AN L0 NORM BASED METHOD FOR FREQUENCY ESTIMATION FROM IRREGULARLY SAMPLED DATA

Md Mashud Hyder and Kaushik Mahata

School of Electrical Engineering and Computer Science, University of Newcastle, Callaghan, NSW 2308, Australia.

md.hyder@studentmail.newcastle.edu.au, kaushik.mahata@newcastle.edu.au

ABSTRACT
We present a frequency estimation method based on a sparse representation of irregular samples with an overcomplete basis. We enforce sparsity by imposing penalties based on an approximate $L_0$-norm. A number of recent theoretical results on compressed sensing justify this choice. Explicitly enforcing the sparsity of the representation is motivated by a desire to obtain a sharp estimate of the frequency spectrum that exhibits super-resolution. Our formulation leads to an optimization problem, which we solve efficiently in an iterative algorithm. The simulation results demonstrate that the proposed algorithm outperforms several other state-of-art methods.

Index Terms—Frequency estimation, spectral estimation, time series, compressed sensing, sparse representation.

1. INTRODUCTION
Consider a complex-valued continuous-time signal

$$y(t) = \sum_{k=1}^{n} a_k e^{j\Omega_k t},$$  \hspace{1cm} (1)

which is being sampled at time instants $\{t_k\}_{k=0}^{M-1}$. The problem is to estimate the complex-valued amplitudes $\{a_k\}_{k=1}^{n}$ and the frequencies $\{\Omega_k\}_{k=1}^{n}$ from the samples $\{y(t_k)\}_{k=1}^{M}$. This setting is motivated by some applications, experimental conditions cause irregular sampling. For example, earth’s daily rotation and annual revolution often causes periodic gaps in the observation. Prewhitenning techniques estimate frequencies by iterative deconvolution of the observed spectrum. Ad-hoc refinements such as CLEAN [1] or CLEANEST [2] has been used to improve the efficiency of prewhitening techniques. Recently, spectral analysis has been modeled as a linear inverse problem, and it is used as an efficient alternative to parametric methods to achieve high resolution [3, 4, 5, 6]. The spectrum is modeled according to (1), and spectral lines are estimated by locating a few nonzero values in the corresponding amplitude vector, i.e., by reconstructing a sparse vector. This makes (1) is well suited for compressed sensing (CS) [7, 8]. In this paper we pose the frequency estimation problem as a sparse signal recovery problem, and propose an estimation strategy called $L_0$ approximation for frequency estimation (LZA-F). The main contributions of our paper include adaptation of sparse signal representation to frequency estimation through $L_0$ approximation. In our experiments, the proposed approach exhibits a number of advantages over other frequency estimation techniques, which include increased resolution, and improved robustness to noise, and better performance with limited number of time samples.

2. AN APPROACH BASED ON COMPRESSED SENSING
Let $\delta$ be the largest real number such that there exits integers $N_1, \ldots, N_{M-1}$ as

$$t_k = t_0 + N_k \delta.$$  \hspace{1cm} (2)

Note that $\delta$ can even be very small depending on the set $\{t_k\}_{k=0}^{M-1}$. Then

$$y(t_k) = \sum_{\ell=1}^{n} a_{\ell} e^{j\Omega_{\ell}(t_0+N_{\ell}\delta)} = \sum_{\ell=1}^{n} b_{\ell} e^{jN_{\ell}\omega_{\ell}},$$

where we define $N_0 = 0$, $\omega_{\ell} = \delta\Omega_{\ell}$, $\ell = 1, \ldots, n$, and

$$b_{\ell} = a_{\ell} e^{j\Omega_{\ell} t_0}.$$ 

Now choose $N \geq N_{M-1}$, and consider a uniform frequency grid

$$G = \{ \tilde{\omega}_{\ell} = 2\pi\ell/N : \ell = 0, 1, \ldots, N - 1 \},$$

and define the DFT matrix $\Psi \in \mathbb{C}^{N \times N}$ such that

\begin{align*}
[\Psi]_{k,\ell} &= \frac{1}{\sqrt{N}} \exp(ik\tilde{\omega}_{\ell}) \\
&= \frac{1}{\sqrt{N}} \exp(i\ell/k). \hspace{1cm} (3)
\end{align*}

There may not exist any $\ell \in \{0, 1, \ldots, N - 1\}$ such that a given $\omega_k = \tilde{\omega}_{\ell}$, or in other words, it is likely that $\omega_k$ may not lie on the grid $G$. However, making $N$ large enough, the grid $G$ becomes sufficiently dense. Then for every $k \in \{1, \ldots, n\}$, we have $\omega_k \approx \tilde{\omega}_{\ell}$ for some $\ell \in \{0, 1, \ldots, N - 1\}$. Then we can write

$$z := [ y(t_0) \ y(t_0 + \delta) \ \cdots \ y(t_0 + (N-1)\delta) ]' = \Psi x + \epsilon,$$

where $\epsilon$ accounts for measurement noise and unmodelled residuals; the components of $x$ are mostly zero. Note that if $t_1 - t_0$ and $t_2 - t_0$ in (2) are uncommensurate, then there are no integers $N_1$ and $N_2$ such that $t_1 - t_0 = N_1\delta$ and $t_2 - t_0 = N_2\delta$. These equalities, however can be approximately be verified if $\delta$ is sufficiently small and the integers $N_1$ and $N_2$ are accordingly large. Therefore, equation (2) holds only approximately. It would be therefore useful account for such an approximation in the subsequent data model. This error

The research is supported by the Australian Research Council.

Index Terms—Frequency estimation, spectral estimation, time series, compressed sensing, sparse representation.
in the sampling instants reduces to a multiplicative complex exponential term for each element of the matrix $\Psi$. Very small values of $\delta$ can make such an error negligible.

The $k$-th component $x_k$ is non-zero only if $\bar{\omega}_k = \omega_k$ for some $k \in \{1, \ldots, n\}$. This makes $x$ as a sparse vector, and the locations of the dominant peaks in $x$ indicates the frequencies present in the data, while the dominant values give the complex-valued amplitudes.

If we know $z$ we could estimate $x$ by inverting $\Psi$, as the DFT matrix is nonsingular. However, we have

$$
y := [y(t_0), y(t_1), \ldots, y(t_{M-1})]' = \Theta \Psi x + \Theta e,
$$

$$
= \Phi x + e
$$

where $\Theta \in \mathbb{R}^{M \times N}$ is formed such that the $k$-th row of $\Theta$ is the same as the $N_k + 1$-th row of $N \times N$ identity matrix. Note that $\Phi$ can be seen as a random partial Fourier matrix and in that case $\Phi$ has a small restricted isometry constant [9]. In order to guarantee unique identifiability of $x$ from $y$ we must have

$$M \leq N/2,
$$

The ability to recover $x$ uniquely from $y$ is not sufficient to guarantee identifiability of the frequencies $\{\Omega_k\}_{k=1}^p$. Any sampling process leads to aliasing. For this reason a feasible set $\mathcal{O}$ must be known, such that $\Omega_k \in \mathcal{O}$, $k = 1, \ldots, n$. In addition, the intersection between $\mathcal{O}$ and any of its aliased version should be empty, i.e. $\mathcal{O} \cap (\mathcal{O} + \pi k/d) = \emptyset$ for every nonzero $k \in \mathbb{Z}$. Note that above condition does also allow $\mathcal{O}$ to be made up of several smaller mutually disjoint intervals.

The standard approach to estimate $x$ from noisy observation $y$ is to solve an optimization problem:

$$
\hat{x}_P = \arg \min_x \|x\|_p + \lambda \|y - \Phi x\|^2_F.
$$

The parameter $\lambda$ controls the tradeoff between the sparsity of the signal and the residual energy [10]. For $p = 2$, differentiating (5) and equating to zero we have

$$
\hat{x}_{(2)} - \lambda \Phi^* (y - \Phi \hat{x}) = 0 \Rightarrow \hat{x} = \lambda (I + \lambda \Phi^* \Phi)^{-1} \Phi^* y,
$$

where $\Phi^*$ indicates conjugate transpose of $\Phi$. After simplifying (6) using the matrix inversion lemma gives

$$
\hat{x}_{(2)} = \lambda [I - \Phi^*(1/\lambda + \Phi \Phi^*)^{-1}] \Phi^* y = \frac{\lambda}{\lambda + 1} \Phi^* y.
$$

Note that in above we use $\Phi \Phi^* = I$. Now $\Phi^* y = \Psi^* (\Theta^* y)$, which is the DFT of $\Theta^* y$. Also if there is no noise then one takes $\lambda \to \infty$, and thus $\hat{x}_{(2)}$ is same as the DFT of $\Theta^* y$. In any case one can use the FFT algorithm to compute $\hat{x}_{(2)}$.

The minimum two-norm solution $\hat{x}_{(2)}$ is not sparse in general. Sparsity is best enforced by making $p = 0$. Since the choice $p = 0$ leads to a NP-hard problem, several relaxations have been proposed [10, 6], where one sets $p = 1$ and use a Basis Pursuit type approach. However, setting $p = 1$ often makes the algorithms sensitive to additive noise. For this reason we prefer using $p = 0$. As $\|\hat{x}\|_0$ is a non-smooth function, we approximate it using Gaussian functions [11, 12].

Defining the Gaussian function

$$f_\sigma(\alpha) := e^{-\frac{\alpha^2}{2\sigma^2}}, \ \alpha \in \mathbb{R},
$$

it is readily verified that

$$F_\sigma(x) := \sum_{j=1}^N f_\sigma(|x_j|)
$$

behaves like $N - \|x\|_0$ when $\sigma \to 0$. This motivates the following way of solving $x$ given $y$ in (4):

$$
x_*(\sigma) = \arg \min_x \ L_\sigma(x),
$$

$$
L_\sigma(x) := -F_\sigma(x) + \frac{\lambda}{2} \|y - \Phi x\|^2_F.
$$

for small values of $\sigma$. Just like $\|x\|_0$, its approximation $N - F_\sigma(x)$ has many local minima for small values of $\sigma$. Nevertheless, as $\sigma$ increases, $F_\sigma(x)$ becomes smoother. In fact, $x_*$ admits a closed form solution as $\sigma \to \infty$. To see this first note that $\lim_{\sigma \to \infty} f_\sigma(\alpha) = 1$. Using this in (9) we have

$$
\lim_{\sigma \to \infty} x_*(\sigma) = \Phi^* (\Phi \Phi^*)^{-1} y = \Psi^* (\Theta^* y),
$$

the DFT of $\Theta^* y$. Hence, we take a large $\sigma$ initially and solve (9). Subsequently, $\sigma$ is reduced by some small factor and (9) is solved again. The procedure is repeated until a convergence criterion is satisfied. However, it is not recommended to solve (9) (for a fixed $\sigma$) using Gauss-Newton type algorithms because the Hessian of $L_\sigma$ is not positive definite in general.

Differentiating $L_\sigma$ and equating to zero one can show that

$$x_*(\sigma) = \zeta(x_*(\sigma)),
$$

where the mapping $\zeta : \mathbb{C}^N \to \mathbb{C}^N$ is defined as

$$\zeta(x) = \lambda \left[ W(x)/\sigma^2 + \lambda \Phi^* \Phi \right]^{-1} \Phi^* y,
$$

It is tempting to use (10) to set up a fixed point iteration to solve $x_*$. However there is no guarantee that $L_\sigma(\zeta(x)) \leq L_\sigma(x)$, which one would require to ensure the convergence of the fixed point iteration. Nevertheless, it can be shown that $L_\sigma(x)$ is decreasing along $\zeta(x) - x$. Furthermore moving to $\zeta(x)$ from $x$ is same as taking the Newton step associated with the convex-concave procedure to minimize $L_\sigma$. The proofs of these results are omitted for brevity. These observations we propose the algorithm in Table 1 to recover sparse $x$ from $y$. In the algorithm, $\hat{x}^{(i)}$ denotes the value of $x$ updated at $i$th iteration. We set $\sigma = 4 \max\|x^{(0)}\|_1$ for the first iteration [11]. The value of $\lambda$ depends on noise level. Experimental results suggests that $\lambda = 50$ is good choice. For minimizing $L_\sigma$ we use $\zeta(x) - x$ as descent-direction. By choosing a proper factor $\nu i.e. 0 < \nu \leq 1$, the algorithm can be accelerated to converge the optimal solution rapidly. We fixed $\gamma$ to 0.5. The inner-iteration for a given $\sigma$ terminates when $\tau(i) < \eta_\sigma$ (see step 6). A wide range of numerical experiments suggests that the good setting of $\eta_\sigma$ is 0.5. The stopping criterion of LZA-F is based on a small $\sigma$ denoted by $\sigma_0$. Upon convergence of each inner iteration we lower $\sigma$ by a factor $\rho$ (step 6). When $\sigma$ becomes smaller than $\sigma_0$, we do not update $\hat{x}^{(i)}$ further and stop iterations. The final value of $\sigma_0$ is set to 0.05 and $\rho = 0.3$.

The major fraction of the computation is involved in computing $\zeta(x)$. However, by denoting $\kappa = \lambda \sigma^2$, and using the matrix inversion lemma in (10) one can verify that

$$\zeta(x) = \{\kappa W^{-1} - \kappa^2 W^{-1} \Phi^*[I + \kappa \Phi W^{-1} \Phi^*]^{-1} \Phi W^{-1}\} \Phi^* y
$$

$$= \kappa W^{-1} \Phi^* [I + \kappa \Phi W^{-1} \Phi^*]^{-1} y
$$

$$= W^{-1} \Psi^* \Theta^* [I/\kappa + \Theta W^{-1} \Psi^* \Theta^*]^{-1} y.
$$
Consider a discrete-time stochastic process \( x(t) \) with a power spectral density \( \Sigma(e^{j\omega}) \). An IFFT of \( \Sigma(e^{j\omega}) \) yields the covariance sequence of \( u(t) \). Subsequently one can generate \( R \) by arranging the elements of the covariance sequence in the standard Toeplitz covariance matrix. Since \( \Theta \) is essentially a row-selector, \( \Theta R \Theta^* \) is a small block of \( R \) which can be formed in a straightforward manner. Next we need to solve the \( M \times M \) positive definite system

\[
(\Theta R \Theta^* + I/\kappa) z = y.
\]

Subsequently, in the aim of computing \( \zeta(x) \) in (13), we compute \( \hat{z} = \Psi^* (\Theta^* z) \) by computing the FFT of \( \Theta^* z \). Again, forming \( \Theta^* z \) does not require any multiplication. Finally the diagonal structure of \( W \) allows computing \( W^{-1} \hat{z} \) using \( N \) multiplications. It is worthwhile to note that for uniform sampling case \( \Theta R \Theta^* + I/\kappa \) will be a Toeplitz matrix. Hence solving (14) can be accomplished by a fast Toeplitz system solver.

### 3. SIMULATION RESULTS

The artificial data set of this simulation is similar to the astronomical data set described in [2] (Data set A), which consist in three sinusoids with periods 370, 300 and 100 days and amplitudes 3, 2.828 and 3, respectively. An initial data set was generated with 200 points sampled every 10 days. The initial dataset is resampled so that a gap of 100 days is introduced after every 360 days. To make the problem a bit trickier, we follow a strategy used in [13]. A fourth sinusoid was added with period 122.5 days and amplitude 3, such that the sidelobes caused by the annual gaps (for periods 370 and 122.5 day) superimpose at a period of 184 days (1/122.5 - 1/365 = 1/370 + 1/365 = 1/184), generating a high false peak in the Fourier spectrum. Hence there are four frequencies in the actual signal \( [1/370, 1/300, 1/122.5, 1/100] = [0.0027, 0.0033, 0.0082, 0.01] \) cycles/day. When generating signal we add random phase \( [0, 2\pi] \) to every sinusoid. It is further assumed that the time samples \( \{y_n\} \) are corrupted by uncorrelated zero mean noise sequence. Since the smallest sampling instants is \( \delta = 10 \) days, hence \( f_{\text{max}} \) should be set to less than \( 0.5 + 1/10 = 5.10^{-2} \) c/d. We set \( f_{\text{max}} = 5^{-2} \) and \( N = 2000 \). We compare the performance of LZA-F with SparSpec \(^1\) [13], Iterative Reweighted Least-Squares (IRLS) for complex valued variable [6] and \( \ell_1 \). For \( \ell_1 \) we use the following convex optimization problem,

\[
x_+ = \min ||x||_1 \text{ subject to } ||\Phi^*(y - \Phi x)||_\infty < \mu \tag{15}
\]

where \( \mu = 3 \). Our simulations are performed in MATLAB7 environment using an Intel Core 2 Duo, 2.66 GHz processor with 4GB of memory, under Mac OS X Version 10.5.5 operating system.

In Figure 1 we compare the frequency spectrum obtained using different methods with 120 time samples. Note that \( \ell_1 \) and IRLS can produce picks around the original locations. However, they generate many sidelobes. Only LZA-F has sharp estimation of actual frequencies. Figure 2 shows the probability of detecting frequencies as a function of \( M \) when \( \text{SNR} = 15 \) dB. Performance-wise, LZA-F exhibits clear superiority over the others. Where SparSpec needs 150 samples to estimate frequencies LZA-F takes only 120 samples.

In Figure 3 demonstrate the performance of different algorithms in presence of varying noise. The interesting observation is that at higher SNR all algorithms perform similarly, however as SNR decreases most of them fails to locate frequencies. Note that above 20dB SNR, number of samples changes linearly for all algorithms. However, after 20 dB the requirement of \( M \) increases rapidly for

---

three algorithms whereas LZA-F still perform linearly. Moreover, at lower SNR other algorithms can not resolve frequencies whatever $M$ increases. Because, at lower SNR they generate lots of spurious peaks. However, LZA-F still able to generate sharp estimation of all four frequencies are averaged to present the figure. Finally Figure 4 shows the bias property of different algorithms as a function of SNR. We take 150 samples and compute frequencies recovered by different algorithms. The bias in all four frequencies are averaged to present the figure.

4. REFERENCES


Fig. 2. Percentage of recovery (%).

Fig. 3. Minimum number of samples needed to recover actual frequencies as a function of SNR. Actual frequencies $[0.0027, 0.0033, 0.0082, 0.01]c/d$.

Fig. 4. Frequency bias as a function of SNR. Actual frequencies $[0.0027, 0.0033, 0.0082, 0.01]c/d$, $M = 150$. 4025