# Dictionary Learning Based on Sparse Distribution Tomography

**Pedram Pad** [*1] **Farnood Salehi** [*2] **Elisa Celis** [2] **Patrick Thiran** [2] **Michael Unser** [1]

## Abstract

We propose a new statistical dictionary learning algorithm for sparse signals that is based on an $\alpha$-stable innovation model. The parameters of the underlying model—that is, the atoms of the dictionary, the sparsity index $\alpha$ and the dispersion of the transform-domain coefficients—are recovered using a new type of probability distribution tomography. Specifically, we drive our estimator with a series of random projections of the data, which results in an efficient algorithm. Moreover, since the projections are achieved using linear combinations, we can invoke the generalized central limit theorem to justify the use of our method for sparse signals that are not necessarily $\alpha$-stable. We evaluate our algorithm by performing two types of experiments: image in-painting and image denoising. In both cases, we find that our approach is competitive with state-of-the-art dictionary learning techniques.

Beyond the algorithm itself, two aspects of this study are interesting in their own right. The first is our statistical formulation of the problem, which unifies the topics of dictionary learning and independent component analysis. The second is a generalization of a classical theorem about isometries of $\ell_p$-norms that constitutes the foundation of our approach.

## 1. Introduction

The problem of finding the mixing matrix $\mathbf{A}$ from a set of observation vectors $\boldsymbol{y}$ in the model

$$\boldsymbol{y} = \mathbf{A}\boldsymbol{x} \tag{1}$$

is only solvable if one can benefit from strong hypotheses on the signal vector $\boldsymbol{x}$. For instance, one may assume that the entries of $\boldsymbol{x}$ are statistically independent, which results in a class of methods refered to as *independent component analysis (ICA)* (Hyvärinen et al., 2004). A more recent trend is to assume that the vector $\boldsymbol{x}$ is sparse, so that the recovery can be recast as a deterministic dictionary learning problem, the prototypical example being *sparse component analysis (SCA)* (Gribonval & Lesage, 2006). Extensive research has been conducted on these problems in the past three decades.

**Prior work:** In the literature, ICA precedes SCA and can be traced back to (Herault & Jutten, 1986). In fact, ICA constitutes the non-Gaussian generalization of the much older *principal component analysis (PCA)*, which is widely used in classical signal processing. ICA is usually formalized as an optimization problem involving a cost function that measures the independence of the estimated $x_i$ (i.e., the entries of the vector $\boldsymbol{x}$). A common measure of independence, which is inspired by information theory, is the mutual information of the entries of $\boldsymbol{x}$. However, due to its computational complexity, other measures such as the kurtosis, which measures the non-Gaussianity of the components, are often used (Hyvärinen & Oja, 2000; Naik & Kumar, 2011) (except in special cases such as the analysis of stable $AR(1)$ processes by (Pad & Unser, 2015)). The main drawback of ICA is that the system (1) needs to be determined; i.e., $\mathbf{A}$ should be square—otherwise the complexity is so high that the methods can only be implemented for small problems (Lathauwer et al., 2007; Lathauwer & Castaing, 2008).

SCA, on the other hand, is usually achieved by putting constraints on the sparsity of the representation or by optimizing a sparsity-promoting cost function. Thanks to the emergence of very efficient algorithms, SCA has found wide use in different applications (see (Mairal et al., 2010; Marvasti et al., 2012)). The underlying framework for SCA is deterministic—this is the primary difference with ICA, which aims to decouple signals that are realizations of stochastic processes.

**$\alpha$-stable distributions:** In this paper, we aim to achieve the best of both worlds: the use of a statistical formulation—in the spirit of ICA—with a restriction to a parametric class of stochastic models that is well adapted to the notion of sparsity. Specifically, we assume that the entries of the vector $\boldsymbol{x}$ are random variables that are i.i.d. symmetric-$\alpha$-stable. The family of $\alpha$-stable distributions is a general-

[*]Equal contribution [1]Biomedical Imaging Group, EPFL, Lausanne, Switzerland [2]Computer Communications and Applications Laboratory 3, EPFL, Lausanne, Switzerland. Correspondence to: Pedram Pad <pedram.pad@epfl.ch>.

*Proceedings of the $34^{th}$ International Conference on Machine Learning*, Sydney, Australia, PMLR 70, 2017. Copyright 2017 by the author(s).

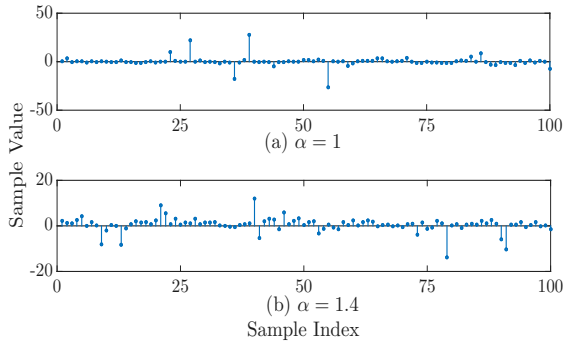


*Figure 1.* Illustration of the effect of $\alpha$ on the sparsity of the signal. Two realizations of i.i.d. S$\alpha$S signals.

ization of the Gaussian probability density function (PDF). Since $\alpha$-stability is preserved through linear transformation, this class of models has a central position in the study of stochastic processes (Samoradnitsky & Taqqu, 1994; Nikias & Shao, 1995; Shao & Nikias, 1993). The family is parametrized by $\alpha \in (0, 2]$, which controls the rate of decay of the distribution. The extreme case of $\alpha = 2$ corresponds to the Gaussian distribution—the only non-sparse member of the family. By contrast, the other members of the S$\alpha$S family for $\alpha < 2$ are heavy-tailed with unbounded variance. This property implies that an i.i.d. sequence of such random variables generates a sparse signal (Amini et al., 2011; Gribonval et al., 2012). By decreasing $\alpha$, the distribution becomes more heavy-tailed and thus the signal becomes more sparse (the effect of $\alpha$ is illustrated in Figure 1).

This class of random variables has also been widely used in practice. Typical applications include: modeling of ultrasound RF signals, (Achim et al., 2015), signal detection theory (Kuruoglu et al., 1998), communications (Middleton, 1999), image processing (Achim & Kuruoglu, 2005), audio processing (Georgiou et al., 1999), sea surface (Gallagher, 2001), network traffic (Resnick, 1997), and finance (Nolan, 2003; Ling, 2005).

**Main contributions:** Our main contribution in this paper is a new dictionary learning algorithm based on the signal modeling mentioned above. The proposed method has the following advantages:

1. all parameters can be estimated from the data (it is hyperparameter-free),
2. it learns the dictionary without the need to recover the signal $\boldsymbol{x}$, and
3. it is fast and remarkably robust.

Once the matrix $\mathbf{A}$ is estimated, it is then possible to efficiently recover $\boldsymbol{x}$ by using standard procedures (Bickson & Guestrin, 2010).

We also show that the proposed algorithm provides an efficient estimator of the spectral measure of a stable random vector. An enabling component of our method is a new theorem that generalizes a classical result about isometries of $\ell_p$-norms.

**Organization:** In the next section, we briefly review S$\alpha$S random variables and present our mathematical model. In Section 3, we establish our main result which then yields an algorithm for finding the matrix $\mathbf{A}$ as well as the sparsity index $\alpha$. In Section 4, we present the simulation results and compare their performance with existing algorithms. In Section 5, we summarize the paper and give some suggestions for future work.

## 2. Preliminaries and problem formulation

We begin by recalling some basic properties of symmetric-$\alpha$-stable random variables. We then proceed with the formulation of the estimation problem that we solve in Section 3. The notation that we use throughout the paper is as follows: we use italic symbols for random variables, capital boldface symbols for matrices and lowercase boldface symbols for vectors. Thus, $\mathbf{X}$ is a deterministic matrix, $\boldsymbol{X}$ is a random matrix and $x$ is a random variable. Likewise, $\boldsymbol{x}$ and $\mathbf{x}$ denote a random and a deterministic vector respectively.

### 2.1. Symmetric-$\alpha$-stable random variables

For any $\alpha \in (0, 2]$ and $\gamma > 0$, a random variable $X$ with characteristic function

$$\widehat{p}_X(\omega) = \exp(-\gamma|\omega|^\alpha) \tag{2}$$

is called a symmetric-$\alpha$-stable (S$\alpha$S) random variable with dispersion $\gamma$ and stability parameter $\alpha$ (Nikias & Shao, 1995). This class of random variables is a generalization of the Gaussian model: For $\alpha = 2$, $X$ is a Gaussian random variable with zero mean and variance $2\gamma$. As their name suggests, $\alpha$-stable variables share the property of *stability under linear combination* (Nikias & Shao, 1995); i.e., if $X_1, \ldots, X_n$ are $n$ i.i.d. copies of $X$ and $a_1, \ldots, a_n \in \mathbb{R}$ are $n$ real numbers, then the random variable

$$Y = a_1 X_1 + \cdots + a_n X_n \tag{3}$$

has the same distribution as

$$\left(|a_1|^\alpha + \cdots + |a_n|^\alpha\right)^{\frac{1}{\alpha}} X. \tag{4}$$

In other words, the random variable $Y$ is an S$\alpha$S random variable with dispersion $\gamma \|\mathbf{a}\|_\alpha^\alpha$ where $\|\mathbf{a}\|_\alpha = \left(|a_1|^\alpha + \cdots + |a_n|^\alpha\right)^{\frac{1}{\alpha}}$ is the $\alpha$-(pseudo)norm of the vector $\mathbf{a} = (a_1, \ldots, a_n)$. This property makes S$\alpha$S random variables convenient for the study of linear systems.

The other property of S$\alpha$S random variables with $\alpha < 2$ is

their heavy-tailed PDF. When $\alpha < 2$, we have

$$\lim_{|x|\to\infty} |x|^{1+\alpha} p_X(x) = C(\alpha, \gamma), \tag{5}$$

where $p_X$ is the PDF of $X$ and $C(\alpha, \gamma)$ is a positive constant that depends on $\alpha$ and $\gamma$ (Nikias & Shao, 1995). This implies that the variance of S$\alpha$S random variables is unbounded for $\alpha < 2$. Also, note that a smaller $\alpha$ results in heavier tails.

Infinite-variance random variables are considered to be appropriate candidates for sparse signals (Amini et al., 2011; Gribonval et al., 2012). Because an i.i.d. sequence of heavy-tailed random variables has most of its energy concentrated on a small fraction of samples, they are good candidates to model signals that exhibit sparse behavior.

Yet, the truly fundamental aspect of $\alpha$-stable random variables is their role in the generalized central limit theorem. As we know, the limit distribution of normalized sums of i.i.d. finite-variance random variables are Gaussian. Likewise, any properly normalized sum of heavy-tailed i.i.d. random variables converges to an $\alpha$-stable random variable whee the $\alpha$ depends on the weight of their tail (Meerschaert & Scheffler, 2001). This implies that a linear combination of a large number of samples of a sparse signal is well represented by $\alpha$-stable random variables.

### 2.2. Problem formulation

Our underlying signal model is

$$\boldsymbol{y} = \mathbf{A}\boldsymbol{x} \tag{6}$$

where $\boldsymbol{x}$ is an unknown $n \times 1$ random vector with S$\alpha$S i.i.d. entries and $\alpha < 2$, $\boldsymbol{y}$ is an $m \times 1$ observation vector and $\mathbf{A}$ is an $m \times n$ dictionary matrix,. We are given $K$ realizations of $\boldsymbol{y}$; namely, $\mathbf{y}_1, \ldots, \mathbf{y}_K$, and our goal is to estimate $\mathbf{A}$.

## 3. Dictionary learning for S$\alpha$S signals

In the problem of dictionary learning, the maximum information that we can asymptotically try to retrieve from $\mathbf{y}_1, \ldots, \mathbf{y}_K$ is the exact distribution of $\boldsymbol{y}$. However, even if we knew $\boldsymbol{y}$, identifying $\mathbf{A}$ is still not tractable in general—for instance, in the case of Gaussian vectors, $\mathbf{A}$ is only identifiable up to right-multiplication by a unitary matrix. Moreover, obtaining an acceptable estimate of the distribution of $\boldsymbol{y}$ requires, in general, a vast amount of data and processing power (since it is a $m$-dimensional function with $m$ possibly large). In this section, we leverage the property of stability under linear combination of S$\alpha$S random variables explained in Section 2.1 to propose a new algorithm to estimate $\mathbf{A}$ for the dictionary learning problem stated in Section 2.2.

### 3.1. New cost function for sparse S$\alpha$S signals

Recall that, the random vector $\boldsymbol{y}$ (see Equations (2) and (6)) is an $m$-dimensional $\alpha$-stable vector with characteristic function

$$\widehat{p}_{\boldsymbol{y}}(\boldsymbol{\omega}) = \exp\left(-\gamma \|\mathbf{A}^\top \boldsymbol{\omega}\|_\alpha^\alpha\right) \tag{7}$$

for $\boldsymbol{\omega} \in \mathbb{R}^m$. Thus, knowing $\|\mathbf{A}^\top \mathbf{u}\|_\alpha$ for all $\mathbf{u} \in \mathcal{S}^{m-1}$, where $\mathcal{S}^{m-1}$ is the $(m-1)$-dimensional unit sphere, i.e.,

$$\mathcal{S}^{m-1} = \{\mathbf{u} \in \mathbb{R}^m \mid \|\mathbf{u}\|_2 = 1\}, \tag{8}$$

is equivalent to knowing the distribution of $\boldsymbol{y}$. Note that $\mathbf{u}^\top \boldsymbol{y} = \mathbf{u}^\top \mathbf{A}\boldsymbol{x}$ (see Equations (3) and (4)) is an S$\alpha$S random variable with dispersion

$$\gamma(\mathbf{u}) = \gamma \|\mathbf{A}^\top \mathbf{u}\|_\alpha^\alpha. \tag{9}$$

Thus, knowing the dispersion of the marginal distributions of $\boldsymbol{y}$ for all $\mathbf{u} \in \mathcal{S}^{m-1}$ is equivalent to knowing the distribution of $\boldsymbol{y}$. In other words, in the case of S$\alpha$S random vectors, knowing their marginal dispersions is equivalent to knowing the Radon transform of their PDFs or, equivalently, their joint characteristic function (Helgason, 2010). Due to the relationship between the Radon transform and the field of tomography, we call our algorithm *sparse distribution tomography* (SparsDT).

Another interesting fact is that, in the non-Gaussian case ($\alpha < 2$), knowing the marginal dispersions of $\boldsymbol{y}$, i.e., $\gamma(\mathbf{u})$, identifies the matrix $\mathbf{A}$ uniquely, up to negation and permutation of the columns. Formally, we have the following theorem, which is proved in Appendix A:

**Theorem 1** *Let $\mathbf{A}$ be an $m \times n$ matrix where columns are pairwise linearly independent. If $\alpha \in (0, 2)$ and $\mathbf{B}$ is an $m \times n$ matrix for which we have*

$$\|\mathbf{A}^\top \mathbf{u}\|_\alpha^\alpha = \|\mathbf{B}^\top \mathbf{u}\|_\alpha^\alpha \tag{10}$$

*for all $\mathbf{u} \in \mathbb{R}^m$, then $\mathbf{B}$ is equal to $\mathbf{A}$ up to negation and permutation of its columns.*

**Remark 1** *This theorem can be seen as a generalization of the result in (Rolewicz, 1985) that states that the isometries of $\ell_p$-norms are generalized permutation matrices (permutation matrices with some of their rows negated). To the best of our knowledge, this result is novel and could be of independent interest.*

This theorem suggests that in order to find $\mathbf{A}$ all we need is to find $\gamma(\mathbf{u})$ for $\mathbf{u} \in \mathbb{R}^m$. Intuitively, we can say that as $\mathbf{A}$ has a finite number of parameters (entries), $\mathbf{A}$ is identifiable based on the knowledge of $\gamma(\mathbf{u})$ for an appropriate finite set of vectors $\mathbf{u} = \mathbf{u}_1, \ldots, \mathbf{u}_L$ (for some $L \geq mn$). We can then solve the set of non-linear equations

$$\begin{aligned} \gamma\|\mathbf{B}^\top \mathbf{u}_1\|_\alpha^\alpha &= \gamma(\mathbf{u}_1), \\ &\vdots \\ \gamma\|\mathbf{B}^\top \mathbf{u}_L\|_\alpha^\alpha &= \gamma(\mathbf{u}_L), \end{aligned} \tag{11}$$

for $\mathbf{B}$ to obtain $\mathbf{A}$.

Now, the problem is to find $\gamma(\mathbf{u})$ for a given $\mathbf{u} \in \mathbb{R}^m$. Recall that $\gamma(\mathbf{u})$ is the dispersion of the S$\alpha$S random variable $\mathbf{u}^T\boldsymbol{y}$. As $\mathbf{y}_1, \ldots, \mathbf{y}_K$ are realizations of $\boldsymbol{y}$, $\mathbf{u}^T\mathbf{y}_1, \ldots, \mathbf{u}^T\mathbf{y}_K$ are realizations of $\mathbf{u}^T\boldsymbol{y}$. There is a rich literature on the estimation of the parameters of $\alpha$-stable random variables through their realizations, see, e.g, (Nikias & Shao, 1995). We use the estimation from (Achim et al., 2015) in the following equation

$$\log \hat{\gamma}(\mathbf{u}) = \frac{\alpha}{K} \sum_{k=1}^{K} \log |\mathbf{u}^T\mathbf{y}_k| - (\alpha - 1)\psi(1) \tag{12}$$

where $\psi$ is the digamma function ($\psi(1)$ is the negative of the Euler-Mascheroni constant and is approximately $0.5572$), and $\hat{\gamma}(\mathbf{u})$ denotes the estimation of $\gamma(\mathbf{u})$. Note that $\hat{\gamma}(\mathbf{u})$ tends to $\gamma(\mathbf{u})$ when the number of observations, $K$, tends to infinity. This means that we can obtain the exact value of $\gamma(\mathbf{u})$ asymptotically.

However, non-exact values for $\gamma(\mathbf{u}_\ell)$, for $\ell = 1, \ldots, L$ (which is the case when $K$ is finite), can lead to the non-existence of a solution for the system of equations (11). To overcome this problem, instead of solving this system of equations exactly, we minimize the following objective function

$$\begin{aligned} \mathcal{E}(\mathbf{B}) &= \frac{1}{L} \sum_{\ell=1}^{L} d\left(\gamma \|\mathbf{B}^\top \mathbf{u}_\ell\|_\alpha^\alpha, \hat{\gamma}(\mathbf{u}_\ell)\right) \\ &= \frac{1}{\alpha L} \sum_{\ell=1}^{L} \left|\log(\gamma \|\mathbf{B}^\top \mathbf{u}_\ell\|_\alpha^\alpha) - \log(\hat{\gamma}(\mathbf{u}_\ell))\right| \end{aligned} \tag{13}$$

where $\log \hat{\gamma}(\mathbf{u}_1), \ldots, \log \hat{\gamma}(\mathbf{u}_L)$ are $L$ numbers calculated from (12). The cost function $d(a, b) = \frac{1}{\alpha}|\log a - \log b|$ is a continuous positive function from $\mathbb{R}^2$ to $\mathbb{R}$, whose global minimum is 0 and is reached over the line $a = b$. When $\hat{\gamma}(\mathbf{u}) = \gamma(\mathbf{u})$, $\mathbf{B} = \mathbf{A}$ minimizes $\mathcal{E}(\mathbf{B})$. Thus, if $\hat{\gamma}(\mathbf{u})$ is close enough to $\gamma(\mathbf{u})$, due to the continuity of $d$, we expect that the minimizer of $\mathcal{E}$ will be close to $\mathbf{A}$. Therefore, our approach to dictionary learning is to solve

$$\begin{aligned} \widehat{\mathbf{A}} &= \underset{\mathbf{B}}{\operatorname{argmin}}\ \mathcal{E}(\mathbf{B}) \\ &= \underset{\mathbf{B}}{\operatorname{argmin}} \frac{1}{\alpha L} \sum_{\ell=1}^{L} \left|\log\left(\gamma \|\mathbf{B}^\top \mathbf{u}_\ell\|_\alpha^\alpha\right) - \log \hat{\gamma}(\mathbf{u}_\ell)\right|. \end{aligned} \tag{14}$$

The only parameter that needs to be set now is the stability parameter $\alpha$. Note that the dispersion parameter $\gamma$ in Equation (14) does not need to be set as it will be automatically merged into the learned dictionary. Recall that there are well-known methods for estimating $\alpha$ from data; among which we use

$$\hat{\alpha}(\mathbf{u}) = \left( \frac{6}{\pi^2 K} \sum_{k=1}^{K} \left(\log |\mathbf{u}^\top \mathbf{y}_k| - \log \hat{\kappa}(\mathbf{u})\right)^2 - \frac{1}{2} \right)^{-\frac{1}{2}} \tag{15}$$

from (Achim et al., 2015), where

$$\log \hat{\kappa}(\mathbf{u}) = \frac{1}{K} \sum_{k=1}^{K} \log |\mathbf{u}^\top \mathbf{y}_k|. \tag{16}$$

This gives us an estimate for $\alpha$ for any given $\mathbf{u} \in \mathbb{R}^m$. Hence, the estimated value of $\alpha$ is the average over all $\hat{\alpha}(\mathbf{u}_\ell)$ for $\ell = 1, \ldots, L$, i.e.,

$$\hat{\alpha} = \frac{1}{L} \sum_{\ell=1}^{L} \hat{\alpha}(\mathbf{u}_\ell). \tag{17}$$

Now, using this estimate, Equation (12) becomes

$$\log \hat{\gamma}(\mathbf{u}) = \hat{\alpha} \log \hat{\kappa}(\mathbf{u}) - (\hat{\alpha} - 1)\psi(1). \tag{18}$$

We also replace $\alpha$ with $\hat{\alpha}$ in Equation (13) which results in a parameter-free cost function. This is in contrast with most existing cost functions that have parameters one must set.

### 3.2. Learning algorithm

To solve the minimization problem in Equation (14), we propose a variation on a gradient-descent algorithm with an adaptive step size that has a changing cost function. To do so, we first derive the gradient of $\mathcal{E}$ at $\mathbf{B}$. Using matrix calculus (see Appendix B), we find that

$$\nabla \mathcal{E}(\mathbf{B}) = \frac{1}{\alpha L} \sum_{\ell=1}^{L} \operatorname{sgn}\left(\log(\gamma \|\mathbf{B}^\top \mathbf{u}_\ell\|_\alpha^\alpha) - \log \hat{\gamma}(\mathbf{u}_\ell)\right) \cdot \frac{\nabla \|\mathbf{B}^\top \mathbf{u}_\ell\|_\alpha^\alpha}{\|\mathbf{B}^\top \mathbf{u}_\ell\|_\alpha^\alpha} \tag{19}$$

where $\operatorname{sgn}(\cdot)$ is the sign function (i.e., $\operatorname{sgn}(e) = 1$ if $e > 0$ and $\operatorname{sgn}(e) = 0$ otherwise) and

$$\nabla \|\mathbf{B}^\top \mathbf{u}\|_\alpha^\alpha = \alpha \begin{bmatrix} \operatorname{sgn}\left(\boldsymbol{b}_1^\top \mathbf{u}\right) \left|\boldsymbol{b}_1^\top \mathbf{u}\right|^{\alpha-1} \mathbf{u}^\top \\ \vdots \\ \operatorname{sgn}\left(\boldsymbol{b}_n^\top \mathbf{u}\right) \left|\boldsymbol{b}_n^\top \mathbf{u}\right|^{\alpha-1} \mathbf{u}^\top \end{bmatrix}^\top \tag{20}$$

where $\boldsymbol{b}_i$ is the $i$th column of $\mathbf{B}$.

The cost function in Equation (13) is non-convex in $\mathbf{B}$. In order to avoid getting trapped in local minima, we iteratively change the cost function inside the gradient descent algorithm. The idea is that instead of keeping $\mathbf{u}_1, \ldots, \mathbf{u}_L$ fixed throughout the optimization process, we regenerate them randomly with a uniform distribution on $\mathcal{R}^m$ after some

In our simulations we also implemented other natural candidates for $d(a, b)$ and all of them gave approximately the same performance. Due to the limited space, we do not present results for other cost functions.

iterations of steepest descent. We repeat this process until convergence. Note that (11) holds for any $\mathbf{u}_1, \ldots, \mathbf{u}_L$ and thus changing this set does not change the end result of (11).

**Remark 2** *Using this idea always results in convergence to the global minimum in our computer simulations. A plausible explanation of this phenomenon is that each set of $\mathbf{u}_1, \ldots, \mathbf{u}_L$ yields a non-convex cost function with different local minima. Yet they all have the same global minimum. Therefore, switching between them during the optimization process prevents getting trapped in any of the local minima, which ultimately results in finding the global minimum of the cost function.*

The pseudocode of our dictionary learning method is given in Algorithm 1. There, $\eta$ is the step size of the gradient descent that increases or decreases by factors of $\kappa^+$ or $\kappa^-$ upon taking a good or poor step. The adaptive step size is especially helpful for $\alpha \leq 1$, where the cost function is not smooth. The algorithm does not depend on the exact choice of convergence criteria.

**Remark 3** *Algorithm 1 can also be seen as an efficient method for estimating the spectral measure of stable random vectors. In fact, the problem of estimating $\mathbf{A}$ from a set of realizations of $\boldsymbol{y}$ can also be seen as parameter estimation for a stable random vector $\boldsymbol{y}$ with a symmetric distribution around the origin. Such random vectors are parametrized by a measure $\Gamma$ on $\mathcal{S}^{m-1}$ that is called the spectral measure. In our problem, we have $\Gamma(\cdot) = \sum_{i=1}^{n} \|\boldsymbol{a}_i\|_\alpha \, \delta_{\boldsymbol{a}_i}(\cdot)$ where the $\delta_{\boldsymbol{a}_i}$s are unit point masses at $\frac{\boldsymbol{a}_i}{\|\boldsymbol{a}_i\|_2}$ and $a_i$ is the $i^{th}$ column of $\mathbf{A}$. Some methods have been proposed to solve this problem, e.g., (Nolan et al., 2001). However, they tend to be computationally intractable for dimensions greater than 3.*

**Remark 4** *According to the generalized central limit theorem, under some mild conditions, the distribution of the sum of symmetric heavy-tailed random variables tends to a S$\alpha$S distribution as the number of summands tends to infinity. This means that we can represent $\mathbf{u}^\top \boldsymbol{y} = \mathbf{u}^\top \mathbf{A} \boldsymbol{x}$ with an S$\alpha$S random variable for large enough $n$ irrespective of the distribution of the $x_i$s provided that the latter are heavy tailed. Therefore, we expect Algorithm 1 to find applications for other classes of sparse signals, provided that $n$ is sufficiently large.*

## 4. Empirical results

In this section, we analyze the performance of the proposed algorithm SparsDT and compare it with existing methods. Recall that the actual dictionary is $\mathbf{A}$ and the learned dictionary is $\widehat{\mathbf{A}}$. We run two types of the experiments: We first test the algorithm on synthetic S$\alpha$S data and then we test it on real images.

**Algorithm 1** SparseDT

**initialize:** $\eta > 0$
**initialize:** $\kappa^+ \geq 1$ and $\kappa^- \leq 1$
**initialize:** **generate** $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n \sim \mathcal{N}(0, \mathbf{I}_{m\times m})$ **and** $\mathbf{B} \leftarrow \left[\boldsymbol{b}_1 | \ldots | \boldsymbol{b}_n\right]$
**repeat**
  **initialize:** **generate** $\mathbf{u}_1, \ldots, \mathbf{u}_L \sim \mathcal{N}(0, \mathbf{I}_{m\times m})$
  **estimate** $\hat{\alpha}$ **from** (15)
  $E \leftarrow \mathcal{E}(\mathbf{B})$
  **repeat**
    $\mathbf{B}_{old} \leftarrow \mathbf{B}$
    $E_{old} \leftarrow E$
    $\mathbf{B} \leftarrow \mathbf{B} - \eta \, \nabla\mathcal{E}(\mathbf{B})$
    $E \leftarrow \mathcal{E}(\mathbf{B})$
    **if** $E \leq E_{old}$ **then**
      $\eta \leftarrow \kappa^+ \cdot \eta$
    **else**
      $\mathbf{B} \leftarrow \mathbf{B}_{old}$
      $E \leftarrow E_{old}$
      $\eta \leftarrow \kappa^- \cdot \eta$
    **end if**
  **until** $\mathbf{B}$ converges (for this choice of $\mathbf{u}_1, \ldots, \mathbf{u}_L$)
**until** $\mathbf{B}$ converges
**return** $\mathbf{B}$

### 4.1. Benchmarks

We compare our algorithm with three commonly used algorithms that are available in the Python package SPAMS. These constrained optimization problems are as follows:

1. $\ell_2/\ell_1$: Maximizing the data fidelity while controling the sparsity of representation with parameter $\lambda_1$:

$$\widehat{\mathbf{A}}_{\ell_2/\ell_1} = \underset{\mathbf{B}}{\operatorname{argmin}} \frac{1}{2K} \sum_{k=1}^{K} \|\mathbf{y}_k - \mathbf{B}\mathbf{x}_k\|_2^2 \quad s.t. \quad \|\mathbf{x}_i\|_1 \leq \lambda_1.$$

2. $\ell_1/\ell_2$: Maximizing the sparsity of representation while controling the data fidelity with parameter $\lambda_2$:

$$\widehat{\mathbf{A}}_{\ell_1/\ell_2} = \underset{\mathbf{B}}{\operatorname{argmin}} \frac{1}{2K} \sum_{k=1}^{K} \|\mathbf{x}_k\|_1 \quad s.t. \quad \|\mathbf{y}_k - \mathbf{B}\mathbf{x}_k\|_2 \leq \lambda_2.$$

3. $\ell_1 + \ell_2$: Combining sparsity and data fidelity in the cost function using Lagrange multipliers:

$$\widehat{\mathbf{A}}_{\ell_1+\ell_2} = \underset{\mathbf{B}}{\operatorname{argmin}} \frac{1}{2K} \sum_{k=1}^{K} \|\mathbf{y}_k - \mathbf{B}\mathbf{x}_k\|_2^2 + \lambda_3 \|\mathbf{x}_k\|_1 + \lambda_4 \|\mathbf{x}_k\|_2^2.$$

One of the challenges in utilizing these benchmarks is determining the regularization parameters $\lambda_1, \ldots, \lambda_4$. In our

http://spams-devel.gforge.inria.fr/

Other cost functions are also available in the package SPAMS, but those retained here yield the best results in our experiments.

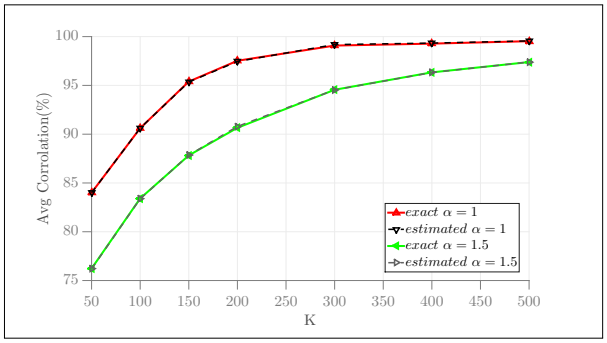


*Figure 2.* Impact of the number of samples $K$ on the average correlation for $\mathbf{A}_{16\times 24}$.

experiments, the regularization parameters are optimized (by grid search) in order to maximize the performance of each of the benchmarks above. This is in contrast to our algorithm, which has no regularization parameter to tune.

### 4.2. Experimental results for synthetic data

We first test the algorithms on synthetic data. In order to quantify the performance of the algorithms, we use several metrics. One is the average correlation of the dictionaries. Specifically, we calculate the correlation between all columns of $\widehat{\mathbf{A}}$ and $\mathbf{A}$, and then match each column of $\widehat{\mathbf{A}}$ with one of the columns of $\mathbf{A}$ (a one-to-one map) such that the average correlation between the corresponding columns is maximized. Additionally, we say that the dictionary is found if the average correlation of the columns is larger than 0.97.

**Effect of $K$:** We demonstrate the effect of the number of samples $K$ on the performance of our proposed algorithm SparsDT. Intuitively, the precision of the estimation increases with the number of samples $K$, and, as $K$ goes to infinity, the estimation error goes to zero, which ultimately gives the exact $\mathbf{A}$. We demonstrate this effect with the following experiment: We take $m = 16$, $n = 24$ and $\alpha = 1$ and 1.5. Then, for each $K$, we run the experiment for 50 random matrices $\mathbf{A}$, and, for each case, we run Algorithm 1 with both exact and estimated $\alpha$ (from (17)). The results are depicted in Figure 2, where the vertical axis is the average correlation of the estimated dictionary with the exact one, and the horizontal axis is the number of samples $K$. Interestingly, the performance of the algorithm is almost the same when using the exact or estimated value of $\alpha$, which suggests that the estimator of $\alpha$ is robust and accurate. Moreover, we see that the average correlation is an increasing function of $K$, as expected. Also note that the convergence is faster for $\alpha = 1$, which corresponds to the setting with more sparsity.

**Comparison against benchmarks:** We compare SparsDT against the $\ell_2/\ell_1$, $\ell_1/\ell_2$ and $\ell_1 + \ell_2$ methods described above. We compare the algorithms with regard to their success rate (i.e., the percentage of the dictionaries found by the algorithm), and the time that they take to find the dictionary (in the cases of success only). We again take $m = 16$, $n = 24$ and generate 100 random matrices $\mathbf{A}$. In Table 1, the results for $\alpha = 1.2$ and $K = 500$ are given. Finally, in Figure 3 we compare the algorithms success rate for different $\alpha$, we take $m = 16$, $n = 24$, and $K = 1000$. These results indicate that SparsDT outperforms the other methods in the rate of success. Also, its average learning time is typically much less than the others, except for $\ell_2/\ell_1$ which does not find the correct dictionary at best in 10% of the time. The range of $\alpha$ that was observed in our experiments is $\alpha \in [1, 1.6]$, which is also the range where our algorithm works well and which is interesting for many applications including image processing. We do not recommend using the method for $\alpha > 1.7$ because the convergence properties degrade as we get closer to 2 (a larger value of $K$ is then needed to reach high success rates).

| | Comparison metrics | |
|---|---|---|
| **Algorithm** | **% found** | **Avg. time (s)** |
| SparsDT | 100 | 5.19 |
| $\ell_2/\ell_1$ | 50 | 0.07 |
| $\ell_1/\ell_2$ | 75 | 95.75 |
| $\ell_1 + \ell_2$ | 94 | 19.89 |

*Table 1.* Performance of Algorithms on S$\alpha$S Signals. $\alpha = 1.2$, $\mathbf{A}_{16\times 24}$ matrix, $K = 500$.

### 4.3. Experimental results for real images

Since images often have sparse representations, we apply our algorithm to problems in image processing applications. Our experiments are missing pixel recovery (in-painting) and denoising, based on dictionary learning. We use a

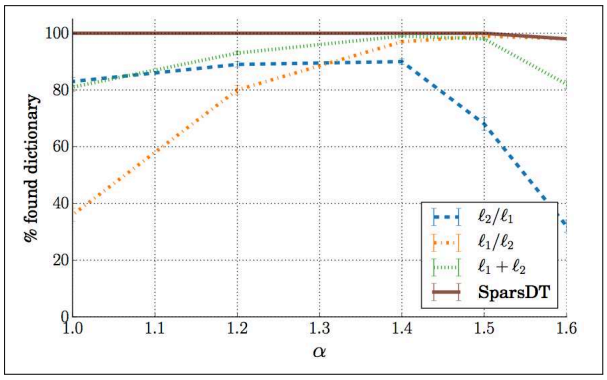


*Figure 3.* Impact of $\alpha \in [1, 1.6]$ on the success rate of the algorithms.

| Algorithm | PSNR (dB) |
|---|---|
| SparsDT | 29.61 |
| $\ell_2/\ell_1$ | 28.98 |
| $\ell_1/\ell_2$ | 28.74 |
| $\ell_1+\ell_2$ | 28.98 |

*Table 2.* Performance of different methods for denoising images contaminated by Gaussian noise.

database of face images provided by AT&T and crop them to have size $112 \times 91$ so we can chop each image to 208 patches of size $7 \times 7$, which correspond to $\mathbf{y}_i$ in our model.

In this situation, the data is not exactly S$\alpha$S, so we must adapt our choice of $\mathbf{u}$ in Step 5 of Algorithm 1. Specifically, in Equation (17) we eliminate projection vectors $\mathbf{u}$ that result in $\alpha$ greater than 2 (as $\alpha$ is required to be less than 2). In addition, we only select $\mathbf{u}$ that results in an $\alpha$ close to our estimated $\hat{\alpha}$ in (17). The number of iterations are chosen such that all algorithms converge.

**Missing pixel recovery:** In this experiment, we reconstruct an image from 50% of its pixels. We take out the image shown in Figure 4, remove 50% of its pixels uniformly at random, and learn the dictionary using the patches of the other images in the collection. We assume 248 atoms in the dictionary. Then, using the learned dictionary, we reconstruct the image using orthogonal matching pursuit (for a detailed analysis see (Sahoo & Makur, 2015)). The results for different dictionary learning methods are depicted in Figure 4; SparsDT outperforms existing methods both visually and in term of PSNR.

**Image denoising:** In this experiment, we use the dictionaries learned in the previous experiment to denoise the image in Figure 4. More precisely, we add Gaussian noise with standard deviation 10 to the original image and use orthogonal matching pursuit to denoise it. The performance of each method in PSNR can be seen in Table 2. As we see, SparsDT outperforms the other methods by at least 0.6 dB.

## 5. Summary and future work

In this paper, we consider a stochastic generation model of sparse signals that involves an S$\alpha$S innovation. Then, by designing an estimator of the spectral measure of so-defined stable random vectors, we propose a new dictionary learning algorithm. The proposed algorithm (SparsDT) turns out to be quite robust; it works well on sparse real-world signals, even when these do not rigorously follow the S$\alpha$S model. This surprising fact can be explained by invoking the generalized central limit theorem. We validate SparsDT on several image-processing tasks and found it to outperform popular dictionary learning methods often significantly.

www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html

Moreover, SparsDT has no parameter to tune, contrary to other algorithms.

Extending this work to non-symmetric $\alpha$-stable random variables is a possible direction of future research. Given the excellent numerical behavior of the algorithm, it is of interest to get a good handle on the accuracy of the estimation in terms of the number of samples and the dimensionality of signals.

## A. Proof of Theorem 1

Denote the $j$th column of $\mathbf{A}$ and $\mathbf{B}$ by $\boldsymbol{a}_j$ and $\boldsymbol{b}_j$, respectively. Also, denote the set of indices $j$ for which $\boldsymbol{b}_j \neq 0$ by $\mathcal{B} \subseteq \{1,\dots,n\}$. Note that due to the assumption of the pairwise linear independence of columns of $\mathbf{A}$, $\boldsymbol{a}_j \neq 0$ for all $j \in \{1,\dots,n\}$. Since

$$\left\|\mathbf{A}^\top \mathbf{u}\right\|_\alpha^\alpha = \sum_{j=1}^{n} \left|\mathbf{u}^\top \boldsymbol{a}_j\right|^\alpha = \sum_{j\in\mathcal{B}} \left|\mathbf{u}^\top \boldsymbol{b}_j\right|^\alpha = \left\|\mathbf{B}^\top \mathbf{u}\right\|_\alpha^\alpha .$$

for all $\mathbf{u} \in \mathbb{R}^m$, the partial derivatives of any order of the two side of the equation are also equal. In particular, we have

$$\frac{\partial^d}{\partial {u_i}^d} \left\|\mathbf{A}^\top \mathbf{u}\right\|_\alpha^\alpha = \frac{\partial^d}{\partial {u_i}^d} \left\|\mathbf{B}^\top \mathbf{u}\right\|_\alpha^\alpha \tag{21}$$

for all $i = 1,\dots,m$ and $d \in \mathbb{N}$, where $u_i$ is the $i$th entry of $\mathbf{u}$.

First we prove the theorem for $0 < \alpha \leq 1$. In (21), we set $d = 1$ and obtain

$$\begin{aligned} &\sum_{j=1}^{n} \left|\mathbf{u}^\top \boldsymbol{a}_j\right|^{\alpha-1} \operatorname{sgn}\left(\mathbf{u}^\top \boldsymbol{a}_j\right) a_{ij} \\ &= \sum_{j\in\mathcal{B}} \left|\mathbf{u}^\top \boldsymbol{b}_j\right|^{\alpha-1} \operatorname{sgn}\left(\mathbf{u}^\top \boldsymbol{b}_j\right) b_{ij}. \end{aligned} \tag{22}$$

Exploiting this equation, we prove the following lemma:

**Lemma 1** *Under the assumptions of Theorem 1, for any $j' \in \{1,\dots,n\}$, there exists $j \in \mathcal{B}$ and $t_{j'} \neq 0$, such that $t_{j'} \boldsymbol{a}_{j'} = \boldsymbol{b}_j$.*

**Proof:** Take $i'$ such that $a_{i'j'} \neq 0$. Also, for all $r = 1,\dots,n$, define

$$\mathcal{K}_r^a = \left\{\mathbf{u} \in \mathbb{R}^m | \mathbf{u}^\top \boldsymbol{a}_r = 0\right\} \tag{23}$$

which is an $(m-1)$-dimensional subspace of $\mathbb{R}^m$. Since for any $j \neq j'$, $\boldsymbol{a}_{j'}$ and $\boldsymbol{a}_j$ are linearly independent, the subspace $\mathcal{K}_{j'}^a \cap \mathcal{K}_j^a$ is $(m-2)$-dimensional. This implies that their $(m-1)$-dimensional Lebesgue measure is zero; and the same holds for the union $\bigcup_{j\neq j'} \left(\mathcal{K}_{j'}^a \cap \mathcal{K}_j^a\right)$. Since

$$\mathcal{K}_{j'}^a \setminus \bigcup_{j\neq j'} \mathcal{K}_j^a = \mathcal{K}_{j'}^a \setminus \bigcup_{j\neq j'} \left(\mathcal{K}_{j'}^a \cap \mathcal{K}_j^a\right), \tag{24}$$

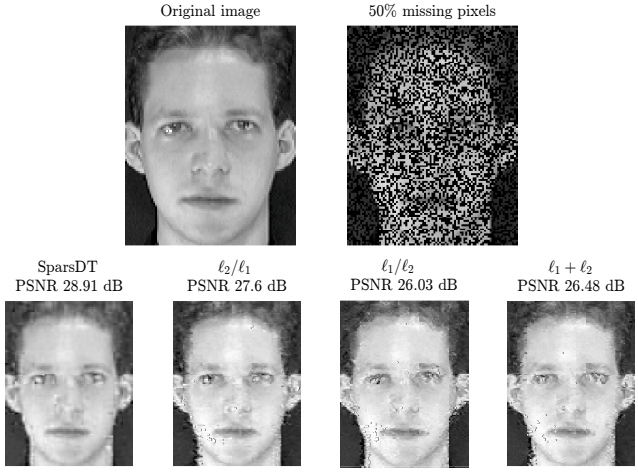


*Figure 4.* Comparing the proposed method with the existing ones for recovering missing pixels.

we conclude that the $(m-1)$-dimensional Lebesgue measure of $\mathcal{K}^a_{j'} \setminus \bigcup_{j\neq j'} \mathcal{K}^a_j$ is infinity.

Note that any $\mathbf{u} \in \mathcal{K}^a_{j'} \setminus \bigcup_{j\neq j'} \mathcal{K}^a_j$ is only orthogonal to $\boldsymbol{a}_{j'}$ and not to any other column of $\mathbf{A}$. This yields that if we set $i = i'$ in the left-hand side of (22), for any $\mathbf{u} \in \mathcal{K}^a_{j'} \setminus \bigcup_{j\neq j'} \mathcal{K}^a_j$, the only discontinuous term at $\mathbf{u}$ is the $j'$th one (because the function $|x|^{\alpha-1}\text{sgn}(x)$ has a single point of discontinuity at $x=0$). As a result, the sum itself is discontinuous over $\mathcal{K}^a_{j'} \setminus \bigcup_{j\neq j'} \mathcal{K}^a_j$. Hence, the same should hold for the right-hand side of the equation.

Similar to (23), define

$$\mathcal{K}^b_r = \left\{\mathbf{u} \in \mathbb{R}^m | \mathbf{u}^\top \boldsymbol{b}_r = 0\right\}. \tag{25}$$

The set of discontinuity points of the right-hand side of (22) is a subset of $\bigcup_{j\in\mathcal{B}} \mathcal{K}^b_j$. Therefore, we have

$$\mathcal{K}^a_{j'} \setminus \bigcup_{j\neq j'} \mathcal{K}^a_j \subseteq \bigcup_{j\in\mathcal{B}} \mathcal{K}^b_j \tag{26}$$

which can also be written as

$$\mathcal{K}^a_{j'} \setminus \bigcup_{j\neq j'} \mathcal{K}^a_j \subseteq \bigcup_{j\in\mathcal{B}} \left(\mathcal{K}^a_{j'} \cap \mathcal{K}^b_j\right) \tag{27}$$

Now, if none of the columns of $\mathbf{B}$ is linearly dependent to $\boldsymbol{a}_{j'}$, all $\mathcal{K}^a_{j'} \cap \mathcal{K}^b_j$ will be $(m-2)$-dimensional spaces, and their $(m-1)$-dimesnional Lebesgue measure is zero. This implies that the $(m-1)$-dimensional Lebesgue measure of the right-hand side of (27) is also zero, which contradicts the result after (24). Therefore, there exists a $j \in \mathcal{B}$ such that $\boldsymbol{b}_j$ is linearly dependent to $\boldsymbol{a}_{j'}$, which completes the proof of the lemma. □

The first consequence of Lemma 1 is that none of the columns of $\mathbf{B}$ is the zero vector and thus $\mathcal{B} = \{1, \ldots, n\}$.

Also, since all pairs of columns of $\mathbf{A}$ are linearly independent, the correspondence between a column of $\mathbf{A}$ and a column of $\mathbf{B}$ that are linearly dependent is one-to-one. Thus, we can simplify (22) to be

$$\sum_{j=1}^{n} (1-|t_j|) \left|\mathbf{u}^\top \boldsymbol{a}_j\right|^{\alpha-1} \text{sgn}\left(\mathbf{u}^\top \boldsymbol{a}_j\right) a_{ij} = 0, \tag{28}$$

which holds for all $\mathbf{u}$. This implies that the left hand-side of the above equation is a continuous function. However, as we saw in the proof of the lemma, every $\mathbf{u} \in \mathcal{K}^a_{j'} \setminus \bigcup_{j\neq j'} \mathcal{K}^a_j$ is a discontinuity point of the left-hand unless $1-|t'_j| = 0$ which completes the proof for the case of $0 < \alpha \leq 1$.

For the case of $1 < \alpha < 2$, we set $d = 2$ in (21) and obtain

$$\sum_{j=1}^{n} \left|\mathbf{u}^\top \boldsymbol{a}_j\right|^{\alpha-2} a_{ij}^2 = \sum_{j=1}^{n} \left|\mathbf{u}^\top \boldsymbol{b}_j\right|^{\alpha-2} b_{ij}^2. \tag{29}$$

Replacing (22) by (29), the same reasoning as for $0 < \alpha \leq 1$ works to prove the theorem for $1 < \alpha < 2$.

## B. Derivation of the gradient of $\mathcal{E}(\mathbf{B})$

To calculate the gradient of $\mathcal{E}(\mathbf{B})$, we first calculate the gradient of $\|\mathbf{B}^\top \mathbf{u}\|_\alpha^\alpha$ using the definition of the gradient, i.e.

$$\begin{aligned}\langle \nabla \|\mathbf{B}^\top \mathbf{u}\|_\alpha^\alpha, \mathbf{C}\rangle &= \frac{\partial}{\partial \epsilon} \left\|(\mathbf{B}^\top - \epsilon \mathbf{C}^\top)\mathbf{u}\right\|_\alpha^\alpha \Big|_{\epsilon=0} \\ &= \alpha \sum_{j=1}^{n} \boldsymbol{c}_j^\top \mathbf{u}\, \text{sgn}\left(\boldsymbol{b}_j^\top \mathbf{u}\right) \left|\boldsymbol{b}_j^\top \mathbf{u}\right|^{\alpha-1}.\end{aligned}$$

Here, $\langle \mathbf{D}, \mathbf{C}\rangle = \text{tr}(\mathbf{D}^\top \mathbf{C})$ is the standard inner product on the space of matrices. Writing the last equation in the matrix form, we obtain (20). Now, using the fact $\frac{\mathrm{d}}{\mathrm{d}x}|\log x| = \text{sgn}(\log x)\frac{1}{x}$ and the chain rule for differentiation yields (19).

## Acknowledgements

The research was partially supported by the Hasler Foundation under Grant 16009, by the European Research Council under Grant 692726 (H2020-ERC Project GlobalBioIm) and by the SNF Project Grant (205121 163385).

## References

Achim, A. and Kuruoglu, E. Image denoising using bivariate $\alpha$-stable distributions in the complex wavelet domain. *IEEE Signal Processing Letters*, 12(1):17–20, 2005.

Achim, A., Basarab, A., Tzagkarakis, G., Tsakalides, P., and Kouamé, D. Reconstruction of ultrasound RF echoes modeled as stable random variables. *IEEE Transactions on Computational Imaging*, 1(2):86–95, June 2015.

Amini, A., Unser, M., and Marvasti, F. Compressibility of deterministic and random infinite sequences. *IEEE Transactions on Signal Processing*, 59(11):5193–5201, November 2011.

Bickson, D. and Guestrin, C. Inference with multivariate heavy-tails in linear models. In *Advances in Neural Information Processing Systems*, pp. 208–216, 2010.

Gallagher, C. A method for fitting stable autoregressive models using the autocovariation function. *Statistics & probability letters*, 53(4):381–390, 2001.

Georgiou, P., Tsakalides, P., and Kyriakakis, C. Alpha-stable modeling of noise and robust time-delay estimation in the presence of impulsive noise. *IEEE transactions on multimedia*, 1(3):291–301, 1999.

Gribonval, R. and Lesage, S. A survey of sparse component analysis for blind source separation: principles, perspectives, and new challenges. In *ESANN'06 proceedings-14th European Symposium on Artificial Neural Networks*, pp. 323–330. d-side publi., 2006.

Gribonval, R., Cevher, V., and Davies, M. E. Compressible distributions for high-dimensional statistics. *IEEE Transactions on Information Theory*, 58(8):5016–5034, August 2012.

Helgason, S. *Integral Geometry and Radon Transforms*. Springer, 2010.

Herault, J. and Jutten, C. Space or time adaptive signal processing by neural network models. In *Neural networks for computing*, volume 151, pp. 206–211. AIP Publishing, 1986.

Hyvärinen, A. and Oja, E. Independent component analysis: algorithms and applications. *Neural networks*, 13(4): 411–430, 2000.

Hyvärinen, A., Karhunen, J., and Oja, E. *Independent component analysis*, volume 46. John Wiley & Sons, 2004.

Kuruoglu, E. E., Fitzgerald, W. J., and Rayner, P. J. Near optimal detection of signals in impulsive noise modeled with a symmetric/spl alpha/-stable distribution. *IEEE Communications Letters*, 2(10):282–284, 1998.

Lathauwer, L. De and Castaing, J. Blind identification of underdetermined mixtures by simultaneous matrix diagonalization. *IEEE Transactions on Signal Processing*, 56 (3):1096–1105, 2008.

Lathauwer, L. De, Castaing, J., and Cardoso, J. Fourth-order cumulant-based blind identification of underdetermined mixtures. *IEEE Transactions on Signal Processing*, 55 (6):2965–2973, 2007.

Ling, S. Self-weighted least absolute deviation estimation for infinite variance autoregressive models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(3):381–393, 2005.

Mairal, J., Bach, F., Ponce, J., and Sapiro, G. Online learning for matrix factorization and sparse coding. *Journal of Machine Learning Research*, 11(Jan):19–60, 2010.

Marvasti, F., Amini, A., Haddadi, F., Soltanolkotabi, M., Khalaj, B.H., Aldroubi, A., Sanei, S., and Chambers, J. A unified approach to sparse signal processing. *EURASIP journal on advances in signal processing*, 2012(1):1, 2012.

Meerschaert, M. and Scheffler, H. *Limit distributions for sums of independent random vectors: Heavy tails in theory and practice*, volume 321. John Wiley & Sons, 2001.

Middleton, D. Non-gaussian noise models in signal processing for telecommunications: new methods an results for class a and class b noise models. *IEEE Transactions on Information Theory*, 45(4):1129–1149, 1999.

Naik, G. and Kumar, D. An overview of independent component analysis and its applications. *Informatica*, 35(1), 2011.

Nikias, C. L. and Shao, M. *Signal Processing with Alpha-Stable Distributions and Applications*. Wiley, New York, 1995.

Nolan, JP. Modeling financial data with stable distributions. *Handbook of Heavy Tailed Distributions in Finance, Handbooks in Finance: Book*, 1:105–130, 2003.

Nolan, JP., Panorska, AK., and McCulloch, JH. Estimation of stable spectral measures. *Mathematical and Computer Modelling*, 34(9):1113–1122, 2001.

Pad, P. and Unser, M. Optimality of operator-like wavelets for representing sparse AR(1) processes. *IEEE Transactions on Signal Processing*, 63(18):4827–4837, September 2015.

Resnick, S. Heavy tail modeling and teletraffic data: special invited paper. *The Annals of Statistics*, 25(5):1805–1869, 1997.

Rolewicz, S. *Metric Linear Spaces*. Mathematics and its applications (D. Reidel Publishing Company).: East European series. D. Reidel, 1985.

Sahoo, S. and Makur, A. Signal recovery from random measurements via extended orthogonal matching pursuit. *IEEE Trans. Signal Processing*, 63(10):2572–2581, 2015.

Samoradnitsky, G. and Taqqu, M. *Stable non-Gaussian random processes: stochastic models with infinite variance*, volume 1. CRC press, 1994.

Shao, M. and Nikias, C. L. Signal processing with fractional lower order moments: stable processes and their applications. *Proceedings of the IEEE*, 81(7):986–1010, Jul 1993.