PROSPARSE DENOISE: PRONY'S BASED SPARSE PATTERN RECOVERY IN THE PRESENCE OF NOISE

Jon Oñativia †*, Yue M. Lu* ⋆ *and Pier Luigi Dragotti* †

† Communications and Signal Processing Group (CSP), Imperial College London, UK
{ jon.onativia, p.dragotti } @imperial.ac.uk

⋆ Signals, Information, and Networks Group (SING), Harvard University, USA
yuelu@seas.harvard.edu

ABSTRACT

We present a novel algorithm — ProSparse Denoise — that can solve the sparsity recovery problem in the presence of noise when the dictionary is the union of Fourier and identity matrices. The algorithm is based on a proper use of Cadzow routine and Prony's method and exploits the duality of Fourier and identity matrices. The algorithm has low complexity compared to state of the art algorithms for sparse recovery since it relies on the Fast Fourier Transform (FFT) algorithm. We provide conditions on the noise that guarantees the correct recovery of the sparsity pattern. Our approach outperforms state of the art algorithms such as Basis Pursuit Denoise and Subspace Pursuit when the dictionary is the union of Fourier and identity matrices.

Index Terms— Sparse representation, union of bases, Prony's method, denoising.

1. INTRODUCTION

We consider the problem of finding the sparse representation of a signal in the union of Fourier and canonical bases when the observation has been corrupted with additive noise. Let the noiseless observation be given by

$$y[n] = \frac{1}{\sqrt{N}} \sum_{k=1}^{K_p} a_k \, e^{j \frac{2\pi}{N} m_k n} + \sum_{k=1}^{K_q} b_k \, \delta[n - n_k], \quad (1)$$

where $0 \leq n < N$. The parameters $0 \leq m_1 < \ldots < m_{K_p} < N$ and $0 \leq n_1 < \ldots < n_{K_q} < N$ are integers that correspond to the indices of the atoms that form the observed signal and $a_k, b_k \in \mathbb{C} \setminus \{0\}$ their amplitudes. When noise is present, the observed signal is given by

$$\tilde{y}[n] = y[n] + \varepsilon[n], \qquad 0 \leq n < N, \quad (2)$$

This work is supported by the European Research Council (ERC) starting investigator award Nr. 277800 (RecoSamp).

where $\varepsilon[n] = \varepsilon^R[n] + j\,\varepsilon^I[n]$ are i.i.d. random variables for $0 \leq n < N$ with the real and imaginary parts drawn from $\mathcal{N}(0, \sigma_\varepsilon^2)$. The observed signal can be written in matricial form as follows $\boldsymbol{y} = \boldsymbol{D}\,\boldsymbol{x} + \boldsymbol{\varepsilon} = [\boldsymbol{F},\, \boldsymbol{I}]\,[\boldsymbol{x}_p^T,\, \boldsymbol{x}_q^T]^T + \boldsymbol{\varepsilon}$ where $\boldsymbol{y}, \boldsymbol{\varepsilon}, \boldsymbol{x}_p, \boldsymbol{x}_q \in \mathbb{C}^N$ and $\boldsymbol{x} \in \mathbb{C}^{2N}$.

The Compressed Sensing framework [3, 4, 8] presents a wide range of algorithms to solve the noisy sparsity recovery problem. The two traditional algorithms to solve this problem are a variation of the Basis Pursuit algorithm known as Basis Pursuit Denoising (BPDN), and the Orthogonal Matching Pursuit (OMP) algorithm [3, 5, 7, 8, 10, 12, 16]. In [17], the Least Absolute Shrinkage and Selection Operator (LASSO) method was presented to solve the sparse recovery problem. More recently, a new set of greedy algorithms have been developed that present lower complexity than OMP with similar performance guarantees: Compressive Sampling Matching Pursuit (CoSaMP) [13] and Subspace Pursuit [6].

Recently a new method which exploits in full the special structure of the dictionary $\boldsymbol{D} = [\boldsymbol{F},\, \boldsymbol{I}]$ has been proposed. The algorithm is based on a variation of Prony's method and is therefore called ProSparse (Prony's based sparsity) [9]. In the noiseless setting, ProSparse can recover signals over a much wider set of sparsity levels than what BP can achieve [14, 15].

In this paper, we present a new algorithm that handles the noisy sparse recovery problem of [3] and that is still based around Prony's ideas. In particular we couple the traditional Prony's method with Cadzow signal enhancement algorithm [2] and call the new method ProSparse Denoise. The key insight is that, for the observed signal given as in (1) and (2), if we assume that the spikes are also part of the noise, we can apply Cadzow's algorithm to try to recover the original Fourier atoms. Using duality we can also recover the spikes by treating the Fourier atoms as noise. The approach leads to a very fast algorithm that also outperforms state of the art methods.

The rest of the paper is organised as follows: Section 2 presents the Cadzow denoising algorithm particularised to our setup. Section 3 describes the novel ProSparse Denoise algo-

Algorithm 1 *Fast Cadzow*—Denoising algorithm for circulant matrices

Input: N noisy samples $\tilde{y}[n]$, number of complex exponentials K.
Output: Denoised samples $y'[n]$.

1: $\hat{y}[m] = \text{DFT}_N\{\tilde{y}[n]\}$.
2: Set to zero $N-K$ samples $\hat{y}[m]$ that correspond to the smallest values of $|\hat{y}[m]|$.
3: $y'[n] = \text{IDFT}_N\{\hat{y}[m]\}$

rithm. Section 4 is an analysis of the algorithm from a probabilistic point of view to establish a guaranteed performance condition for the simple case where there is one Fourier atom and one spike. Section 5 presents simulation results. We then conclude in Section 6.

2. FINITE-DIMENSIONAL CADZOW AND CIRCULANT MATRICES

In our context, the Cadzow algorithm is used to denoise samples that are given by a sum of exponentials corrupted with additive noise. If we consider that the spikes are also part of the noise, the noisy samples $\tilde{y}[n]$ can be written as follows:

$$\tilde{y}[n] = \sum_{k=1}^{K_p} \alpha_k \, u_k^n + \eta[n], \tag{3}$$

where the signal of interest corresponds to the sum of K_p exponentials and the term $\eta[n]$ includes the spikes and the actual noise term $\varepsilon[n]$ from (2). The Cadzow algorithm is based on building the following Toeplitz matrix:

$$\tilde{\boldsymbol{Y}}^{toep} = \begin{bmatrix} \tilde{y}[M] & \tilde{y}[M-1] & \dots & \tilde{y}[0] \\ \tilde{y}[M+1] & \tilde{y}[M] & \dots & \tilde{y}[1] \\ \vdots & \vdots & \ddots & \vdots \\ \tilde{y}[P] & \tilde{y}[P-1] & \dots & \tilde{y}[P-M] \end{bmatrix}, \tag{4}$$

where the matrix is built with a number of rows and columns which are larger than the number of exponentials K_p, and is made as square as possible. We note that in the absence of noise this matrix has rank K_p (see [1, 9]). Therefore, Cadzow algorithm imposes the following two properties sequentially at each iteration: 1. Find the closest matrix of rank K_p (in the Frobenius norm sense) by computing the SVD of $\tilde{\boldsymbol{Y}}^{toep}$ and setting to zero the smallest singular values. 2. Impose a Toeplitz structure by averaging over the diagonal elements. These two properties are applied until a stopping condition is reached. This condition can be a maximum number of iterations or the difference between the singular values of different iterations being below a threshold. An interesting property of this algorithm is that it can also be used to remove the spikes of a signal that is made of Fourier atoms and spikes.

2.1. Fast circulant Cadzow algorithm

In our particular case, the parameters u_k in (3) are complex-valued and lie on a grid of size N on the unit circle: $u_k = \exp\left(j\frac{2\pi}{N}m_k\right)$, where $k = 1, \dots, K_p$, and $0 \leq m_1 < \dots < m_{K_p} < N$. Since we have access to N samples $\tilde{y}[n]$, and due to the periodicity of the Fourier atoms, the Toeplitz matrix can be extended to the following noisy circulant matrix of size $N \times N$:

$$\tilde{\boldsymbol{Y}}^{circ} = \begin{bmatrix} \tilde{y}[0] & \tilde{y}[N-1] & \dots & \tilde{y}[1] \\ \tilde{y}[1] & \tilde{y}[0] & \dots & \tilde{y}[2] \\ \vdots & \vdots & \ddots & \vdots \\ \tilde{y}[N-1] & \tilde{y}[N-2] & \dots & \tilde{y}[0] \end{bmatrix}. \tag{5}$$

This matrix is of full rank N due to the presence of noise. The *noiseless* matrix $\boldsymbol{Y}^{circ}$ with samples $y[n]$ that are only due to the complex exponentials, satisfies the rank deficiency and the Toeplitz structure properties that are required to apply Cadzow denoising algorithm. Therefore, we can also apply the Cadzow strategy to denoise the circulant matrix $\tilde{\boldsymbol{Y}}^{circ}$. It is easy to verify that the noiseless matrix is of rank K_p since it can be diagonalised as follows: $\boldsymbol{Y}^{circ} = \boldsymbol{F}\,\boldsymbol{\Lambda}\,\boldsymbol{F}^H$, where $\boldsymbol{\Lambda} = \text{diag}\left(\hat{y}[m]\right)_{m=0}^{N-1}$ and $\hat{y}[m] = \text{DFT}_N\{y[n]\} = \sum_{n=0}^{N-1} y[n] \exp\left(j2\pi mn/N\right)$ has exactly K_p non-zero elements at $m = m_k$, $k = 1, \dots, K_p$. Due to the circulant structure of this matrix, it follows that imposing the first property to $\tilde{\boldsymbol{Y}}^{circ}$ (rank deficiency) yields a circulant matrix, and therefore the resulting algorithm is not iterative. That is, we do not need to impose the second property since it is already satisfied and the Cadzow denoising algorithm stops after imposing the first property in the first iteration. The resulting denoising algorithm is described in Algorithm 1. This algorithm is fast since it only needs to compute one DFT and one inverse DFT which are performed using the Fast Fourier Transform (FFT) algorithm. The complexity of this algorithm is $\mathcal{O}(N \log N)$. This is considerably better than the complexity of the original Cadzow approach since the latter requires computing an SVD at each iteration which has a complexity of $\mathcal{O}(N^3)$.

3. NOISY SPARSE RECOVERY BASED ON CADZOW AND PRONY

The strategy to remove spikes that has been presented in the previous section is of particular interest to recover sparse vectors in the scenario where the dictionary is the union of Fourier and identity matrices. Note that when the spikes are removed, part of the energy, and therefore of the amplitudes, of the Fourier atoms is also removed. Therefore, instead of removing all the spikes at once and getting an estimate of the Fourier atoms, we can follow a different strategy where only one spike is removed at each iteration. The general idea of the algorithm is that if we compute the difference between the original observation $\boldsymbol{y}$ and the Cadzow denoised vector $\boldsymbol{y}'$, this residual will mainly contain the spikes and the noise. We also assume that the power of the noise is small compared to that of the the spikes. We estimate one spike at each iteration, remove the contribution of this spike, and iterate again until all the spikes have been removed. Once the spikes have been

Algorithm 2 *ProSparse Denoise*—Union of Fourier and identity matrices

Input: Noisy vector $\boldsymbol{y} = [\boldsymbol{F},\ \boldsymbol{I}]\,\boldsymbol{x} + \boldsymbol{\varepsilon}$ and sparsity (K_p, K_q).
Output: (K_p, K_q)-sparse vector $\tilde{\boldsymbol{x}} = [\boldsymbol{x}_p^T,\ \boldsymbol{x}_q^T]^T$.

```
Initialise spikes x_q = 0.
Initialise indices Ω = {0, 1, ..., N − 1}.
Denoise y' = Cadzow(y, K_p).
for i = 1 to K_q do
    Compute residual r = y − y'.
    Estimate spike location n_0 = arg max_{n∈Ω} {|r[n]|}.
    Store spike x_q[n_0] = r[n_0].
    Remove spike location from indices Ω ← Ω \ {n_0}.
    Remove spikes to the observation y' = y − x_q.
    Denoise y' ← Cadzow(y', K_p).
end for
Estimate Fourier atoms x_p = Prony(y', K_p).
```

removed, the Fourier atoms are estimated from this *cleaned* vector $\boldsymbol{y}'$ by applying Prony's method.

This approach provides an estimate for the entire vector $\boldsymbol{x}$, that is, it estimates both, the Fourier atoms and the spikes. However, in practice, ProSparse Denoise uses this algorithm only to estimate the Fourier atoms. The same approach is then applied to $\overline{\boldsymbol{F}\,\boldsymbol{y}}$ to estimate the spikes since, in $\overline{\boldsymbol{F}\,\boldsymbol{y}}$, the spikes correspond to complex exponentials. We then build the entire sparse vector $\boldsymbol{x}$ by merging the Fourier atoms estimated from $\boldsymbol{y}$ and the spikes estimated from $\overline{\boldsymbol{F}\,\boldsymbol{y}}$. Note that both computations can be performed in parallel. The complete method is described in Algorithm 2.

4. PERFORMANCE ANALYSIS

The signal model presented in (1) and (2) allows us to perform some probabilistic analysis of the performance of the algorithm for a worst case scenario. In particular, we can establish a sufficient condition for the algorithm to succeed for the simple case where there is only one Fourier atom and one spike. This condition is then validated with numerical simulations.

The first step of the denoising algorithm operates in the Fourier domain and selects the Fourier atoms by picking the samples with the largest amplitudes. When there is one Fourier atom and one spike, the worst case scenario corresponds to the case where the noise and the Fourier transform of the spike sum destructively with the Fourier atom, the amplitude of the noise at this location is maximum, and there is another location where the noise also reaches the maximum amplitude. We first note that since our noise model is based on complex-valued i.i.d. Gaussian random varibles, the absolute value of the noise term, that is $|\varepsilon[n]| = \left|\varepsilon^R[n] + j\,\varepsilon^I[n]\right|$, follows a Rayleigh distribution. It can be verified that for large N, the maximum value of a collection of N random variables that follow a Rayleigh distribution with parameter σ_ε is equal to $\sigma_\varepsilon\sqrt{2\log N}$ with high probability.

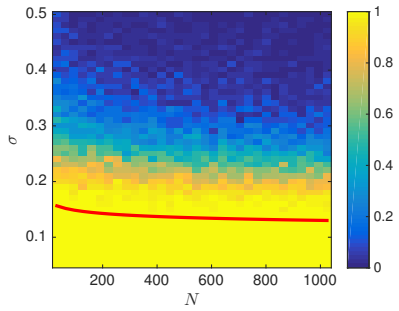


Fig. 1: Guaranteed performance of noisy sparse recovery with 1 Dirac and 1 Fourier atom for different levels of noise. 50 iterations per noise level. The success is measured by checking that the support is correctly retrieved at each iteration. The amplitudes of the atoms are fixed and are equal to $1/\sqrt{2}$ for both the real and imaginary parts. The location of the Fourier atom is $m_1 = 1$ and the location of the spike, n_1, satisfies (7). The red line depicts the equation $\sigma = (1 - 1/\sqrt{N})(\sqrt{8\log N})$ from (8).

Let us assume that the Fourier atom and the spike have the same amplitude: $a_1^R = a_1^I = b_1^R = b_1^I = 1/\sqrt{2}$. If the Fourier atom is located at $m = m_1$, the noiseless sample of the Fourier transform at $m = m_1$ is given by

$$\hat{y}[m_1] = e^{j\pi/4} + \frac{1}{\sqrt{N}}\, e^{j(\pi/4 - 2\pi n_1 m_1/N)}. \tag{6}$$

The Fourier atom and the spike sum destructively when the phase of the second complex exponential in the previous equation has a difference of π radians (modulo 2π) with the phase of the first complex exponential, that is:

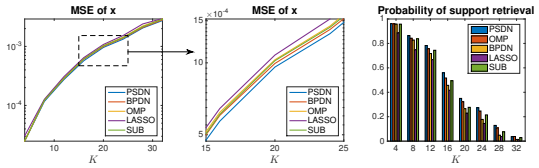
$$\mathrm{mod}\,(n_1 m_1, N) = \frac{N}{2}. \tag{7}$$

This establishes a condition on m_1 and n_1 to have a destructive interaction between the Fourier atom and the Fourier transform of the spike. In that case, the resulting amplitude is given by $|y[m_1]| = 1 - \frac{1}{\sqrt{N}}$.

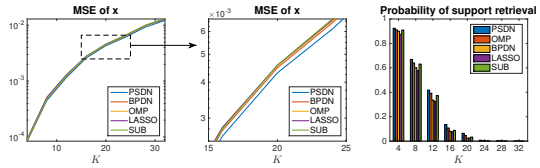
In the worst case scenario, the amplitude of the noise at location $m = m_1$ is maximum, that is $|\varepsilon[m_1]| = \sigma_\varepsilon\sqrt{2\log N}$, and sums destructively with the Fourier atom's amplitude. Moreover, we assume that another location $m \neq m_1$ also presents a noise sample with this same amplitude. We can therefore establish the following sufficient success condition that guarantees that in the worst case scenario we are still able to detect the Fourier atom: $1 - \frac{1}{\sqrt{N}} - \sigma_\varepsilon\sqrt{2\log N} > \sigma_\varepsilon\sqrt{2\log N}$, which leads to

$$\sigma_\varepsilon < \frac{1 - 1/\sqrt{N}}{\sqrt{8\log N}}. \tag{8}$$

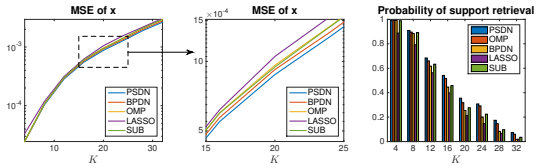
Figure 1 illustrates an empirical validation of the derivation in Equation (8). For different sizes of the problem, and different levels of noise, the sparse vector is reconstructed applying the algorithm described in the previous section. The red line depicts the bound established in Equation (8). We can observe that when the noise has a power below this bound the algorithm always succeeds, confirming the predicted result.



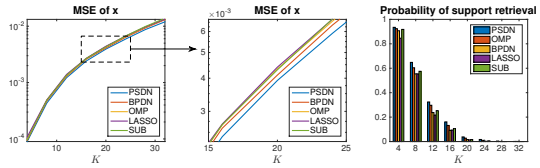
(a) SNR = 10 dB, bias = 50%.



(b) SNR = 5 dB, bias = 50%.



(c) SNR = 10 dB, bias = 25%.



(d) SNR = 5 dB, bias = 25%.

Fig. 2: Union of Fourier and identity bases, each of size $N \times N$ with $N = 256$. Simulation results with $K_q = \text{bias} \cdot K$ spikes and $K_p = K - K_q$ Fourier atoms. 1000 realisations per sparsity level (100 non-zero locations uniformly drawn at random and for each realisation of the non-zero locations 10 realisations of the amplitudes drawn from $\mathcal{N}(0, 1)$ for the real and imaginary parts). Results denoted by PSDN correspond to the proposed algorithm.

5. SIMULATION RESULTS

The noisy sparse recovery algorithm has been tested against state of the art algorithms such as BPDN, OMP, LASSO and Subspace Pursuit (denoted by SUB). Figure 2 presents the results where we have measured the MSE of the recovered sparse vector and the probability that the algorithms recover the exact support of the original sparse signal. Specifically, for a given sparsity level and SNR, 100 different supports of the sparse vector $\boldsymbol{x} \in \mathbb{C}^{2N}$ are generated uniformly at random; and for each realisation of the support, 10 different realisations of the amplitudes of $\boldsymbol{x}$ and of the the noise vector $\boldsymbol{\varepsilon}$ are generated such that the observation $\boldsymbol{y} = \boldsymbol{D}\,\boldsymbol{x} + \boldsymbol{\varepsilon}$ satisfies the specified SNR. All the amplitudes are Gaussian distributed and complex-valued. At each realisation, all the algorithms reconstruct a sparse vector from the noisy observation $\boldsymbol{y}$. Simulations have been performed for 4 different scenarios that correspond to SNR levels of 5 and 10 dB with a bias in the number of atoms from one dictionary with respect to the other of 25 and 50 %. A bias of 25 % means that there are $K/4$ spikes and $3K/4$ Fourier atoms (the different values of K are chosen so that these numbers are always integers). From the results, it is clear that ProSparse Denoise consistently outperforms state of the art algorithms at all noise and sparsity levels.

Besides the gain in performance, it is also important to note that this novel algorithm is faster than the other algorithms in the majority of scenarios. Execution times have been measured during these simulations and are summarised in Table 1. These measurements are obtained by averaging over the multiple realisations of each sparsity level. The experiments have been run using a commercial laptop (tested on a 2.5GHz Intel Core i5 CPU) and all the algorithms were implemented in MATLAB. We used the CVX package to implement the BPDN optimisation problem because it was giving the best performance compared to other optimisation toolboxes [11]. LASSO was tested using MATLAB's implementation. OMP has been implemented for the simulations and the implementation of Subspace Pursuit downloaded from the authors' website. For sparsity levels that go beyond 16 for $N = 256$ ProSparse Denoise is the fastest of all the algorithms.

Table 1: Average execution time of sparse recovery algorithms, $N = 256$.

K	PSDN	BPDN	OMP	LASSO	SUB
4	0.0098	33.0813	0.2442	0.1503	**0.0055**
8	0.0112	29.3087	0.4779	0.1565	**0.0087**
12	0.0136	27.8360	0.7159	0.1722	**0.0110**
16	**0.0151**	27.2135	0.9560	0.1921	0.0165
20	**0.0171**	26.7477	1.1906	0.2038	0.0195
24	**0.0202**	26.7861	1.4324	0.2183	0.0230
28	**0.0216**	25.8907	1.6440	0.2318	0.0251
32	**0.0203**	22.9313	1.6568	0.2218	0.0250

6. SUMMARY

A novel algorithm has been presented to solve the sparse recovery problem in the noisy scenario. This new approach is based on an extension of the Cadzow denoising algorithm for the finite-dimensional case. This extension is combined with an iterative spike removal algorithm to obtain a *cleaned* signal that only contains Fourier atoms. These atoms are then estimated using Prony's method. The overall algorithm is able to solve the sparsity problem faster, and with higher precision, than state of the art algorithms. We note that traditional compressed sensing methods are more flexible, in the sense that they can solve the sparsity problem for generic dictionaries. However, our method outperforms these algorithms because it fully exploits the particular structure of the dictionary at hand: the union of Fourier and identity matrices.

References

[1] T. Blu, P. L. Dragotti, M. Vetterli, P. Marziliano, and L. Coulot. "Sparse Sampling of Signal Innovations". *IEEE Signal Processing Magazine* 25.2 (Mar. 2008), pp. 31–40.

[2] J. A. Cadzow. "Signal Enhancement–A Composite Property Mapping Algorithm". *IEEE Transactions on Accoustics, Speech and Signal Processing* 36.1 (Jan. 1988), pp. 49–62.

[3] E. J. Candès, J. Romberg, and T. Tao. "Robust Uncertainty Principles: Exact Signal Reconstruction From Highly Incomplete Frequency Information". *IEEE Transactions on Information Theory* 52.2 (Feb. 2006), pp. 489–509.

[4] E. J. Candès and M. B. Wakin. "An Introduction To Compressive Sampling". *IEEE Signal Processing Magazine* 25.2 (Mar. 2008), pp. 21–30.

[5] S. S. Chen, D. L. Donoho, and M. A. Saunders. "Atomic Decomposition by Basis Pursuit". *SIAM Journal on Scientific Computing* 20.1 (1998), pp. 33–61.

[6] W. Dai and O. Milenkovic. "Subspace Pursuit for Compressive Sensing Signal Recontruction". *IEEE Transactions on Information Theory* 55.5 (May 2009), pp. 2230–2249.

[7] G. Davis, S. G. Mallat, and Z. Zhang. "Adaptive Time-frequency Decompositions". *SPIE Journal on Optical Engineering* 33.7 (July 1994), pp. 2183–2191.

[8] D. L. Donoho. "Compressed Sensing". *IEEE Transactions on Information Theory* 52.4 (Apr. 2006), pp. 1289–1306.

[9] P. L. Dragotti and Y. M. Lu. "On Sparse Representation in Fourier and Local Bases". *IEEE Transactions on Information Theory* 60.12 (Dec. 2014), pp. 7888–7899.

[10] M. Elad. *Sparse and Redundant Representations: From Theory to Applications in Signal and Image Processing*. 1st ed. Springer, Aug. 2010.

[11] M. Grant and S. Boyd. "CVX: Matlab software for disciplined convex programming, version 2.1". *http://cvxr.com/cvx* (Sept. 2015).

[12] S. G. Mallat and Z. Zhang. "Matching Pursuits with Time-Frequency Dictionaries". *IEEE Transactions on Signal Processing* 41.12 (Dec. 1993), pp. 3397–3415.

[13] D. Needell and J. A. Tropp. "CoSaMP: Iterative signal recovery from incomplete and inaccurate samples". *Applied and Computational Harmonic Analysis* 26.3 (May 2009), pp. 301–321.

[14] J. Oñativia, P. L. Dragotti, and Y. M. Lu. "Sparsity According to Prony, average performance analysis". *SPARS Workshop 2015 (Cambridge, UK)*. July 2015.

[15] J. Oñativia, Y. M. Lu, and P. L. Dragotti. "Sparsity According to Prony: Average Performance Analysis and Phase Transition". *to be submitted* (2015).

[16] Y. C. Pati, R. Rezaiifar, and P. Krishnaprasad. "Orthogonal Matching Pursuit: Recursive Function Approximation with Applications to Wavelet Decomposition". *27th Asilomar Conference on Signals, Systems and Computers*. Vol. 40-44. Nov. 1993.

[17] R. Tibshirani. "Regression Shrinkage and Selection via the Lasso". *Journal of the Royal Statistical Society. Series B (Methodological)* 58.1 (1996), pp. 267–288. URL: `http://www.jstor.org/stable/2346178`.