An Empirical-Bayes Approach to Recovering Linearly Constrained Non-Negative Sparse Signals

Jeremy P. Vila, *Student Member, IEEE*, and Philip Schniter, *Fellow, IEEE*

***Abstract*—We propose two novel approaches for the recovery of an (approximately) sparse signal from noisy linear measurements in the case that the signal is *a priori* known to be non-negative and obey given linear equality constraints, such as a simplex signal. This problem arises in, e.g., hyperspectral imaging, portfolio optimization, density estimation, and certain cases of compressive imaging. Our first approach solves a linearly constrained non-negative version of LASSO using the max-sum version of the generalized approximate message passing (GAMP) algorithm, where we consider both quadratic and absolute loss, and where we propose a novel approach to tuning the LASSO regularization parameter via the expectation maximization (EM) algorithm. Our second approach is based on the sum–product version of the GAMP algorithm, where we propose the use of a Bernoulli non-negative Gaussian-mixture signal prior and a Laplacian likelihood and propose an EM-based approach to learning the underlying statistical parameters. In both approaches, the linear equality constraints are enforced by augmenting GAMP's generalized-linear observation model with noiseless pseudo-measurements. Extensive numerical experiments demonstrate the state-of-the-art performance of our proposed approaches.**

***Index Terms*—Belief propagation, compressed sensing, estimation, expectation maximization algorithms.**

I. Introduction

We consider the recovery of an (approximately) sparse signal $\boldsymbol{x} \in \mathbb{R}^N$ from the noisy linear measurements

$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{w} \in \mathbb{R}^M, \tag{1}$$

where $\boldsymbol{A}$ is a known sensing matrix, $\boldsymbol{w}$ is noise, and M may be $\ll N$. In this paper, we focus on non-negative (NN) signals (i.e., $x_n \geq 0 \forall n$) that obey known linear equality constraints $\boldsymbol{B}\boldsymbol{x} = \boldsymbol{c} \in \mathbb{R}^P$. A notable example is *simplex*-constrained signals, i.e., $\boldsymbol{x} \in \Delta_+^N \triangleq \{\boldsymbol{x} \in \mathbb{R}^N : x_n \geq 0\ \forall n, \mathbf{1}^\top \boldsymbol{x} = 1\}$, occurring in hyperspectral image unmixing [2], portfolio optimization [3], [4], density estimation [5], [6], and other applications. We also consider the recovery of NN sparse signals without the linear constraint $\boldsymbol{B}\boldsymbol{x} = \boldsymbol{c}$ [7]–[9], which arises in imaging applications [10] and elsewhere [11].

One approach to recovering linearly constrained NN sparse $\boldsymbol{x}$ is to solve the ℓ_1-penalized constrained NN least-squares (LS) problem (2) (see, e.g., [4]) for some $\lambda \geq 0$:

$$\widehat{\boldsymbol{x}} = \arg\min_{\boldsymbol{x} \geq 0} \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|_2^2 + \lambda\|\boldsymbol{x}\|_1 \text{ s.t. } \boldsymbol{B}\boldsymbol{x} = \boldsymbol{c}. \tag{2}$$

Although this problem is convex [12], finding a solution can be computationally challenging in the high-dimensional regime. Also, while a larger λ is known to promote more sparsity in $\widehat{\boldsymbol{x}}$, determining the best choice of λ can be difficult in practice. For example, methods based on cross-validation, the L-curve, or Stein's unbiased risk estimator can be used (see [13] for discussions of all three), but they require much more computation than solving (2) for a fixed λ. For this reason, (2) is often considered under the special case $\lambda = 0$ [14], where it reduces to linearly constrained NN-LS.

For the recovery of K-sparse simplex-constrained signals, a special case of the general problem under consideration, the Greedy Selector and Simplex Projector (GSSP) was proposed in [6]. GSSP, an instance of projected gradient descent, iterates

$$\widehat{\boldsymbol{x}}^{i+1} = \mathcal{P}_K\left(\widehat{\boldsymbol{x}}^i - \mathsf{step}^i \nabla_{\boldsymbol{x}} \|\boldsymbol{y} - \boldsymbol{A}\widehat{\boldsymbol{x}}^i\|_2^2\right), \tag{3}$$

where $\mathcal{P}_K(\cdot)$ is the Euclidean projection onto the K-sparse simplex, $\widehat{\boldsymbol{x}}^i$ is the iteration-i estimate, step^i is the iteration-i step size, and $\nabla_{\boldsymbol{x}}$ is the gradient w.r.t $\boldsymbol{x}$. For algorithms of this sort, rigorous approximation guarantees can be derived when $\boldsymbol{A}$ obeys the restricted isometry property [15]. Determining the best choice of K can, however, be difficult in practice.

In this paper, we propose two methods for recovering a linearly constrained NN sparse vector $\boldsymbol{x}$ from noisy linear observations $\boldsymbol{y}$ of the form (1), both of which are based on the Generalized Approximate Message Passing (GAMP) algorithm [16], an instance of loopy belief propagation that has close connections to primal-dual optimization algorithms [17], [18]. When run in "max-sum" mode, GAMP can be used to solve optimization problems of the form $\widehat{\boldsymbol{x}} = \arg\min_{\boldsymbol{x}} \sum_{m=1}^M h_m([\boldsymbol{A}\boldsymbol{x}]_m) + \sum_{n=1}^N g_n(x_n)$, where $\widehat{\boldsymbol{x}}$ can be interpreted as the *maximum a posteriori* (MAP) estimate of $\boldsymbol{x}$ under the assumed signal prior (4) and likelihood (5):

$$f(\boldsymbol{x}) \propto \prod_{n=1}^N \exp\left(-g_n(x_n)\right) \tag{4}$$

$$f(\boldsymbol{y}|\boldsymbol{A}\boldsymbol{x}) \propto \prod_{m=1}^M \exp\left(-h_m\left([\boldsymbol{A}\boldsymbol{x}]_m\right)\right). \tag{5}$$

Manuscript received November 10, 2013; revised March 28, 2014; accepted July 02, 2014. Date of publication July 10, 2014; date of current version August 13, 2014. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Joseph Tabrikian. This work has been supported in part by NSF grants IIP-0968910, CCF-1018368, CCF-1218754, DARPA/ONR grant N66001-10-1-4090, and an allocation of computing time from the Ohio Supercomputer Center. Portions of this work were presented at the IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing [1].

The authors are with the Department of Electrical and Computer Engineering, The Ohio State University, Columbus, OH 43210 USA (e-mail: schniter@ece.osu.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSP.2014.2337841

When run in "sum-product" mode, GAMP returns an approximation of the minimum mean-squared error (MMSE) estimate of $\boldsymbol{x}$ under the same assumptions. In either case, the linear equality constraints $\boldsymbol{Bx} = \boldsymbol{c}$ can be enforced through the use of noiseless pseudo-measurements, as described in the sequel.

The first of our proposed approaches solves (2) using max-sum GAMP while tuning λ using a novel expectation-maximization (EM) [19] procedure. We henceforth refer to this approach as EM-NNL-GAMP, where NNL is short for "non-negative LASSO.[1]" We demonstrate, via extensive numerical experiments, that 1) the *runtime* of our approach is much faster than the state-of-the-art TFOCS solver [22] for a fixed λ, and that 2) the MSE *performance* of our λ-tuning procedure is on par with TFOCS under *oracle* tuning. We also consider the special case of $\lambda = 0$, yielding "non-negative least squares GAMP" (NNLS-GAMP), whose performance and runtime compare favorably to Matlab's `lsqlin` routine. In addition, we consider a variation on (2) that replaces the quadratic loss $\frac{1}{2}\|\boldsymbol{y} - \boldsymbol{Ax}\|_2^2$ with the absolute loss $\|\boldsymbol{y} - \boldsymbol{Ax}\|_1$ for improved robustness to outliers in $\boldsymbol{w}$ [23], and demonstrate the potential advantages of this technique on a practical dataset.

The second of our proposed approaches aims to solve not an optimization problem like (2) but rather an inference problem: compute the *MMSE estimate* of a linearly constrained NN sparse vector $\boldsymbol{x}$ from noisy linear observations $\boldsymbol{y}$. This is in general a daunting task, since computing the true MMSE estimate requires i) knowing both the true signal prior $f(\boldsymbol{x})$ and likelihood $f(\boldsymbol{y}|\boldsymbol{Ax})$, which are rarely available in practice, and ii) performing optimal inference w.r.t that prior and likelihood, which is rarely possible in practice for computational reasons.

However, when the coefficients in $\boldsymbol{x}$ are i.i.d and the observation matrix $\boldsymbol{A}$ in (1) is sufficiently large and random, recent work [24] has demonstrated that near-MMSE estimation is indeed possible via the following methodology: place an i.i.d Gaussian-mixture (GM) model with parameters $\boldsymbol{q}$ on the coefficients $\{x_n\}$, run sum-product GAMP based on that model, and tune the model parameters $\boldsymbol{q}$ using an appropriately designed EM algorithm. For such $\boldsymbol{A}$, the asymptotic optimality of GAMP as an MMSE-inference engine was established in [16], [25], and the ability of EM-GAMP to achieve consistent estimates of $\boldsymbol{q}$ was established in [26].

In this work, we show that the EM-GM-GAMP approach from [24] can be extended to *linearly constrained non-negative* signal models through the use of a non-negative Gaussian-mixture (NNGM) model and noiseless pseudo-measurements, and we detail the derivation and implementation of the resulting algorithm. Moreover, we demonstrate, via extensive numerical experiments, that EM-NNGM-GAMP's reconstruction MSE is state-of-the-art and that its runtime compares favorably to existing methods.

Both of our proposed approaches can be classified as "empirical-Bayes" [27] in the sense that they combine Bayesian and frequentist approaches: GAMP performs (MAP or MMSE) Bayesian inference with respect to a given prior, where the parameters of the prior are treated as deterministic and learned using the EM algorithm, a maximum-likelihood (ML) approach.

Notation: For matrices, we use boldface capital letters like $\boldsymbol{A}$, and we use $\boldsymbol{A}^{\top}$, $\mathrm{tr}(\boldsymbol{A})$, and $\|\boldsymbol{A}\|_F$ to denote the transpose, trace, and Frobenius norm, respectively. For vectors, we use boldface small letters like $\boldsymbol{x}$, and we use $\|\boldsymbol{x}\|_p = \left(\sum_n |x_n|^p\right)^{1/p}$ to denote the ℓ_p norm, with $x_n = [\boldsymbol{x}]_n$ representing the nth element of $\boldsymbol{x}$. Deterministic quantities are denoted using serif typeface (e.g., $x, \boldsymbol{x}, \boldsymbol{X}$), while random quantities are denoted using san-serif typeface (e.g., $\mathsf{x}, \boldsymbol{\mathsf{x}}, \boldsymbol{\mathsf{X}}$). For random variable x, we write the pdf as $f_{\mathsf{x}}(x)$, the expectation as $\mathrm{E}\{\mathsf{x}\}$, and the variance as $\mathrm{var}\{\mathsf{x}\}$. For a Gaussian random variable x with mean m and variance v, we write the pdf as $\mathcal{N}(x; m, v)$ and, for the special case of $\mathcal{N}(x; 0, 1)$, we abbreviate the pdf as $\varphi(x)$ and write the complimentary cdf as $\Phi_c(x)$. Meanwhile, for a Laplacian random variable x with location m and scale v, we write the pdf as $\mathcal{L}(x; m, v)$. For the point mass at $x = 0$, we use the Dirac delta distribution $\delta(x)$. Finally, we use $\mathbb{R}$ for the real field and $\int_+ g(x)dx$ for the integral of $g(x)$ over $x \in [0, \infty)$.

II. GAMP Overview

As described in Section I, the generalized approximate message passing (GAMP) algorithm [16] is an inference algorithm capable of computing either MAP or approximate-MMSE estimates of $\boldsymbol{x} \in \mathbb{R}^N$, where $\boldsymbol{x}$ is a realization of random vector $\boldsymbol{\mathsf{x}}$ with a prior of the form (6), from generalized-linear observations $\boldsymbol{y} \in \mathbb{R}^M$ that yield a likelihood of the form (7),

$$f_{\boldsymbol{\mathsf{x}}}(\boldsymbol{x}) \propto \prod_{n=1}^{N} f_{\mathsf{x}_n}(x_n) \tag{6}$$

$$f_{\boldsymbol{\mathsf{y}}|\boldsymbol{\mathsf{z}}}(\boldsymbol{y}|\boldsymbol{Ax}) \propto \prod_{m=1}^{M} f_{\mathsf{y}_m|\mathsf{z}_m}\left(y_m|[\boldsymbol{Ax}]_m\right), \tag{7}$$

where $\boldsymbol{\mathsf{z}} \triangleq \boldsymbol{A\mathsf{x}}$ represents "noiseless" transform outputs.

GAMP generalizes Donoho, Maleki, and Montanari's Approximate Message Passing (AMP) algorithms [28], [29] from the case of AWGN-corrupted linear observations to the generalized-linear model (7). As we shall see, this generalization is useful when enforcing the linear equality constraints $\boldsymbol{Bx} = \boldsymbol{c}$ and when formulating non-quadratic variations of (2).

GAMP is derived from particular approximations of loopy belief propagation (based on Taylor-series and central-limit-theorem arguments) that yield computationally simple "first-order" algorithms bearing strong similarity to primal-dual algorithms [17], [18]. Importantly, GAMP admits rigorous analysis in the large-system limit (i.e., $M, N \to \infty$ for fixed ratio M/N) under i.i.d sub-Gaussian $\boldsymbol{A}$ [16], [25], where its iterations obey a state evolution whose fixed points are optimal whenever they are unique. Meanwhile, for finite-sized problems and generic $\boldsymbol{A}$, max-sum GAMP yields the MAP solution whenever it converges, whereas sum-product GAMP minimizes a certain mean-field variational objective [17]. Although performance guarantees for generic finite-dimensional $\boldsymbol{A}$ are lacking except in special cases (e.g., [18]), in-depth empirical studies have demonstrated that (G)AMP performs relatively well for the $\boldsymbol{A}$ typically used in compressive sensing applications (see, e.g., [24]).

[1] In the absence of the constraint $\boldsymbol{Bx} = \boldsymbol{c}$, the optimization problem (2) can be recognized as a non-negatively constrained version of the LASSO [20] (also known as basis-pursuit denoising [21]). Similarly, in the special case of $\lambda = 0$, (2) reduces to non-negative LS [14].

Table I summarizes the GAMP algorithm. Effectively, GAMP converts the computationally intractable MAP and MMSE high-dimensional vector inference problems to a sequence of scalar inference problems. In the end, its complexity is dominated by four[2] matrix-vector multiplies per iteration: steps (R1), (R2), (R9), (R10). Furthermore, GAMP can take advantage of fast implementations of the matrix-vector multiplies (e.g., FFT) when they exist. For max-sum GAMP, scalar inference is accomplished by lines (R3) and (R11), which involve the proximal operator

$$\mathrm{prox}_g(\widehat{v};\mu^v) \triangleq \arg\min_{x\in\mathbb{R}} g(x) + \frac{1}{2\mu^v}|x-\widehat{v}|^2 \tag{8}$$

for generic scalar function $g(\cdot)$, as well as lines (R4) and (R12), which involve the derivative of the prox operator (8) with respect to its first argument. Meanwhile, for sum-product GAMP, scalar inference is accomplished by lines (R5) and (R6), which compute the mean and variance of GAMP's iteration-t approximation to the marginal posterior on z_m,

$$f_{\mathsf{z}_m|\mathsf{p}_m}(z|\widehat{p}_m(t);\mu_m^p(t)) \propto f_{\mathsf{y}_m|\mathsf{z}_m}(y_m|z)\mathcal{N}(z;\widehat{p}_m(t),\mu_m^p(t)), \tag{9}$$

and by lines (R13) and (R14), which compute the mean and variance of the GAMP-approximate marginal posterior on x_n,

$$f_{\mathsf{x}_n|\mathsf{r}_n}(x|\widehat{r}_n(t);\mu_n^r(t)) \propto f_{\mathsf{x}_n}(x)\mathcal{N}(x;\widehat{r}_n(t),\mu_n^r(t)). \tag{10}$$

We now provide background on GAMP that helps to explain (9), (10) and Table I. First and foremost, GAMP can be interpreted as an iterative thresholding algorithm, in the spirit of, e.g., [30], [31]. In particular, when the GAMP-assumed distributions are matched to the true ones, the variable $\widehat{r}_n(t)$ produced in (R10) is *an approximately AWGN-corrupted version of the true coefficient* x_n (i.e., $\widehat{\mathsf{r}}_n(t) = \mathsf{x}_n + \widetilde{\mathsf{r}}_n(t)$ with $\widetilde{\mathsf{r}}_n(t) \sim \mathcal{N}(0,\mu_n^r(t))$ independent of x_n) where $\mu_n^r(t)$ is computed in (R9) and the approximation becomes exact in the large-system limit with i.i.d sub-Gaussian $\boldsymbol{A}$ [16], [25]. Note that, under this AWGN corruption model, the pdf of x_n given $\widehat{\mathsf{r}}_n(t)$ takes the form in (10). Thus, in sum-product mode, GAMP sets $\widehat{x}_n(t+1)$ at the scalar MMSE estimate of x_n given $\widehat{r}_n(t)$, as computed via the conditional mean in (R13), and it sets $\mu_n^x(t+1)$ as the corresponding MMSE, as computed via the conditional variance in (R14). Meanwhile, in max-sum mode, GAMP sets $\widehat{x}_n(t+1)$ at the scalar MAP estimate of x_n given $\widehat{r}_n(t)$, as computed by the prox step in (R11), and it sets $\mu_n^x(t+1)$ in accordance with the sensitivity of this proximal thresholding, as computed in (R12). This explains (10) and lines (R9)–(R14) in Table I.

We now provide a similar explanation for (9) and lines (R1)–(R6) in Table I. When the GAMP distributions are matched to the true ones, $\widehat{p}_m(t)$ produced in (R2) is *an approximately AWGN-corrupted version of the true transform output* z_m (i.e., $\widehat{\mathbf{p}}_m(t) = \mathbf{z}_m + \widetilde{\mathbf{p}}_m(t)$ with $\widetilde{\mathbf{p}}_m(t) \sim \mathcal{N}(0,\mu_m^p(t))$ independent of $\widehat{\mathbf{p}}_m(t)$) where $\mu_m^p(t)$ is computed in (R1) and the approximation becomes exact in the large-system limit with i.i.d sub-Gaussian $\boldsymbol{A}$ [16], [25]. Under this model, the pdf of $\mathbf{z}_m$ given $\widehat{\mathbf{p}}_m(t)$ and $\mathbf{y}_m$ takes the form in (9). Thus, in sum-product mode, GAMP sets $\widehat{z}_m(t)$ at the scalar MMSE estimate of $\mathbf{z}_m$ given $\widehat{p}_m(t)$ and y_m, as computed via the conditional mean in (R5), and it sets $\mu_m^z(t)$ as the corresponding MMSE, as computed via the conditional variance in (R6). Meanwhile, in max-sum mode, GAMP sets $\widehat{z}_m(t)$ at the scalar MAP estimate of $\mathbf{z}_m$ given $\widehat{p}_m(t)$ and y_m, as computed by the prox operation in (R3), and it sets $\mu_m^z(t)$ in accordance with the sensitivity of this prox operation, as computed in (R4).

TABLE I
THE GAMP ALGORITHM FROM [16] WITH MAX ITERATIONS $T_{\max}$ AND STOPPING TOLERANCE ϵ_{gamp}

```
inputs: $\forall m,n : f_{\mathsf{x}_n}, f_{\mathsf{y}_m|\mathsf{z}_m}, A_{mn}, T_{\max}, \epsilon_{\mathsf{gamp}} > 0, \mathsf{MaxSum} \in \{0,1\}$
definitions:
  $f_{\mathsf{z}_m|\mathsf{p}_m}(z\,|\,\widehat{p};\mu^p) \triangleq \frac{f_{\mathsf{y}_m|\mathsf{z}_m}(y_m|z)\,\mathcal{N}(z;\widehat{p},\mu^p)}{\int_z f_{\mathsf{y}_m|\mathsf{z}_m}(y_m|z)\,\mathcal{N}(z;\widehat{p},\mu^p)}$   (D1)
  $f_{\mathsf{x}_n|\mathsf{r}_n}(x\,|\,\widehat{r};\mu^r) \triangleq \frac{f_{\mathsf{x}_n}(x)\,\mathcal{N}(x;\widehat{r},\mu^r)}{\int_x f_{\mathsf{x}_n}(x)\,\mathcal{N}(x;\widehat{r},\mu^r)}$   (D2)
initialize:
    $\forall n : \widehat{x}_n(1) = \int_x x\, f_{\mathsf{x}_n}(x)$   (I1)
    $\forall n : \mu_n^x(1) = \int_x |x - \widehat{x}_n(1)|^2 f_{\mathsf{x}_n}(x)$   (I2)
    $\forall m : \widehat{s}_m(0) = 0$   (I3)
for $t = 1 : T_{\max}$,
    $\forall m : \mu_m^p(t) = \sum_{n=1}^N |A_{mn}|^2 \mu_n^x(t)$   (R1)
    $\forall m : \widehat{p}_m(t) = \sum_{n=1}^N A_{mn}\widehat{x}_n(t) - \mu_m^p(t)\,\widehat{s}_m(t-1)$   (R2)
  if MaxSum then
    $\forall m : \widehat{z}_m(t) = \mathrm{prox}_{-\ln f_{\mathsf{y}_m|\mathsf{z}_m}}(\widehat{p}_m(t);\mu_m^p(t))$   (R3)
    $\forall m : \mu_m^z(t) = \mu_m^p(t)\,\mathrm{prox}'_{-\ln f_{\mathsf{y}_m|\mathsf{z}_m}}(\widehat{p}_m(t);\mu_m^p(t))$   (R4)
  else
    $\forall m : \widehat{z}_m(t) = \mathrm{E}\{\mathsf{z}_m|\mathsf{p}_m = \widehat{p}_m(t);\mu_m^p(t)\}$   (R5)
    $\forall m : \mu_m^z(t) = \mathrm{var}\{\mathsf{z}_m|\mathsf{p}_m = \widehat{p}_m(t);\mu_m^p(t)\}$   (R6)
  end if
    $\forall m : \mu_m^s(t) = \big(1 - \mu_m^z(t)/\mu_m^p(t)\big)/\mu_m^p(t)$   (R7)
    $\forall m : \widehat{s}_m(t) = \big(\widehat{z}_m(t) - \widehat{p}_m(t)\big)/\mu_m^p(t)$   (R8)
    $\forall n : \mu_n^r(t) = \big(\sum_{m=1}^M |A_{mn}|^2 \mu_m^s(t)\big)^{-1}$   (R9)
    $\forall n : \widehat{r}_n(t) = \widehat{x}_n(t) + \mu_n^r(t)\sum_{m=1}^M A_{mn}^* \widehat{s}_m(t)$   (R10)
  if MaxSum then
  $\forall n : \widehat{x}_n(t+1) = \mathrm{prox}_{-\ln f_{\mathsf{x}_n}}(\widehat{r}_n(t);\mu_n^r(t))$   (R11)
  $\forall n : \mu_n^x(t+1) = \mu_n^r(t)\,\mathrm{prox}'_{-\ln f_{\mathsf{x}_n}}(\widehat{r}_n(t);\mu_n^r(t))$   (R12)
  else
  $\forall n : \widehat{x}_n(t+1) = \mathrm{E}\{\mathsf{x}_n|\mathsf{r}_n = \widehat{r}_n(t);\mu_n^r(t)\}$   (R13)
  $\forall n : \mu_n^x(t+1) = \mathrm{var}\{\mathsf{x}_n|\mathsf{r}_n = \widehat{r}_n(t);\mu_n^r(t)\}$   (R14)
  end if
  if $\sum_{n=1}^N |\widehat{x}_n(t+1) - \widehat{x}_n(t)|^2 < \epsilon_{\mathsf{gamp}} \sum_{n=1}^N |\widehat{x}_n(t)|^2$, break   (R15)
end
outputs: $\forall m,n : \widehat{z}_m(t), \mu_m^z(t), \widehat{r}_n(t), \mu_n^r(t), \widehat{x}_n(t+1), \mu_n^x(t+1)$
```

Indeed, what sets GAMP (and its simpler incarnation AMP) apart from other iterative thresholding algorithms is that the thresholder inputs $\widehat{r}_n(t)$ and $\widehat{p}_m(t)$ are (approximately) AWGN corrupted observations of x_n and $\mathbf{z}_m$, respectively, ensuring that the scalar thresholding steps (R3)-(R6) and (R11)-(R14) are well justified from the MAP or MMSE perspectives. Moreover, it is the "Onsager" correction "$-\mu_m^p(t)\widehat{s}_m(t-1)$" in (R2) that ensures the AWGN nature of the corruptions; without it, AMP reduces to classical iterative thresholding [28], which performs much worse [32]. Computing the Onsager correction involves (R7)-(R8). To our knowledge, the simplest interpretation of the variables $\widehat{s}_m(t)$ and $\mu_m^s(t)$ computed in (R7)–(R8) comes from primal-dual optimization theory, as established in [18]: whereas $\widehat{x}_n(t)$ are estimates of the primal variables, $\widehat{s}_m(t)$ are estimates of the dual variables; and whereas $\mu_n^x(t)$ relates to the primal sensitivity at the point $\widehat{x}_n(t)$, $\mu_m^s(t)$ relates to the dual sensitivity at $\widehat{s}_m(t)$.

[2] Two matrix multiplies per iteration, those in (R1) and (R9), can be eliminated using the "scalar variance" modification of GAMP, with vanishing degradation in the large-system limit [16].

III. Observation Models

To enforce the linear equality constraint $\boldsymbol{Bx} = \boldsymbol{c} \in \mathbb{R}^P$ using GAMP, we extend the observation model (1) to

$$\underbrace{\begin{bmatrix} \boldsymbol{y} \\ \boldsymbol{c} \end{bmatrix}}_{\triangleq \overline{\boldsymbol{y}}} = \underbrace{\begin{bmatrix} \boldsymbol{A} \\ \boldsymbol{B} \end{bmatrix}}_{\triangleq \overline{\boldsymbol{A}}} \boldsymbol{x} + \begin{bmatrix} \boldsymbol{w} \\ \boldsymbol{0} \end{bmatrix} \tag{11}$$

and exploit the fact that GAMP supports a likelihood function that varies with the measurement index m. Defining $\overline{\boldsymbol{z}} \triangleq \overline{\boldsymbol{A}}\boldsymbol{x}$, the likelihood associated with the augmented model (11) can be written as

$$f_{\overline{\mathsf{y}}_m|\overline{\mathsf{z}}_m}(\overline{y}_m|\overline{z}_m) = \begin{cases} f_{\mathsf{y}|\mathsf{z}}(\overline{y}_m|\overline{z}_m) & m = 1,\ldots,M \\ \delta(\overline{y}_m - \overline{z}_m) & m = M+1,\ldots,M+P, \end{cases} \tag{12}$$

where $f_{\mathsf{y}|\mathsf{z}}$ corresponds to the first M measurements, i.e., (1).

Note that, for either max-sum or sum-product GAMP, the quantities in (R3)–(R6) of Table I then become

$$\widehat{z}_m(t) = c_{m-M} \quad m = M+1,\ldots,M+P \tag{13}$$
$$\mu^z_m(t) = 0 \quad m = M+1,\ldots,M+P, \tag{14}$$

where c_{m-M} are elements of $\boldsymbol{c}$.

A. Additive White Gaussian Noise

When the noise $\boldsymbol{w}$ is modeled as additive white Gaussian noise (AWGN) with variance ψ, the likelihood $f_{\mathsf{y}|\mathsf{z}}$ in (12) takes the form

$$f_{\mathsf{y}|\mathsf{z}}(y|z) = \mathcal{N}(y; z, \psi). \tag{15}$$

In this case, for either max-sum or sum-product GAMP, the quantities in (R3)-(R6) of Table I become [16] (omitting the t index for brevity)

$$\widehat{z}_m = \widehat{p}_m + \frac{\mu^p_m}{\mu^p_m + \psi}(y_m - \widehat{p}_m) \quad m = 1,\ldots,M \tag{16}$$
$$\mu^z_m = \frac{\mu^p_m \psi}{\mu^p_m + \psi} \quad m = 1,\ldots,M. \tag{17}$$

B. Additive White Laplacian Noise

The additive white Laplacian noise (AWLN) observation model is an alternative to the AWGN model that is more robust to outliers [23]. Here, the noise $\boldsymbol{w}$ is modeled as AWLN with rate parameter $\psi > 0$, and the corresponding likelihood $f_{\mathsf{y}|\mathsf{z}}$ in (12) takes the form

$$f_{\mathsf{y}|\mathsf{z}}(y|z) = \mathcal{L}(y; z, \psi) \triangleq \frac{\psi}{2}\exp\left(-\psi|y-z|\right), \tag{18}$$

and so, for the max-sum case, (R3) in Table I becomes

$$\widehat{z}_m = \underset{z_m \in \mathbb{R}}{\arg\min}\, |z_m - y_m| + \frac{(z_m - \widehat{p}_m)^2}{2\mu^p_m \psi}. \tag{19}$$

The solution to (19) can be recognized as a y_m-shifted version of "soft-thresholding" function, and so the max-sum quantities in (R3) and (R4) of Table I become, using $\widetilde{p}_m \triangleq \widehat{p}_m - y_m$,

$$\widehat{z}_m = \begin{cases} \widehat{p}_m - \psi\mu^p_m & \widetilde{p}_m \geq \psi\mu^p_m \\ \widehat{p}_m + \psi\mu^p_m & \widetilde{p}_m \leq -\psi\mu^p_m \\ y_m & \text{else} \end{cases} \quad m = 1,\ldots,M, \tag{20}$$

$$\mu^z_m = \begin{cases} 0 & |\widetilde{p}_m| \leq \psi\mu^p_m \\ \mu^p_m & \text{else} \end{cases} \quad m = 1,\ldots,M. \tag{21}$$

Meanwhile, as shown in Appendix B-A, the sum-product GAMP quantities (R5) and (R6) (i.e., the mean and variance of the GAMP approximated z_m posterior (9)) become

$$\begin{aligned} \widehat{z}_m = {} & y_m + \frac{\underline{C}_m}{C_m}\left(\underline{p}_m - \sqrt{\mu^p_m}h(\underline{\kappa}_m)\right) \\ & + \frac{\overline{C}_m}{C_m}\left(\overline{p}_m + \sqrt{\mu^p_m}h(\overline{\kappa}_m)\right) \end{aligned} \tag{22}$$

$$\begin{aligned} \mu^z_m = {} & \frac{\underline{C}_m}{C_m}\left(\mu^p_m g(\underline{\kappa}_m) + \left(\underline{p}_m - \sqrt{\mu^p_m}h(\underline{\kappa}_m)\right)^2\right) \\ & + \frac{\overline{C}_m}{C_m}\left(\mu^p_m g(\overline{\kappa}_m) + \left(\overline{p}_m + \sqrt{\mu^p_m}h(\overline{\kappa}_m)\right)^2\right) \\ & - (y_m - \widehat{z}_m)^2, \end{aligned} \tag{23}$$

where $\underline{p}_m \triangleq \widetilde{p}_m + \psi\mu^p_m$, $\overline{p}_m \triangleq \widetilde{p}_m - \psi\mu^p_m$,

$$\underline{C}_m \triangleq \frac{\psi}{2}\exp\left(\psi\widetilde{p}_m + \frac{1}{2}\psi^2\mu^p_m\right)\Phi_c(\underline{\kappa}_m) \tag{24}$$

$$\overline{C}_m \triangleq \frac{\psi}{2}\exp\left(-\psi\widetilde{p}_m + \frac{1}{2}\psi^2\mu^p_m\right)\Phi_c(\overline{\kappa}_m), \tag{25}$$

$C_m \triangleq \underline{C}_m + \overline{C}_m$, $\underline{\kappa}_m \triangleq \underline{p}_m / \sqrt{\mu^p_m}$, $\overline{\kappa}_m \triangleq -\overline{p}_m / \sqrt{\mu^p_m}$ and

$$h(a) \triangleq \frac{\varphi(a)}{\Phi_c(a)} \tag{26}$$

$$g(a) \triangleq 1 - h(a)\left(h(a) - a\right). \tag{27}$$

IV. Non-Negative GAMP

A. NN Least Squares GAMP

We first detail the NNLS-GAMP algorithm, which uses max-sum GAMP to solve the $\lambda = 0$ case of (2). Noting that the $\boldsymbol{x} \geq \boldsymbol{0}$ constraint in (2) can be thought of as adding an infinite penalty to the quadratic term when any $x_n < 0$ and no additional penalty otherwise, we model the elements of $\boldsymbol{x}$ as i.i.d random variables with the (improper) NN prior pdf

$$f_{\mathsf{x}}(x) = \begin{cases} 1 & x \geq 0 \\ 0 & x < 0 \end{cases}, \tag{28}$$

and we assume the augmented model (12) with AWGN likelihood (15) (of variance $\psi = 1$), in which case max-sum GAMP performs the unconstrained optimization

$$\underset{\boldsymbol{x}}{\arg\min} - \sum_{n=1}^{N} \ln \mathbb{1}_{x_n \geq 0} - \ln \mathbb{1}_{\boldsymbol{Bx}=\boldsymbol{c}} + \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{Ax}\|_2^2, \tag{29}$$

where $\mathbb{1}_A \in \{0,1\}$ is the indicator function of the event A. Hence, (29) is equivalent to the constrained optimization (2) when $\lambda = 0$.

Under the i.i.d NN uniform prior (28), it is readily shown that the max-sum GAMP steps (R11) and (R12) become

$$\widehat{x}_n = \begin{cases} 0 & \widehat{r}_n \leq 0 \\ \widehat{r}_n & \widehat{r}_n > 0 \end{cases}, \tag{30}$$

$$\mu_n^x = \begin{cases} 0 & \widehat{r}_n \leq 0 \\ \mu_n^r & \widehat{r}_n > 0 \end{cases}. \tag{31}$$

B. NN LASSO GAMP

Next we detail the NNL-GAMP algorithm, which uses max-sum GAMP to solve the $\lambda > 0$ case of (2). For this, we again employ the augmented model (12) and AWGN likelihood (15) (with variance ψ), but we now use i.i.d exponential x_n, i.e.,

$$f_{\mathsf{x}}(x) = \begin{cases} \chi \exp(-\chi x) & x \geq 0 \\ 0 & \text{else} \end{cases} \tag{32}$$

for $\chi > 0$. With these priors and the augmented observation model (11), NNL-GAMP solves the optimization problem

$$\widehat{\boldsymbol{x}} = \arg\min_{\boldsymbol{x} \geq 0} \frac{1}{2\psi} \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|_2^2 + \chi \|\boldsymbol{x}\|_1 \text{ s.t. } \boldsymbol{B}\boldsymbol{x} = \boldsymbol{c}, \tag{33}$$

which reduces to (2) under $\lambda = \chi\psi$.

It is then straightforward to show that the max-sum lines (R11) and (R12) in Table I reduce to

$$\widehat{x}_n = \begin{cases} \widehat{r}_n - \chi\mu_n^r & \widehat{r}_n \geq \chi\mu_n^r \\ 0 & \text{else} \end{cases} \tag{34}$$

$$\mu_n^x = \begin{cases} \mu_n^r & \widehat{r}_n \geq \chi\mu_n^r \\ 0 & \text{else.} \end{cases} \tag{35}$$

C. NN Gaussian Mixture GAMP

Finally, we detail the NNGM-GAMP algorithm, which employs sum-product GAMP under the i.i.d Bernoulli non-negative Gaussian mixture (NNGM) prior pdf for $\boldsymbol{\mathsf{x}}$, i.e.,

$$f_{\mathsf{x}}(x) = (1-\tau)\delta(x) + \tau \sum_{\ell=1}^{L} \omega_\ell \mathcal{N}_+(x; \theta_\ell, \phi_\ell), \tag{36}$$

where $\mathcal{N}_+(\cdot)$ denotes the non-negative Gaussian pdf,

$$\mathcal{N}_+(x; \theta, \phi) = \begin{cases} \frac{\mathcal{N}(x;\theta,\phi)}{\Phi_c(-\theta/\sqrt{\phi})} & x \geq 0 \\ 0 & x < 0, \end{cases} \tag{37}$$

$\tau \in (0, 1]$ is the sparsity rate, and ω_ℓ, θ_ℓ, and ϕ_ℓ are the weight, location, and scale, respectively, of the ℓth mixture component. For now, we treat the NNGM parameters $[\tau, \boldsymbol{\omega}, \boldsymbol{\theta}, \boldsymbol{\phi}]$ and the model order L as fixed and known.

As shown in Appendix C-A, the sum-product GAMP quantities in (R13) and (R14) of Table I then become

$$\widehat{x}_n = \frac{\tau}{\zeta_n} \sum_{\ell=1}^{L} \beta_{n,\ell} \left(\gamma_{n,\ell} + \sqrt{\nu_{n,\ell}} h(\alpha_{n,\ell})\right) \tag{38}$$

$$\mu_n^x = \frac{\tau}{\zeta_n} \sum_{\ell=1}^{L} \beta_{n,\ell} \Big(\nu_{n,\ell} g(\alpha_{n,\ell}) + \left(\gamma_{n,\ell} + \sqrt{\nu_{n,\ell}} h(\alpha_{n,\ell})\right)^2\Big) - \widehat{x}_n^2, \tag{39}$$

where ζ_n is the normalization factor

$$\zeta_n \triangleq (1-\tau)\mathcal{N}(0; \widehat{r}_n, \mu_n^r) + \tau \sum_{\ell=1}^{L} \beta_{n,\ell}, \tag{40}$$

$h(\cdot)$ and $g(\cdot)$ were defined in (26) and (27), respectively, and

$$\alpha_{n,\ell} \triangleq \frac{-\gamma_{n,\ell}}{\sqrt{\nu_{n,\ell}}} \tag{41}$$

$$\gamma_{n,\ell} \triangleq \frac{\widehat{r}_n/\mu_n^r + \theta_\ell/\phi_\ell}{1/\mu_n^r + 1/\phi_\ell}, \tag{42}$$

$$\nu_{n,\ell} \triangleq \frac{1}{1/\mu_n^r + 1/\phi_\ell} \tag{43}$$

$$\beta_{n,\ell} \triangleq \frac{\omega_\ell \mathcal{N}(\widehat{r}_n; \theta_\ell, \mu_n^r + \phi_\ell)\, \Phi_c(\alpha_{n,\ell})}{\Phi_c(-\theta_\ell/\sqrt{\phi_\ell})}. \tag{44}$$

From (10) and (36), it follows that GAMP's approximation to the posterior activity probability $\Pr\{\mathsf{x}_n \neq 0 \mid \boldsymbol{y}\}$ is

$$\pi_n = \frac{1}{1 + \left(\frac{\tau}{1-\tau} \frac{\sum_{\ell=1}^{L} \beta_{n,\ell}}{\mathcal{N}(0; \widehat{r}_n, \mu_n^r)}\right)^{-1}}. \tag{45}$$

V. EM Learning of the Prior Parameters

In the sequel, we will use $\boldsymbol{q}$ to refer to the collection of prior parameters. For example, if NNGM-GAMP was used with the AWGN observation model, then $\boldsymbol{q} = [\tau, \boldsymbol{\omega}, \boldsymbol{\theta}, \boldsymbol{\phi}, \psi]$. Since the value of $\boldsymbol{q}$ that best fits the true data is typically unknown, we propose to learn it using an EM procedure [19]. The EM algorithm is an iterative technique that is guaranteed to converge to a local maximum of the likelihood $f(\boldsymbol{y}; \boldsymbol{q})$.

To understand the EM algorithm, it is convenient to write the log-likelihood as [24]

$$\ln f(\boldsymbol{y}; \boldsymbol{q}) = \mathcal{Q}_{\widehat{p}}(\boldsymbol{y}; \boldsymbol{q}) + D\left(\widehat{p} \| f_{\boldsymbol{\mathsf{x}}|\boldsymbol{\mathsf{y}}}(\cdot|\boldsymbol{y}; \boldsymbol{q})\right), \tag{46}$$

where $\widehat{p}$ is an arbitrary distribution on $\boldsymbol{\mathsf{x}}$, $D(\widehat{p}\|\widehat{q})$ is the Kullback-Leibler (KL) divergence between $\widehat{p}$ and $\widehat{q}$, and

$$\mathcal{Q}_{\widehat{p}}(\boldsymbol{y}; \boldsymbol{q}) \triangleq E_{\widehat{p}}\{\ln f_{\boldsymbol{\mathsf{x}},\boldsymbol{\mathsf{y}}}(\boldsymbol{\mathsf{x}}, \boldsymbol{y}; \boldsymbol{q})\} + H(\widehat{p}), \tag{47}$$

where $H(\widehat{p})$ is the entropy of $\boldsymbol{\mathsf{x}} \sim \widehat{p}$. Importantly, the non-negativity of KL divergence implies that $\mathcal{Q}_{\widehat{p}}(\boldsymbol{y}; \boldsymbol{q})$ is a lower bound on (46). Starting from the initialization $\boldsymbol{q}^0$, the EM algorithm iteratively improves its estimate $\boldsymbol{q}^i$ at each iteration $i \in \mathbb{N}$: first, it assigns $\widehat{p}^i(\cdot) = f_{\boldsymbol{\mathsf{x}}|\boldsymbol{\mathsf{x}}\boldsymbol{\mathsf{y}}}(\cdot|\boldsymbol{y}; \boldsymbol{q}^i)$ to tighten the bound, and then it sets $\boldsymbol{q}^{i+1}$ to maximize (47) with $\widehat{p} = \widehat{p}^i$.

Since the exact posterior pdf $f_{\boldsymbol{\mathsf{x}}|\boldsymbol{\mathsf{y}}}(\cdot|\boldsymbol{y}; \boldsymbol{q}^i)$ is difficult to calculate, in its place we use GAMP's approximate posterior $\prod_n f_{\mathsf{x}_n|\mathsf{r}_n}(\cdot|\widehat{r}_n; \mu_n^r; \boldsymbol{q}^i)$ from (10), resulting in the EM update

$$\boldsymbol{q}^{i+1} = \arg\max_{\boldsymbol{q}} \widehat{\mathrm{E}}\left\{\ln f(\boldsymbol{\mathsf{x}}, \boldsymbol{y}; \boldsymbol{q}) | \boldsymbol{y}; \boldsymbol{q}^i\right\}, \tag{48}$$

where $\widehat{\mathrm{E}}$ denotes expectation using GAMP's approximate posterior. Also, because calculating the joint update for $\boldsymbol{q}$ in (48) can be difficult, we perform the maximization (48) one component at a time, known as "incremental EM" [33]. Note that, even when using an approximate posterior and updating incrementally, the EM algorithm iteratively maximizes a lower-bound to the log-likelihood.

Whereas [24] proposed the use of (48) to tune *sum-product* GAMP, where the marginal posteriors $f_{\mathsf{x}_n|\mathsf{r}_n}(\cdot|\widehat{r}_n;\mu_n^r;\boldsymbol{q}^i)$ from (10) are computed for use in steps (R13)–(R14) of Table I, we hereby propose the use of (48) to tune *max-sum* GAMP. The reasoning behind our proposal goes as follows. Although max-sum GAMP does not compute marginal posteriors (but rather joint MAP estimates), its large-system-limit analysis (under i.i.d sub-Gaussian $\boldsymbol{A}$) [25] shows that $\widehat{r}_n(t)$ can be modeled as an AWGN-corrupted measurement of the true x_n with AWGN variance $\mu_n^r(t)$, revealing the *opportunity* to compute marginal posteriors via (10) as an additional step. Doing so enables the use of (48) to tune max-sum GAMP.

A. EM Update of AWGN Variance

We first derive the EM update of the AWGN noise variance ψ (recall (15)). This derivation differs from the one in [24] in that here we use $\boldsymbol{\mathsf{x}}$ as the hidden variable (rather than $\boldsymbol{\mathsf{z}}$), since experimentally we have observed gains in the low-SNR regime (e.g., SNR < 10 dB). Because we can write $f(\boldsymbol{x},\boldsymbol{y};\boldsymbol{q}) = D\prod_{m=1}^{M} f_{\mathsf{y|z}}(y_m|\boldsymbol{a}_m^\mathsf{T}\boldsymbol{x};\psi)$ with a ψ-invariant term D, the incremental update of ψ from (48) becomes

$$\psi^{i+1} = \underset{\psi>0}{\arg\max} \sum_{m=1}^{M} \widehat{\mathrm{E}}\left\{\ln f_{\mathsf{y|z}}\left(y_m|\boldsymbol{a}_m^\mathsf{T}\boldsymbol{\mathsf{x}};\psi\right)|\boldsymbol{y};\psi^i\right\}. \quad (49)$$

In Appendix A, we show that (49) reduces to

$$\psi^{i+1} = \frac{1}{M}\|\boldsymbol{y}-\boldsymbol{A}\widehat{\boldsymbol{x}}\|_2^2 + \frac{1}{M}\sum_{m=1}^{M}\sum_{n=1}^{N} a_{mn}^2\mu_n^x. \quad (50)$$

B. EM Update of Laplacian Rate Parameter

As in the AWGN case above, the incremental update of the Laplacian rate ψ from (48) becomes

$$\psi^{i+1} = \underset{\psi>0}{\arg\max} \sum_{m=1}^{M} \widehat{\mathrm{E}}\left\{\ln f_{\mathsf{y|z}}\left(y_m|\boldsymbol{a}_m^\mathsf{T}\boldsymbol{\mathsf{x}};\psi\right)|\boldsymbol{y};\psi^i\right\}, \quad (51)$$

but where now $f_{\mathsf{y|z}}$ is given by (18). In Appendix B-B, we show that (51) reduces to

$$\psi^{i+1} = M\left(\sum_{m=1}^{M} \widehat{\mathrm{E}}\left\{\left|\boldsymbol{a}_m^\mathsf{T}\boldsymbol{\mathsf{x}} - y_m\right|\,|\boldsymbol{y};\psi^i\right\}\right)^{-1} \quad (52)$$

where

$$\begin{aligned}\widehat{\mathrm{E}}\left\{\left|\boldsymbol{a}_m^\mathsf{T}\boldsymbol{\mathsf{x}}-y_m\right|\,|\boldsymbol{y};\psi^i\right\} \approx{}& \Phi_c\left(\frac{\widetilde{z}_m}{\mu_m^p}\right)\left(\widetilde{z}_m+\sqrt{\mu_m^p}h\left(\frac{-\widetilde{z}_m}{\sqrt{\mu_m^p}}\right)\right)\\ &-\Phi_c\left(\frac{-\widetilde{z}_m}{\mu_m^p}\right)\left(\widetilde{z}_m-\sqrt{\mu_m^p}h\left(\frac{\widetilde{z}_m}{\sqrt{\mu_m^p}}\right)\right) \quad (53)\end{aligned}$$

for $\widetilde{z}_m \triangleq \boldsymbol{a}_m^\mathsf{T}\widehat{\boldsymbol{x}} - y_m$, μ_m^p defined in line (R1) of Table I, and $h(\cdot)$ defined in (26).

C. EM Update of Exponential Rate Parameter

Noting that $f(\boldsymbol{x},\boldsymbol{y};\boldsymbol{q}) = D\prod_{n=1}^{N} f_{\mathsf{x}}(x_n;\chi)$ with χ-invariant D, the incremental EM update of the exponential rate parameter χ is

$$\begin{aligned}\chi^{i+1} &= \underset{\chi>0}{\arg\max}\sum_{n=1}^{N}\widehat{\mathrm{E}}\left\{\ln f_{\mathsf{x}}(\mathsf{x}_n;\chi)|\boldsymbol{y};\chi^i\right\}, \quad (54)\\ &= \underset{\chi>0}{\arg\max}\, N\log\chi - \chi\sum_{n=1}^{N}\widehat{\mathrm{E}}\left\{\mathsf{x}_n|\boldsymbol{y};\chi^i\right\} \quad (55)\end{aligned}$$

which, after zeroing the derivative of (55) w.r.t. χ, reduces to

$$\chi^{i+1} = N\left(\sum_{n=1}^{N}\widetilde{r}_n + \sqrt{\mu_n^r}h\left(-\frac{\widetilde{r}_n}{\sqrt{\mu_n^r}}\right)\right)^{-1} \quad (56)$$

for $\widetilde{r}_n \triangleq \widehat{r}_n - \chi\mu_n^r$, μ_n^r defined in line (R9) of Table I, and $h(\cdot)$ defined in (26). The derivation of (56) uses the fact that the posterior used for the expectation in (55) simplifies to $f_{\mathsf{x|r}}(x_n|\widehat{r}_n;\mu_n^r) = \mathcal{N}_+(x_n;\widetilde{r}_n,\mu_n^r)$. Note that this procedure, when used in conjunction with the AWGN variance learning procedure, automatically "tunes" the LASSO regularization parameter λ in (2), a difficult problem (see, e.g., [13]).

D. EM Updates for NNGM Parameters and Model-Order Selection

Noting that $f(\boldsymbol{x},\boldsymbol{y};\boldsymbol{q}) = D\prod_{n=1}^{N} f_{\mathsf{x}}(x_n;\boldsymbol{\omega},\boldsymbol{\theta},\boldsymbol{\phi})$ with $[\boldsymbol{\omega},\boldsymbol{\theta},\boldsymbol{\phi}]$-invariant D, the incremental EM updates become

$$\theta_k^{i+1} = \underset{\theta_k\in\mathbb{R}}{\arg\max}\sum_{n=1}^{N}\widehat{\mathrm{E}}\left\{\ln f_{\mathsf{x}}\left(\mathsf{x}_n;\theta_k,\boldsymbol{q}^i_{\backslash\theta_k}\right)|\boldsymbol{y};\boldsymbol{q}^i\right\}, \quad (57)$$

$$\phi_k^{i+1} = \underset{\phi_k>0}{\arg\max}\sum_{n=1}^{N}\widehat{\mathrm{E}}\left\{\ln f_{\mathsf{x}}\left(\mathsf{x}_n;\phi_k,\boldsymbol{q}^i_{\backslash\phi_k}\right)|\boldsymbol{y};\boldsymbol{q}^i\right\}, \quad (58)$$

$$\boldsymbol{\omega}^{i+1} = \underset{\boldsymbol{\omega}>0:\sum_k\omega_k=1}{\arg\max}\sum_{n=1}^{N}\widehat{\mathrm{E}}\left\{\ln f_{\mathsf{x}}\left(\mathsf{x}_n;\boldsymbol{\omega},\boldsymbol{q}^i_{\backslash\boldsymbol{\omega}}\right)|\boldsymbol{y};\boldsymbol{q}^i\right\}, \quad (59)$$

where we use "$\boldsymbol{q}^i_{\backslash\boldsymbol{\omega}}$" to denote the vector $\boldsymbol{q}^i$ with $\boldsymbol{\omega}$ components removed (and similar for $\boldsymbol{q}^i_{\backslash\theta_k}$ and $\boldsymbol{q}^i_{\backslash\phi_k}$). As derived in Appendix C-B, the updates above can be approximated as

$$\theta_k^{i+1} = \frac{\sum_{n=1}^{N}\pi_n\overline{\beta}_{n,k}\left(\gamma_{n,k}+\sqrt{\nu_{n,k}}h(\alpha_{n,k})\right)}{\sum_{n=1}^{N}\pi_n\overline{\beta}_{n,k}} \quad (60)$$

$$\phi_k^{i+1} = \frac{\sum_{n=1}^{N}\pi_n\overline{\beta}_{n,k}\left(\gamma_{n,k}+\sqrt{\nu_{n,k}}h(\alpha_{n,k})-\theta_k\right)^2}{\sum_{n=1}^{N}\pi_n\overline{\beta}_{n,k}} + \frac{\sum_{n=1}^{N}\pi_n\overline{\beta}_{n,k}\nu_{n,k}g(\alpha_{n,k})}{\sum_{n=1}^{N}\pi_n\overline{\beta}_{n,k}} \quad (61)$$

$$\omega_k^{i+1} = \frac{\sum_{n=1}^{N}\pi_n\overline{\beta}_{n,k}}{\sum_{n=1}^{N}\pi_n}, \quad (62)$$

where the quantities $\alpha_{n,\ell},\gamma_{n,\ell},\nu_{n,\ell},\beta_{n,\ell},\pi_n$ were defined in (41)–(45) and $\overline{\beta}_{n,k} \triangleq \beta_{n,k}/\sum_\ell\beta_{n,\ell}$. The EM update of the

NNGM sparsity rate τ (recall (36)) is identical to that for the GM sparsity rate derived in [24]:

$$\tau^{i+1} = \frac{1}{N}\sum_{n=1}^{Q} N\pi_n. \tag{63}$$

Since the quantities in (60)–(63) are already computed by NNGM-GAMP, the EM updates do not significantly increase the complexity beyond that of NNGM-GAMP itself.

The number of components L in the NNGM model (36) can be selected using the standard penalized log-likelihood approach to model-order-selection [34], i.e., by maximizing

$$\ln f(\boldsymbol{y}; \widehat{\boldsymbol{q}}_L) - \eta(L), \tag{64}$$

where $\widehat{\boldsymbol{q}}_L$ is the ML estimate of $\boldsymbol{q}$ under the hypothesis L (for which we would use the EM estimate) and $\eta(L)$ is a penalty term such as that given by the Bayesian information criterion (BIC). Since this model-order-selection procedure is identical to that proposed for EM-GM-GAMP in [24], we refer interested readers to [24] for more details. In practice, we find that the fixed choice of $L = 3$ performs sufficiently well (see Section VI).

E. EM Initialization

With EM, a good initialization is essential to avoiding bad local minima. For EM-NNL-GAMP, we suggest setting the initial exponential rate parameter $\chi^0 = 10^{-2}$, as this seems to perform well over a wide range of problems (see Section VI).

For EM-NNGM-GAMP, we suggest the initial sparsity rate

$$\tau^0 = \min\left\{\frac{M}{N}\rho_{\text{SE}}\left(\frac{M}{N}\right), 1-\epsilon\right\} \tag{65}$$

where $\epsilon > 0$ is set arbitrarily small and $\rho_{\text{SE}}(\cdot)$ is the theoretical noiseless phase-transition-curve (PTC) for ℓ_1 recovery of sparse non-negative signals, shown in [28] to have the closed-form expression

$$\rho_{\mathsf{SE}}(\delta) = \max_{c\geq 0} \frac{1 - \left(\frac{1}{\delta}\right)\left[(1+c^2)\Phi(-c) - c\varphi(c)\right]}{1 + c^2 - \left[(1+c^2)\Phi(-c) - c\varphi(c)\right]} \tag{66}$$

where $\Phi(\cdot)$ and $\varphi(\cdot)$ denote the cdf and pdf of the standard normal distribution. We then propose to set the initial values of the NNGM weights $\{\omega_\ell\}$, locations $\{\theta_\ell\}$, and scales $\{\phi_\ell\}$ at the values that best fit the uniform pdf on $[0, \sqrt{3\varphi^0}]$, which can be computed offline similar to the standard EM-based approach described in [35, p. 435]. Under the AWGN model (15), we propose to set the initial variance of the noise and signal, respectively, as

$$\psi^0 = \frac{\|\boldsymbol{y}\|_2^2}{(\mathsf{SNR}+1)M}, \; \varphi^0 = \frac{\|\boldsymbol{y}\|_2^2 - M\psi^0}{\|\boldsymbol{A}\|_F^2\tau^0}, \tag{67}$$

where, without knowledge of the true $\mathsf{SNR} \triangleq \|\boldsymbol{Ax}\|_2^2/\|\boldsymbol{w}\|_2^2$, we suggest using the value $\mathsf{SNR} = 100$. Meanwhile, under the i.i.d Laplacian noise model (18), we suggest to initialize the rate as $\psi^0 = 1$ and φ^0 again as in (67).

TABLE II
NNLS-GAMP VS. LSQLIN: AVERAGE COMPARATIVE NMSE [dB] AND RUNTIME [SEC] FOR SIMPLEX SIGNAL RECOVERY

		$N = 100$			$N = 250$			$N = 500$		
			time			time			time	
		$\overline{\mathsf{NMSE}}$	NNLS-GAMP	`lsqlin`	$\overline{\mathsf{NMSE}}$	NNLS-GAMP	`lsqlin`	$\overline{\mathsf{NMSE}}$	NNLS-GAMP	`lsqlin`
SNR	10	-161.8	0.068	**0.050**	-161.8	**0.080**	0.550	-161.8	**0.159**	5.414
	100	-161.7	0.069	**0.021**	-154.3	**0.080**	0.205	-161.5	**0.154**	1.497
	1000	-162.1	0.068	**0.011**	-161.7	0.079	**0.074**	-161.5	**0.151**	0.504

VI. NUMERICAL RESULTS

The subsections below describe numerical experiments used to ascertain the performance of the proposed methods[3] to existing methods for non-negative signal recovery.

A. Validation of NNLS-GAMP and NNL-GAMP

We first examine the performance of our proposed algorithms on the linearly constrained NNLS problem (2) with $\lambda = 0$. In particular, we compare the performance of NNLS-GAMP to Matlab's solver `lsqlin`. To do this, we drew realizations of K-sparse simplex $\boldsymbol{x} \in \Delta_+^N$, where the nonzero elements $\{\underline{x}_k\}_{k=1}^K$ were placed uniformly at random and drawn from a symmetric Dirichlet distribution with concentration a, i.e.,

$$f(\underline{x}_1, \ldots, \underline{x}_{K-1}) = \begin{cases} \frac{\Gamma(aK)}{\Gamma(a)^K}\prod_{k=1}^K \underline{x}_k^{a-1}, & \underline{x}_k \in [0,1] \\ 0 & \text{else} \end{cases} \tag{68a}$$

$$f(\underline{x}_K | \underline{x}_1, \ldots, \underline{x}_{K-1}) = \delta(1 - \underline{x}_1 - \ldots - \underline{x}_K), \tag{68b}$$

where $\Gamma(\cdot)$ is the gamma function. For this first experiment, we used $a = 1$, in which case $\{\underline{x}_k\}_{k=1}^{K-1}$ are i.i.d uniform on [0,1], as well as $K = N$ (i.e., no sparsity). We then constructed noisy measurements $\boldsymbol{y} \in \mathbb{R}^M$ according to (1) using $\boldsymbol{A}$ with i.i.d $\mathcal{N}(0, M^{-1})$ entries, $\mathsf{SNR} \triangleq \|\boldsymbol{Ax}\|_2^2/\|\boldsymbol{w}\|_2^2 = [10, 100, 1000]$, and sampling ratio $M/N = 3$. Table II reports the resulting *comparative* $\overline{\mathsf{NMSE}} \triangleq \|\widehat{\boldsymbol{x}}_{\mathsf{NNLS-GAMP}} - \widehat{\boldsymbol{x}}_{\mathtt{lsqlin}}\|_2^2/\|\boldsymbol{x}\|_2^2$ and runtime averaged over $R = 100$ realizations for signal lengths $N = [100, 250, 500]$. From the table, we see that NNLS-GAMP and `lsqlin` return identical solutions (up to algorithmic tolerance[4]), but that NNLS-GAMP's runtime scales like $O(N^2)$ while `lsqlin`'s scales like $O(N^3)$, making NNLS-GAMP much faster for larger problem dimensions N. Moreover, we see that NNLS-GAMP's runtime is invariant to SNR, whereas `lsqlin`'s runtime quickly degrades as the SNR decreases.

Next, we examine the performance of our proposed algorithms on the non-negative LASSO problem (2) with $\lambda > 0$. In particular, we compare NNL-GAMP to TFOCS[5] [22]. For this, K-sparse non-negative $\boldsymbol{x}$ and noisy observations $\boldsymbol{y}$ were constructed as before, but now with $M = 1000, N = 500, K < N$, and $\mathsf{SNR} = 20$ dB. Table III shows the runtimes and comparative $\overline{\text{NMSE}}$ between NNL-GAMP and TFOCS for various combinations of sparsity K and regularization weight λ. Table III shows that the solutions returned by the two algorithms were identical (up to algorithmic tolerance) but that NNL-GAMP ran about 4 to 8 times faster than TFOCS.

[3] We implemented the proposed algorithms using the GAMPmatlab [36] package available at http://sourceforge.net/projects/gampmatlab/.

[4] The algorithms under test include user-adjustable stopping tolerances. As these tolerances are decreased, we observe that the comparative $\overline{\text{NMSE}}$ also decreases, at least down to Matlab's numerical precision limit.

[5] We used Matlab code from http://cvxr.com/tfocs/download/.

TABLE III
NNL-GAMP VS. TFOCS: AVERAGE COMPARATIVE $\overline{\mathrm{NMSE}}$ [dB] AND RUNTIME [SEC] FOR K-SPARSE NON-NEGATIVE SIGNAL RECOVERY

		$K = 50$			$K = 100$			$K = 150$		
			time			time			time	
		$\overline{\mathrm{NMSE}}$	NNL-GAMP	TFOCS	$\overline{\mathrm{NMSE}}$	NNL-GAMP	TFOCS	$\overline{\mathrm{NMSE}}$	NNL-GAMP	TFOCS
λ	0.01	-135.7	**0.024**	0.091	-139.9	**0.025**	0.119	-140.8	**0.025**	0.104
	0.001	-125.4	**0.026**	0.130	-122.9	**0.026**	0.148	-117.0	**0.027**	0.175
	0.0001	-113.2	**0.035**	0.256	-113.4	**0.036**	0.262	-112.4	**0.036**	0.292

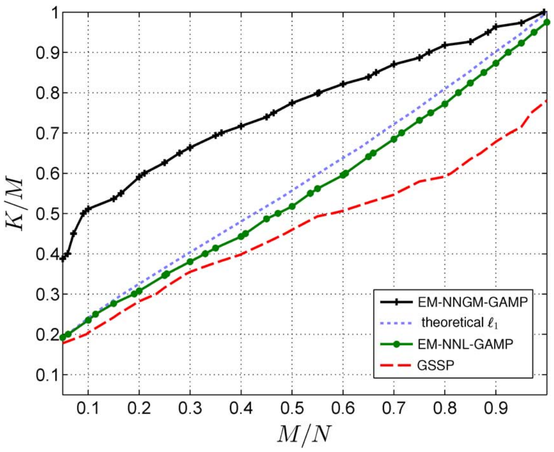


Fig. 1. Empirical PTCs and ℓ_1-SNN theoretical PTC for noiseless recovery of length-$N = 500$, K-sparse, simplex signals with Dirichlet concentration $a = 1$ from M measurements.

B. Noiseless Empirical Phase Transitions

It has been established (see, e.g., [28]) that, for the recovery of a non-negative K-sparse signal $\boldsymbol{x} \in \mathbb{R}^N$ from noiseless observations $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} \in \mathbb{R}^M$, there exists a sharp phase-transition separating problem sizes (M, N, K) that are perfectly solvable (with very high probability) from those that are not. The precise location of the phase-transition curve (PTC) differs among algorithms, presenting an avenue for comparison.

Below, we present empirical PTCs for the recovery of K-sparse N-length simplex signals from M noiseless measurements. To compute each PTC, we fixed $N = 500$ and constructed a 20×20 uniformly spaced grid on the $\frac{M}{N}$-versus-$\frac{K}{M}$ plane for $\frac{M}{N} \in [0.05, 1]$ and $\frac{K}{M} \in [0.05, 1]$. At each grid point, we drew $R = 100$ independent realizations of the pair $(\boldsymbol{A}, \boldsymbol{x})$, where $\boldsymbol{A}$ was drawn from i.i.d $\mathcal{N}(0, M^{-1})$ entries and $\boldsymbol{x} \in \mathbb{R}^N$ had K nonzero elements $\{\underline{x}_k\}_{k=1}^K$ (placed uniformly at random) drawn from a symmetric Dirichlet distribution (68) with concentration parameter a. For the rth realization of $(\boldsymbol{A}, \boldsymbol{x})$, we attempted to recover non-negative sparse $\boldsymbol{x}$ from the augmented observations $\left[\begin{smallmatrix}\boldsymbol{y}\\ 1\end{smallmatrix}\right] = \left[\begin{smallmatrix}\boldsymbol{A}\\ \boldsymbol{1}^\top\end{smallmatrix}\right]\boldsymbol{x}$, which implicitly enforce the simplex constraint. The resulting recovery $\widehat{\boldsymbol{x}}$ was considered to be "successful" if $\mathsf{NMSE} \triangleq \|\boldsymbol{x} - \widehat{\boldsymbol{x}}\|_2^2 / \|\boldsymbol{x}\|_2^2 < 10^{-6}$. Using $S_r = 1$ to record a success and $S_r = 0$ a failure, the average success rate was then computed as $\overline{S} \triangleq \frac{1}{R}\sum_{r=1}^R S_r$, and the corresponding empirical PTC was plotted as the $\overline{S} = 0.5$ level-curve using Matlab's `contour` command.

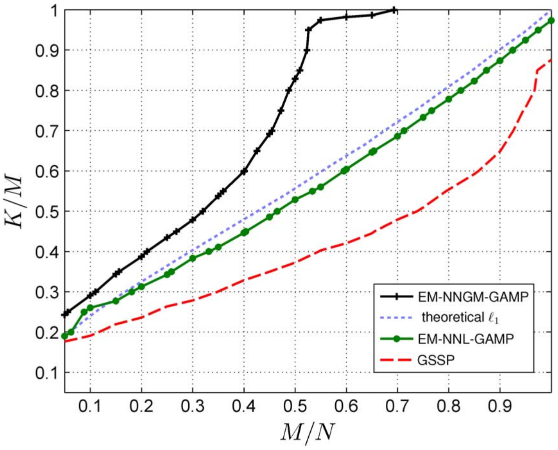


Fig. 2. Empirical PTCs and ℓ_1-SNN theoretical PTC for noiseless recovery of length-$N = 500$, K-sparse, simplex signals with Dirichlet concentration $a = 100$ from M measurements.

Figs. 1 and 2 show the empirical PTCs under the Dirichlet concentration $a = 1$ (i.e., i.i.d uniform $\{\underline{x}_k\}_{k=1}^{K-1}$) and $a = 100$ (i.e., $\underline{x}_k \approx \frac{1}{K} \forall k$), respectively, for our proposed EM-tuned NNGM-GAMP and NNL-GAMP algorithms, in comparison to the GSSP[6] approach (3) proposed in [6]. We did not consider NNLS-GAMP and `lsqlin` because, for $\boldsymbol{A}$ drawn i.i.d Gaussian, the solution to the non-negative LS problem "$\arg\min_{\boldsymbol{x} \geq \boldsymbol{0}} \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|_2^2$" is not guaranteed to be unique when $M < N$ [14, Thm. 1], which is the setting considered here. Figs. 1 and 2 also show $\rho_{\mathsf{SE}}\left(\frac{M}{N}\right)$ from (66), i.e., the theoretical large-system-limit PTC for ℓ_1-based recovery of sparse non-negative (SNN) signals.

Looking at Figs. 1 and 2, we see that the empirical PTCs of EM-NNL-GAMP are close to the theoretical ℓ_1 PTC, as expected, and significantly better than those of GSSP. More striking is the far superior PTCs of EM-NNGM-GAMP. We attribute EM-NNGM-GAMP's success to three factors: i) the generality of the NNGM prior (36), ii) the ability of the proposed EM approach to accurately learn the prior parameters, and iii) the ability of sum-product GAMP to exploit the learned prior. In fact, Fig. 2 shows EM-NNGM-GAMP reliably reconstructing K-sparse signals from only $M = K$ measurements in the compressive (i.e., $M < N$) regime.

C. Sparse Non-Negative Compressive Imaging

As a practical example, we experimented with the recovery of a sparse non-negative image. For this, we used the $N = 256 \times 256$ satellite image shown on the left of Fig. 3, which contained $K = 6678$ nonzero pixels and $N - K = 58858$ zero-valued pixels, and thus was approximately 10% sparse. Measurements $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{w} \in \mathbb{R}^M$ were collected under i.i.d Gaussian noise $\boldsymbol{w}$ whose variance was selected to achieve an $\mathsf{SNR} = 60$ dB. Here, $\boldsymbol{x}$ represents the (rasterized) image and $\boldsymbol{A}$ a linear measurement operator configured as $\boldsymbol{A} = \boldsymbol{\Phi}\boldsymbol{\Psi}\boldsymbol{S}$, where $\boldsymbol{\Phi} \in \{0, 1\}^{M \times N}$ was constructed from rows of the $N \times N$ identity matrix selected uniformly at random, $\boldsymbol{\Psi} \in \{-1, 1\}^{N \times N}$ was a Hadamard transform, and $\boldsymbol{S} \in \mathbb{R}^{N \times N}$ was a diagonal matrix with ± 1 diagonal entries chosen uniformly at random. Note

[6] For GSSP, we used code provided by its authors, but found that its performance was greatly enhanced by initializing the algorithm at the Basis Pursuit solution (as computed by SPGL1 [37]) and using the stepsize $100/\|\boldsymbol{A}\|_F^2$.

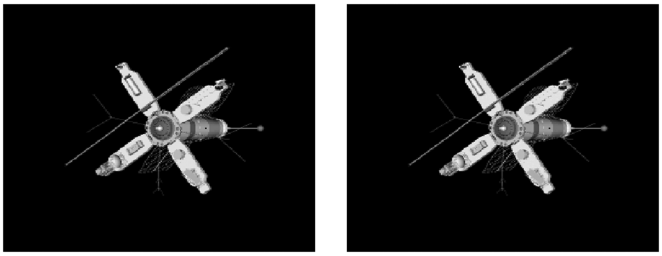


Fig. 3. Sparse non-negative image of a satellite: original image on left and EM-NNGM-GAMP recovery at $\frac{M}{N}=\frac{1}{4}$ on right.

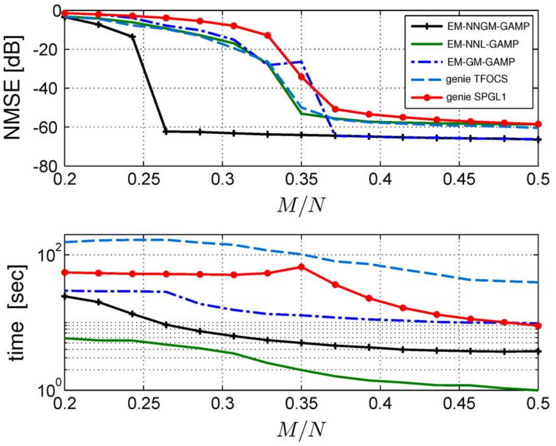


Fig. 4. Recovery NMSE (top) and runtime (bottom) versus $\frac{M}{N}$ for the sparse NN satellite image for the proposed EM-NNGM-GAMP and EM-NNL-GAMP compared to EM-GM-GAMP, non-negative LASSO via oracle-tuned TFOCS, and standard LASSO via oracle-tuned SPGL1.

that multiplication by $\boldsymbol{A}$ can be executed using a fast binary algorithm, making it attractive for hardware implementation. For this experiment, no linear equality constraints exist and so the observation model was not augmented as in (11).

As a function of the sampling ratio $\frac{M}{N}$, Fig. 4 shows the NMSE and runtime averaged over $R = 100$ realizations of $\boldsymbol{A}$ and $\boldsymbol{w}$ for the proposed EM-NNGM-GAMP and EM-NNL-GAMP in comparison to EM-GM-GAMP from [24], genie-tuned non-negative LASSO via TFOCS [22],[7] and genie-tuned standard LASSO implemented via SPGL1[8] [37]. NNLS methods were not considered because of the non-uniqueness of their solutions in the $M < N$ regime (recall [14, Thm. 1]).

Fig. 4 shows that the proposed EM-NNGM-GAMP algorithm provided the most accurate signal recoveries for all undersampling ratios. Remarkably, its phase-transition occurred at $\frac{M}{N} \approx 0.25$, whereas that of the other algorithms occurred at $\frac{M}{N} \approx 0.35$. The gain of EM-NNGM-GAMP over EM-GM-GAMP can be attributed to the former's exploitation of signal non-negativity, whereas the gain of EM-NNGM-GAMP over non-negative LASSO (either via EM-NNL-GAMP or genie-tuned TFOCS) can be attributed to former's learning/exploitation of the true signal distribution. Finally, the gain of non-negative LASSO over standard LASSO can be attributed to the former's exploitation of signal non-negativity.

Fig. 4 also demonstrates that the LASSO tuning procedure proposed in Section V works very well: the NMSE of EM-NNL-GAMP is nearly identical to that of oracle-tuned TFOCS for all sampling ratios M/N.

Finally, Fig. 4 shows that EM-NNGM-GAMP was about 3 times as fast as EM-GM-GAMP, between 3 to 15 times as fast as SPGL1 (implementing standard LASSO), and between 10 to 20 times as fast as TFOCS (implementing non-negative LASSO). The proposed EM-NNL-GAMP was about 2 to 4 faster than EM-NNGM-GAMP, although it did not perform as well in terms of NMSE.

D. Portfolio Optimization

As another practical example, we consider portfolio optimization under the return-adjusted Markowitz mean-variances (MV) framework [3]: if $\boldsymbol{x} \in \Delta_+^N$ is a portfolio and $\boldsymbol{r}_{M+1} \in \mathbb{R}^N$ is a random vector that models the returns of N commodities at the future time $M+1$, then we desire to design $\boldsymbol{x}$ so that the future sum-return $\boldsymbol{r}_{M+1}^\top \boldsymbol{x}$ has relatively high mean and low variance. Although $\boldsymbol{r}_{M+1}$ is unknown at design time, we assume knowledge of the past M returns $\boldsymbol{A} \triangleq [\boldsymbol{r}_1, \ldots, \boldsymbol{r}_M]^\top$, which can be time-averaged to yield $\boldsymbol{\mu} \triangleq \frac{1}{M}\sum_{m=1}^M \boldsymbol{r}_m = \frac{1}{M}\boldsymbol{A}^\top \boldsymbol{1}$, and then (assuming stationarity) design $\boldsymbol{x}$ that minimizes the variance around a target sum-return of ρ, i.e.,

$$\widehat{\boldsymbol{x}} = \underset{\boldsymbol{x}\in\Delta_+^N}{\arg\min} \|\boldsymbol{1}\rho - \boldsymbol{A}\boldsymbol{x}\|_2^2 + \lambda\|\boldsymbol{x}\|_1 \text{ s.t. } \boldsymbol{\mu}^\top \boldsymbol{x} = \rho. \tag{69}$$

In (69), the use of sparsity promoting ℓ_1 regularization [4] aims to help the portfolio designed from past data $\{\boldsymbol{r}_m\}_{m=1}^M$ generalize to the future data $\boldsymbol{r}_{M+1}$. Without ℓ_1 regularization, the solutions to (69) are often outperformed by the "naïve" portfolio $\boldsymbol{x}_{\mathsf{naive}} \triangleq \frac{1}{N}\boldsymbol{1}$ in practice [38].

Noting that (69) is a special case of (2), MV portfolio optimization is a natural application for the algorithms developed in this paper. We thus tested our proposed algorithms against[9] `lsqlin` and cross-validated (CV)[10] TFOCS using the FF49 portfolio database,[11] which consists of monthly returns for $N = 49$ securities from July 1971 (i.e., $\boldsymbol{r}_1$) to July 2011 (i.e., $\boldsymbol{r}_{481}$). In particular, starting from July 1981 and moving forward in yearly increments, we collected the past $M = 120$ months of return data in $\boldsymbol{A}(i) \triangleq [\boldsymbol{r}_{12(i-1)+1}, \ldots, \boldsymbol{r}_{12(i-1)+M}]^\top$ and computed the corresponding time-average return $\boldsymbol{\mu}(i) \triangleq \frac{1}{M}\boldsymbol{A}(i)^\top \boldsymbol{1}$, where $i \in \{1, \ldots, 30\}$ indexed the years from 1981 to 2010. Then, we chose the target sum-return $\rho(i)$ to be that of the naïve scheme, i.e., $\rho(i) = \frac{1}{N}\boldsymbol{\mu}(i)^\top \boldsymbol{1}$, and computed the portfolio $\widehat{\boldsymbol{x}}(i)$

[7] Using EM-NNL-GAMP's $\widehat{\boldsymbol{x}}$, we ran TFOCS over an 11-point grid of hypothesized non-negative ℓ_1 penalty $\lambda \in \{0.5\|\boldsymbol{A}^\top(\boldsymbol{y} - \boldsymbol{A}\widehat{\boldsymbol{x}})\|_\infty, \ldots, 2\|\boldsymbol{A}^\top(\boldsymbol{y} - \boldsymbol{A}\widehat{\boldsymbol{x}})\|_\infty\}$ and then reported the total runtime and bestNMSE.

[8] We ran SPGL1 in "BPDN mode," i.e., solving $\min_{\boldsymbol{x}} \|\boldsymbol{x}\|_1$ s.t. $\|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|_2 < \sigma$ for hypothesized tolerances $\sigma^2 \in \{0.3, 0.6, \ldots, 1.5\} \times M\psi$ and then reported the total runtime and best NMSE.

[9] We were not able to configure GSSP in a way that maintained $\boldsymbol{\mu}^\top \widehat{\boldsymbol{x}} = \rho$, even approximately, after the simplex projection step in (3).

[10] For CV-TFOCS, we used 4-fold cross-validation to tune λ over a 15-point grid between 0.001 and 0.1.

[11] The FF49 database and other financial datasets can be obtained from http://mba.tuck.dartmouth.edu/pages/faculty/ken.french/data_library.html.

TABLE IV
AVERAGE SHARPE RATIO SR, CONSTRAINT ERROR $\mathcal{E}$ (IN dB), AND RUNTIME (IN sec) VERSUS ALGORITHM FOR THE FF49 DATASET

		SR	time (sec)	$\mathcal{E}$ (dB)
	naïve	0.3135	-	-∞
	lsqlin	0.3725	**0.06**	-307.4
	CV-TFOCS	0.3747	31.92	-56.9
AWGN	NNLS-GAMP	0.3724	0.68	-72.0
	EM-NNL-GAMP	0.3725	1.48	-60.9
	EM-NNGM-GAMP	0.3900	6.98	-41.5
AWLN	NNLS-GAMP	0.3818	1.80	-56.1
	EM-NNL-GAMP	0.3829	5.14	-43.2
	EM-NNGM-GAMP	**0.3995**	2.95	-42.3

from $\{\boldsymbol{A}(i), \boldsymbol{\mu}(i), \rho(i)\}$ for each algorithm under test. The resulting $\widehat{\boldsymbol{x}}(i)$ was evaluated on the *future* $T = 12$ months of return data using the Sharpe ratio $\mathsf{SR}(i) \triangleq \widehat{\rho}(i)/\widehat{\sigma}(i)$, where

$$\widehat{\rho}(i) \triangleq \frac{1}{T}\sum_{t=1}^{T} \boldsymbol{r}_{12(i-1)+M+t}^{\mathsf{T}}\widehat{\boldsymbol{x}}(i), \tag{70}$$

$$\widehat{\sigma}^2(i) \triangleq \frac{1}{T}\sum_{t=1}^{T} \left(\boldsymbol{r}_{12(i-1)+M+t}^{\mathsf{T}}\widehat{\boldsymbol{x}}(i) - \widehat{\rho}(i)\right)^2, \tag{71}$$

For lsqlin, the constraints were specified directly. For NNLS-GAMP, EM-NNL-GAMP, and EM-NNGM-GAMP, the constraints were enforced using (11) with $\boldsymbol{B} = [\boldsymbol{\mu}, \mathbf{1}]^{\mathsf{T}}$ and $\boldsymbol{c} = [\rho, 1]^{\mathsf{T}}$, and for CV-TFOCS, the constraints were enforced using the augmentation

$$\overline{y} \triangleq \begin{bmatrix} \rho(i)\mathbf{1} \\ 500\rho(i) \\ 500 \end{bmatrix} \text{ and } \overline{A} = \begin{bmatrix} \boldsymbol{A}(i) \\ 500\boldsymbol{\mu}(i)^{\mathsf{T}} \\ 500\ \mathbf{1}^{\mathsf{T}} \end{bmatrix}, \tag{72}$$

where the gain of 500 helped to weight the constraints above the loss. Lastly, we tried our GAMP-based approaches using both the AWGN likelihood (15) as well as the AWLN likelihood (18).

Table IV reports the average Sharpe ratios $\mathsf{SR} \triangleq \frac{1}{30}\sum_{i=1}^{30}\mathsf{SR}(i)$ and runtimes for each algorithm under test. In addition, it reports the average squared constraint error $\mathcal{E} \triangleq \frac{1}{30}\sum_{i=1}^{30}|\boldsymbol{\mu}(i)^{\mathsf{T}}\widehat{\boldsymbol{x}}(i) - \rho(i)|^2$, showing that all algorithms near-perfectly met the target sum-return constraint $\boldsymbol{\mu}(i)^{\mathsf{T}}\widehat{\boldsymbol{x}}(i) = \rho(i)$. The table shows that Matlab's lsqlin and AWGN NNLS-GAMP (which solve the same NNLS problem) yielded identical Sharpe ratios, which were ≈19% larger than the naïve value. Meanwhile, CV-TFOCS and AWGN EM-NNL-GAMP (which solve the same NN LASSO problem) yielded very similar Sharpe ratios, also ≈19% larger than the naïve value. As in previous experiments, AWGN EM-NNGM-GAMP outperformed both NNLS and NN LASSO, in this case improving on the naïve Sharpe ratio by 24%. The table also shows that the use of an AWLN likelihood (robust to outliers [23]) resulted in across-the-board improvements in Sharpe ratio. Among the algorithms under test, AWLN EM-NNGM-GAMP yielded the best performance, improving the naïve Sharpe ratio by 27%.

In terms of runtimes, Matlab's lsqlin was by far the fastest algorithm, CV-TFOCS was by far the slowest, and the AMP approaches were in-between. NNLS-GAMP and NNL-GAMP were slower here than in Table II and Table III because the matrix $\boldsymbol{A}$ in this financial experiment had correlated columns and thus required the use of a stronger damping factor in the GAMP-matlab implementation [36].

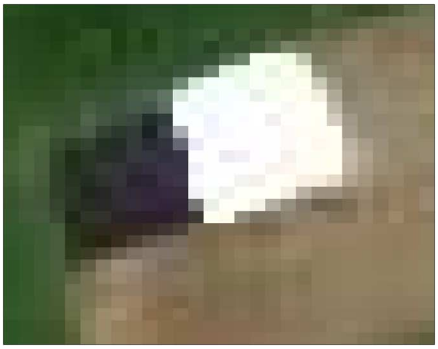


Fig. 5. RGB image of the cropped scene of the SHARE 2012 dataset [40].

E. Hyperspectral Image Inversion

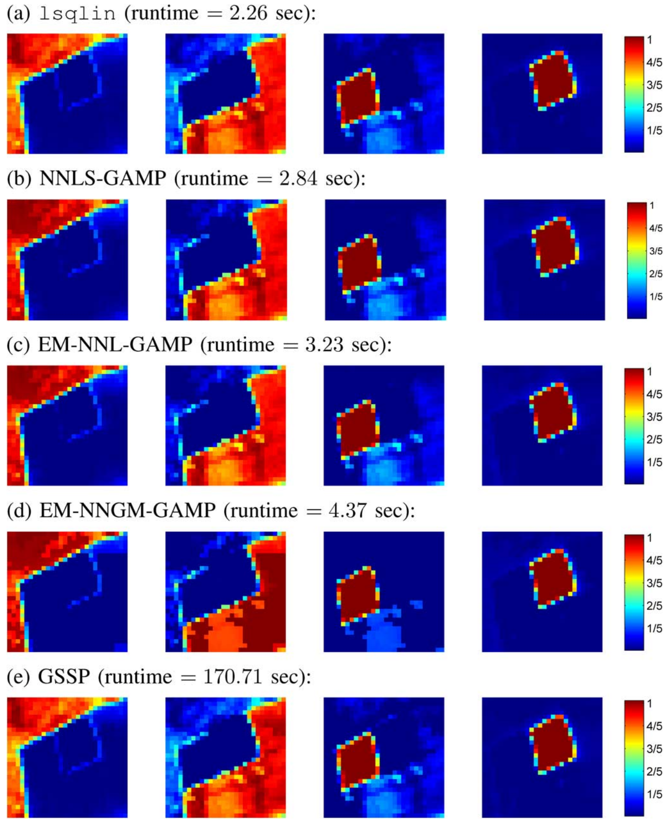
As a final practical example, we consider hyperspectral image inversion [2]. A hyperspectral image is like a color image, but instead of 3 spectral bands (red, green, and blue) it contains $M \gg 3$ spectral bands. With $T = T_1 \times T_2$ spatial pixels, such an image can be represented by a matrix $\boldsymbol{Y} \in \mathbb{R}^{M\times T}$ and, under the macroscopic model, "unmixed" into

$$\boldsymbol{Y} = \boldsymbol{A}\boldsymbol{X} + \boldsymbol{W} \tag{73}$$

where the nth column in $\boldsymbol{A} \in \mathbb{R}^{M\times N}$ is the spectral signature (or "endmember") of the nth material present in the scene, the nth row in $\boldsymbol{X} \in \mathbb{R}_{\geq 0}^{N\times T}$ is the spatial abundance of that material, and $\boldsymbol{W}$ is additive noise. The tth column of $\boldsymbol{X}$, henceforth denoted as $\boldsymbol{x}_t$, describes the distribution of materials within the tth pixel, and so for a valid distribution we need $\boldsymbol{x}_t \in \Delta_+^N$. We will assume that the endmembers $\boldsymbol{A}$ have been extracted from $\boldsymbol{Y}$ (e.g., via the well known VCA algorithm [39]) and therefore focus on image inversion, where the goal is to estimate $\boldsymbol{X}$ in (73) given $\boldsymbol{Y}$ and $\boldsymbol{A}$. In particular, the goal is to estimate a (possibly sparse) simplex-constrained $\boldsymbol{x}_t$ from the observation $\boldsymbol{y}_t = \boldsymbol{A}\boldsymbol{x}_t + \boldsymbol{w}_t$ at each pixel t.

We evaluated algorithm performance using the SHARE 2012 Avon dataset[12] [40], which uses $M = 360$ spectral bands, corresponding to wavelengths between 400 and 2450 nm, over a large rural scene. To do this, we first cropped down to the scene shown in Fig. 5, known to consist primarily of pure grass, dry sand, black felt, and white TyVek [41]. We then extracted the endmembers $\boldsymbol{A}$ from $\boldsymbol{Y}$ using VCA. Finally, we estimated the simplex-constrained columns of $\boldsymbol{X}$ from $(\boldsymbol{Y}, \boldsymbol{A})$ using NNLS-GAMP, EM-NNL-GAMP, EM-NNGM-GAMP, lsqlin (known in the hyperspectral literature as "fully constrained least squares" [42]), and GSSP. For both EM-NNL-GAMP and EM-NNGM-GAMP, we opted to learn the prior parameters separately for each *row* of $\boldsymbol{X}$, since the marginal distributions can be expected to differ across materials. For GSSP, we assumed that each pixel was at most $K = 3$-sparse and used a step size of $3/\|\boldsymbol{A}\|_F^2$, as these choices seemed to yield the best results.

[12] The SHARE 2012 Avon dataset can be obtained from http://www.rit.edu/cos/share2012/.


(a) `lsqlin` (runtime $= 2.26$ sec):

(b) NNLS-GAMP (runtime $= 2.84$ sec):

(c) EM-NNL-GAMP (runtime $= 3.23$ sec):

(d) EM-NNGM-GAMP (runtime $= 4.37$ sec):

(e) GSSP (runtime $= 170.71$ sec):

Fig. 6. Each row shows the $N = 4$ abundance maps estimated by a given algorithm. From left to right, the materials are: grass, dry sand, black felt, and white TyVek. Fig. 5 shows the RGB image of the same scene.

Since we have no knowledge of the true abundances $\boldsymbol{X}$, we are unable to present quantitative results on estimation accuracy. However, a qualitative comparison is made possible using the fact that most pixels in this scene are known to be pure [40] (i.e., contain only one material). In particular, each row of Fig. 6 shows the $N = 4$ abundance maps recovered by a given algorithm, and we see that all recoveries are nearly pure. However, the recoveries of EM-NNGM-GAMP are the most pure, as evident from the deep blue regions in the first and third columns of Fig. 6, as well as the deep red regions in the first and second columns. In terms of runtime, GSSP was by far the slowest algorithm, whereas all the other algorithms were similar (with `lsqlin` beating the others by a small margin).

VII. Conclusions

The problem of recovering a linearly constrained non-negative sparse signal $\boldsymbol{x}$ from noisy linear measurements $\boldsymbol{y}$ arises in many applications. One approach is to pose a sparsity-inducing convex optimization problem like (2) and then apply standard solvers like `lsqlin` (when $\lambda = 0$) or TFOCS (when $\lambda > 0$), although doing so requires also solving the non-trivial problem of optimizing λ [13]. Another approach is to solve for the MMSE estimate of $\boldsymbol{x}$, but doing so is made difficult by the need to estimate the prior distribution of $\boldsymbol{x}$ and then compute the resulting posterior mean.

In this paper, we proposed new solvers for (2) based on the min-sum AMP methodology, yielding NNLS-GAMP (for $\lambda = 0$) and NNL-GAMP (for $\lambda > 0$), and we demonstrated computational advantages relative to standard solvers in the large-N regime. In addition, we proposed a novel EM-based approach to optimizing λ that, in our empirical experiments, worked nearly as well as cross-validation and oracle methods. Moreover, we proposed a new approximate-MMSE estimation scheme that models $\boldsymbol{x}$ using an i.i.d Bernoulli non-negative Gaussian-mixture, learns the distributional parameters via the EM algorithm, and exploits the learned distribution via sum-product AMP. In all of our experiments, the resulting EM-NNGM-GAMP algorithm yielded superior performance while maintaining a reasonable computational efficiency. Finally, for problems where the noise may be non-Gaussian, we developed Laplacian likelihood models for both min-sum and sum-product GAMP, in addition to EM-tuning procedures, and demonstrated performance gains on practical datasets.

Appendix A
EM Update for AWGN Variance

Inserting the Gaussian likelihood (15) into (49), we see that the EM update for the noise variance ψ becomes

$$\psi^{i+1} = \arg\max_{\psi} \frac{M}{2}\ln\frac{1}{\psi} - \frac{1}{2\psi}\widehat{\mathrm{E}}\left\{\|\boldsymbol{y}-\boldsymbol{A}\mathbf{x}\|_2^2|\boldsymbol{y};\psi^i\right\}, \tag{74}$$

where, for the joint posterior $f_{\mathbf{x}|\mathbf{y}}(\boldsymbol{x}|\boldsymbol{y};\psi^i)$, we use the product of the approximate marginal GAMP posteriors from (10). By zeroing the derivative of the objective in (74) w.r.t. ψ, we find that

$$\psi^{i+1} = \frac{1}{M}\widehat{\mathrm{E}}\left\{\|\boldsymbol{y}-\boldsymbol{A}\mathbf{x}\|_2^2|\boldsymbol{y};\psi^i\right\}, \tag{75}$$

where the expectation simplifies to

$$\begin{aligned}
&\widehat{\mathrm{E}}\left\{\|\boldsymbol{y}-\boldsymbol{A}\mathbf{x}\|_2^2|\boldsymbol{y};\psi^i\right\}\\
&\quad= \boldsymbol{y}^{\mathsf{T}}\boldsymbol{y} - \boldsymbol{y}^{\mathsf{T}}\boldsymbol{A}\widehat{\boldsymbol{x}} + \widehat{\mathrm{E}}\left\{\mathbf{x}^{\mathsf{T}}\boldsymbol{A}^{\mathsf{T}}\boldsymbol{A}\boldsymbol{x}|\boldsymbol{y};\psi^i\right\} && (76)\\
&\quad= \boldsymbol{y}^{\mathsf{T}}\boldsymbol{y} - \boldsymbol{y}^{\mathsf{T}}\boldsymbol{A}\widehat{\boldsymbol{x}} + \mathrm{tr}\left(\boldsymbol{A}^{\mathsf{T}}\boldsymbol{A}\boldsymbol{\Sigma}\right) + \widehat{\boldsymbol{x}}^{\mathsf{T}}\boldsymbol{A}^{\mathsf{T}}\boldsymbol{A}\widehat{\boldsymbol{x}} && (77)\\
&\quad= \|\boldsymbol{y}-\boldsymbol{A}\widehat{\boldsymbol{x}}\|_2^2 + \mathrm{tr}(\boldsymbol{A}^{\mathsf{T}}\boldsymbol{A}\boldsymbol{\Sigma}). && (78)
\end{aligned}$$

Here, $\boldsymbol{\Sigma}$ is the posterior covariance matrix of $\mathbf{x}$, which—based on our assumptions—is diagonal with $[\boldsymbol{\Sigma}]_{nn} = \mu_n^x$. Plugging in (78) into (75), we obtain the EM update (50).

Appendix B
Derivation of Laplacian Likelihood Quantities

A. Laplacian Likelihood Steps for Sum-Product GAMP

Inserting the Laplacian likelihood (18) into the GAMP-approximated posterior (9), the posterior mean in line (R5) of Table I becomes (removing the m subscript for brevity)

$$\widehat{z} \triangleq \mathrm{E}\{\mathsf{z}|\mathsf{p}=\widehat{p};\mu^p\} = \frac{1}{C}\int_z z\,\mathcal{L}(z;y;\psi)\mathcal{N}(z;\widehat{p},\mu^p) \tag{79}$$

where the scaling constant C is calculated as

$$C = \int_z \mathcal{L}(z; y, \psi)\mathcal{N}(z; \widehat{p}, \mu^p) \tag{80}$$

$$= \int_{z'} \mathcal{L}(z'; 0, \psi)\mathcal{N}(z'; \widehat{p} - y, \mu^p) \tag{81}$$

$$= \underbrace{\frac{\psi}{2}\int_{-\infty}^{0} \mathcal{N}(z; \widetilde{p}, \mu^p)e^{\psi z}dz}_{\triangleq \underline{C}} + \underbrace{\frac{\psi}{2}\int_{0}^{\infty} \mathcal{N}(z; \widetilde{p}, \mu^p)e^{-\psi z}dz}_{\triangleq \overline{C}} \tag{82}$$

where $\widetilde{p} \triangleq \widehat{p} - y$. The expressions for $\underline{C}$ and $\overline{C}$ reported in (24), (25) result after completing the square inside the exponential terms in the integrands in (82) and simplifying.

Following similar techniques (i.e., shifting z by y and splitting the integral), it can be shown that (79) becomes

$$\widehat{z} = y + \frac{\underline{C}}{C}\int_z z\mathcal{N}_-(z; \widetilde{p}, \mu^p) + \frac{\overline{C}}{C}\int_z z\mathcal{N}_+(z; \widetilde{p}, \mu^p), \tag{83}$$

where $\mathcal{N}_+(\cdot)$ is defined in (26) and where $\mathcal{N}_-(x; a, b^2)$ is the pdf that results from taking a Gaussian with mean a and variance b^2, truncating its support to $x \in (-\infty, 0]$, and normalizing. Supposing that $\mathsf{u} \sim \mathcal{N}(a, b^2)$, [43] shows that

$$\mathrm{E}\{\mathsf{u}|\mathsf{u} > 0\} = \int_u u\mathcal{N}_+(u; a, b^2) = a + bh\left(-\frac{a}{b}\right), \tag{84}$$

$$\mathrm{E}\{\mathsf{u}|\mathsf{u} < 0\} = \int_u u\mathcal{N}_-(u; a, b^2) = a - bh\left(\frac{a}{b}\right), \tag{85}$$

where $h(\cdot)$ is defined in (26). Inserting (84) and (85) into (83) yields the posterior mean expression in (22).

To calculate the posterior variance μ^z used in line (R6) of Table I, we begin with

$$\mathrm{E}\{\mathsf{z}^2|\mathsf{p} = \widehat{p}; \mu^p\}$$

$$= \frac{1}{C}\int_z z^2\mathcal{L}(z; y; \psi)\mathcal{N}(z; \widehat{p}, \mu^p) \tag{86}$$

$$= \frac{1}{C}\int_{z'} (z' + y)^2\mathcal{L}(z'; 0; \psi)\mathcal{N}(z'; \widetilde{p}, \mu^p) \tag{87}$$

$$= 2y(\widehat{z} - y) + y^2 + \frac{1}{C}\int_z z^2\mathcal{L}(z; 0; \psi)\mathcal{N}(z; \widetilde{p}, \mu^p) \tag{87}$$

$$= 2y\widehat{z} - y^2 + \frac{\underline{C}}{C}\int_z z^2\mathcal{N}_-(z; \widetilde{p}, \mu^p) + \frac{\overline{C}}{C}\int_z z^2\mathcal{N}_+(z; \widetilde{p}, \mu^p). \tag{89}$$

Given that $\mathsf{u} \sim \mathcal{N}(a, b^2)$, [43] shows that

$$\mathrm{E}\{\mathsf{u}^2|\mathsf{u} > 0\} = \mathrm{var}\{\mathsf{u}|\mathsf{u} > 0\} + \mathrm{E}\{\mathsf{u}|\mathsf{u} > 0\}^2$$

$$= b^2 g\left(-\frac{a}{b}\right) + \left(a + bh\left(-\frac{a}{b}\right)\right)^2, \tag{90}$$

$$\mathrm{E}\{\mathsf{u}^2|\mathsf{u} < 0\} = \mathrm{var}\{\mathsf{u}|\mathsf{u} < 0\} + \mathrm{E}\{\mathsf{u}|\mathsf{u} < 0\}^2$$

$$= b^2 g\left(\frac{a}{b}\right) + \left(a - bh\left(\frac{a}{b}\right)\right)^2, \tag{91}$$

where $g(\cdot)$ is defined in (27). Inserting (90) and (91) into (89) and noting that $\mathrm{var}\{\mathbf{z}|\mathbf{p} = \widehat{p}; \mu^p\} = \mathrm{E}\{\mathbf{z}^2|\mathbf{p} = \widehat{p}; \mu^p\} - \mathrm{E}\{\mathbf{z}|\mathbf{p} = \widehat{p}; \mu^p\}^2$, we obtain (23).

B. EM update for Laplacian Rate

Inserting the Laplacian likelihood (18) into (51), we see that the EM update for the Laplacian rate parameter ψ becomes.

$$\psi^{i+1} = \arg\max_{\psi} \sum_{m=1}^{M} \widehat{\mathrm{E}}\left\{\ln \mathcal{L}\left(y_m; \boldsymbol{a}_m^{\mathsf{T}}\mathbf{x}, \psi\right) | \boldsymbol{y}; \psi^i\right\} \tag{92}$$

$$= \arg\max_{\psi} M \ln\psi - \psi \sum_{m=1}^{M} \widehat{\mathrm{E}}\left\{\left|\boldsymbol{a}_m^{\mathsf{T}}\mathbf{x} - y_m\right| |\boldsymbol{y}; \psi^i\right\}. \tag{93}$$

Zeroing the derivative of the objective in (93) w.r.t.\ ψ yields the update (52). The expectation in (93) can be written as

$$\widehat{\mathrm{E}}\left\{\left|\boldsymbol{a}_m^{\mathsf{T}}\mathbf{x} - y_m\right| |\boldsymbol{y}; \psi^i\right\} = \int_{\boldsymbol{x}} \left|\boldsymbol{a}_m^{\mathsf{T}}\boldsymbol{x} - y_m\right| f_{\mathbf{x}|\mathbf{y}}(\boldsymbol{x}|\boldsymbol{y}; \psi^i), \tag{94}$$

where $f_{\mathbf{x}|\mathbf{y}}(\boldsymbol{x}|\boldsymbol{y}; \psi^i)$ is taken to be the product of the approximated GAMP marginal posteriors in (10).

In the large system limit, the central limit theorem implies that $\mathsf{z}_m \triangleq \boldsymbol{a}_m^{\mathsf{T}}\mathbf{x}$, when conditioned on $\mathbf{y} = \boldsymbol{y}$, is $\mathcal{N}(\boldsymbol{a}_m^{\mathsf{T}}\widehat{\boldsymbol{x}}, \sum_{n=1}^{N} \boldsymbol{a}_{mn}^2\mu_n^x)$, yielding the approximation

$$\widehat{\mathsf{E}}\left\{\left|\boldsymbol{a}_m^{\mathsf{T}}\mathbf{x} - y_m\right| |\boldsymbol{y}; \psi^i\right\}$$

$$\approx \int_{z_m} |z_m - y_m| \mathcal{N}\left(z_m; \boldsymbol{a}_m^{\mathsf{T}}\widehat{\boldsymbol{x}}, \sum_{n=1}^{N} \boldsymbol{a}_{mn}^2\mu_n^x\right) \tag{95}$$

$$= \int_{z'_m} |z'_m| \mathcal{N}\left(z'_m; \boldsymbol{a}_m^{\mathsf{T}}\widehat{\boldsymbol{x}} - y_m, \sum_{n=1}^{N} \boldsymbol{a}_{mn}^2\mu_n^x\right). \tag{96}$$

Defining $\widetilde{z}_m \triangleq \boldsymbol{a}_m^{\mathsf{T}}\widehat{\boldsymbol{x}} - y_m$, and using a derivation similar to that used for (83), leads to (53).

Appendix C
Derivation of NNGM-GAMP Quantities

A. BNNGM Prior Steps for Sum-Product GAMP

Inserting the Bernoulli NNGM prior (36) into the GAMP approximated posterior (10), the posterior mean in line (R13) of Table I becomes (removing the n subscript for brevity)

$$\widehat{x} \triangleq \mathsf{e}\{\mathsf{x}|\mathsf{r} = \widehat{r}; \mu^r\} = \int_x x f_{\mathsf{x}|\mathsf{r}}(x|\widehat{r}; \mu^r) \tag{97}$$

$$= \frac{1}{\zeta}\int_+ x\mathcal{N}(x; \widehat{r}, \mu^r)\left((1-\tau)\delta(x) + \tau\sum_{\ell=1}^{L}\omega_\ell\mathcal{N}_+(x; \theta_\ell, \phi_\ell)\right)$$

$$= \frac{\tau}{\zeta}\sum_{\ell=1}^{L}\omega_\ell\int_+ x\mathcal{N}(x; \widehat{r}, \mu^r)\mathcal{N}_+(x; \theta_\ell, \phi_\ell), \tag{98}$$

where $\zeta \triangleq \int_x f_{\mathsf{x}}(x)\mathcal{N}(x;\widehat{r},\mu^r)$ is a scaling factor. Using the Gaussian-pdf multiplication rule,[13] we get

$$\widehat{x} = \frac{\tau}{\zeta}\sum_{\ell=1}^{L}\frac{\omega_\ell\mathcal{N}(\widehat{r};\theta_\ell,\mu^r+\phi_\ell)}{\Phi_c(-\theta_\ell/\sqrt{\phi_\ell})}\int_{+} x\mathcal{N}(x;\gamma_\ell,\nu_\ell), \tag{99}$$

with γ_ℓ and ν_ℓ defined in (42) and (43), respectively.

Using similar techniques, the scaling factor

$$\zeta = \int_{+}\mathcal{N}(x;\widehat{r},\mu^r)\left((1-\tau)\delta(x)+\tau\sum_{\ell=1}^{L}\omega_\ell\mathcal{N}_{+}(x;\theta_\ell,\phi_\ell)\right) \tag{100}$$

can be shown to be equivalent to (40). Finally, using the mean of a truncated Gaussian (84), and inserting (40) into (99), we get the NNGM-GAMP estimate (38).

To calculate the variance of the GAMP approximated posterior (10), we note that

$$\begin{aligned}\mu^x &\triangleq \operatorname{var}\{\mathsf{x}|\mathsf{r}=\widehat{r};\mu^r\}\\ &= \int_{+} x^2 f_{x|\mathsf{r}}(\mathsf{x}|\widehat{r};\mu^r) - \mathrm{E}\{\mathsf{x}|\mathsf{r}=\widehat{r};\mu^r\}^2.\end{aligned} \tag{101}$$

Following (97)–(99) and using the Gaussian-pdf multiplication rule, we find the second moment to be

$$\int_{+} x^2 f_{\mathsf{x}|\mathsf{r}}(x|\widehat{r};\mu^r) = \frac{\tau}{\zeta}\sum_{\ell=1}^{L}\frac{\beta_\ell}{\Phi_c(\alpha_\ell)}\int_{+} x^2\mathcal{N}(x;\gamma_\ell,\nu_\ell), \tag{102}$$

where β_ℓ and α_ℓ are given in (44) and (41), respectively.

Leveraging the second moment of a truncated Gaussian (90) in (102), and then inserting (38) and (102) into (101), we obtain the NNGM-GAMP variance estimate (39).

B. EM Updates of NNGM Parameters

We first derive the EM update for θ_k, the kth component location, given the previous parameter estimate $\boldsymbol{q}^i$. The maximizing value of θ_k in (57) is necessarily a value of θ_k that zeros the derivative of the sum, i.e., that satisfies[14]

$$\frac{d}{d\theta_k}\sum_{n=1}^{N}\int_{x_n} f_{\mathsf{x}|\mathsf{r}}\left(x_n|\widehat{r}_n;\mu_n^r,\boldsymbol{q}^i\right)\ln f_{\mathsf{x}}\left(x_n;\theta_k,\boldsymbol{q}^i_{\backslash\theta_k}\right) = 0 \tag{103}$$

$$\Leftrightarrow \sum_{n=1}^{N}\int_{x_n} f_{\mathsf{x}|\mathsf{r}}\left(x_n|\widehat{r}_n;\mu_n^r,\boldsymbol{q}^i\right)\frac{d}{d\theta_k}\ln f_{\mathsf{x}}\left(x_n;\theta_k,\boldsymbol{q}^i_{\backslash\theta_k}\right) = 0. \tag{104}$$

For all $x_n \geq 0$, the derivative in (104) can be written as

$$\frac{d}{d\theta_k}\ln f_{\mathsf{x}}\left(x_n;\theta_k,\boldsymbol{q}^i_{\backslash\theta_k}\right) = \frac{\frac{d}{d\theta_k}\tau^i\omega_k^i\frac{\mathcal{N}\left(x_n;\theta_k;\phi_k^i\right)}{\Phi_c\left(-\theta_k/\sqrt{\phi_k^i}\right)}}{f_{\mathsf{x}}\left(x_n;\theta_k,\boldsymbol{q}^i_{\backslash\theta_k}\right)}. \tag{105}$$

Because plugging (105) into (104) yields an intractable integral, we use the approximation[15] $\Phi_c(-\theta_k/\sqrt{\phi_k^i}) \approx \Phi_c(-\theta_k^i/\sqrt{\phi_k^i})$, yielding (106), shown at the bottom of the page. We also note that (106) is zero at $x_n = 0$ due to the Dirac delta function in the denominator.

Now, plugging in (106) and the approximated GAMP posterior $f_{\mathsf{x}|\mathsf{r}}(x_n|\widehat{r}_n;\mu_n^r,\boldsymbol{q}^i)$ from (10), integrating (104) separately over $[\epsilon,\infty)$ and its complement, and taking $\epsilon \to 0$, we find that the $(-\infty,\epsilon)$ portion vanishes, giving the necessary condition

$$\sum_{n=1}^{N}\int_{+}\frac{\widehat{p}(x_n|x_n\neq 0,\boldsymbol{y};\boldsymbol{q}^i)\omega_k^i\frac{\mathcal{N}\left(x_n;\theta_k,\phi_k^i\right)}{\Phi_c\left(-\theta_k^i/\sqrt{\phi_k^i}\right)}(x_n-\theta_k)}{\zeta_n\Big(\omega_k^i\mathcal{N}_{+}\big(x_n;\theta_k,\phi_k^i\big)+\sum_{\ell\neq k}\omega_\ell^i\mathcal{N}_{+}\big(x_n;\theta_\ell^i,\phi_\ell^i\big)\Big)} = 0, \tag{107}$$

where $\widehat{p}(x_n|x_n\neq 0,\boldsymbol{y};\boldsymbol{q}^i) \triangleq f_{\mathsf{x}|\mathsf{r}}(x_n|\widehat{r}_n,x_n\neq 0;\mu_n^r,\boldsymbol{q}^i)$. Since this integral cannot be evaluated in closed form, we apply the approximation $\mathcal{N}(x_n;\theta_k,\phi_k^i) \approx \mathcal{N}(x_n;\theta_k^i,\phi_k^i)$ in both the numerator and denominator, and subsequently exploit the fact that, for $x_n \geq 0$, $\widehat{p}(x_n|x_n\neq 0,\boldsymbol{y};\boldsymbol{q}^i) = \mathcal{N}(x_n;\widehat{r}_n,\mu_n^r)\sum_\ell\omega_\ell^i\mathcal{N}_{+}(x_n;\theta_\ell^i,\phi_\ell^i)$ from (10) to cancel terms, where we obtain the necessary condition

$$\sum_{n=1}^{N}\int_{+}\frac{\omega_k^i\mathcal{N}\left(x_n;\widehat{r}_n,\mu_n^r\right)\mathcal{N}_{+}\left(x_n;\theta_k^i,\phi_k^i\right)}{\zeta_n}(x_n-\theta_k) = 0. \tag{108}$$

Now using the Gaussian-pdf multiplication rule, we get

$$\sum_{n=1}^{N}\frac{\beta_{n,k}}{\Phi_c(\alpha_{n,k})}\int_{+}\mathcal{N}(x_n;\gamma_{n,k},\nu_{n,k})(x_n-\theta_k) = 0. \tag{109}$$

Following similar techniques as in Appendix C-A and noting that $\beta_{n,k} = \pi_n\overline{\beta}_{n,k}$, we see that the update θ_k^{i+1} in (60) is the value of θ_k that satisfies (109).

Similarly, the maximizing value of ϕ_k in (58) is necessarily a value of ϕ_k that zeroes the derivative, i.e.,

$$\sum_{n=1}^{N}\int_{x_n} f_{\mathsf{x}|\mathsf{r}}\left(x_n|\widehat{r}_n;\mu_n^r,\boldsymbol{q}^i\right)\frac{d}{d\phi_k}\ln f_{\mathsf{x}}\left(x_n;\phi_k,\boldsymbol{q}^i_{\backslash\phi_k}\right) = 0. \tag{110}$$

$$\frac{d}{d\theta_k}\ln f_{\mathsf{x}}\left(x_n;\theta_k,\boldsymbol{q}^i_{\backslash\theta_k}\right) = \left(\frac{x_n-\theta_k}{\phi_k^i}\right)\frac{\tau^i\omega_k^i\mathcal{N}\left(x_n;\theta_k,\phi_k^i\right)/\Phi_c\left(-\theta_k^i/\sqrt{\phi_k^i}\right)}{(1-\tau^i)\delta(x_n)+\tau^i\left(\omega_k^i\mathcal{N}_{+}\left(x_n;\theta_k,\phi_k^i\right)+\sum_{\ell\neq k}\omega_\ell^i\mathcal{N}_{+}\left(x_n;\theta_\ell^i,\phi_\ell^i\right)\right)}. \tag{106}$$

[13] $\mathcal{N}(x;a,A)\mathcal{N}(x;b,B) = \mathcal{N}\left(x;\frac{a/A+b/B}{1/A+1/B},\frac{1}{1/A+1/B}\right)\mathcal{N}(a;b,A+B)$.

[14] By employing the Dirac approximation $\delta(x) = \mathcal{N}(x;0,\varepsilon)$ for fixed arbitrarily small $\varepsilon > 0$, the integrand and its derivative w.r.t θ_k become continuous, justifying the exchange of differentiation and integration via the Leibniz integration rule. We apply the same reasoning for all exchanges of differentiation and integration in the sequel.

[15] This approximation becomes more accurate as $\frac{d}{d\theta_k}\Phi_c(-\theta_k/\sqrt{\phi_k})$ tends to zero,i.e., when $\theta_k/\sqrt{\phi_k}$ gets large, which was observed for the real-world experiments considered in Section VI.

$$\frac{d}{d\phi_k}\ln f_{\mathsf{x}}\left(x_n;\phi_k,\boldsymbol{q}^i_{\backslash\phi_k}\right)=\frac{1}{2}\left(\frac{\left(x_n-\theta^i_k\right)^2}{\phi^2_k}-\frac{1}{\phi_k}\right)\frac{\frac{\tau^i\omega^i_k\mathcal{N}\left(x_n;\theta_k,\phi^i_k\right)}{\Phi_c\left(-\theta^i_k/\sqrt{\phi^i_k}\right)}}{(1-\tau^i)\delta(x_n)+\tau^i\left(\omega^i_k\mathcal{N}_+\left(x_n;\theta^i_k,\phi_k\right)+\sum_{\ell\neq k}\omega^i_\ell\mathcal{N}_+\left(x_n;\theta^i_\ell,\phi^i_\ell\right)\right)}. \tag{111}$$

Using the prior given in (36), and simultaneously applying the approximation $\Phi_c(-\theta^i_k/\sqrt{\phi_k})\approx\Phi_c(-\theta^i_k/\sqrt{\phi^i_k})$, we see that the derivative in (110) can be written as (111), shown at the top of the page. Integrating (110) separately over $(-\infty,\epsilon)$ and $[\epsilon,\infty)$, and taking $\epsilon\to 0$, we see that the $(-\infty,\epsilon)$ portion vanishes, giving

$$\sum_{n=1}^{N}\int_{+}\frac{\frac{\widehat{p}\left(x_n|x_n\neq 0,\boldsymbol{y};\boldsymbol{q}^i\right)\omega^i_k\mathcal{N}\left(x_n;\theta^i_k,\phi_k\right)}{\Phi_c\left(-\theta^i_k/\sqrt{\phi^i_k}\right)}}{\zeta_n\left(\omega^i_k\mathcal{N}\left(x_n;\theta^i_k,\phi_k\right)+\sum_{\ell\neq k}\omega^i_\ell\mathcal{N}\left(x_n;\theta^i_\ell,\phi^i_\ell\right)\right)}\times\left(\frac{\left(x_n-\theta^i_k\right)^2}{\phi_k}-1\right)=0. \tag{112}$$

Again, this integral is difficult to compute, so we apply the approximation $\mathcal{N}(x_n;\theta_k,\phi^i_k)\approx\mathcal{N}(x_n;\theta^i_k,\phi^i_k)$ in both the numerator and denominator. After some cancellation (as in (107)) , we get the necessary condition

$$\sum_{n=1}^{N}\int_{+}\frac{\mathcal{N}\left(x_n;\widehat{r}_n,\mu^r_n\right)\omega^i_k\mathcal{N}_+\left(x_n;\theta^i_k,\phi^i_k\right)}{\zeta_n}\times\left(\frac{\left(x_n-\theta^i_k\right)^2}{\phi_k}-1\right)=0. \tag{113}$$

To find the value of ϕ_k that solves (113), we expand $\left(x_n-\theta^i_k\right)^2=x^2_n-2x_n\theta^i_k+\left(\theta^i_k\right)^2$ and apply the Gaussian-pdf multiplication rule, yielding

$$\sum_{n=1}^{N}\frac{\beta_{n,k}}{\Phi_c(\alpha_{n,k})}\int_{+}\mathcal{N}(x_n;\gamma_{n,k},\nu_{n,k})\times\left(\frac{x^2_n-2x_n\theta^i_k+\left(\theta^i_k\right)^2}{\phi_k}-1\right)=0. \tag{114}$$

Using similar techniques as in Appendix C-A and simplifying, we see that ϕ^{i+1}_k in (61) is the value of ϕ_k that solves (114).

Finally, we calculate the EM update in (59) for positive $\boldsymbol{\omega}$ under the pmf constraint $\sum_{k=1}^{L}\omega_k=1$ by solving the unconstrained optimization problem $\max_{\boldsymbol{\omega},\xi}J(\boldsymbol{\omega},\xi)$, where ξ is a Lagrange multiplier and

$$J(\boldsymbol{\omega},\xi)\triangleq\sum_{n=1}^{N}\widehat{\mathrm{E}}\left\{\ln f_{\mathsf{x}}\left(\mathrm{x}_n;\boldsymbol{\omega},\boldsymbol{q}^i_{\backslash\boldsymbol{\omega}}\right)|\boldsymbol{y};\boldsymbol{q}^i\right\}-\xi\left(\sum_{\ell=1}^{L}\omega_\ell-1\right). \tag{115}$$

First, we set $\frac{d}{d\omega_k}J(\boldsymbol{\omega},\xi)=0$, which yields

$$\sum_{n=1}^{N}\int_{x_n}\frac{f_{\mathsf{x}}(x_n;\boldsymbol{q}^i)\mathcal{N}\left(x_n;\widehat{r}_n,\mu^r_n\right)}{\zeta_n}\frac{d}{d\omega_k}\ln f_{\mathsf{x}}\left(x_n;\boldsymbol{\omega},\boldsymbol{q}^i_{\backslash\boldsymbol{\omega}}\right)=\xi \tag{116}$$

where, for non-negative x_n,

$$\frac{d}{d\omega_k}\ln f_{\mathsf{x}}\left(x_n;\boldsymbol{\omega},\boldsymbol{q}^i_{\backslash\boldsymbol{\omega}}\right)=\frac{\tau^i\mathcal{N}_+\left(x_n;\theta^i_k,\phi^i_k\right)}{f_{\mathsf{x}}\left(x_n;\boldsymbol{\omega},\boldsymbol{q}^i_{\backslash\boldsymbol{\omega}}\right)}. \tag{117}$$

Inserting (117) into (116), we get

$$\sum_{n=1}^{N}\int_{+}\frac{f_{\mathsf{x}}(x_n;\boldsymbol{q}^i)\mathcal{N}\left(x_n;\widehat{r}_n,\mu^r_n\right)}{\zeta_n}\frac{\tau^i\mathcal{N}_+\left(x_n;\theta^i_k,\phi^i_k\right)}{f_{\mathsf{x}}\left(x_n;\boldsymbol{\omega},\boldsymbol{q}^i_{\backslash\boldsymbol{\omega}}\right)}=\xi. \tag{118}$$

As in (107) and (112), the above integral is difficult to evaluate, and so we apply the additional approximation $\boldsymbol{\omega}\approx\boldsymbol{\omega}^i$, which reduces the previous equation to

$$\xi=\sum_{n=1}^{N}\int_{+}\frac{\tau^i\mathcal{N}_+\left(x_n;\theta^i_k,\phi^i_k\right)\mathcal{N}\left(x_n;\widehat{r}_n,\mu^r_n\right)}{\zeta_n}. \tag{119}$$

We then multiply both sides by ω^i_k for $k=1,\ldots,L$, and sum over k. Leveraging the fact $1=\sum_k\omega^i_k$, and simplifying, we obtain the equivalent condition

$$\xi=\sum_{n=1}^{N}\int_{+}\frac{\tau^i\sum_{k=1}^{L}\omega^i_k\mathcal{N}_+\left(x_n;\theta^i_k,\phi^i_k\right)\mathcal{N}\left(x_n;\widehat{r}_n,\mu^r_n\right)}{\zeta_n} \tag{120}$$

$$=\sum_{n=1}^{N}\frac{\tau^i}{\zeta_n}\sum_{k=1}^{L}\beta_{n,k}\int_{+}\frac{\mathcal{N}(x_n;\gamma_{n,k},\phi_{n,k})}{\Phi_c(\alpha_{n,k})}=\sum_{n=1}^{N}\pi_n. \tag{121}$$

Plugging (121) into (119) and multiplying both sides by ω_k, the derivative-zeroing value of ω_k is seen to be

$$\omega_k=\frac{\frac{\sum_{n=1}^{N}\int_{+}\tau^i\omega_k\mathcal{N}_+\left(x_n;\theta^i_k,\phi^i_k\right)\mathcal{N}\left(x_n;\widehat{r}_n,\mu^r_n\right)}{\zeta_n}}{\sum_{n=1}^{N}\pi_n}, \tag{122}$$

where, if we use $\omega_k\approx\omega^i_k$ on the right of (116), then we obtain the approximate EM update ω^{i+1}_k in (62).

Acknowledgment

We would like to thank Jason Parker and Justin Ziniel for suggestions on the Laplacian likelihood and EM-NNL-GAMP.

References

[1] J. Vila and P. Schniter, "An empirical-Bayes approach to recovering linearly constrained non-negative sparse signals," in *Proc. IEEE Workshop Comp. Adv. Multi-Sensor Adapt.Process.*, Dec. 2013, pp. 5–8.

[2] J. Bioucas-Dias, A. Plaza, N. Dobigeon, M. Parente, Q. Du, P. Gader, and J. Chanussot, "Hyperspectral unmixing overview: Geometrical, statistical, and sparse regression-based approaches," *IEEE J. Sel. Topics Appl. Earth Observ.*, vol. 5, no. 2, pp. 354–379, 2012.

[3] H. M. Markowitz, *Portfolio Selection: Efficient Diversification of Investments*. New York, NY, USA: Wiley, 1991.

[4] J. Brodie, I. Daubechies, C. De Mol, D. Giannone, and I. Loris, "Sparse and stable Markowitz portfolios," *Proc. Nat. Acad. Sci.*, vol. 106, no. 30, pp. 12267–12272, 2009.

[5] B. M. Jedynak and S. Khudanpur, "Maximum likelihood set for estimating a probability mass function," *Neural Comput.*, vol. 17, pp. 1508–1530, Jul. 2005.

[6] A. Kyrillidis, S. Becker, V. Cevher, and C. Koch, "Sparse projections onto the simplex," in *Proc. Int. Conf. Mach. Learn.*, June 2013, pp. 235–243.

[7] D. L. Donoho and J. Tanner, "Sparse nonnegative solution of underdetermined linear equations by linear programming," *Proc. Nat. Acad. Sci.*, vol. 102, no. 27, pp. 9446–9451, 2005.

[8] A. M. Bruckstein, M. Elad, and M. Zibulevsky, "On the uniqueness of nonnegative sparse solutions to underdetermined systems of equations," *IEEE Trans. Inf. Theory*, vol. 54, pp. 4813–4820, Nov. 2008.

[9] M. A. Khajehnejad, A. Dimakis, W. Xu, and B. Hassibi, "Sparse recovery of nonnegative signals with minimal expansion," *IEEE Trans. Signal Process.*, vol. 59, pp. 198–208, Jan. 2011.

[10] J. Romberg, "Imaging via compressive sampling," *IEEE Signal Process. Mag.*, vol. 25, pp. 14–20, Mar. 2008.

[11] D. Chen and R. J. Plemmons, "Nonnegativity constraints in numerical analysis," in *The Birth of Numerical Analysis*, A. Bultheel and R. Cools, Eds. Singapore: World Scientific, 2009, pp. 109–140.

[12] D. Bertsekas, *Nonlinear Programming*, 2nd ed. Belmont, MA, USA: Athena Scientific, 1999.

[13] R. Giryes, M. Elad, and Y. C. Eldar, "The projectured GSURE for automatic parameter tuning in iterative shrinkage methods," *Appl. Comput. Harmon. Anal.*, vol. 30, no. 2, pp. 407–422, 2011.

[14] M. Slawski and M. Hein, "Non-negative least squares for high-dimensional linear models: Consistency and sparse recovery without regularization," *Electron. J. Statist.*, vol. 7, pp. 3004–3056, Dec. 2013.

[15] R. Garg and R. Khandekar, "Gradient descent with sparsification: An iterative algorithm for sparse recovery with restricted isometry property," in *Proc. Int. Conf. Mach. Learn.*, New York, NY, USA, 2009, pp. 337–344.

[16] S. Rangan, "Generalized approximate message passing for estimation with random linear mixing," in *Proc. IEEE Int. Symp. Inf. Theory*, St. Petersburg, Russia, Aug. 2011, pp. 2168–2172.

[17] S. Rangan, P. Schniter, E. Riegler, A. Fletcher, and V. Cevher, "Fixed points of generalized approximate message passing with arbitrary matrices," in *Proc. IEEE Int. Symp. Inf. Theory*, Istanbul, Turkey, Jul. 2013, pp. 664–668.

[18] S. Rangan, P. Schniter, and A. Fletcher, "On the convergence of generalized approximate message passing with arbitrary matrices," in *Proc. IEEE Int. Symp. Inf. Theory*, Honolulu, HI, USA, Jul. 2014.

[19] A. Dempster, N. M. Laird, and D. B. Rubin, "Maximum-likelihood from incomplete data via the EM algorithm," *J. Roy. Statist. Soc.*, vol. 39, pp. 1–17, 1977.

[20] R. Tibshirani, "Regression shrinkage and selection via the lasso," *J. Roy. Statist. Soc. B*, vol. 58, no. 1, pp. 267–288, 1996.

[21] 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.

[22] S. Becker, E. Candès, and M. M. Grant, "Templates for convex cone problems with applications to sparse signal recovery," *Math. Program. Comput.*, vol. 3, no. 3, pp. 165–218, 2011.

[23] P. Bloomfield and W. L. Steiger, *Least Absolute Deviations: Theory, Applications, and Algorithms*. Boston, MA, USA: Birkhäuser, 1984.

[24] J. P. Vila and P. Schniter, "Expectation-maximization Gaussian-mixture approximate message passing," *IEEE Trans. Signal Process.*, vol. 61, pp. 4658–4672, Oct. 2013.

[25] A. Javanmard and A. Montanari, "State evolution for general approximate message passing algorithms, with applications to spatial coupling," *Information and Inference*, vol. 2, no. 2, pp. 115–144, 2013.

[26] U. S. Kamilov, S. Rangan, A. K. Fletcher, and M. Unser, "Approximate message passing with consistent parameter estimation and applications to sparse learning," in *Proc. Neural Inf. Process. Syst. Conf.*, Lake Tahoe, NV, USA, Dec. 2012, pp. 2447–2455.

[27] B. Efron, *Large-Scale Inference: Empirical Bayes Methods for Estimation, Testing, and Prediction*. New York, NY, USA: Cambridge Univ. Press, 2010.

[28] D. L. Donoho, A. Maleki, and A. Montanari, "Message passing algorithms for compressed sensing," *Proc. Nat. Acad. Sci.*, vol. 106, pp. 18914–18919, Nov. 2009.

[29] D. L. Donoho, A. Maleki, and A. Montanari, "Message passing algorithms for compressed sensing: II. Analysis and validation," in *Proc. Inf. Theory Workshop*, Cairo, Egypt, Jan. 2010, pp. 1–5.

[30] A. Chambolle, R. A. DeVore, N. Lee, and B. J. Lucier, "Nonlinear wavelet image processing: Variational problems, compression, and noise removal through wavelet shrinkage," *IEEE Trans. Image Process.*, vol. 7, pp. 319–335, Mar. 1998.

[31] I. Daubechies, M. Defrise, and C. D. Mol, "An iterative thresholding algorithm for linear inverse problems with a sparsity constraint," *Commun. Pure Appl. Math.*, vol. 57, pp. 1413–1457, Nov. 2004.

[32] A. Maleki and D. L. Donoho, "Optimally tuned iterative reconstruction algorithms for compressed sensing," *IEEE J. Sel. Topics Signal Process.*, vol. 4, pp. 330–341, Apr. 2010.

[33] R. Neal and G. Hinton, "A view of the EM algorithm that justifies incremental, sparse, and other variants," in *Learning in Graphical Models*, M. I. Jordan, Ed. Cambridge, MA, USA: MIT Press, 1999, pp. 355–368.

[34] P. Stoica and Y. Selén, "Model-order selection: A review of information criterion rules," *IEEE Signal Process. Mag.*, vol. 21, pp. 36–47, Jul. 2004.

[35] C. M. Bishop, *Pattern Recognition and Machine Learning*. New York, NY, USA: Springer, 2007.

[36] S. Rangan, J. T. Parker, P. Schniter, J. Ziniel, J. Vila, and M. Borgerding *et al.*, GAMPmatlab [Online]. Available: https://sourceforge.net/projects/gampmatlab/

[37] E. van den Berg and M. P. Friedlander, "Probing the Pareto frontier for basis pursuit solutions," *SIAM J. Scientif. Comput.*, vol. 31, no. 2, pp. 890–912, 2008.

[38] V. DeMiguel, L. Garlappi, and R. Uppal, "Optimal versus naive diversification: How inefficient is the 1-n portfolio strategy?," *Rev. Financ. Stud.*, vol. 22, pp. 1915–1953, May 2009.

[39] J. Nascimento and J. Bioucas Dias, "Vertex component analysis: A fast algorithm to unmix hyperspectral data," *IEEE Trans. Geosci. Remote Sens.*, vol. 43, no. 4, pp. 898–910, 2005.

[40] A. Giannandrea, N. Raqueno, D. W. Messinger, J. Faulring, J. P. Kerekes, J. van Aardt, K. Canham, S. Hagstrom, E. Ontiveros, A. Gerace, J. Kaufman, K. M. Vongsy, H. Griffith, B. D. Bartlett, E. Ientilucci, J. Meola, L. Scarff, and B. Daniels, "The SHARE 2012 data collection campaign," in *Proc. SPIE*, 2013, vol. 8743, no. 87430F, p. 15.

[41] K. Canham, D. Goldberg, J. Kerekes, N. Raqueno, and D. Messinger, "SHARE 2012: Large edge targets for hyperspectral imaging applications," in *Proc. SPIE*, 2013, vol. 8743, no. 87430G, p. 9.

[42] D. Heinz and C.-I. Chang, "Fully constrained least squares linear spectral mixture analysis method for material quantification in hyperspectral imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 39, no. 3, pp. 529–545, 2001.

[43] N. L. Johnson, S. Kotz, and N. Balakrishnan, *Continuous Univariate Distributions*. New York, NY, USA: Wiley, 1995, vol. 2.

Jeremy P. Vila (S'11) received the B.S. degree in Electrical and Computer Engineering from the Ohio State University in 2010. He is currently a Ph.D. student in the Information Processing Systems Lab in the Department of Electrical and Computer Engineering at OSU. His primary research interests include compressive sensing, statistical signal processing, and machine learning.

Philip Schniter (F'14) received the B.S. and M.S. degrees in Electrical Engineering from the University of Illinois at Urbana-Champaign in 1992 and 1993, respectively, and the Ph.D. degree in Electrical Engineering from Cornell University in Ithaca, NY, in 2000.

From 1993 to 1996 he was employed by Tektronix Inc., Beaverton, OR, as a systems engineer. After receiving the Ph.D. degree, he joined the Department of Electrical and Computer Engineering at The Ohio State University, Columbus, where he is currently a Professor and a member of the Information Processing Systems (IPS) Lab. In 2008–2009, he was a visiting professor at Eurecom, Sophia Antipolis, France, and Supélec, Gif-sur-Yvette, France. His areas of interest currently include statistical signal processing, wireless communications and networks, and machine learning.

In 2003, Dr. Schniter received the National Science Foundation CAREER Award.