Bilinear Generalized Approximate Message Passing—Part II: Applications

Jason T. Parker, *Member, IEEE*, Philip Schniter, *Fellow, IEEE*, and Volkan Cevher, *Senior Member, IEEE*

***Abstract*—In this paper, we extend the generalized approximate message passing (G-AMP) approach, originally proposed for high-dimensional generalized-linear regression in the context of compressive sensing, to the generalized-bilinear case. In Part I of this two-part paper, we derived our Bilinear G-AMP (BiG-AMP) algorithm as an approximation of the sum-product belief propagation algorithm in the high-dimensional limit, and proposed an adaptive damping mechanism that aids convergence under finite problem sizes, an expectation-maximization (EM)-based method to automatically tune the parameters of the assumed priors, and two rank-selection strategies. Here, in Part II, we discuss the specializations of BiG-AMP to the problems of matrix completion, robust PCA, and dictionary learning, and present the results of an extensive empirical study comparing BiG-AMP to state-of-the-art algorithms on each problem. Our numerical results, using both synthetic and real-world datasets, demonstrate that EM-BiG-AMP yields excellent reconstruction accuracy (often best in class) while maintaining competitive runtimes.**

***Index Terms*— Approximate message passing, belief propagation, bilinear estimation, matrix completion, dictionary learning robust principal components analysis, matrix factorization.**

I. Introduction

THIS manuscript is Part II of a two-part work on *Bilinear Generalized Approximate Message Passing* (BiG-AMP), which is an extension of the AMP framework that was originally proposed for linear [5], [6] and generalized linear [7] inference problems encountered in compressive sensing (CS), to the following *generalized bilinear* inference problem: estimate the matrices $\boldsymbol{A} = [a_{mn}] \in \mathbb{R}^{M\times N}$ and $\boldsymbol{X} = [x_{nl}] \in \mathbb{R}^{N\times L}$ from an observation $\boldsymbol{Y} \in \mathbb{R}^{M\times L}$ that is statistically coupled to their product, $\boldsymbol{Z} \triangleq \boldsymbol{AX}$, where $\boldsymbol{A}$ and $\boldsymbol{X}$ are realizations of random matrices $\boldsymbol{A}$ and $\boldsymbol{X}$ with known separable pdfs (or pmfs)

$$p_{\boldsymbol{A}}(\boldsymbol{A}) = \prod_m \prod_n p_{\mathsf{a}_{mn}}(a_{mn}) \tag{1}$$

$$p_{\boldsymbol{X}}(\boldsymbol{X}) = \prod_n \prod_l p_{\mathsf{x}_{nl}}(x_{nl}), \tag{2}$$

and where the likelihood function is known and separable, i.e.,

$$p_{\boldsymbol{Y}|\boldsymbol{Z}}(\boldsymbol{Y} \mid \boldsymbol{Z}) = \prod_m \prod_l p_{\mathsf{y}_{ml}|\mathsf{z}_{ml}}(y_{ml}|z_{ml}). \tag{3}$$

In Part I of the work [1], we proposed and derived the BiG-AMP algorithm, whose general form is summarized in [1, TABLE III]. We also uncovered special cases under which the general approach can be simplified, such as the scalar-variance BiG-AMP under possibly incomplete additive white Gaussian noise (PIAWGN) observations, as summarized in [1, TABLE IV], and its specialization to Gaussian priors, as summarized by the BiG-AMP-Lite algorithm in [1, Table V]. In addition, Part I proposed an adaptive damping mechanism, an expectation-maximization (EM)-based method of tuning the parameters of $p_{\boldsymbol{A}}$, $p_{\boldsymbol{X}}$, and $p_{\boldsymbol{Y}|\boldsymbol{Z}}$ (in case they are unknown), and two methods to select the rank N (in case it is unknown).

In this Part II of the work, we detail the application of BiG-AMP to the problems of matrix completion (MC) in Section II, robust principle components analysis (RPCA) in Section III, and dictionary learning (DL) in Section IV. For each application, we discuss the BiG-AMP's choice of matrix representation, priors, likelihood, initialization, adaptive damping, EM-driven parameter learning, and rank-selection. Also, for each application, we provide an extensive empirical study comparing BiG-AMP to state-of-the-art solvers on both synthetic and real-world datasets. These results demonstrate that BiG-AMP yields excellent reconstruction performance (often best in class) while maintaining competitive runtimes. For each application of BiG-AMP discussed in the sequel, we recommend numerical settings for necessary parameter values, as well as initialization strategies when appropriate. Although we cannot guarantee that our recommendations are universally optimal, they worked well for the range of problems considered in this paper, and we conjecture that they offer a useful starting point for further experimentation. Nevertheless, modifications may be appropriate when applying BiG-AMP outside the range of problems considered here. Our BiG-AMP Matlab code can be found as part of the GAMPmatlab package at

Manuscript received December 09, 2013; revised June 05, 2014; accepted August 19, 2014. Date of publication September 12, 2014; date of current version October 22, 2014. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Ivan W. Selesnick. The work of J. T. Parker on this project has been supported by AFOSR Lab Task 11RY02COR. The work of P. Schniter on this project has been supported by NSF grants IIP-0968910, CCF-1018368, CCF-1218754, by DARPA/ONR grant N66001-10-1-4090, and by an allocation of computing time from the Ohio Supercomputer Center. The work of V. Cevher was supported in part by the European Commission under Grant MIRG-268398, ERC Future Proof, SNF 200021-132548, SNF 200021-146750 and SNF CRSII2-147633. Portions of this work were presented at the Workshop on Signal Processing with Adaptive Sparse Structured Representations, Edinburgh, Scotland, June 2011 [2], the Information Theory and Applications Workshop, La Jolla, CA, USA, February 2012 [3], and at the Asilomar Conference on Signals, Systems, and Computers, Pacific Grove, CA, USA, November 2012 [4].

J. T. Parker is with the Sensors Directorate, Air Force Research Laboratory, Dayton, OH 45433 USA (e-mail: jason.parker.13@us.af.mil).

P. Schniter is with The Ohio State University, Columbus OH 43210 USA (e-mail: schniter@ece.osu.edu).

V. Cevher is with École Polytechnique Fédérale de Lausanne, Lausanne 1015, Switzerland (e-mail: volkan.cevher@epfl.ch).

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.2014.2357773

https://sourceforge.net/projects/gampmatlab/, including examples of BiG-AMP applied to the MC, RPCA, and DL problems.

Notation: Throughout, we use san-serif font (e.g., x) for random variables and serif font (e.g., x) otherwise. We use boldface capital letters (e.g., $\boldsymbol{\mathsf{X}}$ and $\boldsymbol{X}$) for matrices, boldface small letters (e.g., $\boldsymbol{\mathsf{x}}$ and $\boldsymbol{x}$) for vectors, and non-bold small letters (e.g., x and x) for scalars. We then use $p_{\mathsf{x}}(x)$ to denote the pdf of random quantity x, and $\mathcal{N}(x;\widehat{x},\nu^x)$ to denote the Gaussian pdf for a scalar random variable with mean $\widehat{x}$ and variance ν^x. Also, we use $\mathrm{E}\{\mathsf{x}\}$ and $\mathrm{var}\{\mathsf{x}\}$ to denote mean and variance of x, respectively, and $D(p_1\|p_2)$ for the Kullback-Leibler (KL) divergence between pdfs p_1 and p_2. For a matrix $\boldsymbol{X}$, we use $x_{nl} = [\boldsymbol{X}]_{nl}$ to denote the entry in the n^{th} row and l^{th} column, $\|\boldsymbol{X}\|_F$ to denote the Frobenius norm, and $\boldsymbol{X}^\top$ to denote transpose. Similarly, we use x_n to denote the n^{th} entry in vector $\boldsymbol{x}$ and $\|\boldsymbol{x}\|_p = \left(\sum_n |x_n|^p\right)^{1/p}$ to denote the ℓ_p norm.

II. Matrix Completion

A. Problem Setup

In matrix completion (MC) [8], one seeks to recover a rank-$N \ll \min(M,L)$ matrix $\boldsymbol{Z} \in \mathbb{R}^{M\times L}$ after observing a fraction $\delta = \frac{|\Omega|}{ML}$ of its (possibly noise-corrupted) entries, where Ω denotes the set of observations.

BiG-AMP approaches the MC problem by modeling the complete matrix $\boldsymbol{Z}$ as the product $\boldsymbol{Z} = \boldsymbol{A}\boldsymbol{X}$ of random matrices $\boldsymbol{A} \in \mathbb{R}^{M\times N}$ and $\boldsymbol{X} \in \mathbb{R}^{N\times L}$ with priors of the decoupled form in (1), (2), where $\boldsymbol{Z}$ is probabilistically related to the observed matrix $\boldsymbol{Y}$ through a likelihood $p_{\boldsymbol{Y}|\boldsymbol{Z}}(\boldsymbol{Y}|\boldsymbol{Z})$ of the decoupled form in (3). To finally perform MC, BiG-AMP infers $\boldsymbol{A}$ and $\boldsymbol{X}$ from $\boldsymbol{Y}$ under the above model. The corresponding estimates $\widehat{\boldsymbol{A}}$ and $\widehat{\boldsymbol{X}}$ can then be multiplied to yield an estimate $\widehat{\boldsymbol{Z}} = \widehat{\boldsymbol{A}}\widehat{\boldsymbol{X}}$ of the noiseless complete matrix $\boldsymbol{Z}$.

As in several existing Bayesian approaches to matrix completion (e.g., [9]–[12]), we choose Gaussian priors for the factors $\boldsymbol{A}$ and $\boldsymbol{X}$. Although EM-BiG-AMP readily supports the use of priors with row- and/or column-dependent parameters, we focus on simple iid priors of the form

$$p_{\mathsf{a}_{mn}}(a) = \mathcal{N}(a;0,1) \quad \forall m,n \tag{4}$$

$$p_{\mathsf{x}_{nl}}(x) = \mathcal{N}(x;\widehat{x}_0,\nu_0^x) \quad \forall n,l, \tag{5}$$

where the mean and variance in (5) can be tuned using EM-BiG-AMP, as described in the sequel, and where the variance in (4) is fixed to avoid a scaling ambiguity between $\boldsymbol{A}$ and $\boldsymbol{X}$. Section II-F demonstrates that this simple approach is effective in attacking several MC problems of interest. Assuming the observation noise to be additive and Gaussian, we then choose the PIAWGN model from [1, Eq. 80] for the likelihood $p_{\boldsymbol{Y}|\boldsymbol{Z}}$ given by

$$p_{\mathsf{y}_{ml}|\mathsf{z}_{ml}}(y_{ml}|z_{ml}) = \begin{cases} \mathcal{N}(y_{ml};z_{ml},\nu^w) & (m,l)\in\Omega \\ \mathbb{1}_{y_{ml}} & (m,l)\notin\Omega. \end{cases} \tag{6}$$

Note that, by using (4), (5) with $\widehat{x}_0 = 0$ and the scalar-variance approximation from [1, Sec. III-A], the BiG-AMP algorithm from [1, Table III] reduces to the simpler BiG-AMP-Lite algorithm from [1, Table V] with $\nu_0^a = 1$.

B. Initialization

In most cases we advocate initializing the BiG-AMP quantities $\widehat{\boldsymbol{X}}(1)$ and $\widehat{\boldsymbol{A}}(1)$ using random draws from the priors $p_{\boldsymbol{X}}$ and $p_{\boldsymbol{A}}$, although setting either $\widehat{\boldsymbol{X}}(1)$ or $\widehat{\boldsymbol{A}}(1)$ at zero also seems to perform well in the MC application. Although it is also possible to use SVD-based initializations of $\widehat{\boldsymbol{X}}(1)$ and $\widehat{\boldsymbol{A}}(1)$ (i.e., for SVD $\boldsymbol{Y} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{D}^T$, set $\widehat{\boldsymbol{A}}(1) = \boldsymbol{U}\boldsymbol{\Sigma}^{1/2}$ and $\widehat{\boldsymbol{X}}(1) = \boldsymbol{\Sigma}^{1/2}\boldsymbol{D}^T$) as done in LMaFit [13] and VSBL [14], experiments suggest that the extra computation required is rarely worthwhile for BiG-AMP.

As for the initializations $\nu_{nl}^x(1)$ and $\nu_{mn}^a(1)$, we advocate setting them at 10 times the prior variances in (4), (5), which has the effect of weighting the measurements $\boldsymbol{Y}$ more than the priors $p_{\boldsymbol{X}}, p_{\boldsymbol{A}}$ during the first few iterations.

C. Adaptive Damping

For the assumed likelihood (6) and priors (4), (5), the adaptive-damping cost criterion $\widehat{J}(t)$ described in [1, Sec. IV-B] reduces to

$$\begin{aligned}\widehat{J}(t) = &\frac{1}{2}\sum_{n,l}\left[\log\frac{\nu_0^x}{\nu^x(t)} + \left(\frac{\nu^x(t)}{\nu_0^x}-1\right) + \frac{\left(\widehat{x}_{nl}(t)-\widehat{x}_0\right)^2}{\nu_0^x}\right] \\ &+ \frac{1}{2}\sum_{m,n}\left[\log\frac{1}{\nu^a(t)} + (\nu^a(t)-1) + \widehat{a}_{mn}^2(t)\right] \\ &+ \frac{1}{\nu^w}\left(\frac{1}{2}\sum_{(m,l)\in\Omega}\left(y_{ml}-\bar{p}_{ml}(t)\right)^2 + \nu^p(t)\right) \\ &+ |\Omega|\log\sqrt{2\pi\nu^w}.\end{aligned} \tag{7}$$

To derive (7), one can start with the first term in [1, Eq. 108] and leverage the Gaussianity of the approximated posterior on x_{nl}:

$$\begin{aligned}&\sum_{n,l} D\left(p_{\mathsf{x}_{nl}|\mathsf{r}_{nl}}(\cdot|\widehat{r}_{nl}(t);\nu_{nl}^r(t))\middle\|\, p_{\mathsf{x}_{nl}}(\cdot)\right) \\ &\quad= \sum_{n,l}\int_{x_{nl}} \mathcal{N}(x_{nl};\widehat{x}_{nl}(t),\nu_{nl}^x(t))\log\frac{\mathcal{N}(x_{nl};\widehat{x}_{nl}(t),\nu_{nl}^x(t))}{\mathcal{N}(x_{nl};\widehat{x}_0,\nu_0^x)},\end{aligned} \tag{8}$$

which then directly yields the first term in (7). The second term in (7) follows using a similar procedure, and the third and fourth terms follow directly from the PIAWGN model.

In the noise free setting (i.e., $\nu^w \to 0$), the third term in (7) dominates, omitting the need to compute the other terms.

D. EM-BiG-AMP

For the likelihood (6) and priors (4), (5), the distributional parameters $\boldsymbol{\theta} = [\nu^w, \widehat{x}_0, \nu_0^x]^T$ can be tuned using the EM approach from [1, Section V-A].[1] To initialize $\boldsymbol{\theta}$ for EM-BiG-AMP, we adapt the procedure outlined in [15] to our matrix-completion problem, giving the EM initializations $\widehat{x}_0 = 0$ and

$$\nu^w = \frac{\|P_\Omega(\boldsymbol{Y})\|_F^2}{(\mathrm{SNR}^0+1)|\Omega|} \tag{9}$$

[1]For the first EM iteration, we recommend initializing BiG-AMP using $\nu_{nl}^x(1) = \nu_0^x, \widehat{x}_{nl}(1) = \widehat{x}_0, \nu_{mn}^a(1) = 1$, and $\widehat{a}_{mn}(1)$ drawn randomly from $p_{\mathsf{a}_{mn}}$. After the first iteration, we recommend warm-starting BiG-AMP using the values from the previous EM iteration.

$$\nu_0^x = \frac{1}{N}\left[\frac{\|P_\Omega(\boldsymbol{Y})\|_F^2}{|\Omega|} - \nu^w\right], \tag{10}$$

where SNR^0 is an initial estimate of the signal-to-noise ratio that, in the absence of other knowledge, can be set at 100.

E. Rank Selection

For MC rank-selection under the penalized log-likelihood strategy [1, Eq, 113], we recommend using the *small sample corrected AIC* (AICc) [16] penalty $\eta(N) = 2\frac{|\Omega|}{|\Omega|-N_{\text{eff}}-1}N_{\text{eff}}$. For the MC problem, $N_{\text{eff}} = \mathsf{df}+3$, where $\mathsf{df} \triangleq N(M+L-N)$ counts the degrees-of-freedom in a rank-N real-valued $M \times L$ matrix [8] and the three additional parameters come from $\boldsymbol{\theta}$. Based on the PIAWGN likelihood (6) and the standard form of the ML estimate of ν^w (see, e.g., [16, eq. (7)]), the update rule [1, Eq. 113] becomes

$$\widehat{N} = \underset{N=1,\ldots,\bar{N}}{\arg\max}\left[-|\Omega|\log\left(\frac{1}{|\Omega|}\sum_{(m,l)\in\Omega}\left(y_{ml}-\widehat{z}_{ml}(t)\right)^2\right) - 2\frac{|\Omega|\left(N(M+L-N)+3\right)}{|\Omega|-N(M+L-N)-4}\right]. \tag{11}$$

We note that a similar rule (but based on BIC rather than AICc) was used for rank-selection in [17].

MC rank selection can also be performed using the rank contraction scheme described in [1, Sec. V-B2]. We recommend choosing the maximum rank $\overline{N}$ to be the largest value such that $\overline{N}(M+L-\overline{N}) < |\Omega|$ and setting $\tau_{\text{MOS}} = 1.5$. Since the first EM iteration runs BiG-AMP with the large value $N = \overline{N}$, we suggest limiting the number of allowed BiG-AMP iterations during this first EM iteration to $\texttt{nitFirstEM} = 50$. In many cases, the rank learning procedure will correctly reduce the rank after these first few iterations, reducing the added computational cost of the rank selection procedure.

F. Matrix Completion Experiments

We now present the results of experiments used to ascertain the performance of BiG-AMP relative to existing state-of-the-art algorithms for matrix completion. For these experiments, we considered IALM [18], a nuclear-norm based convex-optimization method; LMaFit [13], a non-convex optimization-based approach using non-linear successive over-relaxation; GROUSE [19], which performs gradient descent on the Grassmanian manifold; Matrix-ALPS [20], a greedy hard-thresholding approach; and VSBL [14], a variational Bayes approach. In general, we configured BiG-AMP as described in Section II[2] and made our best attempt to configure the competing algorithms for maximum performance. That said, the different experiments that we ran required somewhat different parameter settings, as we detail in the sequel.

[2]Unless otherwise noted, we used the BiG-AMP parameters $T_{\max} = 1500$ (see [1, Sec. II-H] for descriptions) and the adaptive damping parameters $\texttt{stepInc} = 1.1$, $\texttt{stepDec} = 0.5$, $\texttt{stepMin} = 0.05$, $\texttt{stepMax} = 0.5$, $\texttt{stepWindow} = 1$, and $\beta(1) = \texttt{stepMin}$. (See [1, Sec. IV-B] for descriptions).

1) Low-Rank Matrices: We first investigate recovery of rank-N matrices $\boldsymbol{Z} \in \mathbb{R}^{M\times L}$ from noiseless incomplete observations $\{z_{ml}\}_{(m,l)\in\Omega}$ with indices Ω chosen uniformly at random. To do this, we evaluated the normalized mean square error (NMSE) $\frac{\|\boldsymbol{Z}-\widehat{\boldsymbol{Z}}\|_F^2}{\|\boldsymbol{Z}\|_F^2}$ of the estimate $\widehat{Z}$ returned by the various algorithms under test, examining 10 realizations of $(\boldsymbol{Z},\Omega)$ at each problem size (M,L,N). Here, each realization of $\boldsymbol{Z}$ was constructed as $\boldsymbol{Z} = \boldsymbol{A}\boldsymbol{X}$ for $\boldsymbol{A}$ and $\boldsymbol{X}$ with iid $\mathcal{N}(0,1)$ elements.[3] All algorithms were forced[4] to use the true rank N, run under default settings with very minor modifications,[5] and terminated when the normalized change in either $\widehat{\boldsymbol{Z}}$ or $P_\Omega(\widehat{\boldsymbol{Z}})$ across iterations fell below the tolerance value of 10^{-8}.

Defining "successful" matrix completion as NMSE < -100 dB, Fig. 1 shows the success rate of each algorithm over a grid of sampling ratios $\delta \triangleq \frac{|\Omega|}{ML}$ and ranks N. As a reference, the solid line superimposed on each subplot delineates the problem feasibility boundary, i.e., the values of (δ,N) yielding $|\Omega| = \mathsf{df}$, where $\mathsf{df} = N(M+L-N)$ is the degrees-of-freedom in a rank-N real-valued $M \times L$ matrix; successful recovery above this line is impossible by any method.

Fig. 1 shows that each algorithm exhibits a sharp phase-transition separating near-certain success from near-certain failure. There we see that BiG-AMP yields the best PTC. Moreover, BiG-AMP's PTC is *near optimal* in the sense of coming very close to the feasibility boundary for all tested δ and N. In addition, Fig. 1 shows that BiG-AMP-Lite yields the second-best PTC, which matches that of BiG-AMP except under very low sampling rates (e.g., $\delta < 0.03$). Recall that the only difference between the two algorithms is that BiG-AMP-Lite uses the scalar-variance simplification from [1, Sec. III-A].

Fig. 2 plots median runtime[6] to NMSE $= -100$ dB versus rank N for several sampling ratios δ, uncovering orders-of-magnitude differences among algorithms. For most values of δ and N, LMaFit was the fastest algorithm and BiG-AMP-Lite was the second fastest, although BiG-AMP-Lite was faster than LMaFit at small δ and relatively large N, while BiG-AMP-Lite was slower than GROUSE at large δ and very small N. In all cases, BiG-AMP-Lite was faster than IALM and VSBL, with several orders-of-magnitude difference at high rank. Meanwhile, EM-BiG-AMP was about 3 to 5 times slower than BiG-AMP-Lite. Although none of the algorithm implementations were fully optimized, we believe that the reported runtimes are insightful, especially with regard to the scaling of runtime with rank N.

[3]We chose the i.i.d Gaussian construction due to its frequent appearance in the matrix-completion literature. Similar performance was observed when the low-rank factors $\boldsymbol{A}$ and $\boldsymbol{X}$ were generated in other ways, such as from the left and right singular vectors of an i.i.d Gaussian matrix.

[4]This restriction always improved the performance of the tested algorithms.

[5]GROUSE was run with $\texttt{maxCycles} = 600$ and $\texttt{step\_size} = 0.5$, where the latter was chosen as a good compromise between phase-transition performance and runtime. VSBL was run under $\texttt{MAXITER} = 2000$ and fixed $\beta = 10^9$; adaptive selection of β was found to produce a significant degradation in the observed phase transition. LMaFit was run from the same random initialization as BiG-AMP and permitted at most $\texttt{maxit} = 6000$ iterations. IALM was allowed at most 2000 iterations. A maximum runtime of one hour per realization was enforced for all algorithms.

[6]The reported runtimes do not include the computations used for initialization nor those used for runtime evaluation.

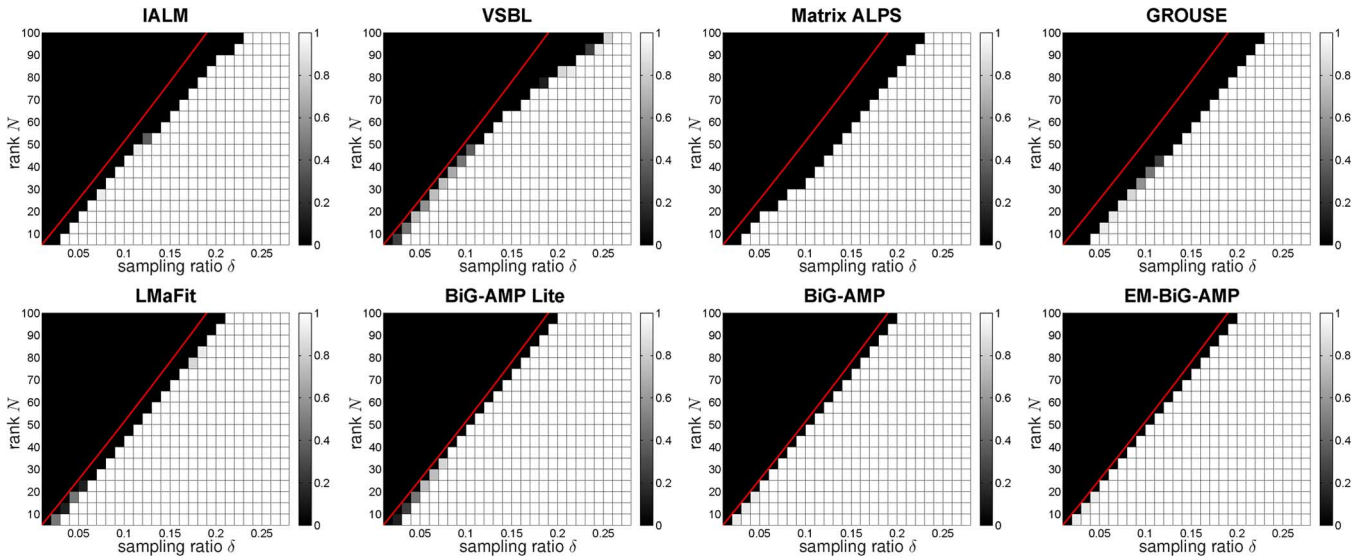

Fig. 1. Empirical success rates for noiseless completion of an $M \times L$ matrix sampled uniformly at random, as a function of sampling ratio $\delta = \frac{|\Omega|}{ML}$ and rank N. Here, "success" is defined as $\mathrm{NMSE} < -100$ dB, success rates were computed from 10 random realizations, and $M = L = 1000$. Points above the red curve are infeasible, as described in the text.

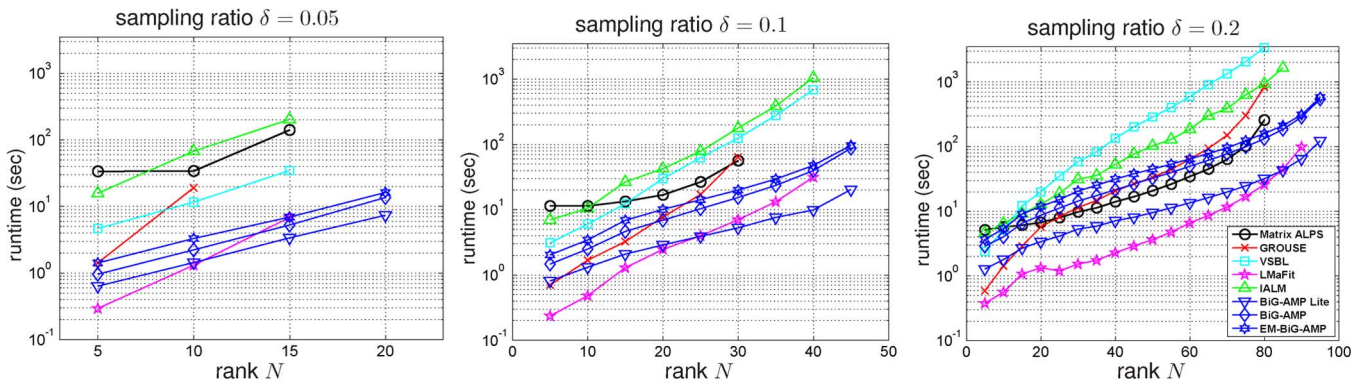

Fig. 2. Runtime to $\mathrm{NMSE} = -100$ dB for noiseless completion of an $M \times L$ matrix sampled uniformly at random, versus rank N, at $M = L = 1000$ and several sampling ratios $\delta = \frac{|\Omega|}{ML}$. All results represent median performance over 10 trials. Missing values indicate that the algorithm did not achieve the required NMSE before termination and correspond to the black regions in Fig. 1.

2) Approximately Low-Rank Matrices: Next we evaluate the performance of recovering approximately low-rank matrices by repeating an experiment from the LMaFit paper [13]. For this, we constructed the complete matrix as $\boldsymbol{Z} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^{\mathsf{T}} \in \mathbb{R}^{500\times 500}$, where $\boldsymbol{U}, \boldsymbol{V}$ were orthogonal matrices (built by orthogonalizing iid $\mathcal{N}(0,1)$ matrices using MATLAB's `orth` command) and $\boldsymbol{\Sigma}$ was a positive diagonal matrix containing the singular values of $\boldsymbol{Z}$. Two versions of $\boldsymbol{\Sigma}$ were considered: one with exponentially decaying singular values $[\boldsymbol{\Sigma}]_{m,m} = e^{-0.3\,m}$, and one with the power-law decay $[\boldsymbol{\Sigma}]_{m,m} = m^{-3}$.

As in [13], we first tried to recover $\boldsymbol{Z}$ from the noiseless incomplete observations $\{z_{ml}\}_{(m,l)\in\Omega}$, with Ω chosen uniformly at random. Fig. 3 shows the performance of several algorithms that are able to learn the underlying rank: LMaFit,[7] VSBL,[8] and EM-BiG-AMP under the penalized log-likelihood rank selection strategy from [1, Sec. V-B1].[9] All three algorithms were allowed a maximum rank of $\bar{N} = 30$. The figure shows that the NMSE performance of BiG-AMP and LMaFit are similar, although BiG-AMP tends to find solutions with lower rank but comparable NMSE at low sampling ratios δ. For this noiseless experiment, VSBL consistently estimates ranks that are too low, leading to inferior NMSEs.

Next, we examined *noisy* matrix completion by constructing the matrix $\boldsymbol{Z} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^{\mathsf{T}}$ as above but then corrupting the measurements with AWGN. Fig. 4 shows NMSE and estimated rank versus the measurement signal-to-noise ratio (SNR) $\sum_{(m,l)\in\Omega}|z_{ml}|^2 / \sum_{(m,l)\in\Omega}|y_{ml} - z_{ml}|^2$ at a sampling rate of $\delta = 0.2$. There we see that, for SNRs $<$50 dB, EM-BiG-AMP and VSBL offer similarly good NMSE performance and nearly identical rank estimates, whereas LMaFit overestimates the rank and thus performs worse in NMSE. Meanwhile, for SNRs $>$50 dB, EM-BiG-AMP and LMaFit offer similarly

[7]LMaFit was run under the settings provided in their source code for this example.

[8]VSBL was allowed at most 100 iterations and run with `DIMRED_THR` $= 10^3$, `UPDATE_BETA` $= 1$, and tolerance $= 10^{-8}$.

[9]Rank-selection rule [1, Eq. 113] was used with up to 5 EM iterations for each rank hypothesis N, a minimum of 30 and maximum of 100 BiG-AMP iterations for each EM iteration, and a BiG-AMP tolerance of 10^{-8}.

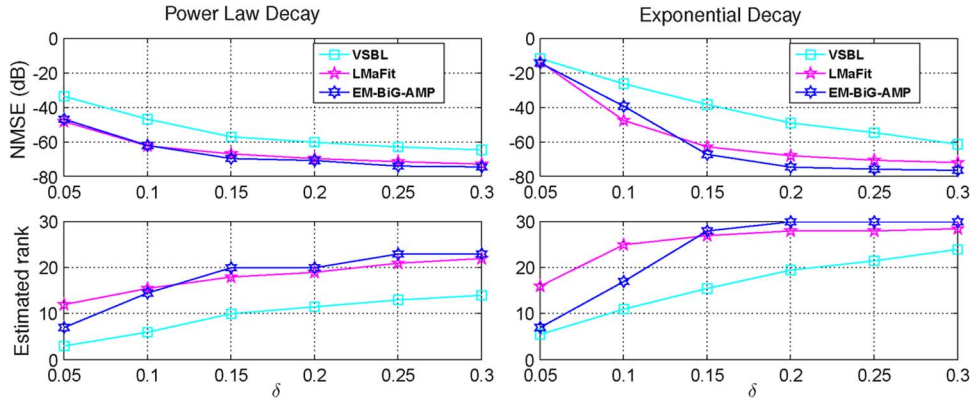

Fig. 3. NMSE (top) and estimated rank (bottom) in noiseless completion of an $M \times L$ matrix sampled uniformly at random, versus sampling ratio $\delta = \frac{|\Omega|}{ML}$. The complete matrices were approximately low-rank, with power-law (left) and exponential (right) singular-value decays and $M = L = 500$. All results represent median performance over 10 random trials.

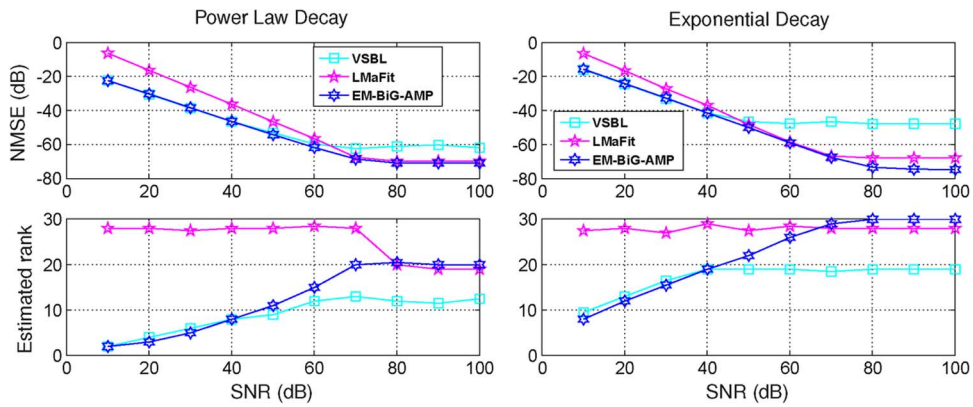

Fig. 4. NMSE (top) and estimated rank (bottom), versus SNR, in noisy completion of an 500×500 matrix sampled uniformly at random at rate $\delta = 0.2$. The complete matrices were approximately low-rank, with power-law (left) and exponential (right) singular-value decays.

good NMSE performance and nearly identical rank estimates, whereas VSBL underestimates the rank and thus performs worse in NMSE. Thus, in these examples, EM-BiG-AMP is the only algorithm to successfully estimate the rank across the full SNR range.

3) Image Completion: We now compare the performance of several matrix-completion algorithms for the task of reconstructing an image from a subset of its pixels. For this, we repeated the experiment in the Matrix-ALPS paper [20], where the 512×512 boat image was reconstructed from 35% of its pixels sampled uniformly at random. Fig. 5 shows the complete (full-rank) image, the images reconstructed by several matrix-completion algorithms[10] under a fixed rank of $N = 40$, and the NMSE-minimizing rank-40 approximation of the complete image, computed using an SVD. In all cases, the sample mean of the observations was subtracted prior to processing and then added back to the estimated images, since this approach generally improved performance. Fig. 5 also lists the median reconstruction NMSE over 10 sampling-index realizations Ω. From these results, it is apparent that EM-BiG-AMP provides the best NMSE, which is only 3 dB from that of the NMSE-optimal rank-40 approximation.

4) Collaborative Filtering: In our final experiment, we investigate the performance of several matrix-completion algorithms on the task of collaborative filtering. For this, we repeated an experiment from the VSBL paper [14] that used the MovieLens 100k dataset, which contains ratings $\{z_{ml}\}_{(m,l)\in\mathcal{R}}$, where $|\mathcal{R}| = 100\,000$ and $z_{ml} \in \{1,2,3,4,5\}$, from $M = 943$ users about $L = 1682$ movies. The algorithms were provided with a randomly chosen training subset $\{z_{ml}\}_{(m,l)\in\Omega}$ of the ratings (i.e., $\Omega \subset \mathcal{R}$) from which they estimated the unseen ratings $\{\widehat{z}_{ml}\}_{(m,l)\in\mathcal{R}\setminus\Omega}$. Performance was then assessed by computing the Normalized Mean Absolute Error (NMAE)

$$\text{NMAE} = \frac{1}{4|\mathcal{R}\setminus\Omega|} \sum_{(m,l)\in\mathcal{R}\setminus\Omega} |z_{ml} - \widehat{z}_{ml}|, \qquad (12)$$

where the 4 in the denominator of (12) reflects the difference between the largest and smallest user ratings (i.e., 5 and 1). When constructing Ω, we used a fixed percentage of the ratings given by each user and made sure that at least one rating was provided for every movie in the training set.

Fig. 6 reports the NMAE and estimated rank $\widehat{N}$ for EM-BiG-AMP under the PIAWGN model (6), LMaFit, and

[10]All algorithms were run with a convergence tolerance of 10^{-4}. VSBL was run with β hand-tuned to maximize performance, as the adaptive version did not converge on this example. GROUSE was run with `maxCycles` $= 600$ and `step_size` $= 0.1$. Matrix-ALPS II with QR was run under default parameters and 300 allowed iterations. Other settings are similar to earlier experiments.

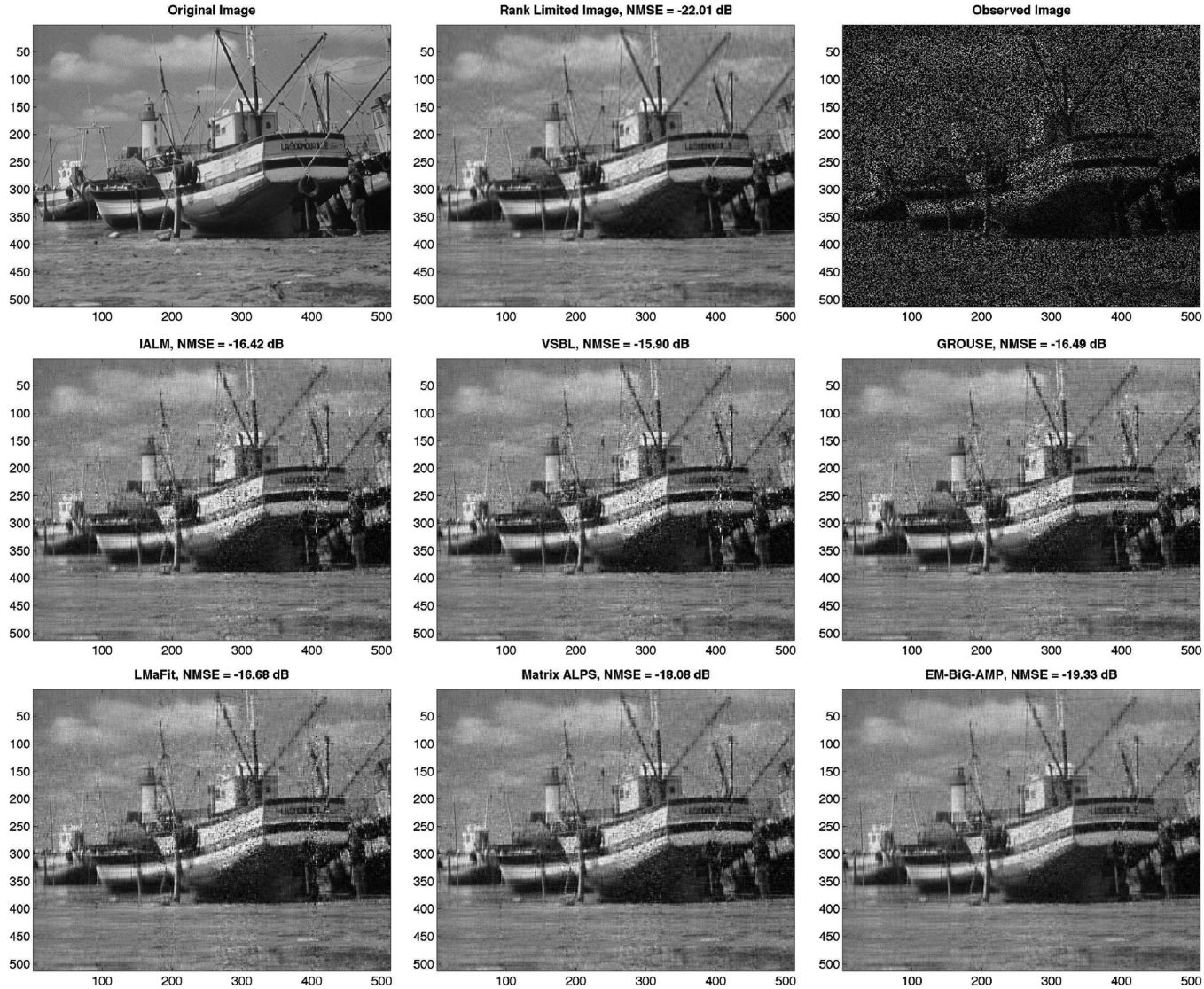

Fig. 5. For the image completion experiment, the complete image is shown on the top left, its best rank-40 approximation is shown on the top middle, and the observed image with 35% of the pixels observed is shown on the top right. The other panes show various algorithms' image reconstructions from 35% of the complete-image pixels (selected uniformly at random) as well as the mean NMSE over 10 trials.

VSBL,[11] all of which include mechanisms for rank estimation. Fig. 6 shows that, under the PIAWGN model, EM-BiG-AMP yields NMAEs that are very close to those of VSBL[12] but slightly inferior at larger training fractions, whereas LMaFit returns NMAEs that are substantially worse all training fractions.[13] Fig. 6 also shows that LMaFit's estimated rank is much higher than those of VSBL and EM-BiG-AMP, suggesting that its poor NMAE performance is the result of overfitting. (Recall that similar behavior was seen for noisy matrix completion in Fig. 4.) In addition, Fig. 6 shows that, as the training fraction increases, EM-BiG-AMP's estimated rank remains very low (i.e., ≤ 2) while that of VSBL steady increases (to >10). This prompts the question: is VSBL's excellent NMAE the result of accurate rank estimation or the use of a heavy-tailed (i.e., student's t) noise prior?

To investigate the latter question, we ran BiG-AMP under

$$p_{\mathrm{y}_{ml}|\mathrm{z}_{ml}}(y_{ml}|z_{ml}) = \begin{cases} \frac{\lambda}{2}\exp\left(-\lambda|y_{ml}-z_{ml}|\right) & (m,l)\in\Omega \\ \mathbb{1}_{y_{ml}} & (m,l)\notin\Omega, \end{cases} \tag{13}$$

i.e., a possibly incomplete additive white Laplacian noise (PIAWLN) model, and used the EM-based approach from [1, Sec. V-A] to learn the rate parameter λ. Fig. 6 shows that, under the PIAWLN model, EM-BiG-AMP essentially matches the NMAE performance of VSBL and even improves on it at very low training fractions. Meanwhile, its estimated rank $\widehat{N}$ remains low for all training fractions, suggesting that the use of a heavy-tailed noise model was the key to achieving

[11]VSBL was allowed at most 100 iterations and was run with `DIMRED_THR` $= 10^3$ and `UPDATE_BETA` $= 1$. Both VSBL and EM-BiG-AMP used a tolerance of 10^{-8}. LMaFit was configured as for the MovieLens experiment in [13]. Each algorithm was allowed a maximum rank of $\overline{N} = 30$.

[12]The NMAE values reported for VSBL in Fig. 6 are slightly inferior to those reported in [14]. We attribute the discrepancy to differences in experimental setup, such as the construction of Ω.

[13]The NMAE results presented here differ markedly from those in the MovieLens experiment in [13] because, in the latter paper, the entire set of ratings was used for both training *and testing*, with the (trivial) result that high-rank models (e.g., $\widehat{N} = 94$) yield nearly zero test error.

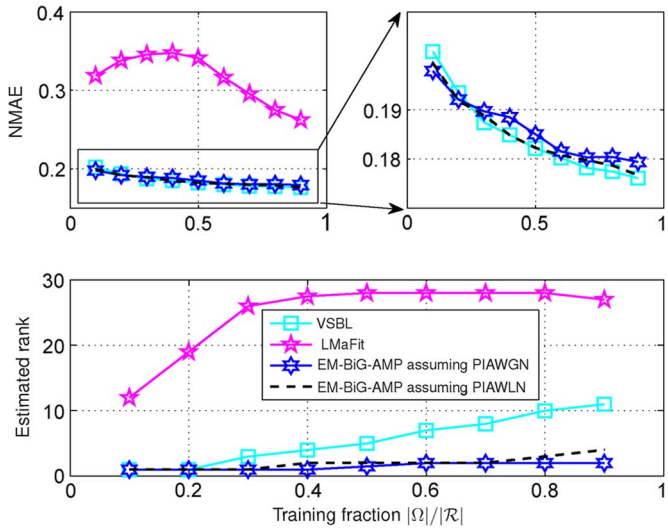

Fig. 6. Median NMAE (top) and estimated rank (bottom) for movie-rating prediction versus fraction of training data $|\Omega|/|\mathcal{R}|$ over 10 trials for the 100k MovieLens data set.

low NMAE in this experiment. Fortunately, the generality and modularity of BiG-AMP made this an easy task.

5) Summary: In summary, the known-rank synthetic-data results above showed the EM-BiG-AMP methods yielding phase-transition curves superior to all other algorithms under test. In addition, they showed BiG-AMP-Lite to be the second fastest algorithm (behind LMaFit) for most combinations of sampling ratio δ and rank N, although it was the fastest for small δ and relatively high N. Also, they showed EM-BiG-AMP was about 3 to 5 times slower than BiG-AMP-Lite but still much faster than IALM and VSBL at high ranks. Meanwhile, the unknown-rank synthetic-data results above showed EM-BiG-AMP yielding excellent NMSE performance in both noiseless and noisy scenarios. For example, in the noisy experiment, EM-BiG-AMP uniformly outperformed its competitors (LMaFit and VSBL).

In the image completion experiment, EM-BiG-AMP again outperformed all competitors, beating the second best algorithm (Matrix ALPS) by more than 1 dB and the third best algorithm (LMaFit) by more than 2.5 dB. Finally, in the collaborative filtering experiment, EM-BiG-AMP (with the PIAWLN likelihood model) matched the best competitor (VSBL) in terms of NMAE, and significantly outperformed the second best (LMaFit).

III. Robust PCA

A. Problem Setup

In robust principal components analysis (RPCA) [21], one seeks to estimate a low-rank matrix observed in the presence of noise and large outliers. The data model for RPCA can be written as

$$\boldsymbol{Y} = \boldsymbol{A}\boldsymbol{X} + \boldsymbol{E} + \boldsymbol{W}, \tag{14}$$

where $\boldsymbol{Z} = \boldsymbol{A}\boldsymbol{X}$—the product of tall $\boldsymbol{A}$ and wide $\boldsymbol{X}$—is the low-rank matrix of interest, $\boldsymbol{E}$ is a *sparse* outlier matrix, and $\boldsymbol{W}$ is a dense noise matrix. We now suggest two ways of applying BiG-AMP to the RPCA problem, both of which treat the elements of $\boldsymbol{A}$ as iid $\mathcal{N}(0,\nu_0^a)$ similar to (4), the elements of $\boldsymbol{X}$ as iid $\mathcal{N}(0,\nu_0^x)$ similar to (5), the *non-zero* elements of $\boldsymbol{E}$ as iid $\mathcal{N}(0,\nu_1)$, and the elements of $\boldsymbol{W}$ as iid $\mathcal{N}(0,\nu_0)$, with $\nu_1 \gg \nu_0$.

In the first approach, $\boldsymbol{E}+\boldsymbol{W}$ is treated as additive noise on $\boldsymbol{Z}$, leading to the likelihood model

$$\begin{aligned} &p_{\mathsf{y}_{ml}|\mathsf{z}_{ml}}(y_{ml}|z_{ml}) \\ &\quad = (1-\lambda)\mathcal{N}(y_{ml};z_{ml},\nu_0) + \lambda\mathcal{N}(y_{ml};z_{ml},\nu_0+\nu_1), \end{aligned} \tag{15}$$

where $\lambda \in [0,1]$ models outlier density.

In the second approach, $\boldsymbol{W}$ is treated as additive noise but $\boldsymbol{E}$ is treated as an additional estimand. In this case, by multiplying both sides of (14) by any (known) unitary matrix $\boldsymbol{Q} \in \mathbb{R}^{M\times M}$, we can write

$$\underbrace{\boldsymbol{Q}\boldsymbol{Y}}_{\triangleq \underline{\boldsymbol{Y}}} = \underbrace{[\boldsymbol{Q}\boldsymbol{A} \quad \boldsymbol{Q}]}_{\triangleq \underline{\boldsymbol{A}}} \underbrace{\begin{bmatrix}\boldsymbol{X}\\ \boldsymbol{E}\end{bmatrix}}_{\triangleq \underline{\boldsymbol{X}}} + \underbrace{\boldsymbol{Q}\boldsymbol{W}}_{\triangleq \underline{\boldsymbol{W}}}, \tag{16}$$

and apply BiG-AMP to the "augmented" model $\underline{\boldsymbol{Y}} = \underline{\boldsymbol{A}}\,\underline{\boldsymbol{X}} + \underline{\boldsymbol{W}}$. Here, $\underline{\boldsymbol{W}}$ remains iid $\mathcal{N}(0,\nu_0)$, thus giving the likelihood

$$p_{\mathsf{y}_{ml}|\mathsf{z}_{ml}}(\underline{y}_{ml}|\underline{z}_{ml}) = \mathcal{N}(\underline{y}_{ml};\underline{z}_{ml},\nu_0). \tag{17}$$

Meanwhile, we choose the following separable priors on $\underline{\boldsymbol{A}}$ and $\underline{\boldsymbol{X}}$:

$$p_{\underline{\mathsf{a}}_{mn}}(\underline{a}_{mn}) = \begin{cases} \mathcal{N}(\underline{a}_{mn};0,\nu_0^a) & n \le N \\ \mathcal{N}(\underline{a}_{mn};q_{mn},0) & n > N \end{cases} \tag{18}$$

$$p_{\underline{\mathsf{x}}_{nl}}(\underline{x}_{nl}) = \begin{cases} \mathcal{N}(\underline{x}_{nl};0,\nu_0^x) & n \le N \\ (1-\lambda)\mathbb{1}_{\underline{x}_{nl}} + \lambda\mathcal{N}(\underline{x}_{nl};0,\nu_1) & n > N. \end{cases} \tag{19}$$

Essentially, the first N columns of $\underline{\boldsymbol{A}}$ and first N rows of $\underline{\boldsymbol{X}}$ model the factors of the low-rank matrix $\boldsymbol{A}\boldsymbol{X}$, and thus their elements are assigned iid Gaussian priors, similar to (4), (5) in the case of matrix completion. Meanwhile, the last M rows in $\underline{\boldsymbol{X}}$ are used to represent the sparse outlier matrix $\boldsymbol{E}$, and thus their elements are assigned a Bernoulli-Gaussian prior. Finally, the last M columns of $\underline{\boldsymbol{A}}$ are used to represent the designed matrix $\boldsymbol{Q}$, and thus their elements are assigned zero-variance priors. Since we find that BiG-AMP is numerically more stable when $\boldsymbol{Q}$ is chosen as a dense matrix, we set it equal to the singular-vector matrix of an iid $\mathcal{N}(0,1)$ matrix. After running BiG-AMP, we can recover an estimate of $\boldsymbol{A}$ by left multiplying the estimate of $\underline{\boldsymbol{A}}$ by $\boldsymbol{Q}^{\mathsf{H}}$.

B. Initialization

We recommend initializing $\widehat{\underline{a}}_{mn}(1)$ using a random draw from its prior and initializing $\widehat{\underline{x}}_{nl}(1)$ at the mean of its prior, i.e., $\widehat{\underline{x}}_{nl}(1) = 0$. The latter tends to perform better than initializing $\widehat{\underline{x}}_{nl}(1)$ randomly, because it allows the measurements $\boldsymbol{Y}$ to determine the initial locations of the outliers in $\boldsymbol{E}$. As in Section II-B, we suggest initializing $\nu_{mn}^{\underline{a}}(1)$ and $\nu_{nl}^{\underline{x}}(1)$ at 10 times the variance of their respective priors to emphasize the role of the measurements during the first few iterations.

C. EM-BiG-AMP

The EM approach from [1, Sec. V-A] can be straightforwardly applied to BiG-AMP for RPCA: after fixing $\nu_0^a = 1$, EM can be used to tune the remaining distributional parameters, $\boldsymbol{\theta} = [\nu_0, \nu_1, \nu_0^x, \lambda]^T$. To avoid initializing ν_0 and ν_0^x with overly large values in the presence of large outliers e_{nl}, we suggest the following procedure. First, define the set $\Gamma \triangleq \{(m,l) : |y_{ml}| \leq \text{median}\{|y_{ml}|\}\}$ and its complement Γ^c. Then initialize

$$\nu_0 = \frac{\frac{1}{|\Gamma|}\sum_{(m,l)\in\Gamma}|y_{ml}|^2}{\text{SNR}^0 + 1} \tag{20}$$

$$\nu_0^x = \frac{1}{N}\text{SNR}^0\nu_0 \tag{21}$$

$$\nu_1 = \frac{1}{|\Gamma^c|}\sum_{(m,l)\in\Gamma^c}|y_{ml}|^2, \tag{22}$$

where, as in Section II-D, we suggest setting $\text{SNR}^0 = 100$ in the absence of prior knowledge. This approach uses the median to avoid including outliers among the samples used to estimate the variances of the dense-noise and low-rank components. Under these rules, the initialization $\lambda = 0.1$ was found to work well for most problems.

D. Rank Selection

In many applications of RPCA, such as video separation, the singular-value profile of $\boldsymbol{AX}$ exhibits a sharp cutoff, in which case it is recommended to perform rank-selection using the contraction strategy from [1, Sec. V-B2].

E. Avoiding Local Minima

Sometimes, when N is very small, BiG-AMP may converge to a local solution that mistakes entire rows or columns of $\boldsymbol{AX}$ for outliers. Fortunately, this situation is easy to remedy with a simple heuristic procedure: the posterior probability that y_{ml} is outlier-corrupted can be computed for each (m,l) at convergence, and if any of the row-wise sums exceeds $0.8M$ or any of the column-wise sums exceeds $0.8L$, then BiG-AMP is restarted from a new random initialization. Experimentally, we found that one or two of such restarts is generally sufficient to avoid local minima.

F. Robust PCA Experiments

In this section, we present a numerical study of the two BiG-AMP formulations of RPCA proposed in Section III, including a comparison to the state-of-the-art IALM [18], LMaFit [13], GRASTA [22], and VSBL [14] algorithms. In the sequel, we use "BiG-AMP-1" when referring to the formulation that treats the outliers as noise, and "BiG-AMP-2" when referring to the formulation that explicitly estimates the outliers.

1) Phase Transition Behavior: We first study the behavior of the proposed BiG-AMP algorithms for RPCA on noise-free synthetic problems. For this, we generated problem realizations of the form $\boldsymbol{Y} = \boldsymbol{Z} + \boldsymbol{E}$, where the low-rank component $\boldsymbol{Z} = \boldsymbol{AX}$ was generated from $\boldsymbol{A}$ and $\boldsymbol{X}$ with iid $\mathcal{N}(0,1)$ entries, and where the sparse corruption matrix $\boldsymbol{E}$ had a fraction δ of non-zero entries that were located uniformly at random with amplitudes drawn iid uniform on $[-10, 10]$. The dimensions of $\boldsymbol{Y} \in \mathbb{R}^{M\times L}$ were fixed at $M = L = 200$, the rank N (of $\boldsymbol{Z}$) was varied from 10 to 90, and the outlier fraction δ was varied from 0.05 to 0.45. We note that, under these settings, the outlier magnitudes are on the same order as the magnitudes of $\boldsymbol{Z}$, which is the most challenging case: much larger outliers would be easier to detect, after which the corrupted elements of $\boldsymbol{Y}$ could be safely treated as incomplete, whereas much smaller outliers could be treated like AWGN.

All algorithms under test were run to a convergence tolerance of 10^{-8} and forced to use the true rank N. GRASTA, LMaFit, and VSBL were run under their recommended settings. [14] Two versions of IALM were tested: "IALM-1," which uses the universal penalty parameter $\lambda_{\texttt{ALM}} = \frac{1}{\sqrt{M}}$, and "IALM-2," which tries 50 hypotheses of $\lambda_{\texttt{ALM}}$, logarithmically spaced from $\frac{1}{10\sqrt{M}}$ to $\frac{10}{\sqrt{M}}$ and uses an oracle to choose the MSE-minimizing hypothesis. BiG-AMP-1 and BiG-AMP-2 were given perfect knowledge of the mean and variance of the entries of $\boldsymbol{A}$, $\boldsymbol{X}$, and $\boldsymbol{E}$ (although their Bernoulli-Gaussian model of $\boldsymbol{E}$ did not match the data generation process) as well as the outlier density λ, while EM-BiG-AMP-2 learned all model parameters from the data. BiG-AMP-1 was run under a fixed damping of $\beta = 0.25$, while BiG-AMP-2 was run under adaptive damping with $\texttt{stepMin} = 0.05$ and $\texttt{stepMax} = 0.5$. Both variants used a maximum of 5 restarts to avoid local minima.

Fig. 7 shows the empirical success rate achieved by each algorithm as a function of corruption-rate δ and rank N, averaged over 10 trials, where a "success" was defined as attaining an NMSE of -80 dB or better in the estimation of the low-rank component $\boldsymbol{Z}$. The red curves in Fig. 7 delineate the problem feasibility boundary: for points (δ, N) above the curve, $N(M + L - N)$, the degrees-of-freedom in $\boldsymbol{Z}$, exceeds $(1-\delta)ML$, the number of uncorrupted observations, making it impossible to recover $\boldsymbol{Z}$ without additional information.

Fig. 7 shows that all algorithms exhibit a relatively sharp phase-transition curve (PTC) separating success and failure regions, and that the BiG-AMP algorithms achieve substantially better PTCs than the other algorithms. The PTCs of BiG-AMP-1 and BiG-AMP-2 are similar (but not identical), suggesting that both formulations are equally effective. Meanwhile, the PTCs of BiG-AMP-2 and EM-BiG-AMP-2 are nearly identical, demonstrating that the EM procedure was able to successfully learn the statistical model parameters used by BiG-AMP. Fig. 7 also shows that all RPCA phase transitions remain relatively far from the feasibility boundary, unlike those for matrix completion (MC) shown in Fig. 1. This behavior, also observed in [21], is explained by the relative difficulty of RPCA over MC: the locations of RPCA outliers (which in this case effectively render the corrupted observations as incomplete) are unknown, whereas in MC they are known.

Fig. 8 plots runtime to $\text{NMSE} = -80$ dB as a function of rank N for various outlier fractions. The results suggest that the BiG-AMP algorithms are moderate in terms of speed, being

[14] For LMaFit, however, we increased the maximum number of allowed iterations, since this improved its performance.

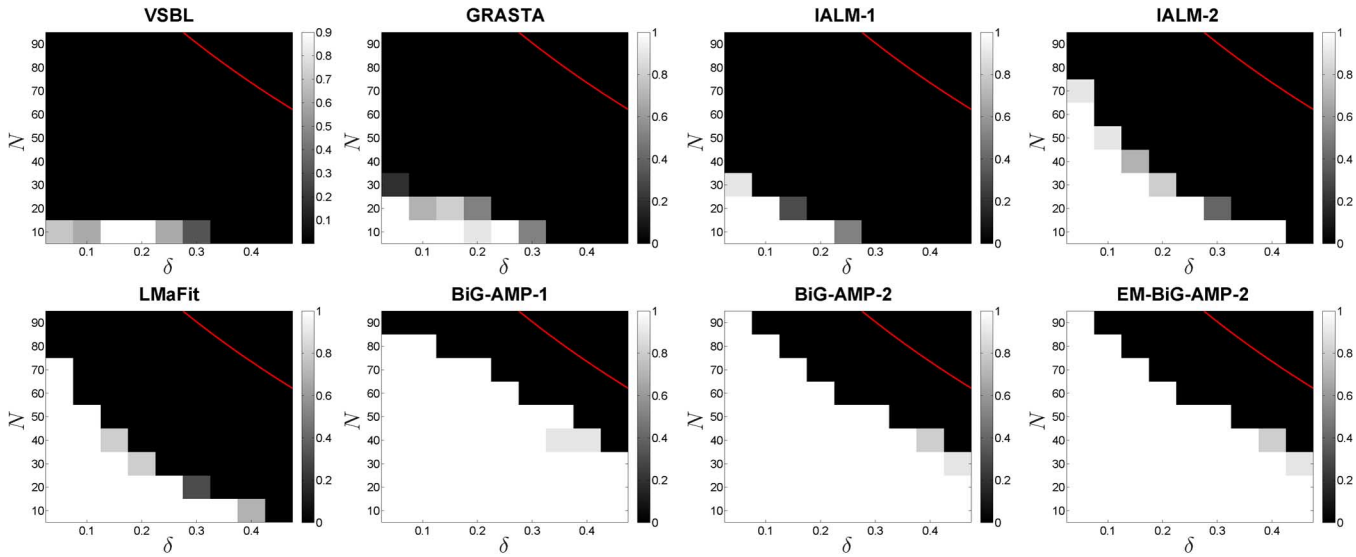

Fig. 7. Empirical success rates for RPCA with a 200 × 200 matrix of rank N corrupted by a fraction δ of outliers with amplitudes uniformly distributed on $[-10, 10]$. Here, "success" is defined as $\mathrm{NMSE} < -80$ dB, and success rates were averaged over 10 problem realizations. Points above the red curve are infeasible, as described in the text.

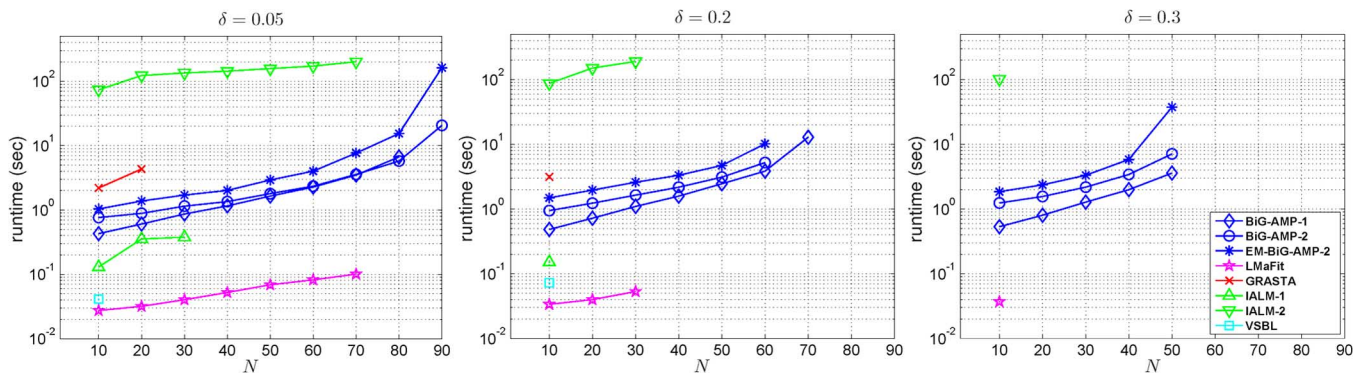

Fig. 8. Runtime to $\mathrm{NMSE} = -80$ dB for RPCA with a 200 × 200 matrix of rank N corrupted by a fraction $\delta \in \{0.05, 0.2, 0.3\}$ of outliers with amplitudes uniformly distributed on $[-10, 10]$. All results represent median performance over 10 trials. Missing values indicate that the algorithm did not achieve the required NMSE before termination and correspond to the black regions in Fig. 7.

faster than GRASTA[15] and much faster than the grid-tuned IALM-2, but slower than IALM-1, VSBL, and LMaFit. Notably, among the non-BiG-AMP algorithms, LMaFit offers both the fastest runtime and the best phase-transition curve on this synthetic test problem.

In summary, the results presented here suggest that BiG-AMP achieves state-of-the-art PTCs while maintaining runtimes that are competitive with existing approaches.

2) Rank Estimation: We now investigate the ability to estimate the underlying rank, N, for EM-BiG-AMP-2 (using the rank-contraction strategy from [1, Sec. V-B2])[16] IALM-1, IALM-2, LMaFit, and VSBL, all of which include either explicit or implicit rank-selection mechanisms. For this, we generated problem realizations of the form $\boldsymbol{Y} = \boldsymbol{Z} + \boldsymbol{E} + \boldsymbol{W}$, where the 200 × 200 rank-N matrix $\boldsymbol{Z}$ and $\delta = 0.1$-sparse outlier matrix $\boldsymbol{E}$ were generated as described in Section III-F-1 and the noise matrix $\boldsymbol{W}$ was constructed with iid $\mathcal{N}(0, 10^{-3})$ elements. The algorithms under test were not provided with knowledge of the true rank N, which was varied between 5 and 90. LMaFit, VSBL, and EM-BiG-AMP, were given an initial rank estimate of $\overline{N} = 90$, which enforces an upper bound on the final estimates that they report.

Fig. 9 reports RPCA performance versus (unknown) true rank N in terms of the estimated rank $\widehat{N}$ and the NMSE on the estimate $\widehat{\boldsymbol{Z}}$. All results represent median performance over 10 Monte-Carlo trials. The figure shows that EM-BiG-AMP-2 and LMaFit returned accurate rank estimates $\widehat{N}$ over the full range of true rank $N \in [5, 90]$, whereas VSBL returned accurate rank estimates only for $N \leq 20$, and both IALM-1 and IALM-2 greatly overestimated the rank at all N. Meanwhile, Fig. 9 shows that EM-BiG-AMP-2 and LMaFit returned accurate estimates of $\widehat{\boldsymbol{Z}}$ for all $N \leq 80$ (with EM-BiG-AMP-2 outperforming LMaFit by several dB throughout this range), whereas VSBL and IALM-1 and IALM-2 returned accurate estimates of $\widehat{\boldsymbol{Z}}$ only for small values of N. We note that the relatively poor MSE performance of LMaFit and EM-BiG-AMP-2 for

[15]We note that, for this experiment, GRASTA was run as a Matlab M-file and not a MEX file, because the MEX file would not compile on the supercomputer used for the numerical results. That said, since BiG-AMP was also run as an unoptimized M-file, the comparison could be considered "fair."

[16]The rank-selection rule [1, Eq. 114] was used with $\tau_{\mathrm{MOS}} = 5$, up to 50 EM iterations, and a minimum of 30 and maximum of 500 BiG-AMP iterations per EM iteration.

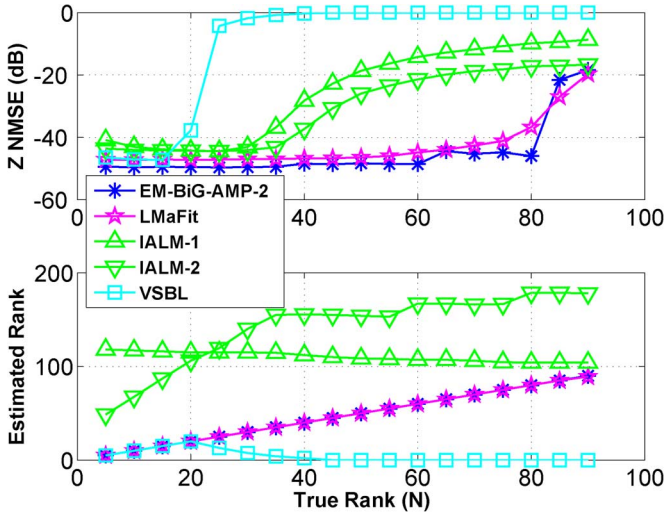

Fig. 9. NMSE (top) and estimated rank $\widehat{N}$ (bottom) versus true rank N for several algorithms performing RPCA on a 200×200 matrix in the presence of additive $\mathcal{N}(0, 10^{-3})$ noise and a fraction $\delta = 0.1$ of outliers with amplitudes uniformly distributed on $[-10, 10]$. All results represent the median over 10 trials.

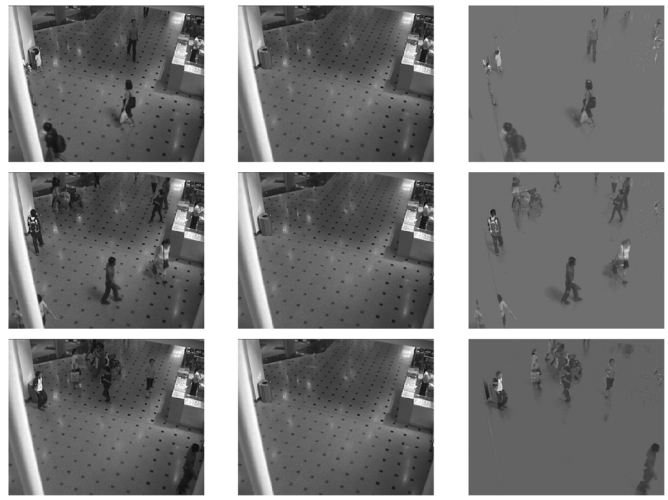

Fig. 10. Three example frames from the "mall" video sequence. The left column shows original frames, the middle column EM-BiG-AMP-2 estimated background, and the right column EM-BiG-AMP-2 estimated foreground.

true rank $N > 80$ is not due to poor rank estimation but rather due to the fact that, at $\delta = 0.1$, these operating points lie above the PTCs shown in Fig. 7.

3) Application to Video Surveillance: We now apply EM-BiG-AMP-2 to a video surveillance problem, where the goal is to separate a video sequence into a static "background" component and a dynamic "foreground" component. To do this, we stack each frame of the video sequence into a single column of the matrix $\boldsymbol{Y}$, run EM-BiG-AMP-2 as described in Section III, extract the background frames from the estimate of the low-rank component $\boldsymbol{Z} = \boldsymbol{AX}$, and extract the foreground frames from the estimate of the (sparse) outlier component $\boldsymbol{E}$. We note that a perfectly time-invariant background would correspond to a rank-one $\boldsymbol{Z}$ and that the noise term $\boldsymbol{W}$ in (14) can be used to account for small perturbations that are neither low-rank nor sparse.

We tested EM-BiG-AMP[17] on the popular "mall" video sequence,[18] processing 200 frames (of 256×320 pixels each) using an initial rank estimate of $\overline{N} = 5$. Fig. 10 shows the result, with original frames in the left column and EM-BiG-AMP-2 estimated background and foreground frames in the middle and right columns, respectively. We note that, for this sequence, the rank-contraction strategy reduced the rank of the background component to 1 after the first EM iteration. Similar results (not shown here for reasons of space) were obtained with other video sequences.

IV. Dictionary Learning

A. Problem Setup

In dictionary learning (DL) [23], one seeks a dictionary $\boldsymbol{A} \in \mathbb{R}^{M \times N}$ that allows the training samples $\boldsymbol{Y} \in \mathbb{R}^{M \times L}$ to be encoded as $\boldsymbol{Y} = \boldsymbol{AX} + \boldsymbol{W}$ for some sparse coefficient matrix $\boldsymbol{X}$ and small perturbation $\boldsymbol{W}$. One chooses $N = M$ to learn a square dictionary or $N > M$ (where N is often a small multiple of M) to learn an overcomplete dictionary. In general, one must have a sufficient number of training examples, $L \gg N$, to avoid over-fitting.[19]

The BiG-AMP methodology is particularly well-suited to the DL problem, since both are inherently bilinear. In this work, for simplicity, we model the entries of $\boldsymbol{A}$ using the iid standard normal prior (4) and the entries of $\boldsymbol{X}$ using the iid zero-mean Bernoulli-Gaussian (BG) prior

$$p_{\mathsf{x}_{nl}}(x) = (1 - \xi)\delta(x) + \xi \mathcal{N}(x; 0, \nu^x), \tag{23}$$

where ξ represents the activity rate and ν^x the active-component variance. However, other priors could be considered, such as truncated Gaussian mixtures with column-dependent prior parameters in the case of non-negative matrix factorization [25]. For the likelihood $p_{\boldsymbol{Y}|\boldsymbol{Z}}$, we again select the PIAWGN model (6), but note that in most applications of DL the observations are completely observed.

B. Initialization

In general, we advocate initializing $\widehat{x}_{nl}(1)$ at the mean of the assumed prior on x_{nl}, and initializing the variances $\nu^x_{nl}(1)$ and $\nu^a_{mn}(1)$ at 10 times the variance of x_{nl} and a_{mn}, respectively. We now discuss several strategies for initializing the dictionary estimates $\widehat{a}_{mn}(1)$. One option is to draw $\widehat{a}_{mn}(1)$ randomly from the assumed prior on a_{mn}, as suggested for MC and RPCA. Although this approach works reasonably well, the prevalence of local minima in the DL problem motivates us to propose two alternative strategies. The first alternative is to exploit prior knowledge of a "good" sparsifying dictionary $\boldsymbol{A}$, in the case that such knowledge exists. With natural images, for example, the discrete cosine transform (DCT) and discrete wavelet transform (DWT) are known to yield reasonably sparse transform coefficients, and so the DCT and DWT matrices make appropriate initializations of $\widehat{\boldsymbol{A}}(1)$.

[17]The maximum allowed damping was reduced to $\texttt{stepMax} = 0.125$ for this experiment. To reduce runtime, a relatively loose tolerance of 5×10^{-4} was used to establish EM and BiG-AMP convergence.

[18]See http://perception.i2r.a-star.edu.sg/bk_model/bk_index.html.

[19]See [24] for a discussion of sample-size requirements for exact recovery of square dictionaries.

The second alternative is to initialize $\widehat{\boldsymbol{A}}(1)$ using an appropriately chosen subset of the columns of $\boldsymbol{Y}$, which is well motivated in the case that there exists a very sparse representation $\boldsymbol{X}$. For example, if there existed a decomposition $\boldsymbol{Y} = \boldsymbol{A}\boldsymbol{X}$ in which $\boldsymbol{X}$ had 1-sparse columns, then the columns of $\boldsymbol{A}$ would indeed match a subset of the columns of $\boldsymbol{Y}$ (up to a scale factor). In the general case, however, it is not apriori obvious which columns of $\boldsymbol{Y}$ to choose, and so we suggest the following greedy heuristic, which aims for a well-conditioned $\widehat{\boldsymbol{A}}(1)$: select (normalized) columns from $\boldsymbol{Y}$ sequentially, in random order, accepting each candidate if the mutual coherences with the previously selected columns and the condition number of the resulting submatrix are all sufficiently small. If all columns of $\boldsymbol{Y}$ are examined before finding N acceptable candidates, then the process is repeated using a different random order. If repeated re-orderings fail, then $\widehat{\boldsymbol{A}}(1)$ is initialized using a random draw from $\boldsymbol{A}$.

C. EM-BiG-AMP

To tune the distributional parameters $\boldsymbol{\theta} = [\nu^w, \nu_0^x, \xi]^\top$, we can straightforwardly apply the EM approach from [1, Sec. V-A]. For this, we suggest initializing $\xi = 0.1$ (since Section IV-E shows that this works well over a wide range of problems) and initializing ν_0^x and ν^w using a variation on the procedure suggested for MC that accounts for the sparsity of x_{nl}:

$$\nu^w = \frac{\|P_\Omega(\boldsymbol{Y})\|_F^2}{(\mathrm{SNR}^0 + 1)|\Omega|} \tag{24}$$

$$\nu_0^x = \frac{1}{N\xi}\left[\frac{\|P_\Omega(\boldsymbol{Y})\|_F^2}{|\Omega|} - \nu^w\right]. \tag{25}$$

D. Avoiding Local Minima

The DL problem is fraught with local minima (see, e.g., [24]), and so it is common to run iterative algorithms several times from different random initializations. For BiG-AMP, we suggest keeping the result of one initialization over the previous if both[20] the residual error $\|\widehat{\boldsymbol{Z}} - \boldsymbol{Y}\|_F$ and the average sparsity (as measured by $\frac{1}{NL}\sum_{nl}\Pr\{\mathsf{x}_{nl} \neq 0 | \boldsymbol{Y} = \boldsymbol{Y}\}$) decrease.

E. Dictionary Learning Experiments

In this section, we numerically investigate the performance of EM-BiG-AMP for DL, as described in Section IV. Comparisons are made with the greedy K-SVD algorithm [26], the SPAMS implementation of the online approach [27], and the ER-SpUD(proj) approach for square dictionaries [24].

1) Noiseless Square Dictionary Recovery: We first investigate recovery of square (i.e., $N = M$) dictionaries from noise-free observations, repeating the experiment from [24]. For this, realizations of the true dictionary $\boldsymbol{A}$ were created by drawing elements independently from the standard normal distribution and subsequently scaling the columns to unit ℓ_2 norm. Meanwhile, realizations of the true $\boldsymbol{X}$ were created by selecting K entries in each column uniformly at random and drawing their values from the standard normal distribution, while setting all other entries to zero. Finally, the observations were constructed as $\boldsymbol{Y} = \boldsymbol{A}\boldsymbol{X}$, from which the algorithms estimated $\boldsymbol{A}$ and $\boldsymbol{X}$ (up to a permutation and scale). The accuracy of the dictionary estimate $\widehat{\boldsymbol{A}}$ was quantified using the relative NMSE metric [24]

$$\mathrm{NMSE}(\widehat{\boldsymbol{A}}) \triangleq \min_{\boldsymbol{J}\in\mathcal{J}} \frac{\|\widehat{\boldsymbol{A}}\boldsymbol{J} - \boldsymbol{A}\|_F^2}{\|\boldsymbol{A}\|_F^2}, \tag{26}$$

where $\boldsymbol{J}$ is a generalized permutation matrix used to resolve the permutation and scale ambiguities.

The subplots on the left of Fig. 11 show the mean NMSE achieved by K-SVD, SPAMS, ER-SpUD(proj), and EM-BiG-AMP,[21] respectively, over 50 problem realizations, for various

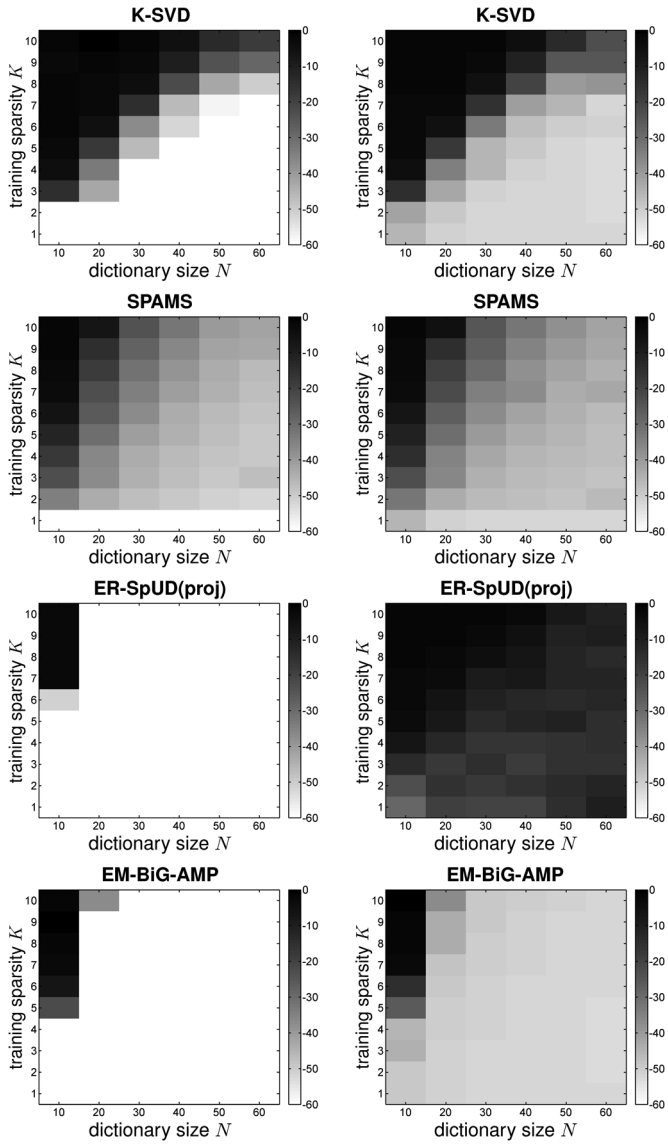

Fig. 11. Mean NMSE (over 10 trials) for recovery of an $N \times N$ dictionary from $L = 5N\log N$ training samples, each of sparsity K, in the noiseless case (left) and under AWGN of 40 dB SNR (right), for several algorithms.

[20] As an alternative, if both the previous and current solutions achieve sufficiently small residual error, then only the average sparsity is considered in the comparison.

[21] EM-BiG-AMP was allowed up to 20 EM iterations, with each EM iteration allowed a minimum of 30 and a maximum of 1500 BiG-AMP iterations. K-SVD was allowed up to 100 iterations and provided with knowledge of the true sparsity K. SPAMS was allowed 1000 iterations and run using the hand-tuned penalty $\lambda = 0.1/\sqrt{N}$. The non-iterative ER-SpUD(proj) was run using code provided by the authors without modification.

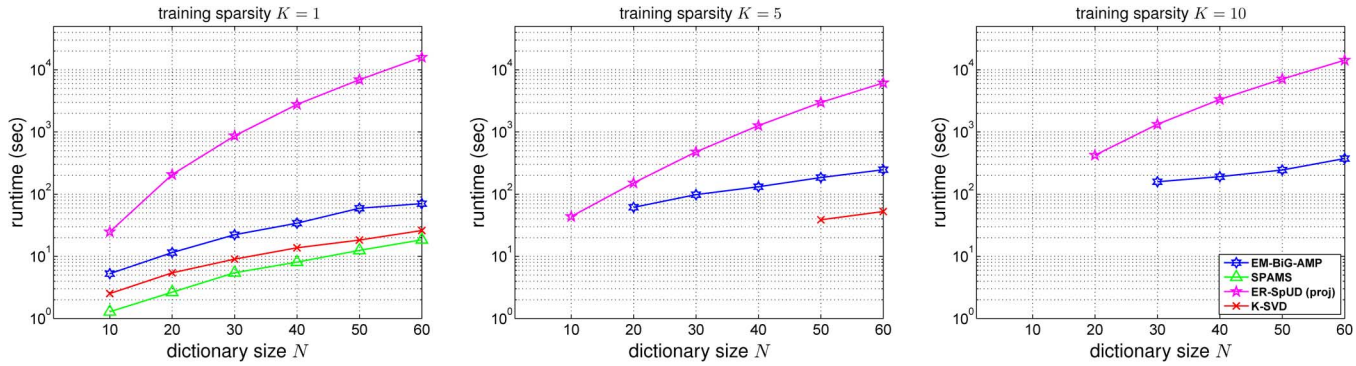

Fig. 12. Median runtime until termination (over 10 trials) versus dictionary size N, for noiseless recovery of a square dictionary from $L = 5N \log N$ K-sparse samples, for several values of training sparsity K. Missing values indicate that the algorithm did not achieve the required NMSE before termination and correspond to the black regions in the panes on the left of Fig. 11.

combinations of dictionary size $N \in \{10, \ldots, 60\}$ and data sparsity $K \in \{1, \ldots, 10\}$, using $L = 5N \log N$ training examples. K-SVD, SPAMS, and EM-BiG-AMP were run with 10 different random initializations for each problem realization. To choose among these initializations, EM-BiG-AMP used the procedure from Section IV-D, while K-SVD and SPAMS used oracle knowledge to choose the NMSE-minimizing initialization.

The left column in Fig. 11 shows that the K-SVD, ER-SpUD(proj), and EM-BiG-AMP algorithms all exhibit a relatively sharp phase-transition curve (PTC) separating success and failure regions, and that ER-SpUD(proj)'s PTC is the best, while EM-BiG-AMP's PTC is very similar. Meanwhile, K-SVD's PTC is much worse and SPAMS performance is not good enough to yield a sharp phase transition,[22] despite the fact that both use oracle knowledge. EM-BiG-AMP, by contrast, was not provided with any knowledge of the DL problem parameters, such as the true sparsity or noise variance (in this case, zero).

For the same problem realizations, Fig. 12 shows the runtime to NMSE $= -60$ dB (measured using MATLAB's `tic` and `toc`) versus dictionary size N. The results show that EM-BG-AMP runs within an order-of-magnitude of the fastest algorithm (SPAMS) and more than two orders-of-magnitude faster than ER-SpUD(proj)[23] for larger dictionaries.

2) Noisy Square Dictionary Recovery: Next we examined the recovery of square dictionaries from AWGN-corrupted observations. For this, we repeated the experiment from the previous section, but constructed the observations as $\boldsymbol{Y} = \boldsymbol{Z} + \boldsymbol{W}$, where $\boldsymbol{Z} = \boldsymbol{AX}$ and $\boldsymbol{W}$ contained AWGN samples with variance adjusted to achieve an $\mathrm{SNR} = \mathrm{E}\{\sum_{m,l} |\mathsf{z}_{ml}|^2\}/\mathrm{E}\{\sum_{m,l} |\mathsf{y}_{ml} - \mathsf{z}_{ml}|^2\}$ of 40 dB.

The right subplots in Fig. 11 show the mean value (over 10 trials) of the relative NMSE from (26) when recovering an $N \times N$ dictionary from $L = 5N \log N$ training samples of sparsity K, for various combinations of N and K. These subplots show that ER-SpUD(proj) falls apart in the noisy case, which is perhaps not surprising given that it is intended only for noiseless problems. Meanwhile, the K-SVD, SPAMS, and EM-BiG-AMP

[22]Our results for SPAMS and ER-SPUD(proj) in the left column of Fig. 11 are nearly identical to those in [24, Fig. 1], while our results for K-SVD are noticeably better.

[23]The simpler "SC" variant of ER-SpUD reduces the computational cost relative to the "proj" variant, but results in a significantly worse PTC (see [24, Fig. 1]) and remains slower than EM-BiG-AMP for larger problems.

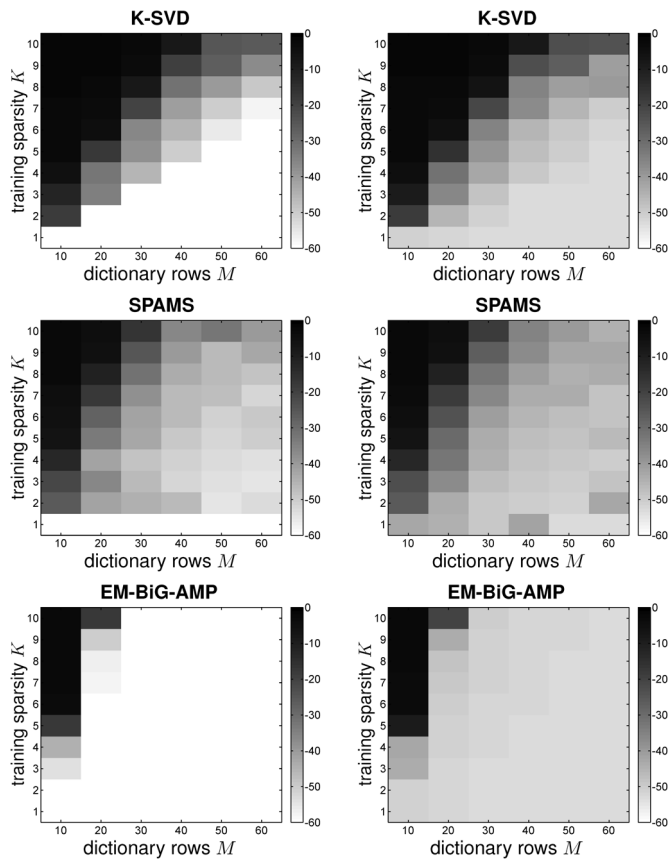

Fig. 13. Mean NMSE (over 10 trials) for recovery of an $M \times (2M)$ dictionary from $L = 10M \log(2M)$ training samples, each of sparsity K, in the noiseless case (left) and under AWGN of 40 dB SNR (right), for several algorithms.

algorithms appear to degrade gracefully in the presence of noise, yielding NMSE ≈ -50 dB at points below the noiseless PTCs.

3) Recovery of Overcomplete Dictionaries: Finally, we consider recovery of overcomplete $M \times N$ dictionaries, i.e., the case where $M < N$. In particular, we investigated the twice overcomplete case, $N = 2M$. For this, random problem realizations were constructed in the same manner as described earlier, except for the dictionary dimensions.

The left column of Fig. 13 shows the mean value (over 10 trials) of the relative NMSE for *noiseless* recovery, while the right column shows the corresponding results for *noisy* recovery. In all cases, $L = 5N \log N = 10M \log(2M)$ training

samples were provided. EM-BiG-AMP, K-SVD, and SPAMS all give very similar results to Fig. 11 for the square-dictionary case, verifying that these techniques are equally suited to the recovery of over-complete dictionaries. ER-SpUD(proj), however, only applies to square dictionaries and hence was not tested here.

4) Summary: In summary, Figs. 11–13 show that, for noiseless square dictionary learning, EM-BiG-AMP yields an empirical PTC that is nearly as good as the state-of-the-art ER-SpUD(proj) algorithm and much better than those of (genie-aided) K-SVD and SPAMS. However, the figures show that, in comparison to ER-SpUD(proj), EM-BiG-AMP is fast for large (square) dictionaries, robust to AWGN, and applicable to non-square dictionaries.

We recall that Krzakala, Mézard, and Zdeborová recently proposed an AMP-based approach to blind calibration and dictionary learning [28] that bears similarity to our scalar-variance BiG-AMP under AWGN-corrupted observations (recall [1, footnote 6]). Although their approach gave good results for blind calibration, they report that it was "not able to solve" the DL problem [28]. We attribute EM-BiG-AMP's success with DL (as evidenced by Figs. 11–13) to the adaptive damping procedure proposed in [1, Sec. IV-B], the initialization procedure proposed in Section IV-B, the EM-learning procedure proposed in Section IV-C, and the re-initialization procedure proposed in Section IV-D.

V. Conclusion

This manuscript is the second part in a two-part work on BiG-AMP, an extension of the G-AMP algorithm [7] originally proposed for high-dimensional generalized-*linear* regression in the context of compressive sensing, to generalized-*bilinear* regression, with applications in matrix completion, robust PCA, dictionary learning, and related matrix-factorization problems. In Part I [1], we proposed and derived the BiG-AMP algorithm, as well as an adaptive damping mechanism to aid convergence under realistic problem sizes, an expectation-maximization (EM)-based method to automatically tune the parameters of the assumed priors, and two rank-selection strategies. Here, in Part II, we detailed the application of BiG-AMP to matrix completion, robust PCA, and dictionary learning, and we presented the results of an extensive numerical investigation into the performance of BiG-AMP on both synthetic and real-world datasets. These results demonstrate that BiG-AMP yields excellent reconstruction accuracy (often best in class) while maintaining competitive runtimes, and that the proposed EM and rank-selection strategies successfully avoid the need to tune algorithmic parameters.

The excellent empirical results reported here motivate future work on the analysis of EM-BiG-AMP, on the extension of EM-BiG-AMP to, e.g., structured-sparse or parametric models, and on the application of EM-BiG-AMP to practical problems in high-dimensional inference. For example, preliminary results on the application of EM-BiG-AMP to hyperspectral unmixing (a form of non-negative matrix factorization) have been reported in [25] and are very encouraging.

Acknowledgment

The authors would like to thank Sundeep Rangan, Florent Krzakala, and Lenka Zdeborová for insightful discussions on various aspects of AMP-based inference. We would also like to thank Subhojit Som, Jeremy Vila, and Justin Ziniel for helpful discussions about EM and turbo methods for AMP.

References

[1] J. T. Parker, P. Schniter, and V. Cevher, "Bilinear generalized approximate message passing—Part I: Derivation," *IEEE Trans. Signal Process.*, vol. 62, no. 22, pp. 5839–5853, 2014.

[2] P. Schniter and V. Cevher, "Approximate message passing for bilinear models," in *Proc. Workshop Signal Process. Adapt. Sparse Struct. Repr. (SPARS)*, Edinburgh, Scotland, Jun. 2011, p. 68.

[3] P. Schniter, J. T. Parker, and V. Cevher, "Bilinear generalized approximate message passing (BiG-AMP) for matrix recovery problems," presented at the Inf. Theory Appl. Workshop (ITA), La Jolla, CA, USA, Feb. 2012.

[4] J. T. Parker and P. Schniter, "Bilinear generalized approximate message passing (BiG-AMP) for matrix completion," presented at the Asilomar Conf., Pacific Grove, CA, USA, Nov. 2012.

[5] D. L. Donoho, A. Maleki, and A. Montanari, "Message passing algorithms for compressed sensing," *Proc. Nat. Acad. Sci.*, vol. 106, no. 45, pp. 18 914–18 919, Nov. 2009.

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

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

[8] E. J. Candés and Y. Plan, "Matrix completion with noise," *Proc. IEEE*, vol. 98, no. 6, pp. 925–936, Jun. 2010.

[9] M. E. Tipping and C. M. Bishop, "Probabilistic principal component analysis," *J. Roy. Statist. Soc. B*, vol. 61, no. 3, pp. 611–622, 1999.

[10] Y. Lim and Y. Teh, "Variational Bayesian approach to movie rating prediction," in *Proc. KDD Cup Workshop*, 2007, pp. 15–21, Citeseer.

[11] R. Salakhutdinov and A. Mnih, "Probabilistic matrix factorization," in *Proc. Neural Inf. Process. Syst. Conf.*, Vancouver, BC, Dec. 2008, pp. 1257–1264.

[12] N. D. Lawrence and R. Urtasun, "Non-linear matrix factorization with Gaussian processes," in *Proc. Int. Conf. Mach. Learning*, Montreal, QC, Canada, Jun. 2009, pp. 601–608.

[13] Z. Wen, W. Yin, and Y. Zhang, "Solving a low-rank factorization model for matrix completion by a nonlinear successive over-relaxation algorithm," *Math. Programm. Comput.* vol. 4, pp. 333–361, 2012 [Online]. Available: http://dx.doi.org/10.1007/s12532-012-0044-1

[14] S. D. Babacan, M. Luessi, R. Molina, and A. K. Katsaggelos, "Sparse Bayesian methods for low-rank matrix estimation," *IEEE Trans. Signal Process.*, vol. 60, no. 8, pp. 3964–3977, Aug. 2012.

[15] J. P. Vila and P. Schniter, "Expectation-maximization Gaussian-mixture approximate message passing," *IEEE Trans. Signal Process.*, vol. 61, no. 19, pp. 4658–4672, Oct. 2013.

[16] P. Stoica and Y. Selen, "Model-order selection: A review of information criterion rules," *IEEE Signal Process. Mag.*, vol. 21, no. 4, pp. 36–47, Jul. 2004.

[17] G. Marjanovic and V. Solo, "On optimization and matrix completion," *IEEE Trans. Signal Process.*, vol. 60, no. 11, pp. 5714–5724, 2012.

[18] Z. Lin, M. Chen, L. Wu, and Y. Ma, "The augmented Lagrange multiplier method for exact recovery of corrupted low-rank matrices," 2010, arXiv preprint: 1009.5055 [Online]. Available: http://arxiv.org/abs/1009.5055

[19] L. Balzano, R. Nowak, and B. Recht, "Online identification and tracking of subspaces from highly incomplete information," in *Proc. Allerton Conf. Commun. Control Comput.*, Sep. 2010, pp. 704–711.

[20] A. Kyrillidis and V. Cevher, "Matrix recipes for hard thresholding methods," *J. Math. Imag. Vis.*, vol. 48, pp. 235–265, 2014.

[21] E. J. Candès, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis?," *J. ACM* vol. 58, no. 3, pp. 11:1–11:37, Jun. 2011 [Online]. Available: http://doi.acm.org/10.1145/1970392.1970395

[22] J. He, L. Balzano, and J. Lui, "Online robust subspace tracking from partial information," 2011, arXiv preprint: 1109.3827 [Online]. Available: http://arxiv.org/abs/1109.3827

[23] R. Rubinstein, A. Bruckstein, and M. Elad, "Dictionaries for sparse representation modeling," *Proc. IEEE*, vol. 98, no. 6, pp. 1045–1057, 2010.
[24] D. A. Spielman, H. Wang, and J. Wright, "Exact recovery of sparsely-used dictionaries," in *JMLR: Workshop Conf. Proc. 25th Annu. Conf. Learn. Theory*, 2012, vol. 23, pp. 37.1–37.18 [Online]. Available: http://jmlr.org/proceedings/papers/v23/spielman12/spielman12.pdf
[25] J. Vila, P. Schniter, and J. Meola, "Hyperspectral image unmixing via bilinear generalized approximate message passing," *Proc. SPIE*, vol. 8743, no. 87430Y, p. 9, 2013.
[26] M. Aharon, M. Elad, and A. Bruckstein, "K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Trans. Signal Process.*, vol. 54, no. 11, pp. 4311–4322, 2006.
[27] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, "Online learning for matrix factorization and sparse coding," *J. Mach. Learn. Res.*, vol. 11, pp. 19–60, 2010.
[28] F. Krzakala, M. Mézard, and L. Zdeborová, "Phase diagram and approximate message passing for blind calibration and dictionary learning," in *Proc. IEEE Int. Symp. Inf. Theory*, 2013, pp. 659–663.

Jason T. Parker (M'06) received the B.S., M.S., and Ph.D. degrees in electrical and computer engineering from The Ohio State University, Columbus, in 2004, 2006, and 2014, respectively. Since 2006, he has been a research engineer with the U.S. Air Force Research Laboratory Sensors Directorate. His research interests currently include compressive sensing, adaptive signal processing, and inverse problems, with applications to radar target detection and imaging.

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. in 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.

In 2003, Dr. Schniter received the National Science Foundation CAREER Award. His areas of interest currently include statistical signal processing, wireless communications and networks, and machine learning.

Volkan Cevher (SM'10) received the B.S. (valedictorian) degree in electrical engineering in 1999 from Bilkent University in Ankara, Turkey, and he received the Ph.D. degree in Electrical and Computer Engineering in 2005 from the Georgia Institute of Technology in Atlanta. He held research scientist positions at the University of Maryland, College Park, from 2006 to 2007 and at Rice University in Houston, Texas, from 2008 to 2009. Currently, he is an Assistant Professor at the Swiss Federal Institute of Technology Lausanne with a complimentary appointment at the Electrical and Computer Engineering Department at Rice University. His research interests include signal processing theory, machine learning, graphical models, and information theory. He received a Best Paper Award at SPARS in 2009 and an ERC StG in 2011.