[Michael B. McCoy, Volkan Cevher, Quoc Tran Dinh, Afsaneh Asaei, and Luca Baldassarre]

Convexity in Source Separation

[Models, geometry, and algorithms]

Source separation, or demixing, is the process of extracting multiple components entangled within a signal. Contemporary signal processing presents a host of difficult source separation problems, from interference cancellation to background subtraction, blind deconvolution, and even dictionary learning. Despite the recent progress in each of these applications, advances in high-throughput sensor technology place demixing algorithms under pressure to accommodate extremely high-dimensional signals, separate an ever larger number of sources, and cope with more sophisticated signal and mixing models. These difficulties are exacerbated by the need for real-time action in automated decision-making systems.

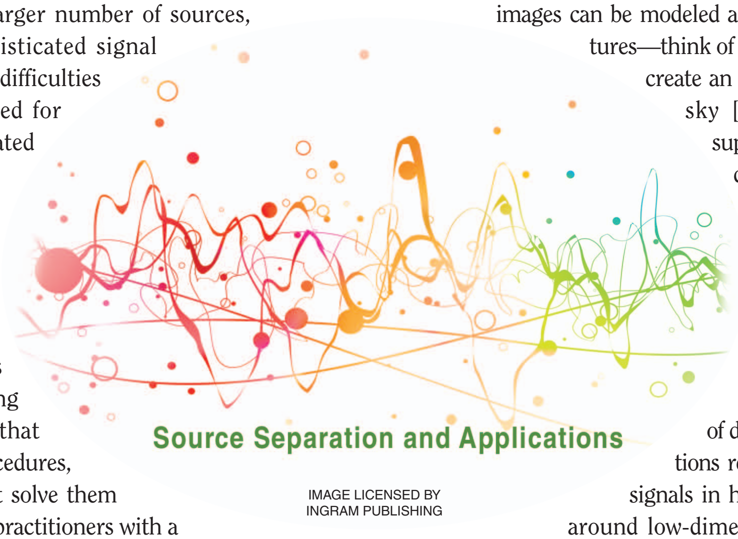


IMAGE LICENSED BY INGRAM PUBLISHING

Recent advances in convex optimization provide a simple framework for efficiently solving numerous difficult demixing problems. This article provides an overview of the emerging field, explains the theory that governs the underlying procedures, and surveys algorithms that solve them efficiently. We aim to equip practitioners with a toolkit for constructing their own demixing algorithms that work, as well as concrete intuition for why they work.

FUNDAMENTALS OF DEMIXING

The most basic model for mixed signals is a superposition model, where we observe a mixed signal $z_0 \in \mathbb{R}^d$ of the form

$$z_0 = x_0 + y_0, \tag{1}$$

and we wish to determine the component signals x_0 and y_0. This simple model appears in many guises. Sometimes, superimposed signals come from basic laws of nature. The amplitudes of electromagnetic waves, for example, sum together at a receiver, making the superposition model (1) common in wireless communications. Similarly, the additivity of sound waves makes superposition models natural in speech and audio processing.

Other times, a superposition provides a useful, if not literally true, model for more complicated nonlinear phenomena. Many images can be modeled as the sum of constituent features—think of stars and galaxies that sum to create an image of a piece of the night sky [1]. In machine learning, superpositions can describe hidden structure [2], while in statistics, superpositions can model gross corruptions to data [3]. These models also appear in texture repair [4], graph clustering [5], and line-spectral estimation [6].

A conceptual understanding of demixing in all of these applications rests on two key ideas. Natural signals in high dimensions often cluster around low-dimensional structures with few degrees of freedom relative to the ambient dimension [7]. Examples include bandlimited signals, array observations from seismic sources, and natural images. By identifying the convex functions that encourage these low-dimensional structures, we can derive convex programs that disentangle structured components from a signal.

Of course, effective demixing requires more than just structure. To distinguish multiple elements in a signal, the components must look different from one another. We capture this idea by saying that two structured families of signal are incoherent if their constituents appear very different from each other. While demixing is impossible without incoherence, sufficient incoherence

Digital Object Identifier 10.1109/MSP.2013.2296605
Date of publication: 7 April 2014

typically leads to provably correct demixing procedures. The two notions of structure and incoherence above also appear at the core of recent developments in information extraction from incomplete data in compressive sensing and other linear inverse problems [8], [9]. The theory of demixing extends these ideas to a richer class of signal models, and it leads to a more coherent theory of convex methods in signal processing.

While this article primarily focuses on mixed signals drawn from the superposition model (1), recent extensions to nonlinear mixing models arise in blind deconvolution, source separation, and nonnegative matrix factorization [10]–[12]. We will see that the same techniques that let us demix superimposed signals reappear in nonlinear demixing problems.

THE ROLE OF CONVEXITY

Convex optimization provides a unifying theme for all of the demixing problems discussed above. This framework is based on the idea that many structured signals possess corresponding convex functions that encourage this structure [9]. By combining these functions in a sensible way, we can develop convex optimization procedures that demix a given observation. The geometry of these functions lets us understand when it is possible to demix a superimposed observation with incoherent components [13]. The resulting convex optimization procedures usually have both theoretical and practical guarantees of correctness and computational efficiency.

To illustrate these ideas, we consider a classical but surprisingly common demixing problem: separating impulsive signals from sinusoidal signals, called the spikes and sines model. This model appears in many applications, including star–galaxy separation in astronomy, interference cancellation in communications, inpainting and speech enhancement in signal processing [1], [14].

While individual applications feature additional structural assumptions on the signals, a simple low-dimensional signal model effectively captures the main idea present in all of these works: sparsity. A vector $x_0 \in \mathbb{R}^d$ is sparse if most of its entries are equal to zero. Similarly, a vector $y_0 \in \mathbb{R}^d$ is sparse-in-frequency if its discrete cosine transform (DCT) Dy_0 is sparse, where $D \in \mathbb{R}^{d \times d}$ is the matrix that encodes the DCT. Sparse vectors capture impulsive signals like pops in audio, while sparse-in-frequency vectors explain smooth objects like natural images. Clearly, such signals look different from one another. In fact, an arbitrary collection of spikes and sines is linearly independent or incoherent provided that the collection is not too big [14].

Is it possible to demix a superimposition $z_0 = x_0 + y_0$ of spikes and sines into its constituents? One approach is to search for the sparsest possible constituents that generate the observation z_0

$$[\check{x}, \check{y}] := \underset{x,y \in \mathbb{R}^d}{\arg\min} \left\{ \|x\|_0 + \lambda \|Dy\|_0 : z_0 = x + y \right\}, \qquad (2)$$

where the ℓ_0-“norm” measures the sparsity of its input, and $\lambda > 0$ is a regularization parameter that trades the relative sparsity of solutions. Unfortunately, solving (2) involves an intractable computational problem. However, if we replace the ℓ_0 penalty with the convex ℓ_1-norm, we arrive at a classical sparse approximation program [14]

$$[\hat{x}, \hat{y}] := \underset{x,y \in \mathbb{R}^d}{\arg\min} \left\{ \|x\|_1 + \lambda \|Dy\|_1 : z_0 = x + y \right\}. \qquad (3)$$

This key change to the combinatorial proposal (2) offers numerous benefits. First, the procedure (3) is a convex program, and a number of highly efficient algorithms are available for its solution. Second, this procedure admits provable guarantees of correctness and noise-stability under incoherence. Finally, the demixing procedure (3) often performs admirably in practice.

Figure 1 illustrates the performance of (3) on both a synthetic signal drawn from the spikes-and-sines model above, as well as on a real astronomical image. The resulting performance for the basic model is quite appealing even for real data that mildly violates the modeling assumptions. Last but not least, this strong baseline performance can be obtained in fractions of seconds with simple and efficient algorithms. The combination of efficient algorithms, rigorous theory, and impressive real-world performance are hallmarks of convex demixing methods.

DEMIXING MADE EASY

This section provides a recipe to generate a convex program that accepts a mixed signal $z_0 = x_0 + y_0$ and returns a set of demixed components. The approach requires two ingredients. First, we must identify convex functions that promote the structure we expect in x_0 and y_0. Second, we combine these functions together into a convex objective. This simple and versatile approach easily extends to multiple signal components and undersampled observations.

STRUCTURE-INDUCING CONVEX FUNCTIONS

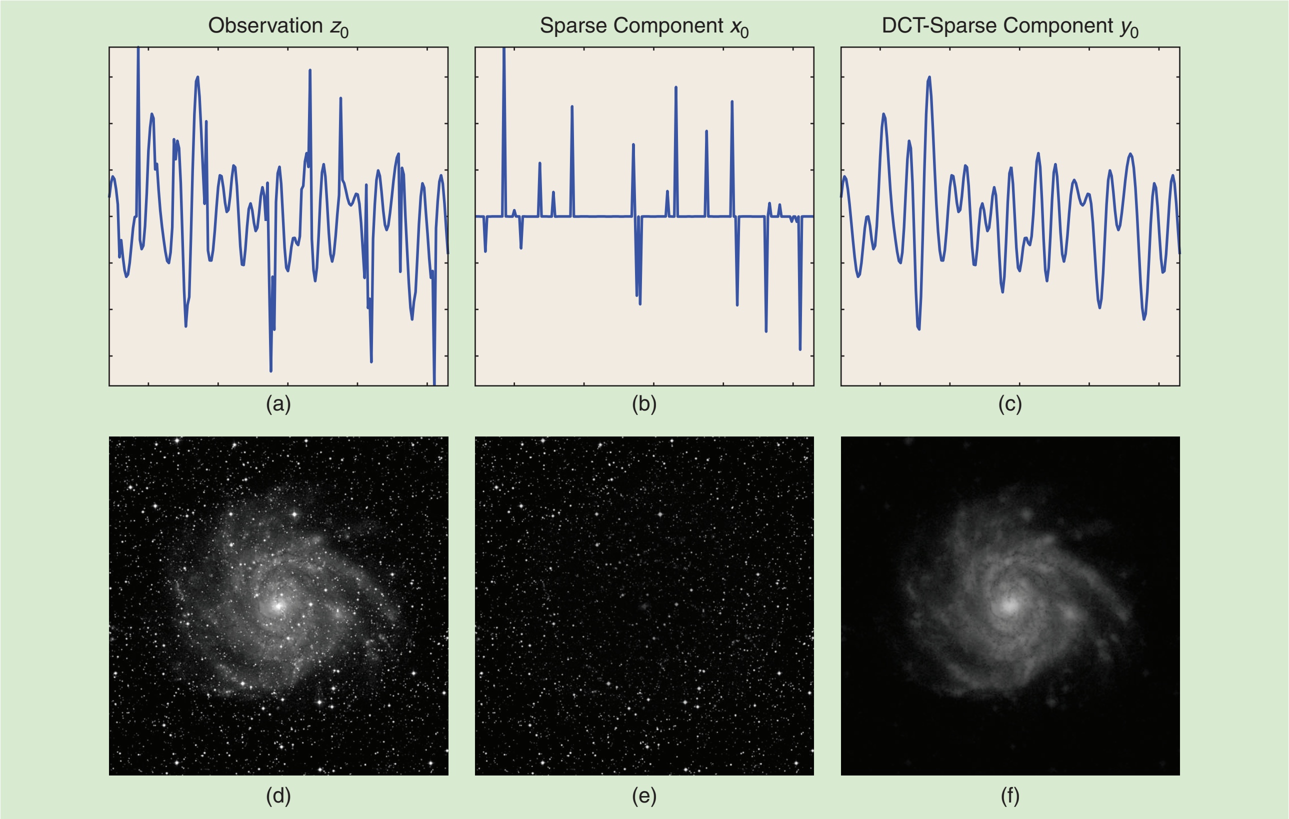
We say that a signal has structure when it has fewer degrees of freedom than the ambient space. Familiar examples of structured objects include sparse vectors, sign vectors, and low-rank matrices. It turns out that each of these structured families have an associated convex function, called an atomic gauge, adapted to their specific features [9].

The general principle is simple. Given a set of atoms $\mathcal{A} \subset \mathbb{R}^d$, we say that a signal $x \in \mathbb{R}^d$ is atomic if it is formed by a sum of a small number of scaled atoms. For example, sparse vectors are atomic relative to the set of standard basis vectors because every sparse vector is the sum of just a few standard basis vectors. For a more sophisticated example, recall that the singular value decomposition implies that low-rank matrices are the sum of a few rank-one matrices. Hence, low-rank matrices are atomic relative to the set of all rank-one matrices.

We can define a function that measures the inherent complexity of signals relative to a given set $\mathcal{A}$. One natural measure is the fewest number of scaled atoms required to write a signal using atoms from $\mathcal{A}$, but unfortunately, computing this quantity can be computationally intractable. Instead, we define the atomic gauge $\|x\|_{\mathcal{A}}$ of a signal $x \in \mathbb{R}^d$ by

$$\|x\|_{\mathcal{A}} := \inf \left\{ \lambda > 0 : x \in \lambda \cdot \operatorname{conv}(\mathcal{A}) \right\},$$

where $\operatorname{conv}(\mathcal{A})$ is the convex hull of $\mathcal{A}$. In other words, the level sets of the atomic gauge are the scaled versions of the convex hull of all the atoms $\mathcal{A}$ [Figure 2(a)].



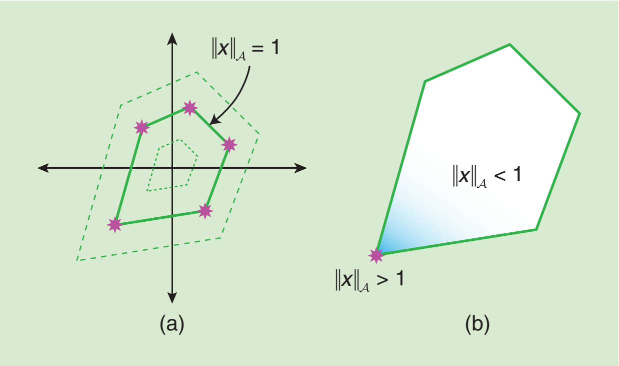
[FIG1] (a)–(c) We obtain perfect separation of spikes from sinusoids by solving (3). The original signal is perfectly separated into (b) its sparse component and (c) its DCT-sparse component. (d)–(f) We also achieve high-quality star-galaxy separation by solving (3) with an astronomical image. (d) The original is separated into (e) a starfield corresponding to a nearly sparse component and (f) a galaxy corresponding to a nearly two-dimensional DCT-sparse component. (Galaxy image courtesy of NASA/JPL-Caltech and used with permission.)

By construction, atomic gauges are "pointy" at atomic vectors. This property means that most deviations away from the atoms result in a rapid increase in the value of the gauge, so that the function tends to penalize deviations away from simple signals [Figure 2(b)]. The pointy geometry plays an important role in the theoretical understanding of demixing, as we will see when we discuss the geometry of demixing below.

A number of common structured families and their associated gauge functions appear in Table 1. More sophisticated examples include gauges for probability measures, cut matrices, and low-rank tensors. We caution, however, that not every atomic gauge is easy to compute, and so we must take care to develop tractable forms of atomic gauges [9], [16]. Surprisingly, it is sometimes easier to compute the value of atomic gauges than it is to compute the (possibly nonunique) decomposition of a vector into its atoms [12]. We will return to the discussion of tractable gauges when we discuss demixing algorithms below.

THE BASIC DEMIXING PROGRAM

Suppose that we know the signal components x_0 and y_0 are atomic with respect to the known atomic sets $\mathcal{A}_x$ and $\mathcal{A}_y$. In this section, we describe how to use the atomic gauge functions $\|\cdot\|_{\mathcal{A}_x}$ and $\|\cdot\|_{\mathcal{A}_y}$ defined above to help us demix the components x_0 and y_0 from the observation z_0.


$\|x\|_{\mathcal{A}} = 1$
$\|x\|_{\mathcal{A}} < 1$
$\|x\|_{\mathcal{A}} > 1$
(a)
(b)

[FIG2] (a) An atomic set $\mathcal{A}$, consisting of five atoms (stars). The "unit ball" of the atomic gauge $\|\cdot\|_{\mathcal{A}}$ is the closed convex hull of $\mathcal{A}$ (heavy line). Other level sets (dashed lines) of the gauge are dilations of the unit ball. (b) At an atom (star), the unit ball of $\|\cdot\|_{\mathcal{A}}$ tends to have sharp corners. Most perturbations away from this atom increase the value of $\|\cdot\|_{\mathcal{A}}$, so the atomic gauge often penalizes complex signals that are comprised of a large number of atoms.

Our intuition developed above indicates that the values $\|x_0\|_{\mathcal{A}_x}$ and $\|y_0\|_{\mathcal{A}_y}$ are relatively small because the vectors x_0 and y_0 are atomic with respect to the atomic sets $\mathcal{A}_x$ and $\mathcal{A}_y$. This suggests that we search for constituents that generate the

[TABLE 1] EXAMPLE SIGNAL STRUCTURES AND THEIR ATOMIC GAUGES [9], [15]. THE TOP TWO ROWS CORRESPOND TO VECTORS WHILE THE BOTTOM THREE REFER TO MATRICES. THE VECTOR NORMS EXTEND TO MATRIX NORMS BY TREATING $m \times n$ MATRICES AS LENGTH-mn VECTORS. THE EXPRESSION $\| \boldsymbol{x} \|_2$ DENOTES THE EUCLIDEAN NORM OF THE VECTOR $\boldsymbol{x}$, WHILE $\sigma_i(\boldsymbol{X})$ RETURNS THE iTH SINGULAR VALUE OF THE MATRIX $\boldsymbol{X}$.

STRUCTURE	ATOMIC SET	ATOMIC GAUGE $\| \cdot \|_{\mathcal{A}}$
SPARSE VECTOR	SIGNED BASIS VECTORS $\{\pm e_i\}$	ℓ_1 NORM $\| x \|_{\ell_1} = \sum_i \lvert x_i \rvert$
BINARY SIGN VECTOR	SIGN VECTORS $\{\pm 1\}^d$	ℓ_∞ NORM $\| x \|_{\ell_\infty} = \max_i \lvert x_i \rvert$
LOW-RANK MATRIX	RANK-1 MATRICES $\{uv^t : \| uv^t \|_F = 1\}$	SCHATTEN 1-NORM $\| X \|_{S_1} = \sum_i \sigma_i(\boldsymbol{X})$
ORTHOGONAL MATRIX	ORTHOGONAL MATRICES $\{O : OO^t = \mathbf{I}\}$	SCHATTEN ∞-NORM $\| \boldsymbol{X} \|_{S_\infty} = \sigma_1(\boldsymbol{X})$
ROW-SPARSE MATRIX	MATRICES WITH ONE NONZERO ROW $\{e_i v^t : \| v \|_2 = 1\}$	ROW-ℓ_1 NORM $\| \boldsymbol{X} \|_{\ell_1/\ell_2}$

observation and have small atomic gauges. That is, we determine the demixed constituents $\hat{x}, \hat{y}$ by solving

$$[\hat{x}, \hat{y}] := \underset{x,y \in \mathbb{R}^d}{\arg\min} \left\{ \| x \|_{\mathcal{A}_x} + \lambda \| y \|_{\mathcal{A}_y} : x + y = z_0 \right\}. \tag{4}$$

The parameter $\lambda > 0$ negotiates a tradeoff between the relative importance of the atomic gauges, and the constraint $\boldsymbol{x} + \boldsymbol{y} = \boldsymbol{z}_0$ ensures that our estimates $\hat{\boldsymbol{x}}$ and $\hat{\boldsymbol{y}}$ satisfy the observation model (1). The hope, of course, is that $\hat{\boldsymbol{x}} = \boldsymbol{x}_0$ and $\hat{\boldsymbol{y}} = \boldsymbol{y}_0$, so that the demixing program (4) actually identifies the true components in the observation $\boldsymbol{z}_0$.

The demixing program (4) is closely related to linear inverse problems and compressive sampling (CS) [8], [9]. Indeed, the summation map $(\boldsymbol{x}, \boldsymbol{y}) \mapsto \boldsymbol{x} + \boldsymbol{y}$ is a linear operator, so demixing amounts to inverting an underdetermined linear system using structural assumptions. The main conceptual difference between demixing and standard CS is that demixing treats the components $\boldsymbol{x}_0$ and $\boldsymbol{y}_0$ as unrelated structures. Also, unlike conventional CS, demixing does not require exact knowledge of the atomic decomposition, but only the value of the gauge.

The only link between the structures that appears in our recipe comes through the choice of tuning parameter λ in (4), which makes these convex demixing procedures easily adaptable to new problems. In general, determining an optimal value of λ may involve fine-tuning or cross-validation, which can be quite computationally demanding in practice. Some theoretical guidance on explicit choices of the regularization parameter appears in [2], [3], and [17].

EXTENSIONS

There are many extensions of the linear superposition model (1). In some applications, we are confronted with a signal that is only partially observed—compressive demixing. In others, we might consider an observation with additive noise, for instance, or a signal with more than two components. The same ingredients that we introduced above can be used to demix signals from these more elaborate models.

For example, if we only see $\boldsymbol{z}_0 = \Phi(\boldsymbol{x}_0 + \boldsymbol{y}_0)$, a linear mapping of the superposition, then we simply update the consistency constraint in the usual demixing program (4) and solve instead

$$[\hat{x}, \hat{y}] := \underset{x,y \in \mathbb{R}^d}{\arg\min} \left\{ \| x \|_{\mathcal{A}_x} + \lambda \| y \|_{\mathcal{A}_y} : \Phi(x + y) = z_0 \right\}. \tag{5}$$

Some applications for this undersampled demixing model appear in image alignment [18], robust statistics [5], and graph clustering [19].

Another straightforward extension involves demixing more than two signals. For example, if we observe $\boldsymbol{z}_0 = \boldsymbol{x}_0 + \boldsymbol{y}_0 + \boldsymbol{w}_0$, the sum of three structured components, we can determine the components by solving

$$\begin{aligned} [\hat{x}, \hat{y}, \hat{w}] := \underset{x,y,w \in \mathbb{R}^d}{\arg\min} & \left\{ \| x \|_{\mathcal{A}_x} + \lambda_1 \| y \|_{\mathcal{A}_y} + \lambda_2 \| w \|_{\mathcal{A}_w} \right. \\ & \left. : x + y + w = z_0 \right\}, \end{aligned} \tag{6}$$

where $\mathcal{A}_w$ is an atomic set tuned to $\boldsymbol{w}_0$, and as before, the parameters $\lambda_i > 0$ trade off the relative importance of the regularizers. This model appears, for example, in image processing applications where multiple basis representations, such as curvelets, ridgelets, and shearlets, explain different morphological components [1]. Further modifications along the lines above extend the demixing framework to a massive number of problems relevant to modern signal processing.

GEOMETRY OF DEMIXING

A critical question we can ask about a demixing program is "When does it work?" Answers to this question can be found by studying the underlying geometry of convex demixing programs. Surprisingly, we can characterize the success and failure of convex demixing precisely by leveraging a basic randomized model for incoherence. Indeed, the geometric viewpoint reveals a tight characterization of the success and failure of demixing in terms of geometric parameters that act as the "degrees of freedom" of the mixed signal. The consequences for demixing are intuitive: demixing succeeds if and only if the dimensionality of the observation exceeds the total degrees of freedom in the signal.

DESCENT CONES AND THE STATISTICAL DIMENSION

Our study of demixing begins with a basic object that encodes the local geometry of a convex function. The descent cone $\mathcal{D}(\mathcal{A}, \boldsymbol{x})$ at a point $\boldsymbol{x}$ with respect to an atomic set $\mathcal{A} \subset \mathbb{R}^d$ consists of the directions where the gauge function $\| \cdot \|_{\mathcal{A}}$ does not increase near $\boldsymbol{x}$. Mathematically, the descent cone is given by

$$\mathcal{D}(\mathcal{A}, x) := \left\{ h : \| x + \tau h \|_{\mathcal{A}} \leq \| x \|_{\mathcal{A}} \text{ for some } \tau > 0 \right\}.$$

The descent cone encodes detailed information about the local behavior of the atomic gauge $\| \cdot \|_{\mathcal{A}}$ near $\boldsymbol{x}$. Since local optimality implies global optimality in convex optimization, we can characterize when demixing succeeds in terms of a configuration of descent cones. See Figure 3 for a precise description of this optimality condition.

To understand when the geometric optimality condition is likely to hold, we need a measure for the "size" of cones. The most apparent measure of size is perhaps the solid angle, which quantifies the amount of space occupied by a cone. The solid angle, however, proves inadequate for describing the intersection of cones even in the simple case of linear subspaces. Indeed, linear subspaces are cones that take up no space at all, but when their dimensions are large enough, any two subspaces will always intersect along a line. Imagine trying to arrange two flat sheets of paper so that they only touch at their centers: it's impossible!

We find a much more informative measure of size, called the statistical dimension, when we measure the proportion of space near a cone, rather than the proportion inside the cone.

DEFINITION 1 (STATISTICAL DIMENSION)

Let $C \subset \mathbb{R}^d$ be a closed convex cone, and denote by $\boldsymbol{\Pi}_C(\boldsymbol{x}) := \arg\min_{\boldsymbol{y} \in C} \| \boldsymbol{x} - \boldsymbol{y} \|$ the closest point in C to $\boldsymbol{x}$. We define the statistical dimension $\delta(C)$ of a convex cone $C \subset \mathbb{R}^d$ by

$$\delta(C) := \mathbb{E} \| \boldsymbol{\Pi}_C(\boldsymbol{g}) \|_2^2, \tag{7}$$

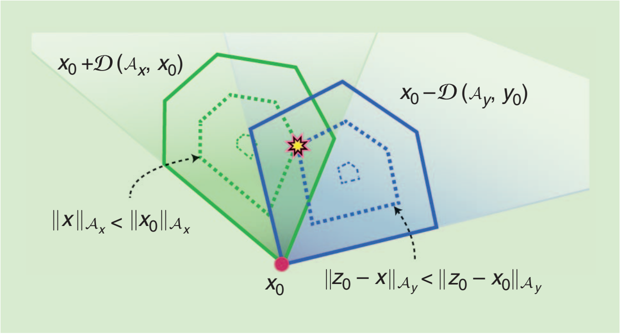
where $\boldsymbol{g} \sim \text{Normal}(0, \mathbf{I})$ is a standard Gaussian random variable and the letter $\mathbb{E}$ denotes the expected value.

The statistical dimension gets its name because it extends many properties of the usual dimension of linear subspaces to convex cones [20], and it is closely related to the Gaussian width used in [9]. Our interest here, however, comes from the interpretation of the statistical dimension as a "size" of a cone. A large statistical dimension $\delta(C) \approx d$ means that $\| \boldsymbol{\Pi}_C(\boldsymbol{g}) \|_2^2$ is usually large, i.e., most points lie near or inside the cone. Conversely, a narrow cone C possesses a small statistical dimension because the nearest point to C is typically close to zero, which drives down the average norm. We will see below that the statistical dimension of descent cones provides the key parameter for understanding the success and failure of demixing procedures.

Of course, a parameter is only useful if we can compute it. Fortunately, the statistical dimension of descent cones is often easy to compute or approximate. Several ready-made statistical dimension formulas and a step-by-step recipe for accurately deriving new formulas appear in [20]. Some useful approximate statistical dimension calculations can also be found in the works [9] and [17]. As an added bonus, recent work indicates that statistical dimension calculations are closely related to the problem of finding optimal regularization parameters [17, Th. 2].

PHASE TRANSITIONS IN CONVEX DEMIXING

The true power of the statistical dimension comes from its ability to predict phase transitions in demixing programs. By phase transition, we mean the peculiar behavior where demixing programs switch from near-certain failure to near-certain success within a narrow range of model parameters. While the optimality condition from Figure 3 characterizes the success and failure of demixing, it is often difficult to certify directly. To understand how demixing operates in typical situations, we need an incoherence model. One proposal to model incoherence assumes that the structured signals are oriented generically relative to one another. This is achieved, for example, by assuming that the structured components are drawn structured relative to a rotated atomic set $Q\mathcal{A}$, where $Q \in \mathbb{R}^{d \times d}$ is a random orthogonal matrix [13]. Surprisingly, this basic randomized model of incoherence leads to a rich theory with precise guarantees that complement other phase transition characterizations in linear inverse problems [21], [22]. Many works propose alternative incoherence models applicable to specific cases, including [3] and [9], but these specific choices do not possess known phase transitions. Under the random model of [13], however, a very general theory is available. The following result appears in [20, Th. III].


$x_0 + \mathcal{D}(\mathcal{A}_x, x_0)$
$x_0 - \mathcal{D}(\mathcal{A}_y, y_0)$
$\|x\|_{\mathcal{A}_x} < \|x_0\|_{\mathcal{A}_x}$
x_0
$\|z_0 - x\|_{\mathcal{A}_y} < \|z_0 - x_0\|_{\mathcal{A}_y}$

[FIG3] The geometric characterization of demixing. When the descent cones $\mathcal{D}(\mathcal{A}_x, \boldsymbol{x}_0)$ and $\mathcal{D}(\mathcal{A}_y, \boldsymbol{y}_0)$ share a line, then there is an optimal point $\hat{\boldsymbol{x}}$ (star) for the demixing program (4) not equal to $\boldsymbol{x}_0$. Conversely, demixing can succeed for some value of $\lambda > 0$ if the two descent cones touch only at the origin. In other words, demixing can succeed if and only if $\mathcal{D}(\mathcal{A}_x, \boldsymbol{x}_0) \cap -\mathcal{D}(\mathcal{A}_y, \boldsymbol{y}_0) = \{0\}$ [13].

THEOREM 1

Suppose that the atomic set of $\boldsymbol{x}_0$ is randomly rotated, i.e., that $\mathcal{A}_x = Q\tilde{\mathcal{A}}_x$ for some random rotation Q and some fixed atomic set $\tilde{\mathcal{A}}_x$. Fix a probability tolerance $\eta \in (0,1)$, and define the normalized total statistical dimension

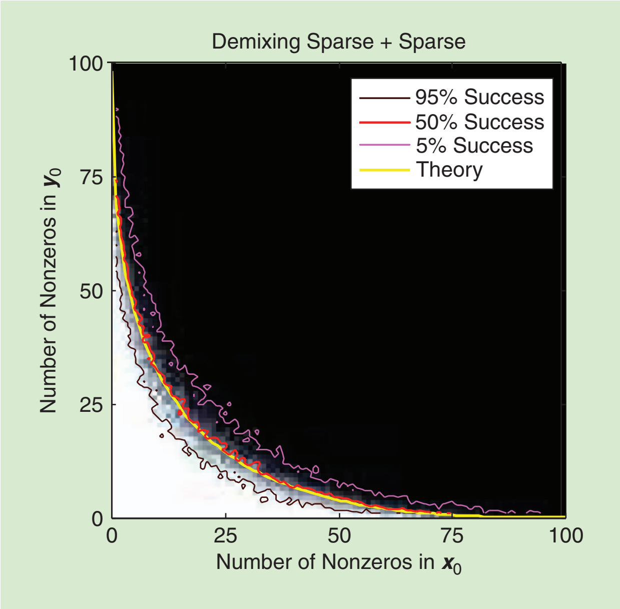
$$\Delta := \frac{1}{d}[\delta(\mathcal{D}(\tilde{\mathcal{A}}_x, \boldsymbol{x}_0)) + \delta(\mathcal{D}(\mathcal{A}_y, \boldsymbol{y}_0))].$$

Then there is a scalar $C > 0$ that depends only on η such that

$$\Delta \le 1 - C/\sqrt{d} \Rightarrow \text{demixing can succeed with probability} \ge 1 - \eta$$
$$\Delta \ge 1 + C/\sqrt{d} \Rightarrow \text{demixing always fails with probability} \ge 1 - \eta.$$

In fact, we can take $C := 4\sqrt{\log(4/\eta)}$. By "demixing can succeed," we mean that there exists a regularization parameter $\lambda > 0$ so that $(\boldsymbol{x}_0, \boldsymbol{y}_0)$ is an optimal point of (4). "Demixing always fails" means that $(\boldsymbol{x}_0, \boldsymbol{y}_0)$ is not an optimal point of (4) for any parameter $\lambda > 0$.

Theorem 1 indicates that demixing exhibits a phase transition as the normalized statistical dimension Δ increases beyond the one. The first implication above tells us that if Δ is just a little less than one, then we can be confident that demixing will succeed for some tuning parameter $\lambda > 0$. On the other hand, the second implication says that if Δ is slightly larger than one, then demixing is hopeless. See Figure 4 for an example of the accuracy of this



[FIG4] **Phase transitions in demixing. Phase transition diagram for demixing two sparse signals using ℓ_1 minimization [20]. This experiment replaces the DCT matrix D in (3) with a random rotation Q. The color map shows the transition from pure success (white) to complete failure (black). The 95%, 50%, and 5% empirical success contours (tortuous curves) appear above the theoretical phase transition curve (yellow), where $\Delta = 1$. See [13] for experimental details. (Figure used with permission from [20].)**

theory for the sparse approximation model (3) from the introduction when the DCT matrix D is replaced with a random rotation Q. The agreement between the empirical 50% success line and the curve where $\Delta = 1$ is remarkable.

This theory extends analogously to the compressive and multiple demixing models (5) and (6). Under a similar incoherence model as above, compressive and multiple demixing are likely to succeed if and only if the sum of the statistical dimensions is slightly less than the number of (possibly compressed) measurements [23, Th. A]. This fact lets us interpret the statistical dimension $\delta(\mathcal{D}(\mathcal{A}, x_0))$ as the degrees of freedom of a signal x_0 with respect to the atomic set $\mathcal{A}$. The message is clear: Incoherent demixing can succeed if and only if the total dimension of the observation exceeds the total degrees of freedom of the constituent signals.

PRACTICAL DEMIXING ALGORITHMS

In theory, many demixing problem instances of the form (4) admit efficient numerical solutions. Indeed, if we can transform these problems into standard linear, cone, or semidefinite formulations, we can apply black-box interior point methods to obtain high-accuracy solutions in polynomial time [24]. In practice, however, the computational burden of interior point methods makes these methods impracticable as the dimension d of the problem grows. Fortunately, a simple and effective iterative algorithm for computing approximate solutions to the demixing program (4) and its extensions can be implemented with just a few lines of high-level code.

SPLITTING THE WORK

The simplest and most popular method for iteratively solving demixing programs goes by the name alternating direction method of multipliers (ADMM). The key object in this algorithm is the augmented Lagrangian function L_ρ defined by

$$\begin{aligned} L_\rho(x, y, w) := {} & \| x \|_{\mathcal{A}_x} + \lambda \| y \|_{\mathcal{A}_y} + \langle w, x + y - z_0 \rangle \\ & + \frac{1}{2\rho} \| x + y - z_0 \|^2, \end{aligned}$$

where $\langle \cdot, \cdot \rangle$ denotes the usual inner product between two vectors and $\rho > 0$ is a parameter that can be tuned to the problem. Starting with arbitrary points $x^1, y^1, w^1 \in \mathbb{R}^d$, the ADMM method generates a sequence of points iteratively as

$$\begin{cases} x^{k+1} & = \arg\min_{x \in \mathbb{R}^d} L_\rho(x, y^k, w^k) \\ y^{k+1} & = \arg\min_{y \in \mathbb{R}^d} L_\rho(x^{k+1}, y, w^k) \\ w^{k+1} & = w^k + (x^{k+1} + y^{k+1} - z_0)/\rho. \end{cases} \tag{8}$$

In other words, the x- and y-updates iteratively minimize the Lagrangian over just one parameter, leaving all others fixed. The alternating minimization of L_ρ gives the method its name. Despite the simple updates, the sequence (x^k, y^k) of iterates generated in this manner converges to the minimizers $(\hat{x}, \hat{y})$ of the demixing program (4) under fairly general conditions [25].

The key to the efficiency of ADMM comes from the fact that the updates are often easy to compute. By completing the square, the x- and y-updates above amount to evaluating proximal operators of the form

$$\begin{aligned} x^{k+1} &= \arg\min_{x \in \mathbb{R}^d} \| x \|_{\mathcal{A}_x} + \frac{1}{2\rho} \| u^k - x \|^2 \quad \text{and} \\ y^{k+1} &= \arg\min_{y \in \mathbb{R}^d} \lambda \| y \|_{\mathcal{A}_y} + \frac{1}{2\rho} \| v^k - y \|^2, \end{aligned} \tag{9}$$

where $u^k := z_0 - y^k - \rho w^k$ and $v^k := z_0 - x^{k+1} - \rho w^k$. When solutions to the proximal minimizations (9) are simple to compute, each iteration of ADMM is highly efficient.

Fortunately, proximal operators are easy to compute for many atomic gauges. For example, when the atomic gauge is the ℓ_1 -norm, the proximal operator corresponds to "soft thresholding"

$$\arg\min_{x \in \mathbb{R}^d} \| x \|_{\ell_1} + \frac{1}{2\rho} \| u - x \|^2 = \text{soft}(u, \rho) = \begin{cases} u_i - \rho, & u_i > \rho, \\ 0, & |u_i| \le \rho, \\ u_i + \rho, & u_i < \rho. \end{cases}$$

If we replace the ℓ_1-norm above with the Schatten-1 norm, then the corresponding proximal operator amounts to soft thresholding the singular values. Numerous other explicit examples of proximal operations appear in [25, Sec. 2.6].

Not all atomic gauges, however, have efficient proximal operations. Even sets with finite number of atoms do not necessarily lead to more efficient proximal maps than sets with an infinite number of atoms. For instance, when the atomic set consists of rank-one matrices with unit Frobenius norm, we have an infinite set of atoms and yet the proximal map can be efficiently obtained via singular value thresholding. On the other hand, when the atomic set consists of rank-one matrices with binary ±1 entries, we have a finite set of atoms and yet the best-known algorithm for computing the proximal map requires an intractable amount of computation.

There is some hope, however, even for difficult gauges. Recent algebraic techniques for approximating atomic gauges provide computable proximal operators in a relatively efficient manner, which opens the door to additional demixing algorithms for richer signal structures [9], [16].

EXTENSIONS

While the ADMM method is the prime candidate for solving problem (4), it is not usually the best method for the extensions (5) or (6). In the first case, if $\boldsymbol{\Phi}$ is a general linear operator, it creates a major computational bottleneck since we need an additional loop to solve the subproblems within the ADMM algorithm. In the latter case, ADMM even loses convergence guarantees [26].

One possible way to handle both (5) and (6) is to use decomposition methods. Roughly speaking, these methods decompose (5) or (6) into smaller components and then solve the convex subproblem corresponding to each term simultaneously. For example, we can use the decomposition method from [27]

$$\begin{cases} \boldsymbol{v}^k & = \boldsymbol{w}^k + \rho(\boldsymbol{\Phi}(\boldsymbol{x}^k + \boldsymbol{y}^k) - \boldsymbol{z}_0) \\ \boldsymbol{x}^{k+1} & = \arg\min_{\boldsymbol{x}\in\mathbb{R}^d} \| \boldsymbol{x} \|_{\mathcal{A}_x} + \langle \boldsymbol{v}^k, \boldsymbol{\Phi}\boldsymbol{x} \rangle + \frac{1}{2\rho}\| \boldsymbol{x} - \boldsymbol{x}^k \|_2^2 \\ \boldsymbol{y}^{k+1} & = \arg\min_{\boldsymbol{y}\in\mathbb{R}^d} \lambda\| \boldsymbol{y} \|_{\mathcal{A}_y} + \langle \boldsymbol{v}^k, \boldsymbol{\Phi}\boldsymbol{y} \rangle + \frac{1}{2\rho}\| \boldsymbol{y} - \boldsymbol{y}^k \|_2^2 \\ \boldsymbol{w}^{k+1} & = \boldsymbol{w}^k + \rho(\boldsymbol{\Phi}(\boldsymbol{x}^{k+1} + \boldsymbol{y}^{k+1}) - \boldsymbol{z}_0). \end{cases} \qquad (10)$$

When the parameter ρ is chosen appropriately, the generated sequence $\{(\boldsymbol{x}^k, \boldsymbol{y}^k)\}$ in (10) converges to the solution of (5). Since the second and the third lines of (10) are independent, it is even possible to solve them in parallel. This scheme easily extends to demixing three or more signals (6).

Another practical method appears in [28]. In essence, this approach combines a dual formulation, Nesterov's smoothing technique, and the fast gradient method [24]. This technique works both for (5) and (6), and it possesses a rigorous $O(1/k)$ convergence rate.

EXAMPLES

The ideas above apply to a large number of examples. Here, we highlight some recent applications of convex demixing in signal processing. The first example, texture inpainting, uses a low-rank and sparse decomposition to discover and repair axis-aligned texture in images. The second example uses the low-rank and diagonal demixing of a sensor array correlation matrix to improve beamforming.

TEXTURE INPAINTING

Many natural and man-made images include highly regular textures. These repeated patterns, when aligned with the image frame, tend to have very low rank. Of course, rarely does a natural image consist solely of a texture. Often, though, a background texture is sparsely occluded by a untextured component. By modeling the occlusion as an additive error, we can use convex demixing to solve for the underlying texture and extract the occlusion [4].

In this model, we treat the observed digital image $Z_0 \in \mathbb{R}^{m\times n}$ as a matrix formed by the sum $Z_0 = X_0 + Y_0$, where the textured component X_0 has low rank and Y_0 is a sparse corruption or occlusion. The natural demixing program in this setting is the rank-sparsity decomposition [2], [3]

$$[\hat{X}, \hat{Y}] = \underset{X,Y\in\mathbb{R}^{m\times n}}{\arg\min} \| X \|_{S_1} + \lambda \| Y \|_1 \quad \text{subject to} \quad X + Y = Z_0, \qquad (11)$$

This unsupervised texture-repair method exhibits a state-of-the-art performance, exceeding even the quality of a supervised procedure built in to Adobe Photoshop on some images [4]. When applied, e.g., to an image of a chessboard, the method flawlessly recovers the checkerboard from the pieces (Figure 5).

BEAMFORMING

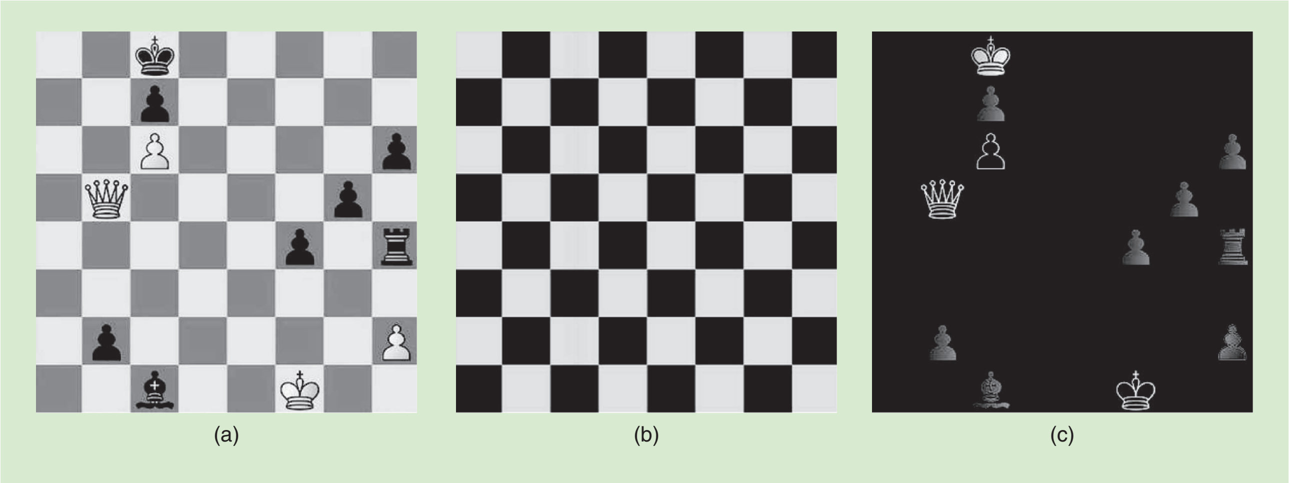
We describe a convex demixing program for signal estimation via beamforming. Beamforming uses an array of n sensors to acquire a source signal from a given direction while suppressing the sources interfering from distinct directions. Denoting the signal of a sensor array with $(S \in C^{n\times 1})$ where is the number of snapshots, the desired signal is estimated with $(w^t\, S)$, where $(w \in C^{n\times 1})$ is known as the beamforming weights. Assuming that the signal impinges on the array from the direction (d), the optimal weights for signal prediction are obtained as $(\mu\, Z_0^{-1}\, d)$ where $(Z_0 = \mathbb{E}[S\, S^t])$ is the correlation matrix and (μ) stands for a correction factor to cancel the distortions [29]. When the sources are independent, the joint expected correlation matrix Z_0 of the sensor array signals takes the form $Z_0 = A_0 A_0^t + Y_0$, where the column space of the $n \times r$ matrix A_0 encodes the bearing information from r sources, and Y_0 is the covariance matrix of the noise at the sensors.

When the number of sources r is much smaller than the number of sensors n, the matrix $X_0 := A_0 A_0^t$ is positive semidefinite and has low rank. Moreover, when the sensor noise is uncorrelated, the matrix Y_0 is diagonal. Using the atomic gauge recipe from above, we can demix X_0 and Y_0 from the empirical covariance matrix $\hat{Z}_0$ by setting

$$\begin{aligned} [\hat{X}, \hat{Y}, \hat{E}] = \underset{X,Y\in\mathbb{R}^{n\times n}}{\arg\min} \quad & \| X \|_{S_1^+} + \| Y \|_{\text{diag}} + \lambda \| E \|_{\text{Fro}}^2 \\ \text{subject to} \quad & X + Y + E = \hat{Z}_0, \end{aligned} \qquad (12)$$

where E absorbs the deviations in the expectation model due to the finite sample size. Here, $\| \cdot \|_{S_1^+}$ is the atomic gauge generated by positive semidefinite rank-one matrices, which is equal to the trace for positive semidefinite matrices, but returns $+\infty$ when its argument has a negative eigenvalue. Similarly, the gauge $\| \cdot \|_{\text{diag}}$ is the atomic gauge generated by the set of all diagonal matrices, and so it is equal to zero on diagonal matrices but $+\infty$ otherwise. The norm $\| \cdot \|_{\text{Fro}}$ is the usual Frobenius norm on a matrix. The results of [11] relate the success of a similar problem to the geometric problem of ellipsoid fitting, and show that, under some incoherence assumptions, the method (12) succeeds.

In beamforming, the array correlation matrix plays a key role in estimating the optimal weights. For instance, minimum variance distortionless response (MVDR) beamforming exploits



[FIG5] Texture inpainting (white to move, checkmate in two). The rank-sparsity decomposition (11) perfectly separates the chessboard from the pieces. (a) The original image. (b) The low-rank component. (c) The sparse component.

the correlation matrix to estimate the source signals at a given direction. The presence of noise corrupts the empirical correlation matrix estimate, which deteriorates the beamforming performance by MVDR.

The approach in [31] assumes a low-rank correlation matrix and discusses source estimation using atomic regularization. Hence, the demixing results perfectly dovetail with this beamforming approach. To see synergy, we simulate a scenario where three sources impinge on a uniform linear array of ten sensors from far-field in free space. The input source-to-interference ratio (SIR) is –5 dB. In addition, we add isotropic noise to the sensor measurements at –10 dB source-to-noise ratio (SNR).

The results are quite encouraging. The average output SIR of the standard MVDR beamformer using the empirical correlation $\hat{\boldsymbol{Z}}_0$ turns out to be 5 dB. The beamforming approach [31] with the empirical correlation estimate yields 6.3 dB SIR, while using the demixed estimate $\hat{X}$ of (12) results in an impressive 9.4 dB SIR—with an approximate improvement of 3 dB in interference suppression for source detection.

HORIZONS: NONLINEAR SEPARATION

We conclude our demixing tutorial with some promising directions for the future. In many applications, the constituent signals are tangled together in a nonlinear fashion [10], [12]. While this situation would seem to rule out the linear superposition model considered above, we can leverage the same convex optimization tools to obtain demixing guarantees and often return to a linear model using a technique called semidefinite relaxation.

We describe the basic idea behind this maneuver with a concrete application: blind deconvolution. Convolved signals appear frequently in communications due, e.g., to multipath channel effects. When the channel is known, removing the channel effects is a difficult but well-understood linear inverse problem. With blind deconvolution, however, we see only the convolved signal $\boldsymbol{z}_0 = \boldsymbol{x}_0 * \boldsymbol{y}_0$ from which we must determine both the channel $\boldsymbol{x}_0 \in \mathbb{R}^m$ and the source $\boldsymbol{y}_0 \in \mathbb{R}^d$.

While the convolution $\boldsymbol{x}_0 * \boldsymbol{y}_0$ involves nonlinear interactions between $\boldsymbol{x}_0$ and $\boldsymbol{y}_0$, the convolution is in fact linear in the matrix formed by the outer product $\boldsymbol{x}_0\boldsymbol{y}_0^t$. In other words, there is a linear operator $C : \mathbb{R}^{m\times d} \to \mathbb{R}^{m+d}$ such that

$$\boldsymbol{z}_0 = C(X_0) \quad \text{where} \quad X_0 := \boldsymbol{x}_0\boldsymbol{y}_0^t.$$

The matrix X_0 has rank one by definition, so it is natural use the Schatten 1-norm to search for low-rank matrices that generate the observed signal

$$\hat{X} = \underset{X\in\mathbb{R}^{m\times d}}{\arg\min} \ \|X\|_{S_1} \quad \text{subject to} \quad \boldsymbol{z}_0 = C(X).$$

This is the basic idea behind the convex approach to blind deconvolution of [10].

The implications of the nonlinear demixing example above are far-reaching. There are large classes of signal and mixing models that support efficient, provable, and stable demixing. Viewing different demixing problems within a common framework of convex optimization, we can leverage decades of research in various diverse disciplines from applied mathematics to signal processing, and from theoretical computer science to statistics. We expect that the diversity of convex demixing models and geometric tools will also inspire the development of new kinds of scalable optimization algorithms that handle nonconventional cost functions [30].

AUTHORS

Michael B. McCoy (mccoy@caltech.edu) received the B.S. degree in electrical engineering (honors) in 2007 from the University of Texas at Austin and the Ph.D. degree in applied and computational mathematics in 2013 from the California Institute of Technology (Caltech), Pasadena. His thesis focused on convex methods for signal decompositions and earned a WP Carey & Co. Inc. Prize for an outstanding doctoral dissertation. He is currently a postdoctoral scholar at Caltech, where his research explores the intersections of optimization, signal processing, statistics, and geometry.

Volkan Cevher (volkan.cevher@epfl.ch) received the B.S. (valedictorian) degree in electrical engineering in 1999 from Bilkent University in Ankara, Turkey, and the Ph.D. degree in electrical and computer engineering in 2005 from the Georgia Institute of Technology in Atlanta. He held research scientist positions at the University of Maryland, College Park, from 2006 to 2007 and at Rice University in Houston, Texas, from 2008 to 2009. Currently, he is an assistant professor at the Swiss Federal Institute of Technology Lausanne and a faculty fellow in the Electrical and Computer Engineering Department at Rice University. His research interests include signal processing theory, machine learning, graphical models, and information theory. He received a Best Paper Award at the Signal Processing with Adaptive Sparse Representations Workshop in 2009 and a European Research Council Starting Grant in 2011.

Quoc Tran Dinh (quoc.trandinh@epfl.ch) received the B.S. degree in applied mathematics and informatics and the M.S. degree in computer science, both from Vietnam National University, Hanoi, in 2001 and 2004, respectively, and the Ph.D. degree in electrical engineering from the Department of Electrical Engineering and Optimization in Engineering Center, KU Leuven, Belgium. He is currently a postdoctoral researcher with the Laboratory for Information and Inference Systems, Ecole Polytechnique Federale de Lausanne, Switzerland. His research interests include methods for convex optimization, sequential convex programming, parametric optimization, optimization in machine learning, and methods for variational inequalities and equilibrium problems.

Afsaneh Asaei (afsaneh.asaei@idiap.ch) received the B.S. degree from Amirkabir University of Technology and the M.S. (honors) degree from Sharif University of Technology, in electrical and computer engineering, respectively. She held a research engineer position at Iran Telecommunication Research Center (ITRC) from 2002 to 2008. She then joined Idiap Research Institute in Martigny, Switzerland, and was a Marie Curie fellow on speech communication with adaptive learning training network. She received the Ph.D. degree in 2013 from Ecole Polytechnique Federale de Lausanne. Her thesis focused on model-based sparsity for reverberant speech processing, and its key idea was awarded the IEEE Spoken Language Processing Grant. Currently, she is a research scientist at Idiap Research Institute. Her research interests lie in the areas of signal processing, machine learning, statistics, acoustics, auditory scene analysis and cognition, and sparse signal recovery and acquisition.

Luca Baldassarre (luca.baldassarre@epfl.ch) received the M.S. degree in physics in 2006 and the Ph.D. degree in machine learning in 2010 from the University of Genoa, Italy. He then joined the Computer Science Department of University College London, United Kingdom, to work with Prof. Massimiliano Pontil on structured sparsity models for machine learning and convex optimization. Currently he is with the Laboratory for Information and Inference Systems at the Ecole Polytechnique Federale de Lausanne, Switzerland. His research interests include model-based machine learning and compressive sensing and large-scale optimization.

REFERENCES

[1] J.-L. Starck, F. Murtagh, and J. M. Fadili, *Sparse Image and Signal Processing*. Cambridge, U.K.: Cambridge Univ. Press, 2010.

[2] V. Chandrasekaran, S. Sanghavi, P. A. Parrilo, and A. S. Willsky, "Rank-sparsity incoherence for matrix decomposition," *SIAM J. Optim.*, vol. 21, no. 2, pp. 572–596, 2011.

[3] E. J. Candès, X. Li, Y. Ma, and J. Wright. (2011, May). Robust principal component analysis? *J. Assoc. Comput. Mach.* [Online]. *58(3)*, pp. 1–37. Available: http://arxiv.org/pdf/0912.3599

[4] X. Liang, X. Ren, Z. Zhang, and Y. Ma, "Repairing sparse low-rank texture," in *Computer Vision–ECCV 2012*. New York: Springer, 2012, pp. 482–495.

[5] Y. Chen, A. Jalali, S. Sanghavi, and C. Caramanis, "Low-rank matrix recovery from errors and erasures," *IEEE Trans. Inform. Theory*, vol. 59, no. 7, pp. 4324–4337, 2013.

[6] B. N. Bhaskar, G. Tang, and B. Recht. (2013). Atomic norm denoising with applications to line spectral estimation, preprint. [Online]. Available: http://arxiv.org/abs/1204.0562

[7] R. G. Baraniuk, V. Cevher, and M. B. Wakin, "Low-dimensional models for dimensionality reduction and signal recovery: A geometric perspective," *Proc. IEEE*, vol. 98, no. 6, pp. 959–971, 2010.

[8] E. J. Candès and M. B. Wakin, "An introduction to compressive sampling," *IEEE Signal Process. Mag.*, vol. 25, no. 2, pp. 21–30, 2008.

[9] V. Chandrasekaran, B. Recht, P. A. Parrilo, and A. S. Willsky, "The convex geometry of linear inverse problems," *Found. Comput. Math.*, vol. 12, no. 6, pp. 805–849, 2012.

[10] A. Ahmed, B. Recht, and J. Romberg, "Blind deconvolution using convex programming," preprint. [Online]. Available: http://arxiv.org/abs/1211.5608

[11] J. Saunderson, V. Chandrasekaran, P. A. Parrilo, and A. S. Willsky, "Diagonal and low-rank matrix decompositions, correlation matrices, and ellipsoid fitting," *SIAM J. Matrix Anal. Appl.*, vol. 33, no. 4, pp. 1395–1416, 2012.

[12] V. Bittorf, C. Ré, B. Recht, and J. A. Tropp, "Factoring nonnegative matrices with linear programs," in *Proc. Advances in Neural Information Processing Systems 25 (NIPS)*, Dec. 2012, pp. 1223–1231.

[13] M. B. McCoy and J. A. Tropp, "Sharp recovery bounds for convex demixing, with applications," *J. Found. Comput. Math*., to be published.

[14] D. L. Donoho and X. Huo, "Uncertainty principles and ideal atomic decomposition," *IEEE Trans. Inform. Theory*, vol. 47, no. 7, pp. 2845–2862, Aug. 2001.

[15] S. S. Chen, D. L. Donoho, and M. A. Saunders, "Atomic decomposition by basis pursuit," *SIAM J. Sci. Comput.*, vol. 20, no. 1, pp. 33–61, 1998.

[16] F. Bach, "Structured sparsity-inducing norms through submodular functions," in *Proc. Advances in Neural Information Processing Systems*, 2010, pp. 118–126.

[17] R. Foygel and L. Mackey. (May, 2013). Corrupted sensing: Novel guarantees for separating structured signals, preprint. [Online]. Available: http://arxiv.org/abs/1305.2524

[18] Y. Peng, A. Ganesh, J. Wright, W. Xu, and Y. Ma, "RASL: Robust alignment by sparse and low-rank decomposition for linearly correlated images," *IEEE Trans. Pattern Anal.*, vol. 34, no. 11, pp. 2233–2246, 2012.

[19] Y. Chen, A. Jalali, S. Sanghavi, and C. Caramanis, "Clustering partially observed graphs via convex optimization," in *Proc. Int. Symp. Information Theory (ISIT)*, 2011, pp. 1001–1008.

[20] D. Amelunxen, M. Lotz, M. B. McCoy, and J. A. Tropp, "Living on the edge: A geometric theory of phase transitions in convex optimization," preprint. [Online]. Available http://arxiv.org/abs/1303.6672

[21] D. L. Donoho and J. Tanner, "Precise undersampling theorems," *Proc. IEEE*, vol. 98, no. 6, pp. 913–924, June 2010.

[22] M. Bayati, M. Lelarge, and A. Montanari, "Universality in polytope phase transitions and message passing algorithms," preprint. [Online]. Available: http://arxiv.org/abs/1207.7321

[23] M. B. McCoy and J. A. Tropp, "The achievable performance of convex demixing," preprint. [Online]. Available: http://arxiv.org/abs/1309.7478

[24] Y. Nesterov, *Introductory Lectures on Convex Optimization: A Basic Course* (Applied Optimization, vol. 87). Norwell, MA: Kluwer, 2004.

[25] P. L. Combettes and V. R. Wajs, "Signal recovery by proximal forward-backward splitting," *Multiscale Model. Simul.*, vol. 4, no. 4, pp. 1168–1200, 2005.

[26] C. Chen, B. S. He, Y. Ye, and X. Yuan, "The direct extension of admm for multi-block convex minimization problems is not necessarily convergent," *Optim. Online*, 2013. [Online]. Available: http://www.optimization-online.org/DB_FILE/2013/09/4059.pdf

[27] G. Chen and M. Teboulle, "A proximal-based decomposition method for convex minimization problems," *Math. Program.*, vol. 64, nos. 1–3, pp. 81–101, Mar. 1994.

[28] I. Necoara and J. Suykens, "Applications of a smoothing technique to decomposition in convex optimization," *IEEE Trans. Automatic Control*, vol. 53, no. 11, pp. 2674–2679, 2008.

[29] H. L. V. Trees, *Optimum Array Processing: Part IV of Detection, Estimation, and Modulation Theory*. Hoboken, NJ: Wiley, 2002.

[30] Q. T. Dinh, A. Kyrillidis, and V. Cevher, "Composite self-concordant minimization," Lab. Inform. Infer. Sys. (LIONS), EPFL, Switzerland, Tech. Rep. 188126, Aug. 2013.

[31] B. Gözcü, A. Asaei, and V. Cevher, "Manifold sparse beamforming," in *Proc. Int. Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, 2013, pp. 113–116.