A Tutorial on Sparse Signal Acquisition and Recovery with Graphical Models

Volkan Cevher, Piotr Indyk, Lawrence Carin, Richard G. Baraniuk

I. Introduction

Many applications in digital signal processing, machine learning, and communications feature a linear regression problem in which unknown data points, hidden variables or codewords are projected into a lower dimensional space via

$$\boldsymbol{y} = \boldsymbol{\Phi}\boldsymbol{x} + \boldsymbol{n}. \tag{1}$$

In the signal processing context, we refer to $\boldsymbol{x} \in \mathbb{R}^N$ as the signal, $\boldsymbol{y} \in \mathbb{R}^M$ as measurements with $M < N$, $\boldsymbol{\Phi} \in \mathbb{R}^{M \times N}$ as the measurement matrix, and $\boldsymbol{n} \in \mathbb{R}^M$ as the noise. The measurement matrix $\boldsymbol{\Phi}$ is a matrix with random entries in data streaming, an overcomplete dictionary of features in sparse Bayesian learning, or a code matrix in communications [1–3].

Extracting $\boldsymbol{x}$ from $\boldsymbol{y}$ in (1) is ill-posed in general since $M < N$ and the measurement matrix $\boldsymbol{\Phi}$ hence has a nontrivial null space; given any vector $\boldsymbol{v}$ in this null space, $\boldsymbol{x} + \boldsymbol{v}$ defines a solution that produces the same observations $\boldsymbol{y}$. Additional information is therefore necessary to distinguish the true $\boldsymbol{x}$ among the infinitely many possible solutions [1, 2, 4, 5]. It is now well-known that sparse representations can provide crucial prior information in this dimensionality reduction; we therefore also refer to the problem of determining $\boldsymbol{x}$ in this particular setting as the *sparse signal recovery*. A signal $\boldsymbol{x}$ has a sparse representation $\boldsymbol{x} = \boldsymbol{\Psi}\boldsymbol{\alpha}$ in a basis $\boldsymbol{\Psi} \in \mathbb{R}^{N \times N}$ when $K \ll N$ coefficients of $\boldsymbol{\alpha}$ can exactly represent or well-approximate the signal $\boldsymbol{x}$. Inspired by communications, coding and information theory problems, we often refer to the application of $\boldsymbol{\Phi}$ on $\boldsymbol{x}$ as *encoding* of $\boldsymbol{x}$; the sparse signal recovery problem is then concerned with the *decoding* of $\boldsymbol{x}$ from $\boldsymbol{y}$ in the presence of noise. In the sequel, we assume the canonical sparsity basis, $\boldsymbol{\Psi} = \boldsymbol{I}$ without loss of generality.

The sparse signal recovery problem has been the subject of extensive research over the last few decades in several different research communities, including applied mathematics, statistics, and theoretical computer science [1–3, 6]. The goal of this research has been to obtain higher compression rates; stable recovery schemes; low encoding, update and decoding times; analytical recovery bounds; and resilience to noise. The momentum behind this research is well-justified: underdetermined linear regression problems in tandem with sparse representations underlie the paradigm of signal compression and denoising in signal processing, the tractability and generalization of learning algorithms in machine learning, the stable embedding and decoding properties of codes in information theory, the effectiveness of data streaming algorithms in theoretical computer science, and neuronal information processing and interactions in computational neuroscience.

An application *du jour* of the sparse signal recovery problem is *compressive sensing* (CS), which integrates the sparse representations with two other key aspects of the linear dimensionality reduction: *information preserving projections* and *tractable recovery algorithms* [1, 2, 4–6]. In CS, sparse signals are represented by a union of the $\binom{N}{K}$, K-dimensional subspaces, denoted as $\boldsymbol{x} \in \Sigma_K$. We call the set of indices corresponding to the nonzero entries the *support* of $\boldsymbol{x}$. While the matrix $\boldsymbol{\Phi}$ is rank deficient, it can be shown to preserve the information in sparse signals if it satisfies the so-called *restricted isometry property* (RIP). Intriguingly, a large class of random matrices have the RIP with high probability. Today's state-of-the-art CS systems can robustly and provably recover K-sparse signals from just $M = \mathcal{O}(K \log(N/K))$ noisy measurements using sparsity-seeking, polynomial-time optimization solvers or greedy algorithms. When $\boldsymbol{x}$ is *compressible*, in that it can be closely approximated as K-sparse, then from the measurements $\boldsymbol{y}$, CS can recover a close approximation to $\boldsymbol{x}$. In this manner we can achieve sub-Nyquist signal acquisition, which requires uniform sampling rates at least two times faster than the signal's Fourier bandwidth to preserve information.

While such measurement rates based on sparsity are impressive and have the potential to impact a broad set of streaming, coding, and learning problems, sparsity is merely a first-order description of signal structure; in many applications we have considerably more a priori information that previous approaches to CS fail to exploit. In particular, modern signal, image, and video coders directly exploit the fact that even compressible signal coefficients often sport a strong additional structure in the support of the significant coefficients. For instance, the image compression standard JPEG2000 does not only use the fact that most of the wavelet coefficients of a natural image are small. Rather, it also exploits the fact that the values and locations of the large coefficients have a particular structure that is characteristic of natural images. Coding this structure using an appropriate model enables JPEG2000 and other similar algorithms to compress images close to the maximum amount possible, and significantly better than a naive coder that just assigns bits to each large coefficient independently [1].

By exploiting a priori information on coefficient structure in addition to signal sparsity, we can make CS better, stronger, and faster. The particular approach we will focus in this tutorial is based on *graphical models* (GM) [3, 7–10]. As we will discover, GMs are not only useful for representing the prior information on $\boldsymbol{x}$, but also lay the foundations for new kinds of measurement systems. GMs enable us to reduce, in some cases significantly, the number of measurements M required to stably recover a signal by permitting only certain configurations of the large and zero/small coefficients via probabilistic dependencies over graphs. During signal recovery, GMs therefore enable us to better differentiate true signal information from recovery artifacts, which leads to a more robust recovery. Moreover, GMs provide powerful tools for designing measurement matrices that result in highly efficient recovery algorithms, with running times that are close to linear in the signal size. GMs achieve all of this in a Bayesian formalism, drawing from a large set of associated tools and recovery algorithms.

We review in this tutorial a broad set of models, tools, and algorithms within the graphical model formalism for sparse signal recovery. A background of graphical models and CS theory in Section II outlines the foundational concepts that the later sections build upon. Section III then explains Bayesian priors for signal sparsity and then extends this framework for concisely representing the dependencies among the sparse coefficients on common GM structures with corresponding recovery algorithms. Section IV indicates how to exploit GMs for designing sparse measurement systems that significantly reduce computational requirements while still providing provable guarantees in sparse recovery. Section V concludes with research challenges and open problems.

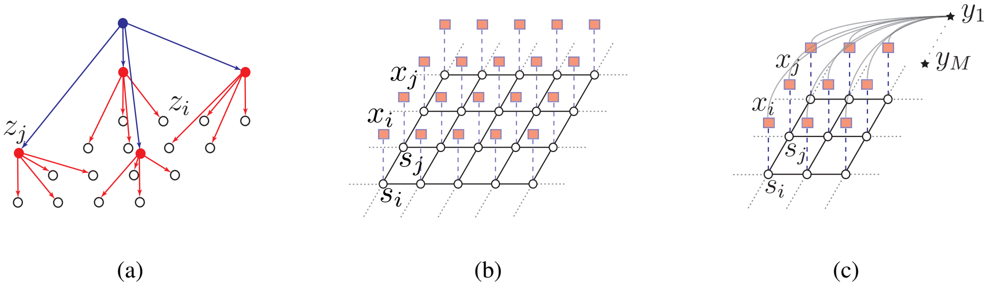
II. Background

A. Graphical models

We take a probabilistic, Bayesian approach to sparse signal recovery with GMs. We assume that $\boldsymbol{x}$ is a realization from a random process with a probability distribution function (pdf) $f(\boldsymbol{x})$, which we call as the *prior*. The information in $\boldsymbol{x}$ is then transferred to the measurements $\boldsymbol{y}$ through a conditional pdf $f(\boldsymbol{y}|\boldsymbol{x})$. For instance, when the noise $\boldsymbol{n}$ is independent and identically distributed (iid) Gaussian with zero mean and variance σ^2, i.e., $n_i \sim \mathcal{N}(n; 0, \sigma^2)$ for $i = 1, \ldots, M$, we find that the conditional pdf is yet another Gaussian distribution, given by $f(\boldsymbol{y}|\boldsymbol{x}) = \mathcal{N}\left(\boldsymbol{y}; \boldsymbol{\Phi}\boldsymbol{x}, \sigma^2\boldsymbol{I}_{M\times M}\right)$, where $\boldsymbol{I}_{M\times M}$ is the $M \times M$ identity matrix. To determine $\boldsymbol{x}$, we exploit *Bayes' rule* to form the *posterior* pdf via $f(\boldsymbol{x}|\boldsymbol{y}) = f(\boldsymbol{y}|\boldsymbol{x})f(\boldsymbol{x})/f(\boldsymbol{y})$. Then, we can compute various point estimates of $\boldsymbol{x}$, denoted as $\widehat{\boldsymbol{x}}$, such as the maximum a posteriori (MAP) estimate $\widehat{\boldsymbol{x}} = \arg\max_{\boldsymbol{x}'} f(\boldsymbol{x}'|\boldsymbol{y})$, the mean estimate $\widehat{\boldsymbol{x}} = \int_{\boldsymbol{X}} \boldsymbol{x}' df(\boldsymbol{x}'|\boldsymbol{y})$, etc.

1) *Graphical representations of pdfs:* Graphical models provide a convenient geometrical representation for describing joint probability distributions of multiple variables [3, 7–11]. They have been applied successfully in a wide variety of statistical problems, in fields as diverse as image processing, coding theory, pattern recognition, and artificial intelligence.

Here, we define some notation for GMs. Let $G = (V, E)$ be a directed/undirected graph formed by a collection of vertices V and edges E. Each vertex $v \in V$ represents a random variable z_v (either discrete



(a) (b) (c)

Fig. 1. *Example graphical models. (a) Quadtree model. (b) Ising model. (c) Ising model with CS measurements.*

or continuous valued). An *undirected edge* $e \in E$ indicates dependence between the corresponding pair of random variables; that is, the conditional density of z_v given $z_1, \dots, z_{v-1}, z_{v+1}, \dots$ is a function of only the neighbors connected to v. A *directed edge* indicates that the conditional density of z_v is a function of only those variables with edges directed *towards* v. Denote the random variables for any subset S of the vertices V as $\boldsymbol{z}_S \triangleq \{z_v | v \in S\}$. A *clique* C is a fully connected subset of vertices, where there exists an edge $e \in C$ connecting every pair of vertices in the subgraph.

A directed acyclic graph (DAG) is a directed graph without *directed* loops (so that for any v, there are no directed paths that begin and end with v). If we denote the set of parents of node v as $\rho(v)$, then it is easy to show that the pdf of a DAG can be factorized as

$$p(\boldsymbol{z}) = \prod_{v \in V} p(z_v | \boldsymbol{z}_{\rho(v)}). \tag{2}$$

An example is shown in Figure 1(a); this can be used to model a quadtree decomposition of radar images [9] or a multiscale wavelet decomposition of natural images [12].

In contrast, Figure 1(b) depicts an undirected graph with loops to represent the Ising model, where the variables $\boldsymbol{s} \subset \boldsymbol{z}$ represent the latent connections between the signal coefficients $\boldsymbol{x} \subset \boldsymbol{z}$. Similarly, Figure 1(c) illustrates an extension of the Ising model to the CS measurement setting, which we will revisit in Section III-B. By defining an appropriate, non-negative *compatibility function* $\chi_C(\boldsymbol{z}_C)$ for each clique $C \subset G$, the Hammersley-Clifford theorem [7] enables the pdf of the graph to be written as

$$p(\boldsymbol{z}) = \frac{1}{Z} \prod_{C} \chi_C(\boldsymbol{z}_C), \tag{3}$$

where Z is the *partition function* that ensures that $p(\boldsymbol{z})$ is properly normalized. The product in (3) is taken over all cliques C in the graph G.

2) *Inference mechanisms:* We refer to the process of estimating local marginal distributions or other summary statistics of $\boldsymbol{z}$, such as the most probable configuration of $f(\boldsymbol{x}|\boldsymbol{y})$, as Bayesian inference.

Inference algorithms on graphical models typically exploit the factorization properties (2) and (3) of probability distributions. By manipulating the intermediate factors, it is often possible to compute the likelihood of a particular z value in an efficient, distributed manner. This strategy is exploited by the sum-product algorithm (also known as belief propagation) and max-product (also known as min-sum) for tree-structured DAGs with rigorous estimation guarantees when the random variables live in a discrete probability space [7]. These algorithms iteratively pass statistical information, denoted as *messages*, among neighboring vertices and converge in finite number of steps. Such guarantees, unfortunately, do not extend to arbitrary DAGs; inference in DAGs is typically NP-hard [11].

The sum-product and max-product algorithms are routinely applied to graphical models with cycles, leading to *loopy belief propagation* methods, even if their convergence and correctness guarantees for DAGs no longer hold in general [3, 11, 13]. Although there are certain local optimality guarantees associated with the fixed points of loopy belief propagation, there are a number of natural inference problems arising in various applications in which loopy belief propagation either fails to converge, or provides poor results. Loopy belief propagation is therefore an *approximate inference* method. Surprisingly, state-of-the-art algorithms for decoding certain kinds of error-correcting codes are equivalent to loopy belief propagation. We revisit this topic in Section IV to discover a crucial role sparse representations play in providing theoretical guarantees for such algorithms.

Approximate inference is also necessary in cases where the random variables are drawn from a continuous space, since corresponding marginal integrals needed for implementing Bayes' rule cannot be analytically performed. Monte Carlo Markov chain sampling methods, such as importance sampling, Metropolis-Hastings and Gibbs sampling, provide computational means of doing approximate inference. The idea is to represent the pdf by a discrete set of samples, which is carefully weighted by some evidence likelihood so that the inference is consistent, with guarantees typically improving with larger sample sizes. See [3] for further background.

Yet another approximate inference method is based on the calculus of variations, also known as *variational Bayes* (VB) [3, 8]. The quintessential VB example is the *mean-field approximation*, which exploits the law of large numbers to approximate large sums of random variables by their means. In particular, the mean-field approximation essentially decouples all the vertices in the graph, and then introduces a parameter, called a variational parameter, for each vertex. It then iteratively updates these variational parameters so as to minimize the cross-entropy between the approximate and true probability distributions. Updating the variational parameters then facilitates inference in an efficient and principled manner, often also providing bounds on the marginal likelihood to be calculated. Another special case of the VB approach is the well-known expectation-maximization (EM) algorithm for MAP and maximum likelihood estimation, which has been extremely successful in a number of applications [3, 8, 14].

B. Compressive sensing (CS)

1) *Sparse signal representations:* No linear nonadaptive dimensionality reducing $\mathbf{\Phi}$ can preserve all of the information in all signals. Hence, researchers restrict the application of $\mathbf{\Phi}$ to not arbitrary signals $\boldsymbol{x} \in \mathbb{R}^N$ but rather to some subset of $\mathbb{R}^N$, and in particular the set of sparse and compressible signals. A sparsity/compressibility prior is then exploited as a tie-breaker to distinguish the true signal among the infinitely many possible solutions of (1).

While sparse signals live in Σ_K, the coefficients of a compressible signal $\boldsymbol{x}$, when sorted in order of decreasing magnitude, decay according to the following power law:

$$\left|\boldsymbol{x}_{\mathcal{I}(i)}\right| \leq R\, i^{-1/r}, \quad i = 1, \ldots, N, \tag{4}$$

where $\mathcal{I}$ indexes the sorted coefficients, and $R > 0$ and $r > 0$ are constants. In the CS setting, we call a signal compressible when $r \leq 1$. Thanks to the power-law decay of their coefficients, compressible signals are well-approximated by K-sparse signals in an appropriate norm. For instance, for all $r < 1$ and $0 < \epsilon \ll 1$, $\|\boldsymbol{x} - \boldsymbol{x}_K\|_1 \leq \epsilon \|\boldsymbol{x}\|_1$ holds independent of N for any $K \geq \lceil (r/\epsilon)^{\frac{r}{1-r}} \rceil$, where $\|\boldsymbol{x}\|_p = \left(\sum_{i=1}^N |x_i|^p\right)^{1/p}$ is the ℓ_p-norm, $\boldsymbol{x}_K = \arg\min_{\|\boldsymbol{x}'\|_0 \leq K} \|\boldsymbol{x} - \boldsymbol{x}'\|_p$ is the best K-sparse approximation of $\boldsymbol{x}$ ($p \geq 1$), and $\|\boldsymbol{x}\|_0$ is a pseudo-norm that counts the number of nonzeros of $\boldsymbol{x}$ [15].

2) *Information preserving projections:* The sparsity or compressibility of $\boldsymbol{x}$ is not sufficient alone for distinguishing $\boldsymbol{x}$ among all possible solutions to (1). The projection matrix $\mathbf{\Phi}$ must also work in tandem with the signal priors so that recovery algorithms can correctly identify the true signal.

The restricted isometry property (RIP) assumption on $\mathbf{\Phi}$ achieves this by requiring $\mathbf{\Phi}$ to approximately preserve the distances between all signal pairs in the sparse signal set [5]. More formally, an $M \times N$ matrix $\mathbf{\Phi}$ has the *K-restricted isometry property* (K-RIP) with constant $\epsilon_K < 1$ if, for all $\boldsymbol{x} \in \Sigma_K$,

$$(1 - \epsilon_K)\|\boldsymbol{x}\|_2^2 \leq \|\mathbf{\Phi}\boldsymbol{x}\|_2^2 \leq (1 + \epsilon_K)\|\boldsymbol{x}\|_2^2. \tag{5}$$

An alternative property is the *restricted strong convexity* (RSC) assumption, which is motivated by convex optimization arguments [16]. In general, the RSC assumption has an explicit dependence on the recovery algorithm's objective function. For instance, if the recovery algorithm's objective is to minimize the measurement error (e.g., $\|\boldsymbol{y} - \mathbf{\Phi}\boldsymbol{x}\|_2^2$), RSC requires $\|\mathbf{\Phi}^t\mathbf{\Phi}\boldsymbol{x}\|_2$ to be strictly positive for all $\boldsymbol{x} \in \Sigma_K$. In different contexts, other conditions on $\mathbf{\Phi}$ are also employed with varying levels of restrictions, such as null space property, spark, unique representation property, etc [6].

While checking whether a measurement matrix $\boldsymbol{\Phi}$ satisfies the K-RIP, RSC, etc. has running time exponential in K and N, random matrices with iid subgaussian entries work with high probability provided $M = \mathcal{O}(K \log(N/K))$. Random matrices also have a so-called universality property in that, for any choice of orthonormal basis matrix $\boldsymbol{\Psi}$, $\boldsymbol{\Phi\Psi}$ also has the K-RIP with high probability. This is useful when the signal is sparse in some basis $\boldsymbol{\Psi}$ other than the canonical basis.

3) *Tractable recovery algorithms:* To recover the signal $\boldsymbol{x}$ from $\boldsymbol{y}$ in (1), we exploit our a priori knowledge of its sparsity or compressibility. For example, to recover strictly sparse signals when there is no measurement noise, we can seek the sparsest $\boldsymbol{x}$ that agrees with the measurements $\boldsymbol{y}$. While this optimization can recover a K-sparse signal from just $M = 2K$ compressive measurements, it is not only a combinatorial, NP-hard problem, but also is not stable in the presence of noise [1].

Tractable recovery algorithms rely on conditions such as the RIP and RSC for stability and correctness and therefore require at least $M = \mathcal{O}(K \log(N/K))$ measurements. They can be grouped into two main camps: *convex optimization* and *greedy approximation*. The first camp relies on ℓ_1-norm minimization as a convex relaxation of seeking sparse solutions:

$$\widehat{\boldsymbol{x}} = \arg\min_{\boldsymbol{x}'} \|\boldsymbol{x}'\|_1 \text{ s.t. } \boldsymbol{y} = \boldsymbol{\Phi x}'. \tag{6}$$

This optimization problem is known as *basis pursuit* and corresponds to a linear program that can be solved in polynomial time [1, 4, 5]. Adaptations to deal with additive noise in (1) have also been proposed; examples include basis pursuit with denoising (BPDN), and the least absolute shrinkage and selection operator (LASSO). The second camp finds the sparsest $\boldsymbol{x}$ agreeing with the measurements $\boldsymbol{y}$ through an iterative, greedy search over the coefficients. Example algorithms include iterative hard thresholding (IHT), compressive sampling matching pursuit (CoSaMP), and Subspace Pursuit (SP) [1].

Currently, convex optimization obtains the best recovery performance in theory, while its greedy counterparts, such as IHT, CoSaMP, and SP, offer desirable computational trade-offs, e.g., $\mathcal{O}\left(N \log^2 N\right)$ vs. $\mathcal{O}\left(M^2 N^{1.5}\right)$ (interior point methods) [1]. Interestingly, algorithms in both camps have similar theoretical recovery guarantees from $M = \mathcal{O}(K \log(N/K))$ when the measurement matrix $\boldsymbol{\Phi}$ satisfies K-RIP with an algorithm dependent ϵ_{2K} (e.g., $\epsilon_{2K} < \sqrt{2} - 1$ for basis pursuit [5]):

$$\|\boldsymbol{x} - \widehat{\boldsymbol{x}}\|_2 \leq C_1 K^{-1/2} \|\boldsymbol{x} - \boldsymbol{x}_K\|_1 + C_2 \|\boldsymbol{n}\|_2, \tag{7}$$

which we refer to as an ℓ_2/ℓ_1 guarantee, where the subscripts are matched to the norms adjacent to the inequality. In (7), $\widehat{\boldsymbol{x}}$ is the algorithm output, C_1 and C_2 are algorithm-dependent constants, and $\boldsymbol{x}_K$ is the signal-dependent best K-sparse approximation. The term $\|\boldsymbol{x} - \boldsymbol{x}_K\|_1$ is known as the irrecoverable energy.

When the signals are compressible, we can never recover them fully under dimensionality reduction; we can only approximately recover them — and only when $r \leq 1$ in (4).

III. Graphical models for structured sparsity

While CS has the potential to revolutionize data acquisition, encoding, and processing in a number of applications, much work remains before it is ready for real-world deployment. In particular, for CS to truly live up its name, it is crucial that the theory and practice leverage concepts from state-of-the-art transform compression algorithms. In virtually all such algorithms, the key ingredient is a signal model for not just the coefficient sparsity but also the coefficient structure. As we saw in Section II, graphical models provide natural framework for capturing such dependencies among sparse signal coefficients. Hence, we review below some of the graphical models that are relevant for structured sparsity and the algorithms that result for sparse signal recovery.

A. Sparsity priors

Our first departure from the standard, deterministic CS approach is to adopt a more general, Bayesian viewpoint as several others have pursued [1, 17, 18]. A probabilistic sparsity prior enforces sparsity or compressibility in each coefficient by heavily weighting zero or small-magnitude values in the probability distribution, while allowing a large range of values with low probabilities to obtain a small number of large coefficients. While this is expected in principle, we also require a generative aspect for such priors in order to build up structured sparsity models; that is, their statistical realizations result in sparse or compressible signals. This, in turn, is crucial for statistical consistency in high-dimensional scalings of the sparse signal recovery problem.

Two-state mixture models are the canonical iid models for generating strictly sparse signals. Among mixture models, the *spike-and-slab* prior stipulates a density with a spike at zero surrounded symmetrically by a uniform distribution with specified boundaries. The mixture probability of the spike at zero approximately determines the percentage of zero coefficients of the signal, and the uniform distribution establishes the distribution of the signal coefficients on the signal support.

To model compressible signals, we can use *compressible priors*, whose statistical realizations exhibit the power-law decay in (4) [15]. For algorithmic recovery guarantees for CS, we must have $r \leq 1$ for such priors (c.f. (4)). For instance, Table III-A demonstrates that N-sample iid realizations of generalized Pareto, Student's t, Frechet, and log-logistics distributions (each parameterized by a shape parameter $q > 0$ and a scale parameter $\lambda > 0$) are compressible with parameter $r = q$. There also exist non-iid compressible priors; multivariate Lomax distribution provides an elementary example whose pdf is given

TABLE I

EXAMPLE DISTRIBUTIONS AND THE COMPRESSIBILITY PARAMETERS OF THEIR IID REALIZATIONS

Distribution	pdf	R	r
Generalized Pareto	$\frac{q}{2\lambda}\left(1+\frac{\lvert x\rvert}{\lambda}\right)^{-(q+1)}$	$\lambda N^{1/q}$	q
Student's t	$\frac{\Gamma((q+1)/2)}{\sqrt{2\pi}\lambda\Gamma(q/2)}\left(1+\frac{x^2}{\lambda^2}\right)^{-(q+1)/2}$	$\left[\frac{2\Gamma((q+1)/2)}{\sqrt{\pi}q\Gamma(q/2)}\right]^{1/q}\lambda N^{1/q}$	q
Fréchet	$(q/\lambda)\,(x/\lambda)^{-(q+1)}\,\mathrm{e}^{-(x/\lambda)^{-q}}$	$\lambda N^{1/q}$	q
Log-Logistic	$\frac{(q/\lambda)(x/\lambda)^{q-1}}{[1+(x/\lambda)^q]^2}$	$\lambda N^{1/q}$	q
Laplacian	$\frac{1}{2\lambda}\mathrm{e}^{-\lvert x\rvert/\lambda}$	$\lambda \log N$	$\log N$

by $\mathrm{MLD}(\boldsymbol{x};q,\lambda) \propto \left(1+\sum_{i=1}^{N}\lambda^{-1}\lvert x_i\rvert\right)^{-q-N}$ [15]. The compressibility parameter MLD is simply $r=1$, irrespective of its shape parameter.

To illustrate how we can exploit probabilistic priors in sparse signal recovery, we focus on two compressible priors on $\boldsymbol{x}$: MLD and Student's t. While MLD is relatively unknown, the Student's t distribution has enjoyed tremendous attention in many fundamental problems, such as statistical modeling of natural images, relevance vector machines (RVM) and automatic relevance determination, and dictionary learning. We consider the case when there is no noise; the observations are then given by $\boldsymbol{y}=\boldsymbol{\Phi x}$ with $M=\mathcal{O}(K\log(N/K))$, which has infinitely many solutions for $\boldsymbol{x}$, as discussed in Section II. We know that such projections preserve the information of the largest K-coefficients of signal realizations. We will assume that the parameters of MLD and the Student's t priors are matched (i.e., the shape parameter of Student's t is 1) so that the statistical realizations of both priors have the same decay profile.

1) *MLD and basis pursuit:* We first exploit the MLD likelihood function for sparse signal recovery. For instance, when we ask for the solution that maximizes the MLD likelihood given $\boldsymbol{y}$ for $\lambda_i=\lambda$, it is easy to see that we obtain the basis pursuit algorithm formulation in (6). In contrast, the conventional probabilistic motivation for the basis pursuit algorithm assumes that $\boldsymbol{x}$ is iid Laplacian, e.g., $f(\boldsymbol{x}) \propto \exp\left(-\|\boldsymbol{x}\|_1/\lambda\right)$ for some $\lambda \in \mathbb{R}^+$. We then face the optimization problem (6), when we seek the most likely signal from the Laplacian prior given $\boldsymbol{y}$. Unfortunately, iid Laplacian realizations cannot be approximated as sparse since their compressibility parameter is $r=\log N$, which is greater than 1 in general (c.f. Table III-A); it can be shown that $\|\boldsymbol{x}-\boldsymbol{x}_K\|_1 \leq \epsilon\|\boldsymbol{x}\|_1$ requires $K \geq (1-\sqrt{\epsilon})N$ with high probability [15]. Hence, while the ℓ_1 minimization correctly recovers the sparse vectors from $M=\mathcal{O}(K\log(N/K))$, it seems that it does not correspond to an iid Laplacian prior on the sparse signal coefficients in a consistent Bayesian framework and may correspond to the MLD prior as this example demonstrates.

2) *Student's t and iterative reweighted least squares:* We now exploit the Student's t likelihood function

for sparse signal recovery, similar to MLD likelihood function above:

$$\widehat{\boldsymbol{x}} = \max_{\boldsymbol{x}'} f(\boldsymbol{x}) = \min_{\boldsymbol{x}'} \sum_i \log\left(1 + \lambda^{-2} x_i'^2\right), \text{ s.t. } \boldsymbol{y} = \boldsymbol{\Phi}\boldsymbol{x}'. \tag{8}$$

Unfortunately, (8) is a non-convex problem. However, we can circumvent the non-convexity in (8) using a simple variational Bayes idea where we iteratively obtain a tractable upperbound on the log-term in (8) using the following inequality: $\forall u, v \in (0, \infty), \;\; \log u \leq \log v + u/v - 1$. After some straightforward calculus, we obtain the iterative algorithm below, indexed by k, where $\widehat{\boldsymbol{x}}_{\{k\}}$ is the k-th iteration estimate ($\widehat{\boldsymbol{x}}_{\{0\}} = 0$):

$$\widehat{\boldsymbol{x}}_{\{k\}} = \min_{\boldsymbol{x}'} \sum_i w_{i,\{k\}} x_i'^2, \text{ s.t. } \boldsymbol{y} = \boldsymbol{\Phi}\boldsymbol{x}'; \text{ where } w_{i,\{k\}} = \left(\lambda^2 + x_{i,\{k\}}'^2\right)^{-1}. \tag{9}$$

The decoding scheme in (9) is well-known as the iterative reweighted least squares (IRLS), where each iteration has an analytical solution [19].

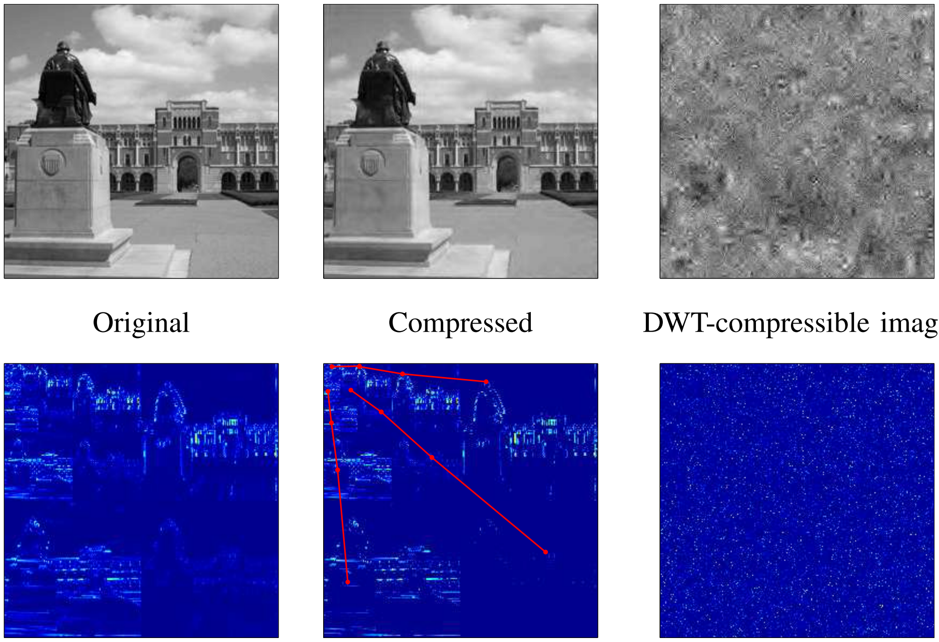
Both priors in the above algorithms are trying to approximate their signal realizations, which have the same decay profile, and yet result in radically different algorithms. Both schemes have provable recovery guarantees; however, they differ in terms of their computational costs: $\mathcal{O}(M^2N)$ (IRLS) vs. $\mathcal{O}(M^2N^{1.5})$ (BP).

B. Structured sparsity via GMs

Compressible priors provide a natural launching point for incorporating further structure among the sparse signal coefficients. Here, by structure, we specifically mean the probabilistic dependence and independence relationships among sparse signal coefficients as summarized by directed and undirected graphs. Such relationships are abundant in many diverse problems, such as speech recognition, computer vision, decoding of low-density parity-check codes, modeling of gene regulatory networks, gene finding and diagnosis of diseases. Graphical models also provide powerful tools for automatically learning these interactions from data. While learning GMs is an active research area, it is beyond the scope of this paper.

1) *Tree graphs:* Wavelet transforms sparsify piecewise smooth phenomena, including many natural and manmade signals and images. However, the significant discrete wavelet transform (DWT) coefficients do not occur in arbitrary positions for such signals. Instead they exhibit a characteristic signal-dependent structure, with the large and small wavelet coefficients clustering along the branches of the wavelet tree. We highlight the persistency of the parent child relationship for several large DWT coefficients on the wavelet image in Figure 2.

Hidden Markov trees (HMT) models and Gaussian scale mixtures (GSMs) on steerable pyramids succinctly and accurately captures this statistical structure [12, 20, 21]. The HMT models the probability



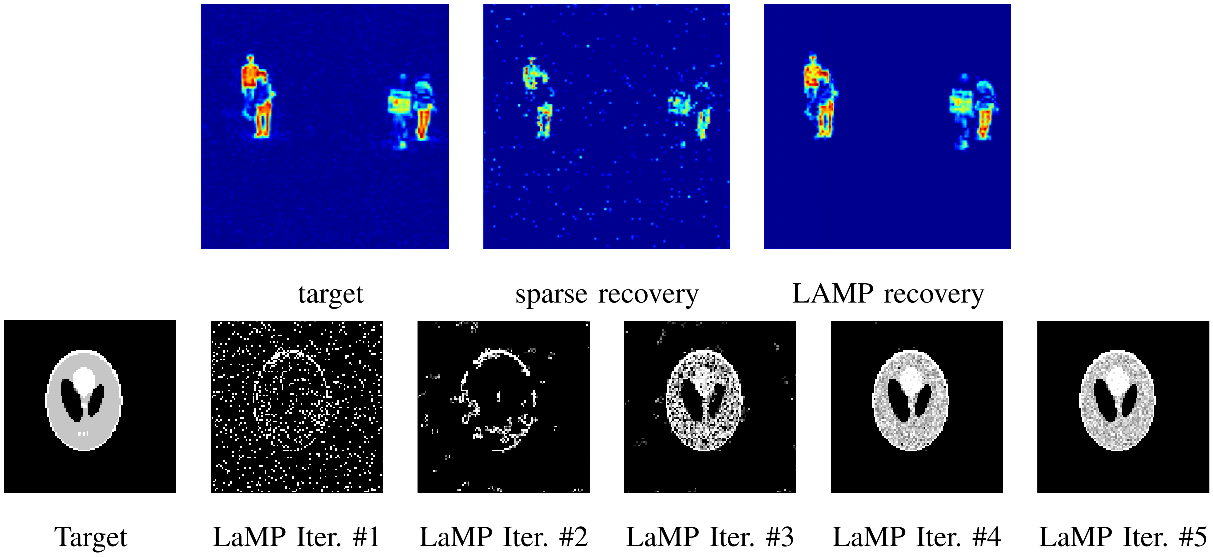
Original Compressed DWT-compressible image

DWT coefficients 10% of DWT coefficents Shuffled DWT coeff.

Fig. 2. *Most of the discrete wavelet transform (DWT) coefficients of natural images are small (blue in the bottom row corresponds to zero amplitude). However, the significant coefficients appear clustered on the wavelet trees [12, 20], as illustrated by the solid red lines. The dependence structure of the coefficients is crucial; by randomly shuffling the DWT coefficients of the original image, we obtain a reconstruction that does not even remotely resemble a natural image, let alone the original.*

density function of each wavelet coefficient as a mixture density with a hidden binary state that determines whether the coefficient is large or small. Wavelet coefficient persistence along the tree branches is captured by a tree-based Markov model that correlates the states of parent and children coefficients. GSMs exploit the Student's t prior or its variants. Both models have been successfully applied to improve the performance of denoising, classification, and segmentation algorithms for wavelet sparse signals. For an explanation of the EM algorithm applied to GSMs, see the original EM paper [14].

Another Bayesian approach leverages the clustering of the DWT coefficients using the spike-and-slab prior along with VB inference. The main idea is to construct a tree-structured, conjugate-exponential family sparse model for wavelet coefficients where the mixture probability of the spike at an individual sparse coefficient on the wavelet tree is controlled by the size of its parent on the tree. Then, the VB updates can be efficiently calculated with convergence guarantees. For instance, [17] demonstrates that the VB approach *simultaneously* improves both the recovery speed and performance over the state-of-the-art greedy and convex optimization based CS recovery approaches, discussed in Section II. Moreover, as


target sparse recovery LAMP recovery

Target LaMP Iter. #1 LaMP Iter. #2 LaMP Iter. #3 LaMP Iter. #4 LaMP Iter. #5

Fig. 3. (Top) *A real background subtracted image is shown where the foreground is sparse and clustered (blue corresponds to zero). We represent the pixels on the image by an Ising model with hidden binary support variables s that are connected to their adjacent pixels over a lattice to explain clustered behavior. Exploiting this structure in sparse signal recovery leads to improved performance over the state-of-the-art for the same number of measurements.* (Bottom) *Reconstructing the Shepp-Logan phantom under 10dB measurement SNR with lattice matching pursuit (LAMP).* $N = 100 \times 100 = 10^4$, $K = 1740$, $M = 2K = 3480$.

opposed to providing single point estimates, probabilistic predictions are made within the VB framework; for instance, the precision of each sparse coefficient can be inferred.

2) *Markov random fields:* Background-subtracted images play a fundamental role in making inferences about objects and activities in a scene in computer vision and, by nature, have structured spatial sparsity corresponding to the foreground innovations. That is, compared to the scale of the scene, the foreground innovations are usually not only sparse and but also clustered in a distinct way, corresponding to the silhouettes of moving humans and vehicles; see Figure 3. Such clustering behavior are also encountered in neuroscience problems that are involved with decoding of natural images in the primary visual cortex (V1) or understanding the statistical behavior of groups of neurons in the retina [1].

Markov random fields (MRF) can capture interactivity and produce collective effects among sparse signal coefficients; previous applications have included sonar image segmentation, scene geometry estimation using multiple sensors, and designing discriminative classifiers. To enforce clustered sparse behavior, we use a special MRF, called the Ising model, with latent support variables $s \in \mathbb{R}^N$ such that $s_i = -1$

when x_i is small and $s_i = 1$ when x_i is large; the support variables thus have the following pdf:

$$f(\boldsymbol{s}) \propto \exp \left\{ \sum_{(i,j) \in E} \lambda_{ij} s_i s_j + \sum_{i \in V} \lambda_i s_i \right\}, \tag{10}$$

where prior parameters consist of the edge interaction parameters λ_{ij} and the vertex bias parameters λ_i; c.f., Section II. Interestingly, when the underlying graph of support variables in the Ising model is tree-structured or chordal (i.e., each cycles in the graph of four or more nodes has a chord, which is an edge joining two nodes that are not adjacent in the cycle), the pdf in (10) can be equivalently represented by HMTs since, for instance, the set of distributions parameterized by directed trees is exactly the same as the set of distributions parameterized by undirected trees (c.f., Sect. 4.5.3 of [11]).

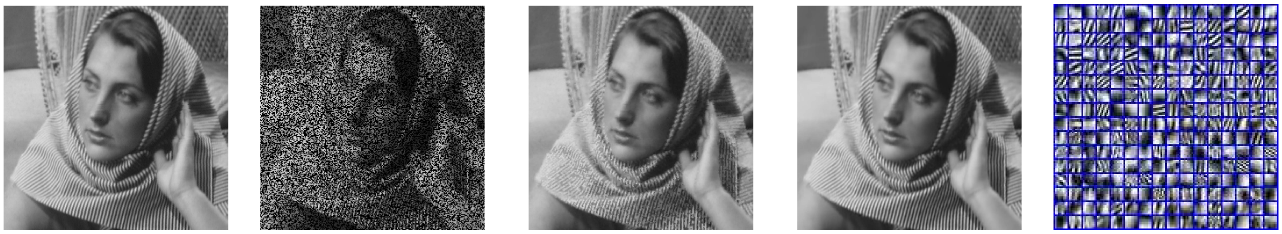
The following optimization problem emerges as result of the MAP formulation of (1) over a lattice graph on the image pixels, as illustrated in Figure 3, when the observations are corrupted by iid Gaussian noise with variance σ^2:

$$[\widehat{\boldsymbol{x}}, \widehat{\boldsymbol{s}}] = \arg \max_{\boldsymbol{x}', \boldsymbol{s}'} \sum_{(i,j) \in E} \lambda_{ij} s'_i s'_j + \sum_{i \in V} \left[\lambda_i s'_i + \log(p(x'_i | s'_i)) \right] - \frac{1}{2\sigma^2} \left| \left| \boldsymbol{y} - \boldsymbol{\Phi} \boldsymbol{x}' \right| \right|_2^2 . \tag{11}$$

For the compressibility of the structured sparse signal $\boldsymbol{x}$ as signal dimensions vary, the pdf's $p(x_i|s_i)$ should be chosen to result in a compressible prior for the signal coefficients, such as Student's t, with proper scale parameters to differentiate the small and large coefficients. Although the objective function is non-convex, a local optimum can be efficiently obtained via variational methods and alternating minimization techniques over the support and the signal coefficients.

In the context of sparse signal recovery, recent work [18] exploits the Ising model and demonstrates that it naturally motivates a greedy search algorithm for background subtraction with clustered sparsity, dubbed Lattice matching pursuit (LAMP). Enabled by the Ising model, the LAMP algorithm (i) converges significantly faster due to the reduction in the search space and (ii) recovers sparse signals from a much smaller number of measurements without sacrificing stability. Similar optimization formulations can also be seen in applications involving face recognition in the presence of occlusions, where the occlusions are not only sparse but also contiguous.

To solve (11), the LAMP algorithm relies on an approach inspired by the RIP assumption on the measurement matrices. The key observation for this iterative recovery scheme is that when the sampling matrix $\boldsymbol{\Phi}$ has K-RIP with constant $\epsilon_K \ll 1$ in (5), then the vector $\boldsymbol{b} = \boldsymbol{\Phi}^T \boldsymbol{y}$ can serve as a rough approximation of the original signal $\boldsymbol{x}$. In particular, the largest K entries of $\boldsymbol{b}$ point toward the largest K entries of the K-sparse signal $\boldsymbol{x}$. Then, given the signal coefficient estimates, LAMP uses graph cuts to obtain the MAP estimates of the latent support variables $\boldsymbol{s}$. By clever book-keeping of the data



Target Incomplete Lin. interp. (27.6dB) IBP (35.2dB) Learned dictionary

Fig. 4. *Natural images exhibit significant self similarities that can be leveraged using the Indian buffet processes (IBP) in sparse signal recovery even if a large number of pixels are missing. The IBP mechanism automatically infers a dictionary, in which the image patches are sparse, and the composition of these patches on the image to significantly improve the recovery performance over linear interpolation.*

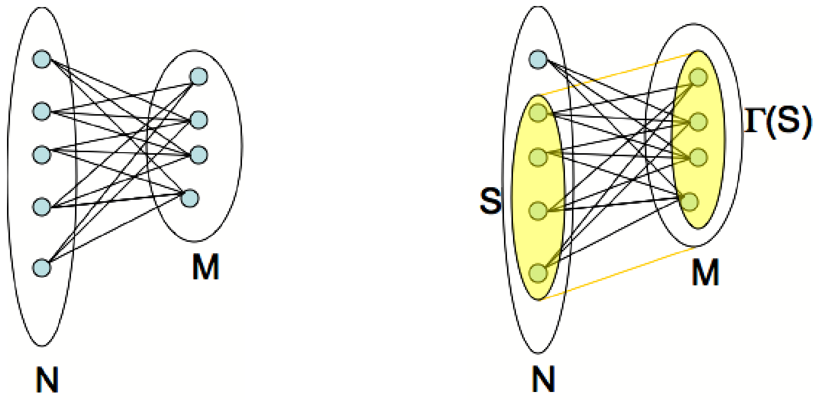
residual along with the auxiliary support estimates, LAMP iteratively refines its signal estimates until convergence; see Figure 3 for a robust recovery example from $M = 2K$, where the signal-to-noise ratio (SNR) in measurements is 10dB.

3) *Structured power-law processes:* In many sparse signal recovery problems, the appropriate signal sparsifying basis or *dictionary* $\mathbf{\Psi}$ is oft-times unknown and must be determined for each individual problem. In dictionary learning problems, researchers develop algorithms to learn a sparsifying dictionary directly from data using techniques where a set of signals are compressible in particular with nonparametric Bayesian priors. By nonparametric, we mean that the number of parameters within the prior distribution is beforehand unspecified. Recent work in this area focused on vision applications and developed structured random processes to capture the power-law distribution of the image patch frequencies and segment sizes. Such distributions lay the foundations of a scale, resolution independent inference mechanism, which is key for compressible structured sparse models.

We highlight two examples. Indian buffet processes (IBP), which provide exchangeable distributions over binary matrices, exploit hierarchical probability models and can be used to infer the dictionary size and its composition for natural images. Recent work exploits the IBP formalism for sparse signal recovery by jointly learning the sparsifying dictionary and demonstrates that significant improvements can be achieved in sparse recovery, denoising, and inpainting of natural images [22]; see Figure 4. Similarly, Pitman-Yor processes provide a statistical framework for unsupervised discovery and segmentation of visual object categories with power-law properties. Further applications of such power-law priors to the sparse signal recovery problem are yet to emerge [23].

IV. Graphical models for structured measurements

While random matrices have information preserving guarantees for the set of sparse signals, they are difficult to store, implement in real-hardware, and are computationally costly in sparse signal recovery.



(a) A bipartite graph.

(b) Expansion.

Fig. 5. An example of a bipartite graphs $G = (A, B, E)$, over the "left" set A of size N and the "right" set B of size M. A graph is an expander if any "small" subset S of A has "many" neighbors $\Gamma(S)$ in B.

Hence, in this section, we focus on another use of graphical models, to design high-performance measurement matrices $\mathbf{\Phi}$ that are *sparse*. Specifically, we consider matrices $\mathbf{\Phi}$ whose entries are mostly equal to zero, while the non-zero entries are equal to 1.[1] Each such matrix can be interpreted as bi-partite graph $G = (A, B, E)$ between a set $A = \{1 \ldots N\}$ of N nodes, and a set $B = \{1 \ldots M\}$ of M nodes: the edge $i - j$ is present in the graph if and only if $\mathbf{\Phi}_{j,i} = 1$. Note that the nodes in A correspond to the coordinates of the signal $\boldsymbol{x}$, while the nodes in B correspond to coordinates in the measurement vector $\boldsymbol{y}$. See Figure 5 for an illustration.

Sparse measurement matrices have several desirable features: (i) matrix-vector products during the encoding of the vector $\boldsymbol{x}$ into $\mathbf{\Phi}\boldsymbol{x}$ can be performed very efficiently, in time proportional to the number of non-zeros in $\mathbf{\Phi}$; (ii) the measurement vector $\mathbf{\Phi}\boldsymbol{x}$ can be quickly updated if one or a few of its coordinates are modified — crucial for processing massive data streams; (iii) the recovery process is quite efficient as well, since it relies on the matrix-vector product as a subroutine. Moreover, the graphical interpretation of such matrices enabled designing recovery algorithms using the *belief propagation approach*, leading to highly accurate recovery methods. Because of these reasons, various forms of sparse recovery using sparse matrices has been recently a subject of extensive research; see the recent survey [2] and the references therein.

To complement the above, we also point out some disadvantages of sparse matrices. One of them is that they are directly applicable only to the case where the signal $\boldsymbol{x}$ is approximately sparse in the

[1] Some of the constructions (e.g., [24]) allow the non-zero values to be equal to -1 as well.

canonical basis, i.e., $\boldsymbol{x} = \boldsymbol{\alpha}$ (see the introduction for the notation). If the signal (as it is often the case) is sparse only after applying a linear transformation $\boldsymbol{\Psi}$, then the actual measurement matrix is equal to $\boldsymbol{\Phi\Psi}$. This requires that the matrix $\boldsymbol{\Psi}$ is known before the measurements are taken. Note that the product matrix $\boldsymbol{\Phi\Psi}$ might not be sparse in general, and the encoding therefore must perform extra computation, corresponding to matrix multiplication; the order of these computations is $\mathcal{O}(NM)$ for general matrices, but it is much lower for special bases, such as discrete Fourier and wavelet transforms.

A. Intuition and the randomized case

We start from an intuitive overview of why properly chosen sparse matrices are capable of preserving enough information about sparse signals to enable their recovery. We focus on the case where $\boldsymbol{x}$ is exactly K-sparse, the matrix $\boldsymbol{\Phi}$ is random, and the goal is to exactly recover $\boldsymbol{x}$ from $\boldsymbol{y} = \boldsymbol{\Phi x}$ with "high" probability.

Consider first a particularly simple distribution over sparse matrices, where the i-th column of $\boldsymbol{\Phi}$ contains exactly single 1, at a position, indexed by $h(i)$, chosen independently and uniformly at random among the rows $\{1 \dots M\}$. Multiplying $\boldsymbol{\Phi}$ by $\boldsymbol{x}$ has then a natural message-passing interpretation; each coordinate $\boldsymbol{x}_i$ is *sent* to the $h(i)$-th coordinate of the measurement vector $\boldsymbol{y}$, and all coordinates sent to a given entry of $\boldsymbol{y}$ are added together. Assuming that M is sufficiently larger than K, we make the following observation: since the vector $\boldsymbol{x}$ is K-sparse and the indices $h(i)$ are chosen randomly from $\{1 \dots M\}$, there is only a small probability that any particular coordinate $\boldsymbol{x}_i$ will collide with any other non-zero entry $x_{i'}$ (where by collision we mean $h(i) = h(i')$). Therefore, $\boldsymbol{x}_i = \boldsymbol{y}_{h(i)}$ and we can recover sparse coefficients by inverse mapping from the measurement vector.

Unfortunately, the above procedure also creates many erroneous entries over the *entire* vector $\boldsymbol{x}$. In short, we note that each zero entry $\boldsymbol{x}_i$ has approximately K/M chance of colliding with some non-zero entry. Each such collision results in a erroneous non-zero entry in the recovered vector. As a result, a recovered vector can have $K/M \times (N - K) \gg K$ erroneous entries, and therefore becomes a poor approximation of a K-sparse signal $\boldsymbol{x}$.

To improve the quality of recovery, we reduce the probability of such erroneous entries via *repetition*. Specifically, instead of having only one 1 per column of the measurement matrix $\boldsymbol{\Phi}$, we can select D random entries of each column and set them to 1.[2] In the graph representation, this means that each node $i \in A$ has a set of D neighbors in B; we denote this set by $\Gamma(i)$. On one hand, this *increases*

[2] There are several different distributions of D-tuples from $\{1 \dots M\}$ than can be used, leading to essentially identical recovery results; see the survey [2].

the probability of a collision, since now $\boldsymbol{y}$ can contain up to DK non-zero entries. However, if we set $M = CDK$ for some constant $C > 1$, then for any fixed node $i \in A$, *each* of the D neighbors of i has at most $DK/M = 1/C$ chance of colliding with another non-zero entry. Therefore, the probability that *all* neighbors of i collide with other non-zero entries is very small, at most $1/C^D$.

Therefore, a sparse matrix $\boldsymbol{\Phi}$ can potentially preserve information for an overwhelming fraction of the coordinates of $\boldsymbol{x}$ by encoding it into $\boldsymbol{y}$. Then, how can we recover $\boldsymbol{x}$ from $\boldsymbol{y}$? If $C > 2$ and $D = d \log N$ for large enough constant d, then one can show that, with high probability, less than half of the neighbors of each node $i \in A$ collide with other non-zero entries. In that case, taking the median or the mode (the most frequently occurring value) of the entries $\{\boldsymbol{y}_j : j \in \Gamma(i)\}$ returns the correct value of $\boldsymbol{x}_i$. This is the basic idea behind *Count-Median* [25] and *Count-Sketch* [24] algorithms, which further extend this approach to general vectors $\boldsymbol{x}$.[3] The resulting algorithms require $M = \mathcal{O}(K \log N)$, and the recovery is performed in time proportional to $\mathcal{O}(N \log N)$.

In order to reduce the number of measurements, the researchers have developed more elaborate algorithms, based on loopy belief propagation [26, 27] and related message-passing approaches [28]. Here, we sketch the *Counter Braids* algorithms of [27]. That algorithm works assuming that the vector $\boldsymbol{x}$ is K-sparse and non-negative.

First, observe that for non-negative signals $\boldsymbol{x}$, the median-based approach of estimating $\boldsymbol{x}_i$ can be further simplified. Specifically, one can observe that for any neighbor j of i in G, we have $\boldsymbol{y}_j \geq \boldsymbol{x}_i$, since collisions with non-negative entries of $\boldsymbol{x}$ can only increase the values of $\boldsymbol{y}_j$. Therefore, we can estimate $\boldsymbol{x}_i$ by taking the minimum–instead of the median–of the set $\{\boldsymbol{y}_j : j \in \Gamma(i)\}$. Note that this can only lead to an overestimation of $\boldsymbol{x}_i$. This variant of the one-shot approach, called *Count-Min*, has been discovered in [25, 29].

The Counter Braids algorithm can be now described as an iterative refinement of Count-Min. Consider the estimation vector $\boldsymbol{x}'$ obtained above, i.e., $\boldsymbol{x}'_i = \min\{\boldsymbol{y}_j : j \in \Gamma(i)\}$. From the discussion, we have $\boldsymbol{x}' \geq \boldsymbol{x}$. The key idea is that we can now use $\boldsymbol{x}'$ to obtain an *underestimation* of $\boldsymbol{x}$. Consider any edge $i - j$. Since $\boldsymbol{y}_j = \sum_{l-j \in G} \boldsymbol{x}_l$, and $\boldsymbol{x}_l \leq \boldsymbol{x}'_l$, it follows that for $\boldsymbol{x}''_i = \boldsymbol{y}_j - \sum_{l-j \in G, l \neq i} \boldsymbol{x}'_l$ we have $\boldsymbol{x}_i \geq \boldsymbol{x}''_i$. Thus, we obtained a refined bound for $\boldsymbol{x}$, this time from below. We can now iterate this approach, alternating between upper and lower bounds on $\boldsymbol{x}$, until they are equal, in which case the recovery is exact, or some other stopping condition is satisfied.

The analysis given in [27] shows that, for a proper distribution over matrices $\boldsymbol{\Phi}$, one can give the

[3] For this case, however, we need to use the median estimator; the mode estimator does not work in this setting. In fact, the ”small” entries of $\boldsymbol{x}$ can make all coordinates of $\boldsymbol{y}$ distinct, in which case the mode estimator can report an arbitrary value

following guarantee. Let $\boldsymbol{x}$ be any K-sparse signal. Then, the Counter Braids algorithm, when given a random matrix $\boldsymbol{\Phi}$ and the measurement vector $\boldsymbol{\Phi x}$, recovers $\boldsymbol{x}$ with high probability, as long as K/N is small enough and $M \geq \kappa K \log(N/K)$ for $\kappa \approx 2.08$.

In a very recent work [28], the authors present a message-passing scheme (for general sparse vectors $\boldsymbol{x}$) which achieves a bound for M matching that of ℓ_1 minimization. Unlike other algorithms described in this section, the algorithm of [28] uses *dense* random matrices.

B. Deterministic case

The above discussion provides an intuition why multiplying a *fixed* vector $\boldsymbol{x}$ by a randomly chosen sparse matrix $\boldsymbol{\Phi}$ should preserve sufficient information about K-sparse approximation to $\boldsymbol{x}$. In order to be able to construct one (deterministic) matrix $\boldsymbol{\Phi}$ that works for *all* vectors $\boldsymbol{x}$, we need concrete matrices that behave in a semi-random fashion. For sparse matrices such property is encapsulated by the notion of *graph expansion*. For any set $S \subset A$, we let $\Gamma(S)$ be the set of neighbors of nodes in S, i.e., $\Gamma(S) = \cup_{i \in S} \Gamma(i)$. We say that the graph G is an (s, α)-*expander* if for any $S \subset A$, $|S| \leq s$, we have $|\Gamma(S)| \geq \alpha |S|$. Note that the *expansion factor* α is always at most D, since each node can have at most D neighbors. However, there exist graphs that achieve $\alpha = (1-\epsilon)D$ for any constant $\epsilon > 0$; such graphs require $D = \mathcal{O}(\log(N/s))$ and $M = |B| = \mathcal{O}(s \log(N/s))$. See Figure 5 for an illustration.

To see why expansion is desirable for sparse recovery, consider the case when ϵ is very close to 0 and $s = K$. Let $\boldsymbol{x}$ be a K-sparse vector , and let S be a set of indices of non-zero entries in $\boldsymbol{x}$. Then we observe that for most nodes $i \in S$, only a small fraction of their neighbors $\Gamma(i)$ collide with any other non-zero entry. Such entries $\boldsymbol{x}_i$ can be thus recovered correctly by the median-based procedure outlined earlier.

To recover *all* entries, we can use an iterative refinement approach, similar to the one from the previous section, and inspired by the "bit-flipping" algorithm for decoding low-density parity check codes. Consider first the case where $\boldsymbol{x}$ is exactly K-sparse. The algorithm [30] starts by setting an initial approximation $\boldsymbol{x}'$ to 0 and iteratively refines $\boldsymbol{x}'$ in order to achieve $\boldsymbol{\Phi x}' = \boldsymbol{y}$. In each step, it tries to reduce $\|\boldsymbol{\Phi x}' - \boldsymbol{y}\|_0$, by finding a pair (i, g) such that incrementing $\boldsymbol{x}'_i$ by g reduces the ℓ_0 difference. It is then shown that if the graph is an $(\mathcal{O}(K), (1-\epsilon)D)$-expander for a sufficiently small value of ϵ, then $\boldsymbol{x}'$ converges to $\boldsymbol{x}$ in $\mathcal{O}(K)$ iterations. The running time of the algorithm is dominated by the preprocessing step, which takes time $\mathcal{O}(ND)$, after which each iteration can be performed in $\mathcal{O}(\log N)$ time or, in some cases, even faster. Since the graph G is an expander, it follows that the number of measurements is $M = \mathcal{O}(KD) = \mathcal{O}(K \log(N/K))$.

When $\boldsymbol{x}$ is not exactly K-sparse, then the ℓ_0 norm is no longer a suitable measure of progress, and we need to use a more refined approach. To this end, we first observe that the graph-theoretic notion of expansion has a natural and useful geometric interpretation. Specifically, consider the following variant of the K-RIP in (5): we say that an $M \times N$ matrix $\boldsymbol{\Phi}$ has the *K-restricted isometry property in the ℓ_1 norm (K-RIP1)* with constant ϵ, if for all K-sparse vectors $\boldsymbol{x}$, we have

$$\|x\|_1(1-\epsilon) \leq \|\boldsymbol{\Phi}x\|_1 \leq \|x\|_1. \tag{12}$$

It has been shown in [31] that if $\boldsymbol{\Phi}$ is a matrix underlying a $(K, D(1-\epsilon/2))$-expander, then $\boldsymbol{\Phi}/D$ satisfies K-RIP1 with constant ϵ. As a result, both ℓ_1 minimization and variants of the iterative algorithms described in Section II can be used for sparse matrices. Unlike the methods using the standard RIP, the algorithms produce $\widehat{\boldsymbol{x}}$ which satisfies a somewhat weaker guarantee of the form

$$\|\boldsymbol{x} - \widehat{\boldsymbol{x}}\|_1 \leq C\|\boldsymbol{x} - \boldsymbol{x}_K\|_1 \tag{13}$$

Perhaps surprisingly, the simplest of the algorithms, called *Sequential Sparse Matching Pursuit* [32], is similar to the aforementioned algorithm of [30]. There are two key differences though. Firstly, each iteration reduces not the ℓ_0 error $\|\boldsymbol{\Phi}\boldsymbol{x}' - \boldsymbol{y}\|_0$, but the ℓ_1 error $\|\boldsymbol{\Phi}\boldsymbol{x}' - \boldsymbol{y}\|_1$. This is because by the RIP1 property, the error $\|\boldsymbol{x}' - \boldsymbol{x}\|_1$ is small when $\|\boldsymbol{\Phi}\boldsymbol{x}' - \boldsymbol{y}\|_1$ is small. Second, in order to be able to apply the RIP1 property, we need to ensure that the vector $\boldsymbol{x}'$ continues to be $\mathcal{O}(K)$-sparse after we perform the updates. Since this property might cease to be true after some number of steps, we need to periodically re-sparsify the vector $\boldsymbol{x}'$ by setting to zero all but the K largest (in absolute value) entries of $\boldsymbol{x}'$. This re-sparsification step is a standard tool in the design of iterative recovery algorithms for general vectors $\boldsymbol{x}$.

See the survey [2] for a more detailed description of the algorithms for sparse matrices.

V. Conclusions

A great deal of theoretic and algorithmic research has revolved around sparsity view of signals over the last decade to characterize new, sub-Nyquist sampling limits as well as tractable algorithms for signal recovery from dimensionality reduced measurements. Despite the promising advances made, real life applications require more realistic signal models that can capture underlying, application dependent order of sparse coefficients, better sampling matrices with information preserving properties that can be implemented in practical systems, and ever faster algorithms with provable recovery guarantees for real-time operation.

On this front, we have seen that graphical models (GM) are emerging to effectively address the core of many of these desiderata. GMs provide a broad scaffold for automatically encoding the probabilistic dependencies of sparse coefficients for sparse signal recovery. By exploiting the GM structure of signals beyond simple sparsity, we can radically reduce the number of measurements, increase noise robustness, and decrease recovery artifacts in signal acquisition. GMs are instrumental in constructing measurement matrices based on expander graphs. These matrices not only stably embed sparse signals into lower dimensions but also lead to faster recovery algorithms with rigorous guarantees. Moreover, the GM-based inference tools, such as variational methods, can estimate a posterior distribution for the sparse signal coefficients, providing confidence bounds that are critical in many applications.

To date, the sparse signal acquisition and recovery problems–surprisingly–have been studied largely in isolation. Real progress in efficient signal recovery, processing and analysis requires that we unify probabilistic, structured sparsity models with sparse measurement matrices to simultaneously reduce sampling requirements and the computational complexity of recovery without compromising the recovery guarantees. This will in turn entail investigation of streaming algorithms, coding theory, and learning theory with a common, connecting element, which we expect to be graphical models.

REFERENCES

[1] R. G. Baraniuk, V. Cevher, and M. Wakin, "Low-dimensional models for dimensionality reduction and signal recovery: A geometric perspective," *Proceedings of the IEEE*, 2010.

[2] A. Gilbert and P. Indyk, "Sparse recovery using sparse matrices," *Proceedings of IEEE*, 2010.

[3] C. M. Bishop, *Pattern recognition and machine learning*. Springer, 2006.

[4] D. L. Donoho, "Compressed sensing," *IEEE Trans. Info. Theory*, vol. 52, pp. 1289–1306, Sept. 2006.

[5] E. J. Candès, "Compressive sampling," in *Proc. International Congress of Mathematicians*, vol. 3, (Madrid, Spain), pp. 1433–1452, 2006.

[6] A. Cohen, W. Dahmen, and R. DeVore, "Compressed sensing and best k-term approximation," *American Mathematical Society*, vol. 22, no. 1, pp. 211–231, 2009.

[7] S. L. Lauritzen, *Graphical Models*. Oxford University Press, 1996.

[8] M. I. Jordan, Z. Ghahramani, T. S. Jaakkola, and L. K. Saul, "An introduction to variational methods for graphical models," *Machine Learning*, vol. 37, no. 2, pp. 183–233, 1999.

[9] A. Willsky, "Multiresolution Markov models for signal and image processing," *Proceedings of the IEEE*, vol. 90, no. 8, pp. 1396–1458, 2002.

[10] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann Publishers, 1988.

[11] D. Koller and N. Friedman, *Probabilistic Graphical Models: Principles and Techniques*. The MIT Press, 2009.

[12] M. S. Crouse, R. D. Nowak, and R. G. Baraniuk, "Wavelet-based statistical signal processing using Hidden Markov Models," *IEEE Trans. Signal Processing*, vol. 46, pp. 886–902, Apr. 1998.

[13] J. S. Yedidia, W. T. Freeman, and Y. Weiss, "Generalized belief propagation," *Advances in Neural Information Processing Systems*, vol. 13, pp. 689–695, 2001.

[14] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 39, no. 1, pp. 1–38, 1977.

[15] V. Cevher, “Learning with compressible priors,” in *NIPS*, (Vancouver, B.C., Canada), 7–12 December 2008.

[16] S. Negahban and M. J. Wainwright, “Estimation of (near) low-rank matrices with noise and high-dimensional scaling,” *Arxiv preprint arXiv:0912.5100*, 2009.

[17] L. He and L. Carin, “Exploiting structure in wavelet-based Bayesian compressive sensing,” 2008. Preprint. Available at http://people.ee.duke.edu/ lcarin/Papers.html.

[18] V. Cevher, M. F. Duarte, C. Hegde, and R. G. Baraniuk, “Sparse signal recovery using Markov random fields,” in *Neural Information Processing Systems (NIPS)*, (Vancouver, B.C., Canada), 8–11 December 2008.

[19] I. Daubechies, R. DeVore, M. Fornasier, and C. S. Gunturk, “Iteratively Reweighted Least Squares Minimization for Sparse Recovery,” *Communications on Pure and Applied Mathematics*, vol. 63, pp. 0001–0038, 2010.

[20] J. K. Romberg, H. Choi, and R. G. Baraniuk, “Bayesian tree-structured image modeling using wavelet-domain Hidden Markov Models,” *IEEE Trans. Image Processing*, vol. 10, pp. 1056–1068, July 2001.

[21] M. J. Wainwright and E. P. Simoncelli, “Scale mixtures of Gaussians and the statistics of natural images,” in *Neural Information Processing Systems (NIPS)* (S. A. Solla, T. K. Leen, and K.-R. Müller, eds.), vol. 12, (Cambridge, MA), pp. 855–861, MIT Press, Dec. 2000.

[22] M. Zhou, H. Chen, J. Paisley, L. Ren, G. Sapiro, and L. Carin, “Non-parametric bayesian dictionary learning for sparse image representations,” in *Neural Information Processing Systems (NIPS)*, 2009.

[23] E. B. Sudderth and M. I. Jordan, “Shared segmentation of natural scenes using dependent Pitman-Yor processes,” in *Advances in Neural Information Processing Systems*, vol. 21, 2009.

[24] M. Charikar, K. Chen, and M. Farach-Colton, “Finding frequent items in data streams,” *Proceedings of International Colloquium on Automata, Languages and Programming (ICALP)*, 2002.

[25] G. Cormode and S. Muthukrishnan, “Improved data stream summaries: The count-min sketch and its applications,” *Proceedings of Latin American Theoretical Informatics Symposium (LATIN)*, 2004.

[26] D. Baron, S. Sarvotham, and R. G. Baraniuk, “Bayesian compressive sensing via belief propagation,” *to appear in IEEE Transactions on Signal Processing*, 2010.

[27] Y. Lu, A. Montanari, B. Prabhakar, S. Dharmapurikar, and A. Kabbani, “Counter braids: a novel counter architecture for per-flow measurement,” in *SIGMETRICS ’08: Proceedings of the 2008 ACM SIGMETRICS international conference on Measurement and modeling of computer systems*, (New York, NY, USA), pp. 121–132, ACM, 2008.

[28] D. Donoho, A. Maleki, and A. Montanari, “Message passing algorithms for compressed sensing,” *Proceedings of the National Academy of Sciences*, 2009.

[29] C. Estan and G. Varghese, “New directions in traffic measurement and accounting: Focusing on the elephants, ignoring the mice,” *ACM Transactions on Computer Systems*, 2003.

[30] S. Jafarpour, W. Xu, B. Hassibi, and A. R. Calderbank, “Efficient and robust compressed sensing using high-quality expander graphs,” *IEEE Transactions on Information Theory*, vol. 33(9), 2009.

[31] R. Berinde, A. Gilbert, P. Indyk, H. Karloff, and M. Strauss, “Combining geometry and combinatorics: A unified approach to sparse signal recovery,” in *Proc. Allerton Conf. Communication, Control, and Computing*, 2008.

[32] R. Berinde and P. Indyk, “Sequential sparse matching pursuit,” *Proceedings of Allerton Conference on Communication, Control, and Computing*, 2009.