Recovery of K-Sparse Non-Negative Signals From K DFT Values and Their Conjugates

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

Abstract— **The goal is to reconstruct a sparse signal from some, but not all, of its Discrete Fourier Transform (DFT) values. If the signal has K non-zero and real values, a unique solution is determined by any K DFT values, their conjugates, and the DC value, if the DFT order is prime. However, no algorithm is known for this unless the K DFT values are at consecutive frequencies (a total of 2K+1 consecutive values). ℓ_1-norm minimization only works if the frequencies are randomly chosen. We present a new algorithm that reconstructs a K-sparse non-negative real-valued signal from any K DFT values, their conjugates, and the DC value, provided that the DFT order is prime and less than 4K. It does not use the ℓ_1 norm or pursuit.**

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 N-point DFT X_k of the length=N signal x_n is

$$X_k = \sum_{n=0}^{N-1} x_n e^{-j2\pi nk/N}, k = 0 \dots N-1. \quad (1)$$

We are given the following facts about the problem:

- x_n is real-valued and non-negative ($x_n \geq 0$);
- x_n=0 unless $n \in \{n_1 \dots n_K\}$ (the n_i are unknown);
- X_k known only for $k \in \{k_1 \dots k_M\}$ (k_i known) and their conjugates $\{N–k_1 \dots N–k_M\}$ (X_{N-k}=X_k^*);
- The DC value X_0=$\sum_{n=0}^{N-1} x_n$ is also known;
- The DFT order $N < 4K$ and N is a prime number.

The goal is to recover the K-sparse signal x_n from its DFT values X_k known at M frequencies k_i, their conjugate frequencies $N–k_i$, and X_0, Since $M < N$ the problem is very underdetermined, so the sparsity of x_n must be used to determine a unique solution.

B. Problem Significance

Reconstruction of signals and images from limited frequency data occurs in various problems such as:

- Limited-angle tomography in medical imaging;
- Synthethic-Aperture Radar (SAR) radar imaging;
- Magnetic Resonance Imaging (MRI) in medicine.

Many signals and images of practical interest have a sparse representation in a wavelet basis. Computation of the wavelet transform can be viewed as convolutions with scaled wavelet and scaling basis functions, which becomes multiplication in the DFT domain. Hence the results of this paper apply to both sparse signals and wavelet-sparsifiable signals.

C. Previous Results

It is known that a K-sparse real-valued signal can be recovered uniquely from any K N-point DFT values X_k, their conjugates X_{N-k}=X_k^*, and X_0, provided that N is prime. In particular, if X_k is known for $|k| \leq K$ (2K+1 consecutive values of X_k), then x_n can be reconstructed using any of the well-known array processing techniques such as Prony's method, MUSIC, or ESPRIT (even if N is not prime).

The necessity of N being prime can be seen as follows. Suppose N is even. Then knowledge of X_{2k} for $|k| \leq K$ would only allow reconstruction of the aliased x_n+$x_{n+N/2}$. Hence there is no way to determine whether x_n or $x_{n+N/2}$ is the non-zero element. Replacing 2 with any other factor shows that N must be prime to ensure uniqueness for *any* choice of k_i.

A common approach to sparse reconstruction is to compute the minimum ℓ_1 norm solution, usually by linear programming. If the DFT frequencies k_I are randomly chosen, and if enough of them are known, then it has been shown that the minimum ell_1 norm solution is in fact x_n. However, in many practical situations we do not have the luxury of choosing the k_i at random-they are pre-specified. And the number M (or 2M+1, depending on how conjugates are counted) must be greater than the minimum of K.

Other approaches include thresholded Landweber iteration and orthogonal matching pursuit. These are much faster computationally, but require more of the problem in order to compute x_n. Since our apporach is completely different from all of these, we refer the reader to the extensive literature on these methods.

Our approach, in contrast, works for *any* choice of frequencies k_i (and N–k_i). However, it also requires that $x_n \geq 0$ and N<4K, so that the system matrix has aspect ratio=2. This seems to be a common choice in the sparse reconstruction literature.

II. NEW ALGORITHM: PRELIMINARIES

A. Non-Negative Indicator Function

Define the polynomial $S(z)$ as

$$S(z)=\prod_{i=1}^{K}(z-e^{j2\pi n_i/N})(z^{-1}-e^{-j2\pi n_i/N}). \quad (2)$$

$S(z)$ is a polynomial of degree 2K whose coefficients S_k have Hermitian symmetry S_{-k}=S^*_k, and S_k=0 for $|k|>K$. Evaluating $S(z)$ on the unit circle z=$e^{j\omega}$:

$$S(e^{j\omega})=\prod_{i=1}^{K}(e^{j\omega}-e^{j2\pi n_i/N})(e^{-j\omega}-e^{-j2\pi n_i/N}). \quad (3)$$

Setting $\omega=2\pi n/N$ shows that

$$S(e^{j2\pi n/N})=\prod_{i=1}^{K}|e^{j2\pi n/N}-e^{j2\pi n_i/N}|^2 \quad (4)$$

from which it is apparent that

$$S(e^{j2\pi n/N})\begin{cases}=0 & \text{for } n=n_i\\ >0 & \text{for } n\neq n_i\end{cases}. \quad (5)$$

The inverse N-point DFT s_n of S_k is seen to be

$$s_n=\frac{1}{N}\sum_{k=-K}^{K}S_k e^{j2\pi nk/N}=\frac{1}{N}S(e^{j2\pi n/N}) \quad (6)$$

(S_k=0 for $|k|>K$). s_n is real since S_{-k}=S^*_k, and

$$s_n\begin{cases}=0 & \text{for } n=n_i\\ >0 & \text{for } n\neq n_i\end{cases} \quad (7)$$

B. Problem I: Given $\{X_k, |k|\leq K\}$

By Parseval's theorem for the DFT, we have

$$\sum_{k=-K}^{K}X^*_k S_k=N\sum_{n=0}^{N-1}x_n s_n=N\sum_{i=1}^{K}x_{n_i}s_{n_i}=0 \quad (8)$$

since s_{n_i}=0 by construction. Furthermore, since x_n and s_n are both non-negative, $\sum_{n=0}^{N-1}x_n s_n$=0 implies $x_n s_n$=0 for all n, so that *any* non-negative K-sparse solution x_n must have s_{n_i}=0 at non-zero locations n_i of x_n. So s_n is an indicator function for these.

Suppose we are given the 2K+1 consecutive values $\{X_k, |k|\leq K\}$. Then s_n can be computed from the $\{X_k, |k|\leq K\}$ by solving for $\{S_k, |k|\leq K\}$ the linear system of inequalities

$$\sum_{k=-K}^{K}X^*_k S_k=0; Ns_n=\sum_{k=-K}^{K}S_k e^{j2\pi nk/N}\geq 0. \quad (9)$$

and looking for zero values of s_n. The locations n_i of the zero values of s_n indicate possible non-zero values of x_n. Once the n_i have been found, x_{n_i} can be computed either by solving a $K\times K$ linear system of equations, or by using a POCS algorithm.

Of course, Problem I can also be solved using Prony's method, MUSIC, or ESPRIT, and without requiring non-negativity of x_n. But now we extend Problem I to allow missing values of X_k in $|k|\leq K$.

C. Problem II: Given $\{X_k, |k|\leq K+2\}$ Missing $X_{k'}$

Now suppose we are given $\{X_k, |k|\leq K+2\}$ but we are missing $X_{\pm k'}$ for some $0\leq k'\leq K$. None of Prony's method, MUSIC, or ESPRIT can be applied. But the above procedure can be applied by defining

$$T(z)=S(z)(z+a+z^{-1})^2 \quad (10)$$

In terms of coefficients of $S(z)$ and $T(z)$:

$$(z+a+z^{-1})^2\sum_{k=-K}^{K}S_k z^k=\sum_{k=-(K+2)}^{K+2}T_k z^k. \quad (11)$$

The constant a is chosen so that $T_{\pm k'}$=0. Then

$$T(e^{j\omega})=S(e^{j\omega})(a+2\cos(\omega))^2. \quad (12)$$

Setting $\omega=2\pi n/N$ as before gives

$$T(e^{j2\pi n/N})=S(e^{j2\pi n/N})(a+2\cos(2\pi n/N))^2. \quad (13)$$

We again recognize the inverse DFTs

$$Nt_n=Ns_n(a+2\cos(2\pi n/N))^2\geq 0. \quad (14)$$

t_n is an indicator for locations of non-zero values of x_n. The DFT T_k of t_n has the same properties as S_k, except that T_k=0 for $|k|>K$+2 and $T_{\pm k'}$=0.

t_n is computed from $\{X_k, |k|\leq K+2, k=\pm k'\}$ by solving for $\{T_k, |k|\leq K+2\}$ the linear system of inequalities

$$\sum_{\substack{k=-(K+2)\\ k=\pm k'}}^{K+2}X^*_k T_k=0; Nt_n=\sum_{k=-K+2}^{K+2}T_k e^{j2\pi nk/N}\geq 0.$$

and looking for zero values of t_n.

By induction, we can extend this to any number of missing values of X_k. Each missing X_k requires two additional values of X_k to compensate for it. This is disappointing, since in theory we should only need K values of X_k (and their conjugates), even if they are non-consecutive. It may seem as though we could use

$$T(z)=S(z)(z-a)(z^{-1}-a^*) \quad (15)$$

which maintains $T(e^{j\omega})\geq 0$. Unfortunately, in this case it is not always possible to choose a so that $T_{\pm k'}$=0. To see why, expand the additional term into

$$(z-a)(z^{-1}-a^*) = -a^*z + (1+|a|^2) - az^{-1} \quad (16)$$

so that the constant term must be greater than the square magnitudes of the leading and trailing coefficients. But the Hermitian quadratic polynomial that makes $T_{\pm k'}$=0 does not always have this property.

III. NEW ALGORITHM: MAIN RESULT

A. Problem Statement

We now consider the main problem: Let x_n be a K-sparse real-valued non-negative signal having an N-point DFT X_k. Given *any* 2K+1 values of X_k

$$\{X_0, X_{\pm k_1}, X_{\pm k_2} \dots X_{\pm k_K}\}, 0 < k_i < N/2 \quad (17)$$

we can recover x_n by solving a linear system of inequalities, provided $N < 4K$. N must be prime to ensure uniqueness, but this can usually be relaxed.

B. Problem Solution

Since we evaluate polynomials at $z = e^{j2\pi n/N}$, we can perform all polynomial calculations mod(z^N–1). So we can redefine $T(z)$ as (compare to above)

$$T(z) = S(z) \prod_{k=1}^{[N/4]} (z + a_k + z^{-1})^2 \quad \mathrm{mod}(z^N - 1) \quad (18)$$

where $[N/4]$ is the greatest integer not exceeding $N/4$. Then the extra term is a polynomial of degree $4[N/4] \leq N$, and the $[N/4]$ constants a_k can be chosen to make $[N/4]$ values of $T_{\pm k_i'}$=0, so that $X_{\pm k_i'}$ are not needed to compute the indicator function t_n.

We already know that a K-sparse x_n requires K values of $X_{\pm k}$ and X_0. So if N=4K, we can zero out K=N/4 values of $T_{\pm k}$, leaving K=N/4 nonzero values of $T_{\pm k}$ and T_0, just what we need to compute x_n.

Using the same argument as before, the indicator function t_n can be computed by solving the linear system of inequalities

$$\sum_{\substack{k=-[N/2] \\ k=\pm k'}}^{[N/2]} X_k^* T_k = 0; N t_n = \sum_{k=-[N/2]}^{[N/2]} T_k e^{j2\pi nk/N} \geq 0.$$

and looking for zero values of t_n.

C. Solving Linear Systems of Inequalities

Superficially, this looks like a linear programming problem, but it isn't. It is the first stage of a linear programming problem using the simplex method, in which a basic feasible solution is computed.

However, this problem can also be solved using a POCS algorithm, consisting of the following steps:

- Project t_n onto set of non-negative signals $t_n \geq 0$ using two FFTs to relate T_k to t_n;
- Set $\{T_k, k \neq \pm k_i\} = 0$. That is, set $T_k = 0$ for those values of k for which X_k is unknown;
- Project non-zero $\{T_k, k = \pm k_i\}$ onto the known hyperplane $\sum_{k=\pm k_i} X_k^* T_k = 0$;
- Continue until $N/4$ values of t_n are smaller than the remaining 3N/4 values of t_n;
- These t_n are possible locations of non-zero x_n.

Once the possible locations n_i of the non-zero values of x_n have been found by thresholding t_n, another POCS algorithm can be used to solve for these x_n:

- Set $z_n = 0$ for $n \neq n_i$ and for $z_{n_i} < 0$;
- Compute DFT Z_k of z_n and set $Z_{\pm k_i} = X_{\pm k_i}$;
- Compute the inverse DFT z_n of Z_k;
- Continue until convergence.

IV. Review of Two FFTs

We review two FFT algorithms for fast computation of the DFT that are pertinent to this algorithm.

A. Rader Prime Factor FFT

The Cooley-Tukey FFT is well-suited for values of N that are highly composite, e.g., powers of two. N prime is a worst case for it, since N cannot be factored. But the Rader FFT can be used instead.

Let N be prime and p be a primitive root of N (often $p = 2$ can be used). Then

$$\{p^i, i = 1 \dots N-1\} = \{i, i = 1 \dots N-1\}, \mathrm{mod}(N). \quad (19)$$

That is, the powers of p mod(N) are a reordering of $1 \dots N-1$. Then the DFT can be rewritten as

$$X_{p^{\tilde{k}}} - x(0) = \sum_{\tilde{n}=1}^{N-1} x_{p^{\tilde{n}}} e^{-j2\pi p^{\tilde{n}+\tilde{k}}} \quad (20)$$

where $p^{\tilde{i}} \equiv i$ mod(N). The DC value X_0 is computed separately as $X_0 = \sum_{n=0}^{N-1} x_n$.

This is a cyclic convolution of order $N-1$, so it can be computed using an $(N-1)$-point DFT. This is especially effective if N is one more than a power of two, e.g., N=257 or 65537.

B. Good-Thomas 2-D to 1-D FFT

The Good-Thomas FFT is a mapping between a 1-D and 2-D DFT using a residue number system to relate the indices. Let $N = N_1 N_2$ where N_1 and N_2

are relatively prime. Then define

$$n \equiv n_1 \bmod(N_1); \quad n \equiv n_2 \bmod(N_2) \tag{21}$$

The N_1N_2-point 1-D DFT X_k of x_n is related to the $(N_1 \times N_2)$-point 2-D DFT X_{k_1,k_2} of x_{n_1,n_2}, where

$$k \equiv N_1k_2 + N_2k_1 \bmod(N_1N_2) \tag{22}$$

So a 2-D DFT can be reformulated as a 1-D DFT, and the algorithm of this paper applied to images.

V. Numerical Example

We present a very small (N=101) example to illustrate the algorithm. The locations of known X_k and non-zero x_n are chosen randomly (while preserving conjugate symmetry locations), although randomness is unnecessary. The non-zero values of x_n are one, to facilitate the results. The total number of known X_k is 55 (27 values, their conjugates, and the DC value), and the number of non-zero x_n is 27.

The result of the method of frames (setting unknown $X_k = 0$) is shown in Figure #1. The unfilled circles are the true values of x_n, and the filled circles are the result of the method of frames. It can be seen that a threshold would recover many, but not all, of the locations of non-zero x_n.

The result of our algorithm is shown in Figure #2. Again the unfilled circles are the actual x_n, and the filled circles are the t_n computed from the known X_k. It can be seen that zero values of the t_n find all of the locatins of non-zero x_n (and some other possible values). The total number of zero values of t_N is 39, 12 more than the actual number of non-zero x_n, but fewer than the 55 known values of X_k. Hence a second POCS algorithm can be used to reconstruct the actual values of the non-zero x_n. The results of this algorithm are not shown.

The Matlab program used to generate these results:

```
clear;N=101;rand(’seed’,0);
K=find(round(rand(1,floor(N/2)))==1);
K=[1 K fliplr(N+2-K)];
rand(’seed’,0);X=round(rand(1,N)-1/4);
FX=fft(X);Y=zeros(1,N);Y(K)=FX(K);
%GOAL: Reconstruct sparse X from Y.
F=rand(1,N); %Initialization
for I=1:1000; %Iteration
%Project S onto S>0:
S=real(ifft(F));S=S.*(S>0);F=fft(S);
%Set F=0 except where Y is nonzero:
F=F.*(Y~=0); %Project F onto F*Y’=0:
F=F-Y*(F*Y’)/(Y*Y’);end; figure
stem(X),hold on,stem(S*100,’filled’)
```

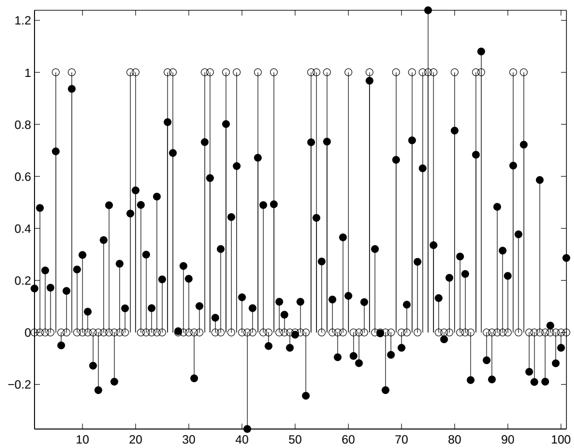

Fig. 1. Result of Method of Frames

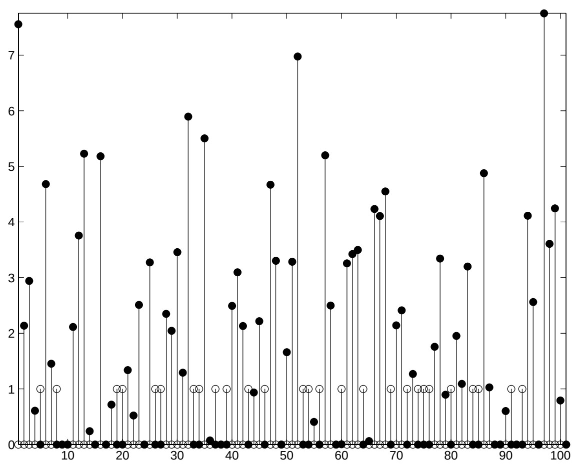

Fig. 2. Result of Algorithm: Indicator

.