solution is not guaranteed. The relation between minimum ℓ_1 norm (computed using linear programming) and minimum ℓ_0 norm (maximum sparsity) has been only recently been clarified: under certain conditions, minimizing the former also minimizes the latter ([19],[20] et seq.). This research is ongoing and will not be summarized here.

D. New Approach

This paper proposes a non-iterative algorithm for the sparse deconvolution problem that computes the maximally-sparse solution in the absence of noise, and generates a solution close to this solution for low noise levels. The algorithm is very fast; only two Toeplitz systems of equations are solved. Unlike other methods (besides computationally-intensive backwards-greedy subset selection), it generates the "correct" answer in the noiseless case.

The algorithm proceeds as follows:

1. h_n is lowpass, so y_n is lowpass, and the L-1 values of y_n needed to complete the *cyclic* convolution of h_n and x_n are extrapolated quickly using the algorithm of [13]-[14]. Alternatively, if h_n is not lowpass but x_n is lowpass, the same procedure can also be used;
2. The low-frequency values X_k of the DFT of x_n are computed as $X_k = Y_k/H_k$ for those frequencies for which $H_k \neq 0$. Only some of the X_k are known;
3. A variation of TLS MUSIC is used to compute the locations of the nonzero values of x_n from the low-frequency values X_k. Since nonzero values can occur only at integer times n, there are only a finite number of possible locations. There is error correction to the nearest possible location for low noise levels.

II. Noiseless Solution Procedure

A. Computation of Missing Data

We present a matrix depiction of the algorithm. We extend (4) to the circulant system of equations

$$\begin{bmatrix} y_L \\ \vdots \\ y_M \\ y_{M+1}+y_1 \\ \vdots \\ y_{M+L-1}+y_{L-1} \end{bmatrix} = \begin{bmatrix} h_L & \cdots & h_1 & 0 & 0 \\ 0 & \ddots & \ddots & \ddots & 0 \\ 0 & 0 & h_L & \ddots & h_1 \\ h_1 & 0 & 0 & \ddots & h_2 \\ h_2 & h_1 & 0 & \ddots & h_3 \\ \ddots & \ddots & \ddots & \ddots & \ddots \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_M \end{bmatrix} \tag{5}$$

Note the following about the circulant system (5):

- The Toeplitz matrix has been extended vertically to an $(M \times M)$ circulant matrix–each row consists of the row above it, circularly shifted one position to the right. The bottom row of the matrix is the top row circularly shifted one position to the left;
- The $\{y_{M+n}+y_n, 1 \leq n \leq (L-1)\}$ are unknown;
- The noise $[v_n]$ is not shown here to save space. Noise will be dealt with in the next section.

The eigendecomposition of the circulant matrix gives

$$\begin{bmatrix} y_L \\ \vdots \\ y_{M+L-1}+y_{L-1} \end{bmatrix} = D^H \cdot \mathrm{DIAG}[H_1 \cdots H_M] D \begin{bmatrix} x_1 \\ \vdots \\ x_M \end{bmatrix}. \tag{6}$$

D is the DFT matrix whose $(n,k)^{th}$ element is $e^{-j2\pi(n-1)(k-1)/M}$. Since the L-1 middle values of H_k are zero, we can eliminate the corresponding columns of D^H and corresponding rows of D. This yields

$$\begin{bmatrix} y_L \\ \vdots \\ y_{M+L-1}+y_{L-1} \end{bmatrix} = F^H \cdot \mathrm{DIAG}[H_k] F \begin{bmatrix} x_1 \\ \vdots \\ x_M \end{bmatrix} \tag{7}$$

where F is the DFT matrix with its middle L-1 rows discarded, and only the *nonzero* values of H_k appear. Finally, discarding the final L-1 rows of F^H gives

$$\begin{bmatrix} y_L \\ \vdots \\ y_M \end{bmatrix} = G^H \cdot \text{DIAG}[H_k] F \begin{bmatrix} x_1 \\ \vdots \\ x_M \end{bmatrix} \quad (8)$$

where G^H is F^H with its final L-1 rows discarded. G^H is the Hermitian transpose of the M×M DFT matrix with middle L-1 rows and final L-1 columns discarded, leaving an (M-L+1)×(M-L+1) matrix.

B. Computation of Low-Frequency Values of X_k

We now compute the low-frequency values X_k of the M-point DFT of x_n from this equation:

$$[X_k] = F \begin{bmatrix} x_1 \\ \vdots \\ x_M \end{bmatrix} = G^{-H} \begin{bmatrix} y_L \\ \vdots \\ y_M \end{bmatrix} / \text{DIAG}[H_k] \quad (9)$$

where G^{-H} refers to solving an (M-L+1)×(M-L+1) linear system of equations and the division by nonzero H_k is pointwise in k. The problem now is to compute x_n from its low-frequency DFT values X_k.

In fact, it is much easier to extrapolate the unknown aliased data $\{y_{M+n} + y_n, 1 \leq n \leq (L-1)\}$ directly from the known data $\{y_n, L \leq n \leq M\}$, as the middle L-1 values of Y_k are zero. This requires the solution of a small (L-1)×(L-1) linear system. This extrapolation can be done in closed form using [21]. The low-frequency values of X_k are computed using

$$X_k = \begin{cases} Y_k/H_k & \text{for } H_k \neq 0; \\ \text{unknown} & \text{for } H_k = 0 \end{cases}. \quad (10)$$

In practice $H_k \approx 0$ is treated as $H_k = 0$.

This seems to contradict the statement that some information about x_n is necessary to solve the partial data problem. There is no contradiction; we have *not* reconstructed X_k, but only its low frequency components. We cannot proceed further without some side information about x_n, as in the next subsection.

We also note that even if h_n is not lowpass, but x_n is, then the high-frequency values of Y_k are still zero. Then we can proceed as above and reconstruct the low-frequency values of X_k, so that we know all of X_k and hence x_n. This constitutes side information.

C. Computation of Sparse Solution x_n

The problem now is to compute x_n from its low-frequency DFT values X_k. Under the assumption that x_n is sparse (specifically, less than (M-L)/2 (roughly half if $L << M$) values of x_n are nonzero), we reconstruct x_n from the low-frequency values of X_k as a line spectrum estimation problem with the time and frequency domains exchanged.

As x_n is nonzero only at K times $\{n_i, 1 \leq i \leq K\}$,

$$X_{m-k} = \sum_{i=1}^{K} x_{n_i} e^{-j2\pi(n_i-1)(m-k-1)/M}. \quad (11)$$

The circulant matrix $[C]$ which has as its first row $\{X_k, 1 \leq k \leq M\}$ has the eigen-decomposition:

$$[C] = D^H \cdot \text{DIAG}[x_1 \cdots x_M] D \quad (12)$$

where D is again the DFT matrix having $(n,k)^{th}$ element $e^{-j2\pi(n-1)(k-1)/M}$.

Only K of the M diagonal values x_n are nonzero. Let F be the submatrix of D in which all but rows numbered $\{n_i, 1 \leq i \leq K\}$ have been deleted, and all but the first K+1 (out of M) columns have been deleted. Then the (K+1)×(K+1) Toeplitz submatrix $[X]$ of $[C]$ can be factored as

$$[X] = \begin{bmatrix} X_1 & \ldots & X_{K+1} \\ \ddots & \ddots & \ddots \\ X^*_{K+1} & \ddots & X_1 \end{bmatrix} = F^H \cdot \text{DIAG}[x_{n_i}] F. \quad (13)$$

This (K+1)×(K+1) matrix clearly has rank K, so it has a null vector $\vec{a} = [a_1 \ldots a_1^*]'$. F^H has full rank since it includes a Vandermonde matrix with distinct rows as a submatrix, and the x_{n_i} are all nonzero by definition. Postmultiplying by null vector $\vec{a}$ gives

$$[X]\vec{a} = F^H \cdot \text{DIAG}[x_{n_i}] F\vec{a} = 0 \rightarrow F\vec{a} = 0. \quad (14)$$

Examination of $F\vec{a} = 0$ shows that $e^{j2\pi n_i/M}$ are the roots of the polynomial having coefficients $\{a_n\}$.

However, there is another way to interpret $F\vec{a} = 0$: Since F is a submatrix of the DFT matrix D, we can compute $D\vec{a}$ and *see what values are zero*. The rows of D corresponding to those values are the rows of F, and this identifies the $\{n_i\}$. Thus, no root finding is required; we need only compute $D\vec{a}$ using an FFT and look for zeros in the result. This is also more numerically stable than computing polynomial roots.

III. Noisy Data: TLS MUSIC

Of course, in practice there is always noise in the data. Here we propose an algorithm we term TLS MUSIC to estimate locations $\{n_i\}$ from noisy data.

A. Other Spectral Estimation Algorithms

Most spectral estimation algorithms, such as Pisarenko method, MUSIC, and ESPRIT, operate not on the data but on the autocorrelation function estimated from the data. This has the advantage that additive white noise tends to be concentrated in the subspace spanned by the singular vectors associated with the minimum singular values, since the autocorrelation of zero-mean white noise is an impulse.

However, autocorrelation-based methods are inappropriate here, for the following three reasons:

- Only a small number of data points are available, not a long time series of data;
- Estimation of autocorrelation from data, which is always inexact due to end effects, is impractical;
- In practice, the additive noise is often neither white nor uncorrelated with the data.

Hence an approach that operates directly on the data, rather than on the autocorrelation, is necessary.

The approach used in this paper has been termed MUSIC since it is conceptually similar to MUSIC, but differs from it in these four (minor) ways:

- It works directly on the data, not autocorrelation;
- Noise is dealt with not by exploiting its (approximately) impulsive autocorrelation added to data correlation, but by perturbing data directly;
- Time and frequency domains are exchanged;
- The finite number of possible locations introduces some error correction for very low noise levels.

A simple likelihood function argument shows that if the noise is an additive zero-mean white Gaussian random process or field, then the likelihood is maximized when the given Fourier data X_k are perturbed as little as possible (in the mean square norm sense) to make the Toeplitz matrix (13) drop rank. Two major approaches are known for this problem.

The first is an iterative algorithm that alternates between the following two constraints:

- Computing the nearest (in Frobenius norm) lower rank matrix using the singular value decomposition, by subtracting the outer product of the minimum singular vectors times the minimum singular value;
- Computing the nearest (in Frobenius norm) Toeplitz matrix by averaging along the diagonals (the Hermitian structure is preserved throughout).

The other is structured total least squares, which iteratively perturbs the matrix closer to singularity, averaging diagonals to preserve Toeplitz structure.

Both of these approaches have been applied successfully to other problems. However, they have two problems rendering them inappropriate here:

- The size of (13) makes repeated computation of its singular value decomposition quite impractical;
- The Frobenius norm (sum of squared magnitudes of matrix elements) weights lower frequency components X_k more, since they occur more often in (13).

B. Background for Noisy Data Algorithm

Consider (14) with noiseless data $\{X_k\}$ replaced with noisy $\{X_k\}$. Then $[X]\vec{a} \neq 0$. What happens now is that $F\vec{a}$ computes the DFT of $\vec{a}$, “filters” it with $\{x_n\}$, and then F^H computes the inverse DFT of the result. In the noiseless case, only K of the $\{x_n\}$ are nonzero, and choosing the DFT of $\vec{a}$ to be zero at those nonzero locations makes $[X]\vec{a}$=0. In the noisy case, all of the $\{x_n\}$ are nonzero, so there is no way to make $[X]\vec{a}$=0, since $\vec{a}$ only has length M+1.

However, if the noise level is not too high, K of the $\{x_n\}$ will be larger than the remaining M-K values. Heuristically, $[X]\vec{a}$ will be minimized by choosing the DFT of $\vec{a}$ to be zero at the locations of the largest $\{x_n\}$ values, and nonzero elsewhere. Hence computing the value of $\vec{a}$ that minimizes $[X]\vec{a}$ (in the mean square norm) can be expected to pick out the locations of the largest $\{x_n\}$, which are assumed to be the true nonzero $\{x_n\}$. This value of $\vec{a}$ is the singular vector of $[X]$ associated with the minimum singular value of $[X]$. In the sequel, we refer to this as the “minimum singular vector” of the matrix $[X]$.

This will not work perfectly, of course, since the DFT of $\vec{a}$ has varying nonzero values. Hence the minimum of $[X]\vec{a}$ will be attained by a weighting of $\{x_n\}$. But it can be expected that the DFT of $\vec{a}$ will be significantly smaller at the K locations of the true nonzero $\{x_n\}$. Note that choosing $\vec{a}$ to be the minimum singular vector of $[X]$ also minimizes the perturbation (in Frobenius norm) of $[X]$ that makes $[X]$ drop rank, without regard to structure. This is the *non-structured* total least squares (TLS) solution. TLS has been used effectively with Prony’s method. Accordingly, we refer to our method as TLS MUSIC.

Note that the SVD of $[X]$ need not be computed– only its minimum singular vector is required. This can be computed fairly quickly using a few iterations of the inverse power method. Each iteration requires solving a linear system of equations with structured (Toeplitz or TBT) matrix $[X]$. Also note that there are only M possible locations of the $\{x_n\}$ (along each dimension). Error correction to nearest possible location happens if the noise level is sufficiently small.

C. Noisy Data Algorithm

1. Extrapolate data $\{y_n\}$ to an aliased $\{y_{M+n} + y_n\}$ using the extrapolation algorithm given in [21] (this is very effective in 2D since it operates in parallel);

2. Compute low-frequency values of $X_k = Y_k/H_k$ for $H_k \neq 0$. X_k is unknown when $H_k = 0$;
3. Assemble the Toeplitz (1D) or TBT (2D) matrix $[X]$ from the known values of the data X_k;
4. Compute the minimum singular vector $\vec{a}$ of $[X]$ using a few iterations of the inverse power method;
5. Compute the M-point (1D) or (M×M)-point (2D) DFT of $\vec{a}$, (unwrapping $\vec{a}$ into rows in the 2D case);
6. Identify the locations of the K smallest (in magnitude) values of the 2D DFT of $\vec{a}$. Use these as estimates of the locations of nonzero values of $\{x_n\}$;
7. Compute estimates of the values of nonzero $\{x_n\}$ by solving an overdetermined linear system of equations in the least-squares sense.

IV. Illustrative Micro-Examples

These examples only illustrate the new algorithm. Numerical simulations are given in the next section.

A. 1D Problem Micro-Example

The goal of this example is to solve the underdetermined convolution linear system of equations

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \end{bmatrix} = \begin{bmatrix} h_4 & h_3 & h_2 & h_1 & 0 & 0 & 0 & 0 \\ 0 & h_4 & h_3 & h_2 & h_1 & 0 & 0 & 0 \\ 0 & 0 & h_4 & h_3 & h_2 & h_1 & 0 & 0 \\ 0 & 0 & 0 & h_4 & h_3 & h_2 & h_1 & 0 \\ 0 & 0 & 0 & 0 & h_4 & h_3 & h_2 & h_1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \end{bmatrix} \tag{15}$$

where we know only two x_n are nonzero and

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \end{bmatrix} = \begin{bmatrix} 1.7071 \\ 3.1213 \\ 4.1213 \\ 3.4142 \\ 1.4142 \end{bmatrix}; \quad \begin{bmatrix} h_1 \\ h_2 \\ h_3 \\ h_4 \end{bmatrix} = \begin{bmatrix} .7071 \\ 1.7071 \\ 1.7071 \\ .7071 \end{bmatrix}. \tag{16}$$

The 8-point DFT of h_n is easily computed. Then the 8-point DFT of y_n is computed by extrapolating the known values of y_n using [13] and is

$$\begin{array}{lcclcc} H_1 & = & 4.8284 & Y_1 & = & 14.4853 \\ H_2 & = & 1.414-j3.414 & Y_2 & = & -6.2426+j5.4142 \\ H_3 & = & -1-j & Y_3 & = & -1-j \\ H_4 & = & 0 & Y_4 & = & 0 \\ H_5 & = & 0 & Y_5 & = & 0 \\ H_6 & = & 0 & Y_6 & = & 0 \\ H_7 & = & -1+j & Y_7 & = & -1+j \\ H_8 & = & 1.414+j3.414 & Y_8 & = & -6.2426-j5.4142 \end{array} \tag{17}$$

from which we compute the low-frequency values

$$X_1 = 3; \quad X_2 = X_8^* = -2-j; \quad X_3 = X_7^* = 1. \tag{18}$$

X_4, X_5, X_6 are unknown since H_4, H_5, H_6 are zero.

Since there is no noise we can use Prony's method:

$$\begin{bmatrix} X_1 & X_2 & X_3 \\ X_2^* & X_1 & X_2 \\ X_3^* & X_2^* & X_1 \end{bmatrix} \begin{bmatrix} a \\ b \\ a^* \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}. \tag{19}$$

Inserting the known values of X_k gives

$$\begin{bmatrix} 3 & -2-j & 1 \\ -2+j & 3 & -2-j \\ 1 & -2+j & 3 \end{bmatrix} \begin{bmatrix} a \\ b \\ a^* \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \tag{20}$$

which has the solution

$$[a \quad b \quad a^*] = [1+j \quad 2 \quad 1-j]. \tag{21}$$

Roots of the polynomial with these coefficients are

$$(1+j)z^2 + 2z + (1-j) = 0 \rightarrow$$
$$z = j = 1e^{j2\pi(3-1)/8}; z = -1 = 1e^{j2\pi(5-1)/8} \tag{22}$$

so the nonzero x_n are at n=3 and n=5. Solving

$$\begin{bmatrix} 1.7071 \\ 3.1213 \\ 4.1213 \\ 3.4142 \\ 1.4142 \end{bmatrix} = \begin{bmatrix} 1.7071 & 0 \\ 1.7071 & .7071 \\ .7071 & 1.7071 \\ 0 & 1.7071 \\ 0 & .7071 \end{bmatrix} \begin{bmatrix} x_3 \\ x_5 \end{bmatrix} \tag{23}$$

from which $x_3 = 1$ and $x_5 = 2$, so finally we have

$$\{x_n\} = \{0, 0, 1, 0, 2, 0, 0, 0\}. \tag{24}$$

Alternatively, we can avoid computing polynomial roots by computing the 8-point DFT of [1+j,2,1-j]:

$[4, \sqrt{2}(1-j), 0, (2-\sqrt{2})(1+j), 0, \sqrt{2}(j-1), 4j, (2+\sqrt{2})(1+j)]$.

The locations of the two zero values of this DFT shows the locations n=3,5 of nonzero values of x_n.

B. 2D Problem Micro-Example

The procedure extends in a straightforward way to 2D (reconstruction of sparse images). Note that:

- The extrapolation of the aliased data can be performed very quickly and in parallel using the algorithm of [21]. The problem decouples into independent (L-1)×(L-1) linear systems of equations;
- The TLS MUSIC algorithm extends directly to 2D. The null vector of a Toeplitz-Block-Toeplitz matrix, instead of Toeplitz, must now be computed.

The goal of this example is to solve the partial-data 2D deconvolution problem

$$\begin{bmatrix} x_{11} & x_{12} & x_{13} & x_{14} \\ x_{21} & x_{22} & x_{23} & x_{24} \\ x_{31} & x_{32} & x_{33} & x_{34} \\ x_{41} & x_{42} & x_{43} & x_{44} \end{bmatrix} * \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} = \begin{bmatrix} * & * & * & * & * \\ * & 3 & 4 & 1 & * \\ * & 4 & 1 & 1 & * \\ * & 4 & 0 & 0 & * \\ * & * & * & * & * \end{bmatrix} \quad (25)$$

where $*$ denote unknown data values (and a 2D convolution) and only three of x_{ij} are known nonzero. The point-spread function has the 2D DFT

$$\text{DFT}\left\{\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}\right\} = \begin{bmatrix} 4 & 2-2j & 0 & 2+2j \\ 2-2j & -2j & 0 & 2 \\ 0 & 0 & 0 & 0 \\ 2+2j & 2 & 0 & 2j \end{bmatrix} \quad (26)$$

Note that $H_{3m} = H_{n3} = 0$ so that the point-spread function is lowpass. The (aliased) missing values of y_{ij} can be extrapolated using the two formulae [21]

$$\begin{aligned} y_{i1} &= y_{i2} - y_{i3} + y_{i4} \\ y_{1j} &= y_{2j} - y_{3j} + y_{4j}. \end{aligned} \quad (27)$$

resulting in the 2D cyclic deconvolution problem

$$\begin{bmatrix} x_{11} & x_{12} & x_{13} & x_{14} \\ x_{21} & x_{22} & x_{23} & x_{24} \\ x_{31} & x_{32} & x_{33} & x_{34} \\ x_{41} & x_{42} & x_{43} & x_{44} \end{bmatrix} \copyright \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 3 & 3 & 0 \\ 0 & 3 & 4 & 1 \\ 4 & 4 & 1 & 1 \\ 4 & 4 & 0 & 0 \end{bmatrix} \quad (28)$$

where © denotes 2D cyclic convolution. The 2D DFT of the aliased data y_{ij} (right side of this) is

$$\text{DFT}\{y_{ij}\} = \begin{bmatrix} 32 & -12j & 0 & 12j \\ -4 & -4+8j & 0 & -8+8j \\ 0 & 0 & 0 & 0 \\ -4 & -8-8j & 0 & -4-8j \end{bmatrix}. \quad (29)$$

Note the zeros in the spectrum of the data caused by the lowpass nature of the point-spread function. The 2D DFT of the image x_{ij} can be computed by dividing the above two 2D DFTs. This yields

$$\text{DFT}\{x_{ij}\} = \begin{bmatrix} 8 & 3-3j & * & 3+3j \\ -1-j & -4-2j & * & -4+4j \\ * & * & * & * \\ -1+j & -4-4j & * & -4+2j \end{bmatrix}. \quad (30)$$

Now construct the 16×16 circulant-block-circulant matrix having for its first block row the circulant matrices having as first rows the rows of the DFT of x_{ij}. Taking the $\{1, 2, 5, 6\}$ rows and columns yields the Toeplitz-block-Toeplitz matrix and null vector

$$\begin{bmatrix} 8 & 3-3j & -1-j & -4-2j \\ 3+3j & 8 & -4+4j & -1-j \\ -1+j & -4-4j & 8 & 3-3j \\ -4+2j & -1+j & 3+3j & 8 \end{bmatrix} \times [1, -1-2j, 1-2j, -1]^T = [0, 0, 0, 0]^T. \quad (31)$$

Unwrapping the null vector into two rows and then computing its 2D DFT yields

$$\text{DFT}\left\{\begin{bmatrix} 1 & -1-2j \\ 1-2j & -1 \end{bmatrix}\right\} = \begin{bmatrix} -4j & 0 & 4 & 4-4j \\ -2-2j & -2 & 0 & -2j \\ 0 & -2+2j & 4j & 2+2j \\ 2-2j & 2j & 4+4j & -6 \end{bmatrix}. \quad (32)$$

The zeros in this DFT are at the locations of the nonzero x_{ij}. These three nonzero values are easily computed, yielding a final answer of

$$[x_{ij}] = \begin{bmatrix} 0 & 3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 4 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}. \quad (33)$$

Note the following about this problem:

- The null vector is anti-Hermitian. This is only coincidence; recall that this is a 2D problem;
- For tiny examples like this, it is possible that the 2D DFT of the null vector will have additional zeros in locations other than those of the nonzero x_{ij}. This is due to coincidence and does not happen for realistic-sized problems.

References

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

[2] N. Chapman and I. Barrodale, "Deconvolution of marine seismic data using the L_1 norm," *Geophys. J.R. Astr. Soc.* 72, 93-100, 1983.

[3] I. Barrodale, C. Zala, N. Chapman, "Comparison of the L_1 and L_2 norms applied to one-at-a-time spike extraction from seismic traces," *Geophysics* 49, 2048-2052, 1984.

[4] B. Drachman, "Two methods to deconvolve: L_1-method using simplex algorithm and L_2-method using least squares and a parameter," *IEEE Trans. Ant. Prop.* TAP-32, 219-225, 1984.

[5] R. Yarlagadda, J.B. Bendar and T.L. Watt, "Fast algorithms for L_p deconvolution," *IEEE Trans. ASSP*, ASSP-33, 174-182, 1985.

[6] J.B. Bednar, R. Yarlagadda, T.L. Watt, "L_1 deconvolution and its application to seismic signal processing," *IEEE Trans. ASSP*, ASSP-34, 1655-1658, 1986.

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

[8] M.S. O'Brien, A.N. Sinclair, S.M. Kramer, "Recovery of a sparse spike time series by ℓ_1 norm deconvolution," *IEEE Trans. ASSP*, ASSP-42, 3353-3365, 1994.

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

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

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

[12] 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.

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

[14] J.J. Kormylo and J.M. Mendel, "Maximum-likelihood seismic deconvolution," *IEEE Trans. Geosci. Rem. Sensing* GE-21, 72-82, 1983.

[15] T.T. Pharn and R.J.P. deFiguereido, "Maximum likelihood estimation of a class of non-Gaussian densities with application to ℓ_p deconvolution," *IEEE Trans. ASSP*, ASSP-37, 73-82, 1989.

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

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

[18] 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.

[19] 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.

[20] K.K. Herrity, A.C. Gilbert, J.A. Tropp, "Sparse approximation via iterative thresholding," Proc. ICASSP 2006.

[21] H. Soltanian-Zadeh and A.E. Yagle, "A fast algorithm for extrapolation of discrete-time periodic band-limited signals," *Signal Processing* 33, 183-196, 1993.