Expectation-Maximization Gaussian-Mixture Approximate Message Passing

Jeremy Vila and Philip Schniter
Dept. of ECE, The Ohio State University, Columbus, OH 43210. (Email: vilaj@ece.osu.edu, schniter@ece.osu.edu)

***Abstract*—When recovering a sparse signal from noisy compressive linear measurements, the distribution of the signal's non-zero coefficients can have a profound affect on recovery mean-squared error (MSE). If this distribution was apriori known, one could use efficient approximate message passing (AMP) techniques for nearly minimum MSE (MMSE) recovery. In practice, though, the distribution is unknown, motivating the use of robust algorithms like Lasso—which is nearly minimax optimal—at the cost of significantly larger MSE for non-least-favorable distributions. As an alternative, we propose an empirical-Bayesian technique that simultaneously learns the signal distribution while MMSE-recovering the signal—according to the learned distribution—using AMP. In particular, we model the non-zero distribution as a Gaussian mixture, and learn its parameters through expectation maximization, using AMP to implement the expectation step. Numerical experiments confirm the state-of-the-art performance of our approach on a range of signal classes.** [1] [2]

I. Introduction

We consider estimating a K-sparse (or compressible) signal $\boldsymbol{x} \in \mathbb{R}^N$ from $M < N$ linear measurements $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{w} \in \mathbb{R}^M$, where $\boldsymbol{A}$ is known and $\boldsymbol{w}$ is additive white Gaussian noise (AWGN). For this problem, accurate (relative to the noise variance) signal recovery is possible with polynomial-complexity algorithms when $\boldsymbol{x}$ is sufficiently sparse and when $\boldsymbol{A}$ satisfies certain restricted isometry properties [1].

A well-known approach to the sparse-signal recovery problem is Lasso [2], which solves the convex problem

$$\hat{\boldsymbol{x}}_{\mathsf{lasso}} = \arg\min_{\hat{\boldsymbol{x}}} \|\boldsymbol{y} - \boldsymbol{A}\hat{\boldsymbol{x}}\|_2^2 + \lambda_{\mathsf{lasso}}\|\hat{\boldsymbol{x}}\|_1, \qquad (1)$$

with λ_{lasso} a tuning parameter. When $\boldsymbol{A}$ is constructed from i.i.d entries, the performance of Lasso can be sharply characterized in the large system limit (i.e., as $K, M, N \to \infty$ with fixed undersampling ratio M/N and sparsity ratio K/M) using the so-called phase transition curve (PTC) [3]. When the observations are noiseless, the PTC bisects the M/N-versus-K/M plane into the region where Lasso reconstructs the signal perfectly (with high probability) and the region where it does not. (See Figs. 1-3.) When the observations are noisy, the same PTC bisects the plane into the regions where Lasso's noise sensitivity (i.e., the ratio of estimation-error power to measurement-noise power under the worst-case signal distribution) is either finite or infinite [4]. An important fact about Lasso's noiseless PTC is that it is invariant to signal distribution. In other words, if we consider the elements of the vector $\boldsymbol{x}$ to be drawn i.i.d from the marginal pdf

$$p_X(x) = \lambda f_X(x) + (1-\lambda)\delta(x), \qquad (2)$$

where $\delta(\cdot)$ is the Dirac delta, $f_X(\cdot)$ is the active-coefficient pdf (with zero probability mass at $x = 0$), and $\lambda \triangleq K/N$, then the Lasso PTC is invariant to $f_X(\cdot)$. While this implies that Lasso is robust to "difficult" sparse-signal distributions, it also implies that Lasso cannot benefit from the sparse-signal distribution being an "easy" one.

At the other end of the spectrum is minimum mean-squared error (MMSE)-optimal signal recovery under *known* marginal pdfs of the form (2). The PTC of MMSE recovery has been recently characterized [5] and shown to be well above that of Lasso. In particular, for *any* $f_X(\cdot)$, the PTC on the M/N-versus-K/M plane equals the line $K/M = 1$ in both the noiseless and noisy cases. Moreover, efficient algorithms for approximate MMSE-recovery have been proposed, such as the Bayesian version of Donoho, Maleki, and Montanari's *approximate message passing* (AMP) algorithm from [6], which performs loopy belief-propagation on the underlying factor graph using central-limit-theorem approximations that become exact in the large-system limit under i.i.d $\boldsymbol{A}$. Although AMP's complexity is remarkably low (e.g., dominated by one application of $\boldsymbol{A}$ and $\boldsymbol{A}^{\mathsf{T}}$ per iteration with typically < 50 iterations to convergence), it offers rigorous performance guarantees in the large-system limit. To handle arbitrary noise distributions and a wider class of matrices $\boldsymbol{A}$, Rangan proposed a *generalized AMP* (GAMP) [7] that forms the starting point of this work. (See Table I.)

In practice, one desires a recovery algorithm that does not need to know $p_X(\cdot)$ a priori, yet offers performance on par with MMSE recovery, which (by definition) knows $p_X(\cdot)$ a priori. Towards this aim, we propose a recovery scheme that aims to *learn* the prior signal distribution $p_X(\cdot)$ (as well as the variance of the AWGN) while simultaneously recovering the signal vector $\boldsymbol{x}$ from the noisy compressed measurements $\boldsymbol{y}$. To do so, we model the active component $f_X(\cdot)$ in (2) using a generic L-term Gaussian mixture (GM) and then learn the prior signal and noise parameters using the expectation-maximization (EM) algorithm [8]. As we will see, the EM expectation is naturally implemented using the GAMP algorithm, which also provides approximately MMSE estimates of $\boldsymbol{x}$.

[1] This work has been supported in part by NSF-I/UCRC grant IIP-0968910, by NSF grant CCF-1018368, and by DARPA/ONR grant N66001-10-1-4090.

[2] Portions of this work were presented in a poster at the Duke Workshop on Sensing and Analysis of High-Dimensional Data, July 2011.

Since we treat the prior pdf parameters as deterministic unknowns, our proposed EM-GM-GAMP algorithm can be considered as an "empirical-Bayesian" approach. Compared with previous empirical-Bayesian approaches (e.g., [9]–[12]), ours has a more flexible signal model, and thus is able to better match a wide range of signal pdfs $p_X(\cdot)$, as we demonstrate numerically in the sequel. Moreover, due to the computationally efficient nature of GAMP, our algorithm is significantly faster than empirical-Bayesian algorithms based on Tipping's relevance vector machine [9]–[11]. Finally, we note that our EM-GM-GAMP algorithm can be considered as a generalization of our previously proposed EM-BG-GAMP algorithm [12] from a Bernoulli-Gaussian signal model to a Bernoulli-GM signal model.

II. Gaussian-Mixture GAMP

We first introduce Gaussian-mixture (GM) GAMP, a key component of our overall algorithm. In GM-GAMP, the signal $\boldsymbol{x} = [x_1, \ldots, x_N]^{\mathsf{T}}$ is assumed to be i.i.d with marginal pdf

$$p_X(x; \lambda, \boldsymbol{\omega}, \boldsymbol{\theta}, \boldsymbol{\phi}) = (1-\lambda)\delta(x) + \lambda \sum_{\ell=1}^{L} \omega_\ell \mathcal{N}(x; \theta_\ell, \phi_\ell), \quad (3)$$

where $\delta(\cdot)$ denotes the Dirac delta, λ the sparsity rate, and, for the k^{th} GM component, ω_k, θ_k, and ϕ_k are the weight, mean, and variance, respectively. The AWGN[3] $\boldsymbol{w}$ is then assumed to be independent of $\boldsymbol{x}$ and have variance ψ:

$$p_W(w; \psi) = \mathcal{N}(w; 0, \psi) \quad (4)$$

Although above and in the sequel we assume real-valued Gaussians, all expressions can be converted to the circular-complex case by replacing $\mathcal{N}$ with $\mathcal{CN}$ and removing all $\frac{1}{2}$'s. We emphasize that, from GM-GAMP's perspective, the prior parameters $\boldsymbol{q} \triangleq [\lambda, \boldsymbol{\omega}, \boldsymbol{\theta}, \boldsymbol{\phi}, \psi]$ are all known.

GAMP can handle an arbitrary probabilistic relationship $p_{Y|Z}(y_m|z_m)$ between the observed output y_m and the noiseless output $z_m \triangleq \boldsymbol{a}_m^{\mathsf{T}}\boldsymbol{x}$, where $\boldsymbol{a}_m^{\mathsf{T}}$ is the m^{th} row of $\boldsymbol{A}$. Our additive Gaussian noise assumption implies $p_{Y|Z}(y|z) = \mathcal{N}(y; z, \psi)$. To complete our description of GM-GAMP, we only need to derive $g_{\mathsf{in}}(\cdot)$, $g'_{\mathsf{in}}(\cdot)$, $g_{\mathsf{out}}(\cdot)$, and $g'_{\mathsf{out}}(\cdot)$ in Table I. Using straightforward calculations, our $p_{Y|Z}(\cdot|\cdot)$ yields [7]

$$g_{\mathsf{out}}(y, \hat{z}, \mu^z; \boldsymbol{q}) = \frac{y - \hat{z}}{\mu^z + \psi} \quad (5)$$

$$-g'_{\mathsf{out}}(y, \hat{z}, \mu^z; \boldsymbol{q}) = \frac{1}{\mu^z + \psi}, \quad (6)$$

and our GM signal prior (3) yields

$$g_{\mathsf{in}}(\hat{r}, \mu^r; \boldsymbol{q}) = \frac{\sum_{\ell=1}^{L} \beta_\ell(\hat{r}, \mu^r; \boldsymbol{q}) \gamma_\ell(\hat{r}, \mu^r; \boldsymbol{q})}{(1-\lambda)\mathcal{N}(0; \hat{r}, \mu^r) + \sum_{\ell=1}^{L} \beta_\ell(\hat{r}, \mu^r; \boldsymbol{q})} \quad (7)$$

$$\mu^r g'_{\mathsf{in}}(\hat{r}, \mu^r; \boldsymbol{q}) = -|g_{\mathsf{in}}(\hat{r}, \mu^r, \boldsymbol{q})|^2 + \frac{\sum_{\ell=1}^{L} \beta_\ell(\hat{r}, \mu^r; \boldsymbol{q}) \left(|\gamma_\ell(\hat{r}, \mu^r; \boldsymbol{q})|^2 + \nu_\ell(\hat{r}, \mu^r; \boldsymbol{q}) \right)}{(1-\lambda)\mathcal{N}(0; \hat{r}, \mu^r) + \sum_{\ell=1}^{L} \beta_\ell(\hat{r}, \mu^r; \boldsymbol{q})} \quad (8)$$

[3]To model heavy-tailed noise, a different choice of $p_W(\cdot;\cdot)$ may be more appropriate. However, the way it is handled in GAMP and learned by EM would remain essentially the same.

definitions:

$$p_{Z|Y}(z|y; \hat{z}, \mu^z) = \frac{p_{Y|Z}(y|z)\,\mathcal{N}(z; \hat{z}, \mu^z)}{\int_{z'} p_{Y|Z}(y|z')\,\mathcal{N}(z'; \hat{z}, \mu^z)} \quad \text{(D1)}$$

$$g_{\mathsf{out}}(y, \hat{z}, \mu^z) = \frac{1}{\mu^z}\left(\mathrm{E}_{Z|Y}\{z|y; \hat{z}, \mu^z\} - \hat{z}\right) \quad \text{(D2)}$$

$$g'_{\mathsf{out}}(y, \hat{z}, \mu^z) = \frac{1}{\mu^z}\left(\frac{\mathrm{var}_{Z|Y}\{z|y; \hat{z}, \mu^z\}}{\mu^z} - 1\right) \quad \text{(D3)}$$

$$p_{X|\boldsymbol{Y}}(x|\boldsymbol{y}; \hat{r}, \mu^r) = \frac{p_X(x)\,\mathcal{N}(x; \hat{r}, \mu^r)}{\int_{x'} p_X(x')\,\mathcal{N}(x'; \hat{r}, \mu^r)} \quad \text{(D4)}$$

$$g_{\mathsf{in}}(\hat{r}, \mu^r) = \int_x x\, p_{X|\boldsymbol{Y}}(x|\boldsymbol{y}; \hat{r}, \mu^r) \quad \text{(D5)}$$

$$g'_{\mathsf{in}}(\hat{r}, \mu^r) = \frac{1}{\mu^r}\int_x |x - g_{\mathsf{in}}(\hat{r}, \mu^r)|^2\, p_{X|\boldsymbol{Y}}(x|\boldsymbol{y}; \hat{r}, \mu^r) \quad \text{(D6)}$$

initialize:

$$\forall n: \hat{x}_n(1) = \int_x x\, p_X(x) \quad \text{(I1)}$$

$$\forall n: \mu_n^x(1) = \int_x |x - \hat{x}_n(1)|^2 p_X(x) \quad \text{(I2)}$$

$$\forall m: \hat{u}_m(0) = 0 \quad \text{(I3)}$$

for $t = 1, 2, 3, \ldots$

$$\forall m: \hat{z}_m(t) = \textstyle\sum_{n=1}^{N} A_{mn}\hat{x}_n(t) \quad \text{(R1)}$$

$$\forall m: \mu_m^z(t) = \textstyle\sum_{n=1}^{N} |A_{mn}|^2 \mu_n^x(t) \quad \text{(R2)}$$

$$\forall m: \hat{p}_m(t) = \hat{z}_m(t) - \mu_m^z(t)\,\hat{u}_m(t-1) \quad \text{(R3)}$$

$$\forall m: \hat{u}_m(t) = g_{\mathsf{out}}(y_m, \hat{p}_m(t), \mu_m^z(t)) \quad \text{(R4)}$$

$$\forall m: \mu_m^u(t) = -g'_{\mathsf{out}}(y_m, \hat{p}_m(t), \mu_m^z(t)) \quad \text{(R5)}$$

$$\forall n: \mu_n^r(t) = \left(\textstyle\sum_{n=1}^{N} |A_{mn}|^2 \mu_m^u(t)\right)^{-1} \quad \text{(R6)}$$

$$\forall n: \hat{r}_n(t) = \hat{x}_n(t) + \mu_n^r(t) \textstyle\sum_{m=1}^{M} A_{mn}^* \hat{u}_m(t) \quad \text{(R7)}$$

$$\forall n: \mu_n^x(t+1) = \mu_n^r(t) g'_{\mathsf{in}}(\hat{r}_n(t), \mu_n^r(t)) \quad \text{(R8)}$$

$$\forall n: \hat{x}_n(t+1) = g_{\mathsf{in}}(\hat{r}_n(t), \mu_n^r(t)) \quad \text{(R9)}$$

end

TABLE I
The GAMP Algorithm [7]

where

$$\beta_\ell(\hat{r}, \mu^r; \boldsymbol{q}) \triangleq \lambda\omega_\ell \mathcal{N}(\hat{r}; \theta_\ell, \phi_\ell + \mu^r) \quad (9)$$

$$\gamma_\ell(\hat{r}, \mu^r; \boldsymbol{q}) \triangleq \frac{\hat{r}/\mu^r + \theta_\ell/\phi_\ell}{1/\mu^r + 1/\phi_\ell} \quad (10)$$

$$\nu_\ell(\hat{r}, \mu^r; \boldsymbol{q}) \triangleq \frac{1}{1/\mu^r + 1/\phi_\ell}. \quad (11)$$

Table I implies that GM-GAMP's marginal posteriors are

$$p(x_n|\boldsymbol{y}; \boldsymbol{q}) = p_X(x_n; \boldsymbol{q})\,\mathcal{N}(x_n; \hat{r}_n, \mu_n^r)/\zeta(\hat{r}_n, \mu_n^r; \boldsymbol{q}) \quad (12)$$

$$= \left((1-\lambda)\delta(x_n) + \lambda\sum_{\ell=1}^{L}\omega_\ell\mathcal{N}(x_n; \theta_\ell, \phi_\ell)\right) \times \mathcal{N}(x_n; \hat{r}_n, \mu_n^r)/\zeta(\hat{r}_n, \mu_n^r; \boldsymbol{q}) \quad (13)$$

$$\zeta(\hat{r}, \mu^r; \boldsymbol{q}) \triangleq \int_x p_X(x; \boldsymbol{q})\,\mathcal{N}(x; \hat{r}, \mu^r). \quad (14)$$

From (13), it is straightforward to show that the posterior support probabilities returned by GM-GAMP are

$$\Pr\{x_n \neq 0 \,|\, \boldsymbol{y}; \boldsymbol{q}\} = \pi(\hat{r}_n, \mu_n^r; \boldsymbol{q}) \quad (15)$$

$$\pi(\hat{r}, \mu^r; \boldsymbol{q}) \triangleq \frac{1}{1 + \left(\frac{\sum_{\ell=1}^{L}\beta_\ell(\hat{r}, \mu^r; \boldsymbol{q})}{(1-\lambda)\mathcal{N}(0; \hat{r}, \mu^r)}\right)^{-1}}. \quad (16)$$

III. EM Learning of the Prior Parameters $\boldsymbol{q}$

We use the expectation-maximization (EM) algorithm [8] to learn the prior parameters $\boldsymbol{q} \triangleq [\lambda, \boldsymbol{\omega}, \boldsymbol{\theta}, \boldsymbol{\phi}, \psi]$. The EM algorithm is an iterative technique that increases a lower bound on the likelihood $p(\boldsymbol{y}; \boldsymbol{q})$ at each iteration, thus guaranteeing that the likelihood converges to a local maximum. In our case, the "hidden data" is chosen as $\{\boldsymbol{x}, \boldsymbol{w}\}$, implying the iteration-i EM update

$$\boldsymbol{q}^{i+1} = \arg\max_{\boldsymbol{q}} \mathrm{E}\left\{ \ln p(\boldsymbol{x}, \boldsymbol{w}; \boldsymbol{q}) \,\middle|\, \boldsymbol{y}; \boldsymbol{q}^i \right\}, \quad (17)$$

where $\mathrm{E}\{\cdot|\boldsymbol{y};\boldsymbol{q}^i\}$ denotes expectation conditioned on the observations $\boldsymbol{y}$ under the parameter hypothesis $\boldsymbol{q}^i$. Since it is impractical to update the entire vector $\boldsymbol{q}$ at once, we update $\boldsymbol{q}$ one element at a time (while holding the others fixed), which can be recognized as the "incremental" technique from [13]. In the sequel, we use "$\boldsymbol{q}^i_{\backslash\lambda}$" to denote the vector $\boldsymbol{q}^i$ with λ^i removed (and similar for the other parameters).

A. EM update for λ

We now derive the EM update for λ given previous parameters $\boldsymbol{q}^i \triangleq [\lambda^i, \boldsymbol{\omega}^i, \boldsymbol{\theta}^i, \boldsymbol{\phi}^i, \psi^i]$. Since $\boldsymbol{x}$ is apriori independent of $\boldsymbol{w}$ and i.i.d, the joint pdf $p(\boldsymbol{x},\boldsymbol{w};\boldsymbol{q})$ decouples into $C\prod_{n=1}^N p_X(x_n;\boldsymbol{q})$ for a λ-invariant constant C, and so

$$\lambda^{i+1} = \arg\max_{\lambda\in(0,1)} \sum_{n=1}^N \mathrm{E}\left\{ \ln p_X(x_n;\lambda,\boldsymbol{q}^i_{\backslash\lambda}) \,\middle|\, \boldsymbol{y};\boldsymbol{q}^i\right\}. \quad (18)$$

The maximizing value of λ in (18) is necessarily a value of λ that zeroes the derivative, i.e., that satisfies

$$\sum_{n=1}^N \int_{x_n} p(x_n|\boldsymbol{y};\boldsymbol{q}^i) \frac{d}{d\lambda} \ln p_X(x_n;\lambda,\boldsymbol{q}^i_{\backslash\lambda}) = 0. \quad (19)$$

For the $p_X(x_n;\boldsymbol{q})$ given in (3), it is readily seen that

$$\frac{d}{d\lambda}\ln p_X(x_n;\lambda,\boldsymbol{q}^i_{\backslash\lambda}) = \frac{\sum_{\ell=1}^L \omega^i_\ell \mathcal{N}(x_n;\theta^i_\ell,\phi^i_\ell) - \delta(x_n)}{p_X(x_n;\lambda,\boldsymbol{q}^i_{\backslash\lambda})} = \begin{cases} \frac{1}{\lambda} & x_n \neq 0 \\ \frac{-1}{1-\lambda} & x_n = 0 \end{cases}. \quad (20)$$

Plugging (20) and (13) into (19), it becomes evident that the point $x_n = 0$ must be treated differently than $x_n \in \mathbb{R}\backslash 0$. Thus, we define the closed ball $\mathcal{B}_\epsilon = [-\epsilon,\epsilon]$ and $\overline{\mathcal{B}_\epsilon} \triangleq \mathbb{R}\setminus\mathcal{B}_\epsilon$, and note that, in the limit $\epsilon\to 0$, the following becomes equivalent to (19), given the definition of $\pi(\hat{r},\mu^r;\boldsymbol{q})$ in (16):

$$\frac{1}{\lambda}\sum_{n=1}^N \underbrace{\int_{x_n\in\overline{\mathcal{B}_\epsilon}} p(x_n\,|\,\boldsymbol{y};\boldsymbol{q}^i)}_{\stackrel{\epsilon\to 0}{=}\,\pi(\hat{r}_n,\mu^r_n;\boldsymbol{q}^i)} = \frac{1}{1-\lambda}\sum_{n=1}^N \underbrace{\int_{x_n\in\mathcal{B}_\epsilon} p(x_n\,|\,\boldsymbol{y};\boldsymbol{q}^i)}_{\stackrel{\epsilon\to 0}{=}\,1-\pi(\hat{r}_n,\mu^r_n;\boldsymbol{q}^i)}. \quad (21)$$

To verify that the left integral converges to the $\pi(\hat{r}_n,\mu^r_n;\boldsymbol{q}^i)$ defined in (16), it suffices to plug (13) into (21) and apply the Gaussian-pdf multiplication rule.[4] Meanwhile, for any ϵ, the right integral must equal one minus the left. Thus, the EM update for λ is the unique value satisfying (21) as $\epsilon\to 0$, i.e.,

$$\lambda^{i+1} = \frac{1}{N}\sum_{n=1}^N \pi(\hat{r}_n,\mu^r_n;\boldsymbol{q}^i). \quad (22)$$

Conveniently, $\{\pi(\hat{r}_n,\mu^r_n;\boldsymbol{q}^i)\}_{n=1}^N$ are GM-GAMP outputs, as we recall from (16).

[4] $\mathcal{N}(x;a,A)\mathcal{N}(x;b,B) = \mathcal{N}(x;\frac{a/A+b/B}{1/A+1/B},\frac{1}{1/A+1/B})\mathcal{N}(0;a-b,A+B)$.

B. EM updates for Gaussian Mixture Parameters

For each $k = 1,\ldots,L$, we incrementally update each GM parameter θ_k, ϕ_k, and $\boldsymbol{\omega}$ while holding the others fixed. The EM updates become

$$\theta^{i+1}_k = \arg\max_{\theta_k\in\mathbb{R}} \sum_{n=1}^N \mathrm{E}\left\{\ln p_X(x_n;\theta_k,\boldsymbol{q}^i_{\backslash\theta_k})|\boldsymbol{y},\boldsymbol{q}^i\right\}, \quad (23)$$

$$\phi^{i+1}_k = \arg\max_{\phi_k>0} \sum_{n=1}^N \mathrm{E}\left\{\ln p_X(x_n;\phi_k,\boldsymbol{q}^i_{\backslash\phi_k})|\boldsymbol{y},\boldsymbol{q}^i\right\} \quad (24)$$

$$\boldsymbol{\omega}^{i+1} = \arg\max_{\boldsymbol{\omega}>0:\,\sum_k\omega_k=1} \sum_{n=1}^N \mathrm{E}\left\{\ln p_X(x_n;\boldsymbol{\omega},\boldsymbol{q}^i_{\backslash\boldsymbol{\omega}})|\boldsymbol{y},\boldsymbol{q}^i\right\}. \quad (25)$$

Following (19), the maximizing value of θ_k in (23) is necessarily a value of θ_k that zeros

$$\sum_{n=1}^N \int_{x_n} p(x_n|\boldsymbol{y},\boldsymbol{q}^i)\frac{d}{d\theta_k}\ln p_X(x_n;\theta_k,\boldsymbol{q}^i_{\backslash\theta_k}) = 0, \quad (26)$$

where $p(x_n|\boldsymbol{y},\boldsymbol{q}^i) = p_X(x_n;\boldsymbol{q}^i)\mathcal{N}(x_n;\hat{r}_n,\mu^r_n)/\zeta(\hat{r}_n,\mu^r_n;\boldsymbol{q}^i)$ from (D4), recalling $p_X(x;\boldsymbol{q})$ from (3) and $\zeta(\hat{r},\mu^r;\boldsymbol{q})$ from (14). Taking the derivative, we find

$$\frac{d}{d\theta_k}\ln p_X(x_n;\theta_k,\boldsymbol{q}^i_{\backslash\theta_k}) = \left(\tfrac{x_n-\theta_k}{\phi^i_k}\right) \times \frac{\lambda^i\omega^i_k\mathcal{N}(x_n;\theta_k,\phi^i_k)}{(1-\lambda^i)\delta(x_n)+\lambda^i(\omega^i_k\mathcal{N}(x_n;\theta_k,\phi^i_k)+\sum_{\ell\neq k}\omega^i_\ell\mathcal{N}(x_n;\theta^i_\ell,\phi^i_\ell))}. \quad (27)$$

Integrating (26) separately over $\mathcal{B}_\epsilon$ and $\overline{\mathcal{B}_\epsilon}$, as in (21), and taking $\epsilon\to 0$, we find that the $\mathcal{B}_\epsilon$ portion vanishes, giving

$$\textstyle\sum_{n=1}^N \int_{x_n} \frac{p(x_n|x_n\neq 0,\boldsymbol{y},\boldsymbol{q}^i)\lambda^i\omega^i_k\mathcal{N}(x_n;\theta_k,\phi^i_k)(x_n-\theta_k)}{\zeta(\hat{r}_n,\mu^r_n;\boldsymbol{q}^i)(\omega^i_k\mathcal{N}(x_n;\theta_k,\phi^i_k)+\sum_{\ell\neq k}\omega^i_\ell\mathcal{N}(x_n;\theta^i_\ell,\phi^i_\ell))} = 0. \quad (28)$$

Since this integral is difficult to evaluate, we apply the approximation $\mathcal{N}(x_n;\theta_k,\phi^i_k) \approx \mathcal{N}(x_n;\theta^i_k,\phi^i_k)$ and exploit the fact that $p(x_n|x_n\neq 0,\boldsymbol{y},\boldsymbol{q}^i) = \mathcal{N}(x_n;\hat{r}_n,\mu^r_n)\sum_\ell\omega^i_\ell\mathcal{N}(x_n;\theta^i_\ell,\phi^i_\ell)$ to cancel terms, giving

$$\sum_{n=1}^N \int_{x_n} \frac{\lambda^i\omega^i_k\mathcal{N}(x_n;\hat{r}_n,\mu^r_n)\mathcal{N}(x_n;\theta^i_k,\phi^i_k)}{\zeta_n(\hat{r}_n,\mu^r_n;\boldsymbol{q}^i)}(x_n-\theta_k) = 0. \quad (29)$$

We then simplify (29) using the Gaussian-pdf multiplication rule,[4] and set θ^{i+1}_k equal to the value of θ_k satisfying (29):

$$\theta^{i+1}_k = \frac{\sum_{n=1}^N \Pr\{x_n\neq 0, k_n = k\,|\,\boldsymbol{y},\boldsymbol{q}^i\}\gamma_k(\hat{r},\mu^r;\boldsymbol{q}^i)}{\sum_{n=1}^N \Pr\{x_n\neq 0, k_n = k\,|\,\boldsymbol{y},\boldsymbol{q}^i\}}, \quad (30)$$

where the joint activity/mixture probabilities are

$$\Pr\{x_n\neq 0, k_n = k\,|\,\boldsymbol{y},\boldsymbol{q}^i\} \triangleq \frac{\beta_k(\hat{r}_n,\mu^r_n;\boldsymbol{q}^i)}{(1-\lambda)\mathcal{N}(0;\hat{r}_n,\mu^r_n)+\sum_{\ell=1}^L\beta_\ell(\hat{r}_n,\mu^r_n;\boldsymbol{q}^i)} \quad (31)$$

with $\beta_k(\hat{r},\mu^r;\boldsymbol{q}^i)$ and $\gamma_k(\hat{r},\mu^r;\boldsymbol{q}^i)$ defined in (9)-(10). Above, "$k_n = k$" represents the event that x_n was generated from mixture component k.

Following (26), the maximizing value of ϕ_k in (24) is necessarily a value of ϕ_k that zeroes the derivative

$$\sum_{n=1}^N \int_{x_n} p(x_n|\boldsymbol{y},\boldsymbol{q}^i)\frac{d}{d\phi_k}\ln p_X(x_n;\phi_k,\boldsymbol{q}^i_{\backslash\phi_k}) = 0. \quad (32)$$

Taking the derivative, we find

$$\frac{d}{d\phi_k}\ln p_X(x_n;\phi_k,\boldsymbol{q}^i_{\backslash\phi_k}) = \frac{1}{2}\Big(\tfrac{|x_n-\theta_k^i|^2}{\phi_k^2}-\tfrac{1}{\phi_k}\Big) \times \tfrac{\lambda^i\omega_k^i\mathcal{N}(x_n;\theta_k^i,\phi_k)}{(1-\lambda^i)\delta(x_n)+\lambda^i(\omega_k^i\mathcal{N}(x_n;\theta_k^i,\phi_k)+\sum_{\ell\neq k}\omega_\ell^i\mathcal{N}(x_n;\theta_\ell^i,\phi_\ell^i))}. \quad (33)$$

Integrating (32) separately over $\mathcal{B}_\epsilon$ and $\overline{\mathcal{B}_\epsilon}$, as in (21), and taking $\epsilon\to 0$, we find that the $\mathcal{B}_\epsilon$ portion vanishes, giving

$$\sum_{n=1}^N\int_{x_n}\tfrac{p(x_n|x_n\neq 0,\boldsymbol{y},\boldsymbol{q}^i)\lambda^i\omega_k^i\mathcal{N}(x_n;\theta_k^i,\phi_k)}{\zeta_n(\hat{r}_n,\mu_n^r;\boldsymbol{q}^i)(\omega_k^i\mathcal{N}(x_n;\theta_k^i,\phi_k)+\sum_{\ell\neq k}\omega_\ell^i\mathcal{N}(x_n;\theta_\ell^i,\phi_\ell^i))} \times\Big(\tfrac{|x_n-\theta_k^i|^2}{\phi_k}-1\Big)=0. \quad (34)$$

Similar to (28), this integral is difficult to evaluate, and so we apply the approximation $\mathcal{N}(x_n;\theta_k^i,\phi_k)\approx\mathcal{N}(x_n;\theta_k^i,\phi_k^i)$, after which certain terms cancel, yielding

$$\sum_{n=1}^N\int_{x_n}\tfrac{\mathcal{N}(x_n;\hat{r}_n,\mu_n^r)\lambda^i\omega_k^i\mathcal{N}(x_n;\theta_k^i,\phi_k^i)}{\zeta_n(\hat{r}_n,\mu_n^r;\boldsymbol{q}^i)}\Big(\tfrac{|x_n-\theta_k^i|^2}{\phi_k}-1\Big)=0. \quad (35)$$

To find the value of ϕ_k satisfying (35), we expand $|x_n-\theta_k^i|^2=|x_n|^2-2\,\mathrm{Re}(x_n^*\theta_k^i)+|\theta_k^i|^2$ and apply the Gaussian-pdf multiplication rule,[4] which gives

$$\phi_k^{i+1}=\tfrac{\sum_{n=1}^N\Pr\{x_n\neq 0,k_n=k|\boldsymbol{y},\boldsymbol{q}^i\}(|\theta_k^i-\gamma_k(\hat{r},\mu^r;\boldsymbol{q}^i)|^2+\nu_k(\hat{r},\mu^r;\boldsymbol{q}))}{\sum_{n=1}^N\Pr\{x_n\neq 0,k_n=k|\boldsymbol{y},\boldsymbol{q}^i\}} \quad (36)$$

where $\Pr\{x_n\neq 0,k|\boldsymbol{y},\boldsymbol{q}^i\}$ was given in (31).

Finally, the value of the pmf-constrained $\boldsymbol{\omega}$ maximizing (25) can be found by solving the unconstrained optimization problem $\max_{\boldsymbol{\omega},\xi}J(\boldsymbol{\omega},\xi)$, where ξ is a Lagrange multiplier and

$$J(\boldsymbol{\omega},\xi)\triangleq\sum_{n=1}^N\mathrm{E}\left\{\ln p_X(x_n;\boldsymbol{\omega},\boldsymbol{q}^i_{\backslash\boldsymbol{\omega}})\,\middle|\,\boldsymbol{y},\boldsymbol{q}^i\right\}-\xi\Big(\sum_{\ell=1}^L\omega_\ell-1\Big). \quad (37)$$

We start by setting $\frac{d}{d\omega_k}J(\boldsymbol{\omega},\xi)=0$, which yields

$$\sum_{n=1}^N\int_{x_n}\frac{p_X(x_n;\boldsymbol{q}^i)\mathcal{N}(x_n;\hat{r}_n,\mu_n^r)}{\zeta(\hat{r}_n,\mu_n^r;\boldsymbol{q}^i)}\frac{\lambda^i\mathcal{N}(x_n;\theta_k^i,\phi_k^i)}{p_X(x_n;\boldsymbol{\omega},\boldsymbol{q}^i_{\backslash\boldsymbol{\omega}})}=\xi. \quad (38)$$

Like (28) and (34), the above is difficult to evaluate, and so we approximate $\boldsymbol{\omega}\approx\boldsymbol{\omega}^i$, which leads to

$$\xi=\sum_{n=1}^N\int_{x_n}\frac{\lambda^i\mathcal{N}(x_n;\theta_k^i,\phi_k^i)\mathcal{N}(x_n;\hat{r}_n,\mu_n^r)}{\zeta(\hat{r}_n,\mu_n^r;\boldsymbol{q}^i)}. \quad (39)$$

Multiplying both sides by ω_k^i, summing over $k=1,\dots,L$, employing the fact $1=\sum_k\omega_k^i$, and simplifying, we obtain

$$\xi=\sum_{n=1}^N\int_{x_n}\frac{\lambda^i\sum_{k=1}^L\omega_k^i\mathcal{N}(x_n;\theta_k^i,\phi_k^i)\mathcal{N}(x_n;\hat{r}_n,\mu_n^r)}{\zeta(\hat{r}_n,\mu_n^r;\boldsymbol{q}^i)} \quad (40)$$

$$=\sum_{n=1}^N\Pr\{x_n\neq 0\,|\,\boldsymbol{y};\boldsymbol{q}^i\}. \quad (41)$$

Plugging (41) into (39) and multiplying both sides by ω_k, the derivative-zeroing value of ω_k is seen to be

$$\omega_k=\tfrac{\sum_{n=1}^N\int_{x_n}\lambda^i\omega_k\mathcal{N}(x_n;\theta_k^i,\phi_k^i)\mathcal{N}(x_n;\hat{r}_n,\mu_n^r)/\zeta(\hat{r}_n,\mu_n^r;\boldsymbol{q}^i)}{\sum_{n=1}^N\Pr\{x_n\neq 0\,|\,\boldsymbol{y};\boldsymbol{q}^i\}}, \quad (42)$$

where, if we use $\omega_k\approx\omega_k^i$ on the right of (42), then we obtain

$$\omega_k^{i+1}=\frac{\sum_{n=1}^N\Pr\{x_n\neq 0,k_n=k\,|\,\boldsymbol{y};\boldsymbol{q}^i\}}{\sum_{n=1}^N\Pr\{x_n\neq 0\,|\,\boldsymbol{y};\boldsymbol{q}^i\}}. \quad (43)$$

Note that the numerator and denominator of (43) can be computed from GM-GAMP outputs via (16) and (31).

C. EM update for ψ

The final parameter to estimate is the noise energy ψ. In this paper, the AWGN noise model given in (4) is identical to EM-BG-GAMP's [12]. There, the (exact) EM update for ψ is[5]

$$\psi^{i+1}=\frac{1}{M}\sum_{m=1}^M\big(|y_m-\hat{z}_m|^2+\mu_m^z\big). \quad (44)$$

IV. Learning the Model Order

So far, the GM model order L has been treated as fixed and known. In practice, one could indeed choose a fixed value L that is thought to be large enough to capture the essential structure of $p_X(\cdot)$ and, given an appropriate initialization of the $3L+2$ parameters $\boldsymbol{q}^0$, apply the previously described EM-GM-GAMP algorithm to jointly estimate $\boldsymbol{x}$ and $\boldsymbol{q}$.

As an alternative, one could instead start with the model order $L=1$ (i.e., a Bernoulli-Gaussian model for $p_X(\cdot)$) and increment L one-by-one, stopping as soon as negligible benefits are observed (e.g., $\|\hat{\boldsymbol{x}}_L-\hat{\boldsymbol{x}}_{L-1}\|_2^2/\|\hat{\boldsymbol{x}}_{L-1}\|_2^2<\texttt{tol}$) or a predefined $L_{\max}$ has been reached. Here, EM-GM-GAMP would be re-run as described in Sections II-III at each new value of L. This latter approach would relieve the user from the potentially difficult task of choosing both L and a many-parameter $\boldsymbol{q}^0$ apriori. In the remainder of this section, we propose a particular implementation of this approach.

When growing the model-order from L to $L+1$, we propose to split the mixture component $k_*\in\{1,\dots,L\}$ with the "worst fit" into two new components. To select the worst-fitting mixture component, one could use the approach of split-and-merge-EM [14], i.e., maximization of local Kullback-Leibler divergence, or similar. Given the mixture-component-to-split k_*, the subset of coefficient indices n that are most probably associated with k_* is identified, i.e.,

$$\mathfrak{N}_{k_*}\triangleq\big\{n\in\mathfrak{N}\colon\arg\max_k\Pr\{x_n\neq 0,k_n=k\,|\,\boldsymbol{y},\boldsymbol{q}\}=k_*\big\}, \quad (45)$$

where $\mathfrak{N}\triangleq\big\{n:\Pr\{x_n\neq 0\,|\,\boldsymbol{y},\boldsymbol{q}\}>0.5\big\}$ is the subset of coefficient indices that are most probably non-zero. To simplify the notation, we henceforth assume, without loss of generality, that $k_*=L$.

To split the L^{th} mixture component, we replace the mean θ_L, variance ϕ_L, and weight ω_L with two new values for each (e.g., θ_L^{new} and $\theta_{L+1}^{\mathsf{new}}$ replace θ_L), resulting in an $L+1$-term parameter vector $\boldsymbol{q}^{\mathsf{new}}$. Rather than considering only a single possibility for $\boldsymbol{q}^{\mathsf{new}}$, we consider S possibilities $\{\boldsymbol{q}_s^{\mathsf{new}}\}_{s=1}^S$ obtained as variations of the following two strategies:

[5] Sometimes we observe that the EM update for ψ works better with the μ_m^z term in (44) weighted by $\frac{M}{N}$ and suppressed until later EM iterations. We conjecture that this is due to bias in the GAMP variance estimates μ_m^z.

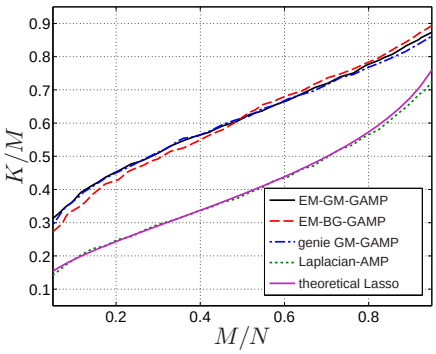

Fig. 1. Noiseless empirical PTCs and Lasso theoretical PTC for Bernoulli-Gaussian signals.

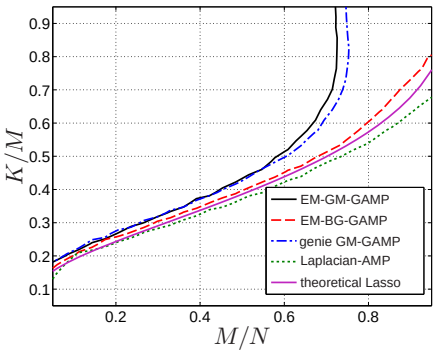

Fig. 2. Noiseless empirical PTCs and Lasso theoretical PTC for Bernoulli-Rademacher signals.

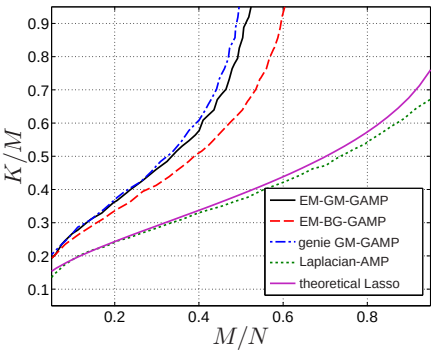

Fig. 3. Noiseless empirical PTCs and Lasso theoretical PTC for Bernoulli signals.

1) Split-mean(a): $\theta_L^{\text{new}} = \theta_L - a$, $\theta_{L+1}^{\text{new}} = \theta_L + a$, $\phi_L^{\text{new}} = \phi_{L+1}^{\text{new}} = \phi_L$, and $\omega_L^{\text{new}} = \omega_{L+1}^{\text{new}} = \omega_L/2$; and
2) Split-variance(b): $\phi_L^{\text{new}} = b\phi_L$ and $\phi_{L+1}^{\text{new}} = b^{-1}\phi_L$, $\theta_L^{\text{new}} = \theta_{L+1}^{\text{new}} = \theta_L$, and $\omega_L^{\text{new}} = \omega_{L+1}^{\text{new}} = \omega_L/2$,

where $a, b > 0$ are design parameters. Note that, by considering several distinct values of a and/or b, we have $S > 2$. Finally, to judge which of the S possible splits is best, one could, e.g., evaluate the corresponding likelihoods, i.e., solve

$$\arg\max_s \sum_n \mathrm{E}\left\{ \ln p_X(\boldsymbol{x}; \boldsymbol{q}_s^{\text{new}}) \,\middle|\, \boldsymbol{y}; \boldsymbol{q} \right\}. \tag{46}$$

Empirically, we have found that the incremental method of learning L described above[6] works very well for "sparse" signals like Bernoulli-Gaussian, Bernoulli-Rademacher, Bernoulli-Uniform, and Bernoulli. (See below for details.) For "heavy-tailed" signals like Student-t, however, it seems better to fix L at a reasonable value (e.g., $L = 4$), keep the means at zero (i.e., $\theta_k^i = 0 \ \forall k, i$), and jointly learn the L weights $\boldsymbol{\omega}$, the L variances $\boldsymbol{\phi}$, and the sparsity rate λ.

V. EM Initialization

Since the EM algorithm is guaranteed to converge only to a local maximum of the likelihood function, proper initialization of $\boldsymbol{q}$ is essential. Here, we describe initialization strategies for the "sparse" and "heavy-tailed" modes described above.

For the "sparse" mode, where initially $L = 1$ (and thus $\omega_1^0 = 0$), we use the same initializations that we proposed for EM-BG-GAMP [12], i.e., $\lambda^0 = \frac{M}{N}\rho_{\text{SE}}(\frac{M}{N})$, where $\rho_{\text{SE}}(\frac{M}{N})$ is the sparsity ratio $\frac{K}{M}$ achieved by the noiseless Lasso PTC [3]

$$\rho_{\text{SE}}(\tfrac{M}{N}) = \max_{c \geq 0} \frac{1 - \frac{2N}{M}[(1+c^2)\Phi(c) - c\phi(c)]}{1 - c^2 - 2[(1+c^2)\Phi(c) - c\phi(c)]}, \tag{47}$$

with $\Phi(\cdot)$ and $\phi(\cdot)$ the cdf and pdf of the normal distribution, respectively, and

$$\psi^0 = \frac{\|\boldsymbol{y}\|_2^2}{(\text{SNR}^0 + 1)M}, \ \phi_1^0 = \frac{\|\boldsymbol{y}\|_2^2 - M\psi^0}{\text{tr}(\boldsymbol{A}^{\mathsf{T}}\boldsymbol{A})\lambda^0}, \ \theta_1^0 = 0, \tag{48}$$

where, without other knowledge, we suggest $\text{SNR}^0 = 100$.

For the "heavy-tailed" mode, we suggest initializing λ^0 and ψ^0 as above and, with $L = 4$, choosing

$$\omega_k^0 = \frac{1}{L}, \ \phi_k^0 = \frac{k}{\sqrt{L}} \frac{(\|\boldsymbol{y}\|_2^2 - M\psi^0)}{\text{tr}(\boldsymbol{A}^{\mathsf{T}}\boldsymbol{A})\lambda^0}, \ \theta_k^0 = 0, \ k = 1 \ldots L. \tag{49}$$

[6]For the simulations, we used $S = 3$ splitting methods in the "sparse mode": $a = \sqrt{\phi_L}, b_1 = 3, b_2 = 6$.

VI. Numerical Results

A. Noiseless Phase Transitions

First, we describe the results of experiments that computed noiseless empirical phase transition curves (PTCs) under three sparse-signal distributions. To evaluate each empirical PTC, we constructed a 30×30 grid of oversampling ratio $\frac{M}{N} \in [0.05, 0.95]$ and sparsity ratio $\frac{K}{M} \in [0.05, 0.95]$ for fixed signal length $N = 1000$. At each grid point, we generated $R = 100$ independent realizations of K-sparse signal $\boldsymbol{x}$ and $M \times N$ measurement matrix with i.i.d $\mathcal{N}(0, M^{-1})$ entries. From the measurements $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}$, we attempted to reconstruct the signal $\boldsymbol{x}$ using various algorithms. A recovery $\hat{\boldsymbol{x}}$ from realization $r \in \{1, \ldots, R\}$ was defined a success (i.e., $S_r = 1$) if the NMSE $\triangleq \|\boldsymbol{x} - \hat{\boldsymbol{x}}\|_2^2 / \|\boldsymbol{x}\|_2^2 < 10^{-4}$, and the average success rate was defined as $\overline{S} \triangleq \frac{1}{R}\sum_{r=1}^R S_r$. The empirical PTC was then plotted, using Matlab's `contour` command, as the $\overline{S} = 0.5$ contour over the sparsity-undersampling grid.

Figures 1-3 show the empirical PTCs for four recovery algorithms: the proposed EM-GM-GAMP algorithm[7] (in "sparse" mode), the EM-BG-GAMP algorithm from [12], a "genie-aided" GM-GAMP with the true $[\lambda, \boldsymbol{\omega}, \boldsymbol{\theta}, \boldsymbol{\phi}, \psi]$, and the Laplacian-AMP from [3]. For comparison, Figs. 1-3 also display the theoretical Lasso PTC (47). The signals were generated as Bernoulli-Gaussian (BG) in Fig. 1 ($\theta = 0$, $\phi = 1$), as Bernoulli-Rademacher (BR) in Fig. 2 (i.e., non-zero coefficients chosen uniformly from $\{-1, 1\}$), and as Bernoulli in Fig. 3 (i.e., all non-zero coefficients set equal to 1).

For all three signal types, Figs. 1-3 show that the empirical PTC of EM-GM-GAMP significantly improves on those of Laplacian-AMP and theoretical Lasso. (The latter two converge in the large system limit [3].) For Bernoulli-Gaussian signals, EM-GM-GAMP performed very similarly to genie-GM-GAMP and EM-BG-GAMP. Such behavior is expected, because all three can accurately model the signal distribution. For BR signals, however, EM-GM-GAMP performed significantly better than EM-BG-GAMP, since it can better model the BR distribution (using an $L = 2$ GM). It even performed slightly better than genie-GM-GAMP here, since it is able to perform realization-specific parameter fitting. For Bernoulli signals, EM-GM-GAMP performed moderately better than EM-BG-GAMP, and nearly the same as genie-GM-GAMP.

[7]Matlab code available at http://www.ece.osu.edu/~schniter/EMturboGAMP

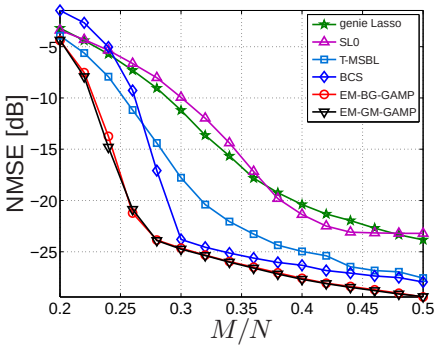

Fig. 4. NMSE for noisy recovery of Bernoulli-Gaussian signals.

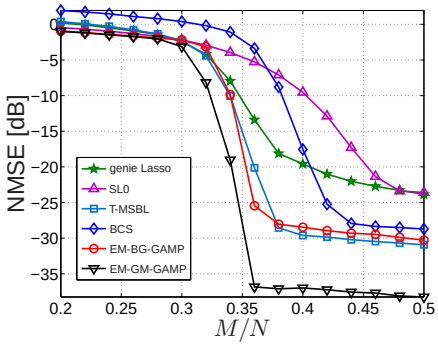

Fig. 5. NMSE for noisy recovery of Bernoulli-Rademacher signals.

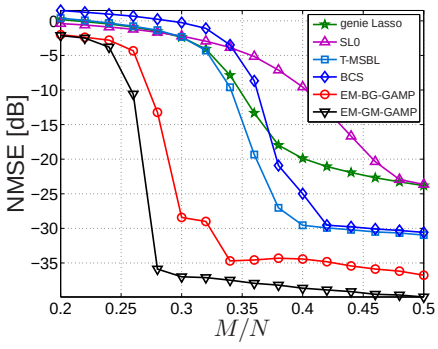

Fig. 6. NMSE for noisy recovery of Bernoulli signals.

B. Noisy Signal Recovery

Figures 4-6 show NMSE for noisy recovery of BG, BR, and Bernoulli signals. To construct these plots, we fixed $N = 1000$, $K = 100$, $\mathsf{SNR} = 25$dB, and varied M. Each data point represents NMSE averaged over $R = 500$ realizations. For comparison, we show the performance of the proposed EM-GM-GAMP (in "sparse" mode), EM-BG-GAMP [12], Bayesian Compressive Sensing (BCS) [11], Sparse Bayesian Learning [10] (via T-MSBL [15]), debiased genie-aided[8] Lasso (via SPGL1 [16]), and Smoothed-ℓ_0 (SL0) [17]. All algorithms were run under the suggested defaults, with `'noise=small'` in T-MSBL.

For BG signals, Fig. 4 shows that EM-GM-GAMP exhibits the best performance (together with EM-BG-AMP). For BR and Bernoulli signals, however, Figs. 5-6 show that EM-GM-GAMP significantly outperforms the other algorithms. Relative to EM-BG-GAMP, EM-GM-GAMP's greatest improvement comes with BR signals, which are not well-modeled using a BG prior. We have verified, using all three signal types, that EM-GM-GAMP's excellent behavior persists at lower SNRs, as well as on sparse L-ary discrete signals with $L > 2$.

Perhaps most impressive is EM-GM-GAMP's performance in recovering heavy-tailed signals. As an example, Fig. 7 shows noisy recovery NMSE for a Student's-t signal with pdf

$$p_X(x;q) \triangleq \frac{\Gamma((q+1)/2))}{\sqrt{2\pi}\Gamma(q/2)} \left(1+x^2\right)^{-(q+1)/2} \qquad (50)$$

under the *non-compressible* parameter choice $q = 1.67$ [18]. Here, EM-GM-GAMP was run in "heavy-tailed" mode and outperformed all other algorithms under test—even genie-aided Lasso. Although the algorithms that perform best on the sparse signals in Figs. 4-6 usually perform worst on heavy-tailed signals like that in Fig. 7, and vice versa, Figs. 4-7 show EM-GM-GAMP excelling on *all* signal types.

In conclusion, we attribute EM-GM-GAMP's excellent performance, on a wide range of signal types, to its ability to near-optimally learn and exploit a wide range of signal priors.

REFERENCES

[1] Y. C. Eldar and G. Kutyniok, *Compressed Sensing: Theory and Applications*. New York: Cambridge Univ. Press, 2012.

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

[8]We ran SPGL1 in 'BPDN' mode: $\min_{\hat{\boldsymbol{x}}} \|\boldsymbol{x}\|_1$ s.t. $\|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|_2 \leq \sigma$, for tolerances $\sigma^2 \in \{0.1, 0.2, \ldots, 1.5\} \times M\psi$, and reported the lowest NMSE.

[3] 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.

[4] D. L. Donoho, A. Maleki, and A. Montanari, "The noise-sensitivity phase transition in compressed sensing," *arXiv:1004.1218*, Apr. 2010.

[5] Y. Wu and S. Verdú, "Optimal phase transitions in compressed sensing," *arXiv:1111.6822*, Nov. 2011.

[6] D. L. Donoho, A. Maleki, and A. Montanari, "Message passing algorithms for compressed sensing: I. Motivation and construction," in *Proc. Inform. Theory Workshop*, (Cairo, Egypt), Jan. 2010.

[7] S. Rangan, "Generalized approximate message passing for estimation with random linear mixing," *arXiv:1010.5141*, Oct. 2010.

[8] 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.

[9] M. E. Tipping, "Sparse Bayesian learning and the relevance vector machine," *J. Mach. Learn. Res.*, vol. 1, pp. 211–244, 2001.

[10] D. P. Wipf and B. D. Rao, "Sparse Bayesian learning for basis selection," *IEEE Trans. Signal Process.*, vol. 52, pp. 2153 – 2164, Aug. 2004.

[11] S. Ji, Y. Xue, and L. Carin, "Bayesian compressive sensing," *IEEE Trans. Signal Process.*, vol. 56, pp. 2346–2356, June 2008.

[12] J. P. Vila and P. Schniter, "Expectation-maximization Bernoulli-Gaussian approximate message passing," in *Proc. Asilomar Conf. Signals Syst. Comput.*, (Pacific Grove, CA), Nov. 2011.

[13] 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.), pp. 355–368, MIT Press, 1999.

[14] N. Ueda, R. Nakano, Z. Ghahramani, and G. E. Hinton, "SMEM algorithm for mixture models," *Neural Comput.*, vol. 12, pp. 2109–2128, Sept. 2000.

[15] Z. Zhang and B. D. Rao, "Sparse signal recovery with temporally correlated source vectors using sparse Bayesian learning," *IEEE J. Sel. Topics Signal Process.*, vol. 5, pp. 912–926, Sept. 2011.

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

[17] H. Mohimani, M. Babaie-Zadeh, and C. Jutten, "A fast approach for overcomplete sparse decomposition based on smoothed norm," *IEEE Trans. Signal Process.*, vol. 57, pp. 289–301, Jan. 2009.

[18] V. Cevher, "Learning with compressible priors," in *Proc. Neural Inform. Process. Syst. Conf.*, (Vancouver, B.C.), Dec. 2009.

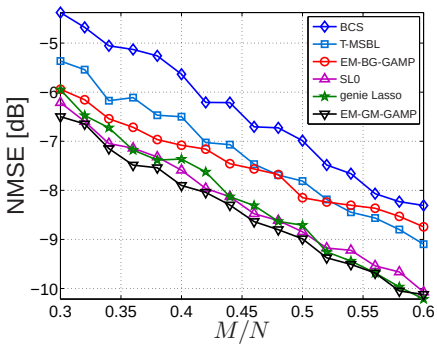

Fig. 7. NMSE for noisy recovery of non-compressible Student's-t signals.