Bilinear generalized approximate message passing (BiG-AMP) for High Dimensional Inference

Phil Schniter

THE OHIO STATE UNIVERSITY

Collaborators: Jason Parker @OSU, Jeremy Vila @OSU, and Volkan Cehver @EPFL

With support from NSF CCF-1218754, NSF CCF-1018368, NSF IIP-0968910,
and DARPA/ONR N66001-10-1-4090

Oct. 10, 2013

Four Important High Dimensional Inference Problems

1. Matrix Completion (MC):

   Recover low-rank matrix $\boldsymbol{Z}$
   from noise-corrupted incomplete observations $\boldsymbol{Y} = \mathcal{P}_\Omega(\boldsymbol{Z} + \boldsymbol{W})$.

2. Robust Principle Components Analysis (RPCA):

   Recover low-rank matrix $\boldsymbol{Z}$ and sparse matrix $\boldsymbol{S}$
   from noise-corrupted observations $\boldsymbol{Y} = \boldsymbol{Z} + \boldsymbol{S} + \boldsymbol{W}$.

3. Dictionary Learning (DL):

   Recover (possibly overcomplete) dictionary $\boldsymbol{A}$ and sparse matrix $\boldsymbol{X}$
   from noise-corrupted observations $\boldsymbol{Y} = \boldsymbol{A}\boldsymbol{X} + \boldsymbol{W}$.

4. Non-negative Matrix Factorization (NMF):

   Recover non-negative matrices $\boldsymbol{A}$ and $\boldsymbol{X}$
   from noise-corrupted observations $\boldsymbol{Y} = \boldsymbol{A}\boldsymbol{X} + \boldsymbol{W}$.

The following generalizations may also be of interest:

- RPCA, DL, or NMF with incomplete observations.
- RPCA or DL with structured sparsity.
- Any of the above with non-additive corruptions (e.g., one-bit or phaseless $\boldsymbol{Y}$).

Contributions

- We propose a novel unified approach to these matrix-recovery problems that leverages the recent framework of approximate message passing (AMP).

- While previous AMP algorithms have been proposed for the linear model:
  - Infer $\boldsymbol{x} \sim \prod_n p_{\mathsf{x}}(x_n)$ from $\boldsymbol{y} = \boldsymbol{\Phi x} + \boldsymbol{w}$ with AWGN $\boldsymbol{w}$ and known $\boldsymbol{\Phi}$. [Donoho/Maleki/Montanari'10]

  or the generalized linear model:
  - Infer $\boldsymbol{x} \sim \prod_n p_{\mathsf{x}}(x_n)$ from $\boldsymbol{y} \sim \prod_m p_{\mathsf{y|z}}(y_m|z_m)$ with hidden $\boldsymbol{z} = \boldsymbol{\Phi x}$ and known $\boldsymbol{\Phi}$. [Rangan'10]

  our work tackles the generalized *bilinear* model:
  - Infer $\boldsymbol{A} \sim \prod_{m,n} p_{\mathsf{a}}(a_{mn})$ and $\boldsymbol{X} \sim \prod_{n,l} p_{\mathsf{x}}(x_{nl})$ from $\boldsymbol{Y} \sim \prod_{m,l} p_{\mathsf{y|z}}(y_{ml}|z_{ml})$ with hidden $\boldsymbol{Z} = \boldsymbol{AX}$. [Schniter/Cevher'11]

- In addition, we propose methods to select the rank of $\boldsymbol{Z}$, to estimate the parameters of $p_{\mathsf{a}}, p_{\mathsf{x}}, p_{\mathsf{y|z}}$, and to handle non-separable priors on $\boldsymbol{A}, \boldsymbol{X}, \boldsymbol{Y}|\boldsymbol{Z}$.

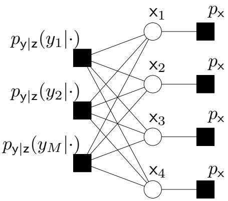
Outline

1. Bilinear Generalized AMP (BiG-AMP)
   - Background on AMP
   - BiG-AMP heuristics
   - Example configurations/applications
2. Practicalities
   - Adaptive damping
   - Parameter tuning
   - Rank selection
   - Non-separable priors
3. Numerical results:
   - Matrix completion
   - Robust PCA
   - Dictionary learning
   - Hyperspectral unmixing (via NMF)

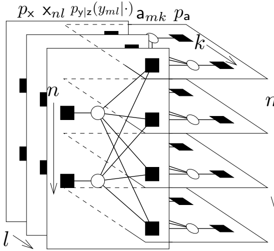
Bilinear Generalized AMP (BiG-AMP)

- BiG-AMP is a Bayesian approach that uses approximate message passing (AMP) strategies to infer $(\boldsymbol{Z}, \boldsymbol{A}, \boldsymbol{X})$.

Generalized Linear:



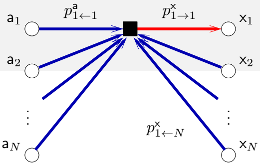
Generalized Bilinear:



- In AMP, beliefs are propagated on a loopy factor graph using approximations that exploit certain blessings of dimensionality:
  1. Gaussian message approximation (motivated by central limit theorem),
  2. Taylor-series approximation of message differences.

- Rigorous analyses of GAMP for CS (with large iid sub-Gaussian Φ) reveal a state evolution whose fixed points are optimal when unique. [Javanmard/Montanari'12]

BiG-AMP sum-product heuristics



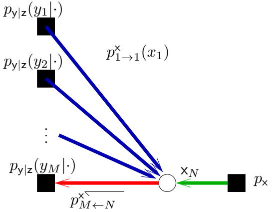
1. Message from i^{th} node of $\boldsymbol{Z}$ to j^{th} node of $\boldsymbol{X}$:

$$p^{\mathsf{x}}_{i\to j}(x_j) \propto \int_{\{a_n\}_{n=1}^N,\{x_n\}_{n\neq j}} p_{\mathsf{y}|\mathsf{z}}\Big(y_i \Big| \overbrace{\textstyle\sum_n a_n x_n}^{z_i|x_j \,\approx\, \mathcal{N} \text{ via CLT!}}\Big)\Big(\prod_n p^{\mathsf{a}}_{i\leftarrow n}(a_n)\Big)\Big(\prod_{n\neq j} p^{\mathsf{x}}_{i\leftarrow n}(x_n)\Big)$$

$$\approx \int_{z_i} p_{\mathsf{y}|\mathsf{z}}(y_i|z_i)\,\mathcal{N}\big(z_i; \hat{z}_i(x_j), \nu^z_i(x_j)\big) \;\approx \mathcal{N} \quad \text{(exact for AWGN!)}$$

(A similar thing then happens with the messages from $\boldsymbol{Z}$ to $\boldsymbol{A}$.)
To compute $\hat{z}_i(x_j), \nu^z_i(x_j)$, the means and variances of $p^{\mathsf{x}}_{i\leftarrow n}$ & $p^{\mathsf{a}}_{i\leftarrow n}$ suffice, and thus we have Gaussian message passing!

2. Although Gaussian, we still have $4MLN$ messages to compute (too many!). Exploiting similarity among the messages $\{p^{\mathsf{x}}_{i\leftarrow j}\}_{i=1}^M$, we employ a Taylor-series approximation whose error vanishes as $M\to\infty$. (Same for $\{p^{\mathsf{a}}_{i\leftarrow j}\}_{i=1}^L$ with $L\to\infty$.) In the end, we only need to compute $\mathcal{O}(ML)$ messages!



Example Configurations

1. Matrix Completion (MC):

   Recover low-rank $\boldsymbol{Z} = \boldsymbol{A}\boldsymbol{X}$ from $\boldsymbol{Y} = \mathcal{P}_\Omega(\boldsymbol{Z} + \boldsymbol{W})$.

   $\mathsf{a}_{ml} \sim \mathcal{N}(0,1)$, $\mathsf{x}_{nl} \sim \mathcal{N}(\mu_\mathsf{x}, v_\mathsf{x})$, and $\mathsf{y}_{ml}|\mathsf{z}_{ml} \sim \begin{cases} \mathcal{N}(\mathsf{z}_{ml}, v_\mathsf{w}) & (m,l) \in \Omega \\ \mathbb{1}_0 & (m,l) \notin \Omega \end{cases}$

2. Robust PCA (RPCA):

   a) Recover low-rank $\boldsymbol{Z} = \boldsymbol{A}\boldsymbol{X}$ from $\boldsymbol{Y} = \boldsymbol{Z} + \boldsymbol{E}$.
      $\mathsf{a}_{mn} \sim \mathcal{N}(0,1)$, $\mathsf{x}_{nl} \sim \mathcal{N}(\mu_\mathsf{x}, v_\mathsf{x})$, $\mathsf{y}_{ml}|\mathsf{z}_{ml} \sim \mathcal{GM}_2(\lambda, \mathsf{z}_{ml}, v_\mathsf{w}+v_\mathsf{s}, \mathsf{z}_{ml}, v_\mathsf{w})$

   b) Recover low-rank $\boldsymbol{Z} = \boldsymbol{A}\boldsymbol{X}$ and sparse $\boldsymbol{S}$ from $\boldsymbol{Y} = [\boldsymbol{A} \ \ \boldsymbol{I}][\boldsymbol{X}^\mathsf{T} \ \ \boldsymbol{S}^\mathsf{T}]^\mathsf{T} + \boldsymbol{W}$.
      $\mathsf{a}_{mn} \sim \mathcal{N}(0,1)$, $\mathsf{x}_{nl} \sim \mathcal{N}(\mu_\mathsf{x}, v_\mathsf{x})$, $\mathsf{s}_{ml} \sim \mathcal{BG}(\lambda, 0, v_\mathsf{s})$, $\mathsf{y}_{ml}|\mathsf{z}_{ml} \sim \mathcal{N}(\mathsf{z}_{ml}, v_\mathsf{w})$

3. Dictionary Learning (DL):

   Recover dictionary $\boldsymbol{A}$ and sparse $\boldsymbol{X}$ from $\boldsymbol{Y} = \boldsymbol{A}\boldsymbol{X} + \boldsymbol{W}$.

   $\mathsf{a}_{mn} \sim \mathcal{N}(0,1)$, $\mathsf{x}_{nl} \sim \mathcal{BG}(\lambda, 0, v_\mathsf{x})$, and $\mathsf{y}_{ml}|\mathsf{z}_{ml} \sim \mathcal{N}(\mathsf{z}_{ml}, v_\mathsf{w})$

4. Non-negative Matrix Factorization (NMF):

   Recover non-negative $\boldsymbol{A}$ and $\boldsymbol{X}$ (up to perm/scale) from $\boldsymbol{Y} = \boldsymbol{A}\boldsymbol{X} + \boldsymbol{W}$.

   $\mathsf{a}_{mn} \sim \mathcal{N}_+(0, \mu_\mathsf{a})$, $\mathsf{x}_{nl} \sim \mathcal{N}_+(0, \mu_\mathsf{x})$, and $\mathsf{y}_{ml}|\mathsf{z}_{ml} \sim \mathcal{N}(\mathsf{z}_{ml}, v_\mathsf{w})$

Example Configurtions (cont.)

5. One-bit Matrix Completion (MC):

   Recover low-rank $\boldsymbol{Z} = \boldsymbol{A}\boldsymbol{X}$ from $\boldsymbol{Y} = \mathcal{P}_\Omega(\mathrm{sgn}(\boldsymbol{Z} + \boldsymbol{W}))$.

   $\mathsf{a}_{ml} \sim \mathcal{N}(0,1)$, $\mathsf{x}_{nl} \sim \mathcal{N}(\mu_\mathsf{x}, v_\mathsf{x})$, and $\mathsf{y}_{ml}|\mathsf{z}_{ml} \sim \begin{cases} \text{probit} & (m,l) \in \Omega \\ \mathbb{1}_0 & (m,l) \notin \Omega \end{cases}$

   ...leveraging previous work on one-bit/classification GAMP [Ziniel/Schniter'13]

6. Phaseless Matrix Completion (MC):

   Recover low-rank $\boldsymbol{Z} = \boldsymbol{A}\boldsymbol{X}$ from $\boldsymbol{Y} = \mathcal{P}_\Omega(\mathrm{abs}(\boldsymbol{Z} + \boldsymbol{W}))$.

   $\mathsf{a}_{ml} \sim \mathcal{N}(0,1)$, $\mathsf{x}_{nl} \sim \mathcal{N}(\mu_\mathsf{x}, v_\mathsf{x})$, and

   $$p_{\mathsf{y}_{ml}|\mathsf{z}_{ml}}(y|z) = \begin{cases} \exp\left(-\frac{|y|^2+|z|^2}{v_\mathsf{w}}\right) I_0\left(\frac{|y||z|}{v_\mathsf{w}}\right) & (m,l) \in \Omega \\ \mathbb{1}_0 & (m,l) \notin \Omega \end{cases}$$

   ...leveraging previous work on phase-retrieval GAMP [Schniter/Rangan'12]

7. and so on ...

Adaptive Damping

- The heuristics used to derive GAMP hold in the large system limit: $M, N, L \to \infty$ with fixed M/N, M/L.
- In practice, M, N, L are finite and the rank N is often very small!
- To prevent BiG-AMP from diverging, we damp the updates using an adjustable step-size parameter $\beta \in (0, 1]$.
- Moreover, we adapt β by monitoring (an approximation to) the cost function minimized by BiG-AMP and adjusting β as needed to ensure decreasing cost, leveraging similar methods from GAMP [Rangan/Schniter/Riegler/Fletcher/Cevher'13].

$$\hat{J}(t) = \sum_{n,l} D\Big(\hat{p}_{\mathsf{x}_{nl}|\mathbf{Y}}(\,\cdot\,|\,\boldsymbol{Y})\Big\|\,p_{\mathsf{x}_{nl}}(\cdot)\Big) \quad \leftarrow \text{KL divergence between posterior \& prior}$$

$$+ \sum_{m,n} D\Big(\hat{p}_{\mathsf{a}_{mn}|\mathbf{Y}}(\,\cdot\,|\,\boldsymbol{Y})\Big\|\,p_{\mathsf{a}_{mn}}(\cdot)\Big)$$

$$- \sum_{m,l} \mathrm{E}_{\mathcal{N}(\mathsf{z}_{ml};\bar{p}_{ml}(t);\nu^{p}_{ml}(t))}\big\{\log p_{\mathsf{y}_{ml}|\mathsf{z}_{ml}}(y_{ml}\,|\,\mathsf{z}_{ml})\big\}.$$

Parameter Tuning via EM

- We treat the parameters $\boldsymbol{\theta}$ that determine the priors $p_{\mathsf{x}}, p_{\mathsf{a}}, p_{\mathsf{y}|\mathsf{z}}$ as deterministic unknowns and compute (approximate) ML estimates using expectation-maximization (EM), as done for GAMP in [Vila/Schniter'13].
- Taking $\boldsymbol{X}$, $\boldsymbol{A}$, and $\boldsymbol{Z}$ to be the hidden variables, the EM recursion becomes

$$
\begin{aligned}
\hat{\boldsymbol{\theta}}^{k+1} &= \arg\max_{\boldsymbol{\theta}} \mathrm{E}\left\{ \log p_{\boldsymbol{X},\boldsymbol{A},\boldsymbol{Z},\boldsymbol{Y}}(\boldsymbol{X},\boldsymbol{A},\boldsymbol{Z},\boldsymbol{Y};\boldsymbol{\theta}) \,\middle|\, \boldsymbol{Y};\hat{\boldsymbol{\theta}}^{k} \right\} \\
&= \arg\max_{\boldsymbol{\theta}} \Bigg\{ \sum_{n,l} \mathrm{E}\left\{ \log p_{\mathsf{x}_{nl}}(\mathsf{x}_{nl};\boldsymbol{\theta}) \,\middle|\, \boldsymbol{Y};\hat{\boldsymbol{\theta}}^{k} \right\} \\
&\qquad + \sum_{m,n} \mathrm{E}\left\{ \log p_{\mathsf{a}_{mn}}(\mathsf{a}_{mn};\boldsymbol{\theta}) \,\middle|\, \boldsymbol{Y};\hat{\boldsymbol{\theta}}^{k} \right\} \\
&\qquad + \sum_{m,l} \mathrm{E}\left\{ \log p_{\mathsf{y}_{ml}|\mathsf{z}_{ml}}(y_{ml} \,|\, \mathsf{z}_{ml};\boldsymbol{\theta}) \,\middle|\, \boldsymbol{Y};\hat{\boldsymbol{\theta}}^{k} \right\} \Bigg\}
\end{aligned}
$$

- For tractability, the $\boldsymbol{\theta}$-maximization is performed one variable at a time.

Rank Selection

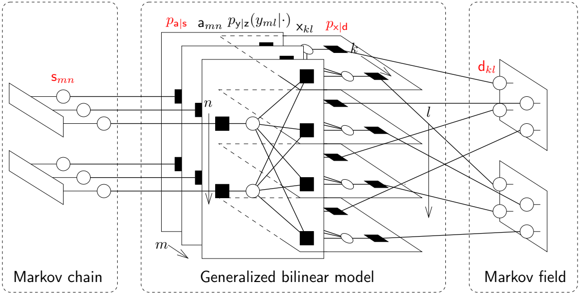
- In practice, the rank of $\boldsymbol{Z}$ (i.e., # columns in $\boldsymbol{A}$ and rows in $\boldsymbol{X}$) is unknown.
- We propose two methods for rank selection:
  1. Penalized log-likelihood maximization:

$$\hat{N} = \underset{N=1,\dots,\overline{N}}{\arg\max}\; 2\log p_{\boldsymbol{Y}|\boldsymbol{Z}}(\boldsymbol{Y} \,|\, \hat{\boldsymbol{A}}_N \hat{\boldsymbol{X}}_N; \hat{\boldsymbol{\theta}}_N) - \eta(N),$$

     where $\eta(N)$ penalizes the effective number of parameters under rank N (e.g., BIC, AIC). Although $\hat{\boldsymbol{A}}_N, \hat{\boldsymbol{X}}_N, \hat{\boldsymbol{\theta}}_N$ are ideally ML estimates under rank N, we use EM-BiG-AMP estimates.
  2. Rank contraction (adapted from LMaFit [Wen/Ying/Zhang'12]):
     Run EM-BiG-AMP at maximum rank $\overline{N}$ and then set $\hat{N}$ to the location of the largest gap between singular values, but only if the gap is sufficiently large. If not, run EM-BiG-AMP and check again.
- For matrix completion we advocate the first strategy (with the AICc rule), while for robust PCA we advocate the second strategy.

Non-Separable Priors

- As described until now, BiG-AMP is limited to separable priors $p_{\boldsymbol{A}}$, $p_{\boldsymbol{X}}$, and $p_{\boldsymbol{Y}|\boldsymbol{Z}}$ (i.e., statistically independent elements).
- We circumvent this by augmenting our model with random variables that ensure *conditional* independence, and then use "turbo AMP" [Schniter'10]
- Example: to facilitate dependence within each column of $\boldsymbol{A}$, we introduce $\boldsymbol{S}$ such that $\boldsymbol{A}|\boldsymbol{S} \sim \prod_{m,n} p_{\mathsf{a}|\mathsf{s}}(a_{mn}|s_{mn})$. Similarly, we introduce $\boldsymbol{D}$ for $\boldsymbol{X}$:



Numerical Results for Matrix Completion

We compared several state-of-the-art techniques:

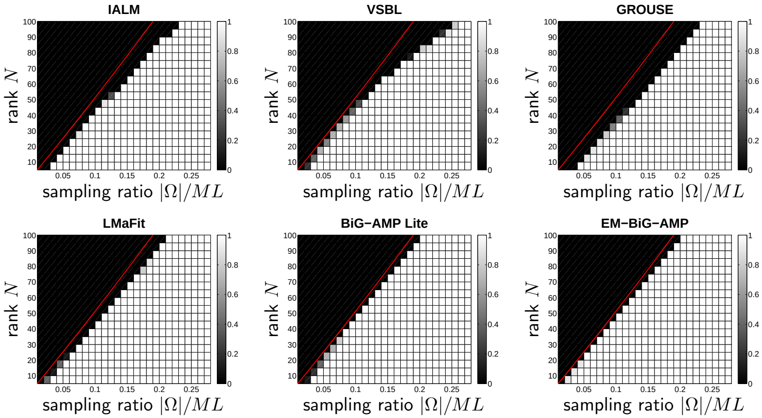
- Inexact Augmented Lagrange Multipler (IALM) [Lin/Chen/Wu/Ma'10]
  - a nuclear-norm based convex-optimization method
- GROUSE [Balzano/Nowak/Recht'10]
  - gradient descent on the Grassmanian manifold
- LMaFit [Wen/Ying/Zhang'12]
  - a non-convex approach based on non-linear successive over-relaxation
- VSBL [Babacan/Luessi/Molina/Katsaggalos'12]
  - a variational Bayesian approach.

to two variations on our proposed techniques:

- EM-BiG-AMP
  - BiG-AMP setup for Matrix Completion, with EM-adjusted $\mu_{\mathsf{x}}, v_{\mathsf{x}}, v_{\mathsf{w}}$.
- BiG-AMP Lite
  - A simplified version, based on Gaussian priors and uniform variances.

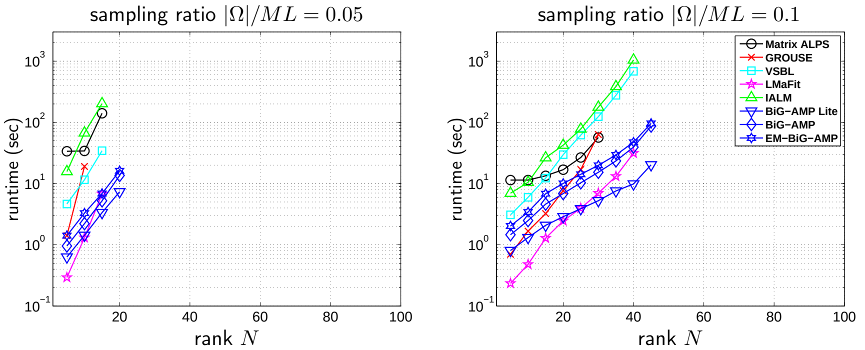
Matrix Completion: Phase Transitions

The following plots show empirical probability that NMSE < -100 dB (over 10 realizations) for noiseless completion of an $M \times L$ matrix with $M = L = 1000$.



Note that BiG-AMP-Lite and EM-BiG-AMP have the best phase transitions.

Matrix Completion: Runtime to NMSE=-100 dB

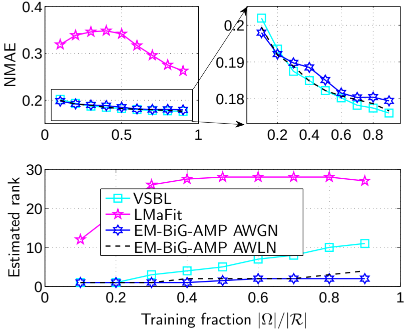


- Although LMaFit is the fastest algorithm at small rank N, BiG-AMP-Lite's superior complexity-scaling-with-N eventually wins out.
- BiG-AMP runs 1 to 2 orders-of-magnitude faster than IALM and VSBL.

Collaborative Filtering: MovieLens 100k

- $M = 943$ users, $L = 1682$ movies, $|\mathcal{R}| = 100\text{k}$ ratings $\in \{1, 2, 3, 4, 5\}$.
- Goal: from (incomplete) training subset Ω, predict test ratings $\mathcal{R} \setminus \Omega$.
- Metric: normalized mean absolute error

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



- Our experiments show that LMaFit overfits due to rank over-estimation.
- VSBL does very well, mainly because its heavy-tailed (student-t) priors are a good match to this dataset.
- EM-BiG-AMP suffers with an AWGN model, but with an additive Laplacian noise model, it matches VSBL and even does better at high undersampling.

Numerical Results for Robust PCA

We several state-of-the-art RPCA techniques

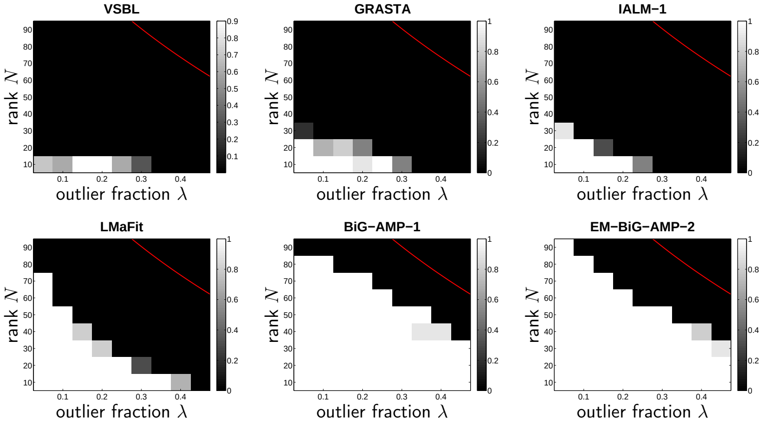
- Inexact Augmented Lagrange Multipler (IALM) [Lin/Chen/Wu/Ma'10]
  - a nuclear-norm and ℓ_1-based convex-optimization method
- GRASTA [He/Balzano/Lui'11]
  - gradient descent on the Grassmanian manifold
- LMaFit [Wen/Ying/Zhang'12]
  - a non-convex approach based on non-linear successive over-relaxation
- VSBL [Babacan/Luessi/Molina/Katsaggalos'12]
  - a variational Bayesian approach.

to two variations on our proposed techniques:

- BiG-AMP-1
  - BiG-AMP under the RPCA model using $\mathcal{BG}$ noise.
- EM-BiG-AMP-2
  - BiG-AMP using AWGN, $\mathcal{BG}$ signal, and EM-adjusted $\lambda, v_{\mathsf{s}}, \mu_{\mathsf{x}}, v_{\mathsf{x}}, v_{\mathsf{w}}$.

Robust PCA: Phase Transitions

Empirical probability of NMSE < -80 dB over 10 realizations for noiseless recovery of the low-rank component of a 200×200 outlier-corrupted matrix.

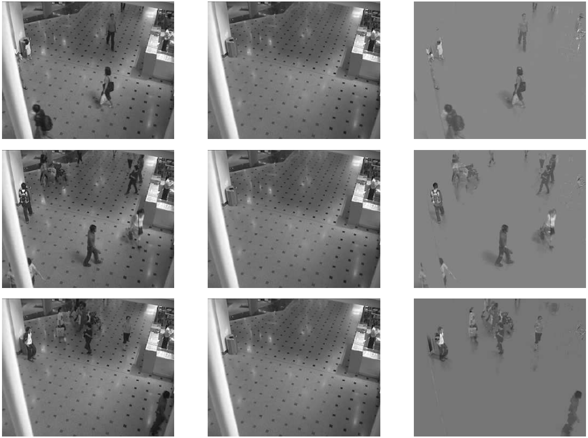


As before, the BiG-AMP methods yield the best phase transitions.

Robust PCA: Video Surveillance

EM-BiG-AMP-2 accurately extracted the low-rank background $\boldsymbol{Z}$ and the sparse foreground $\boldsymbol{S}$ from the "Mall" video sequence $\boldsymbol{Y} = \boldsymbol{Z} + \boldsymbol{S} + \boldsymbol{W}$.

original | background | foreground



Numerical Results for Dictionary Learning

We compared several state-of-the-art techniques

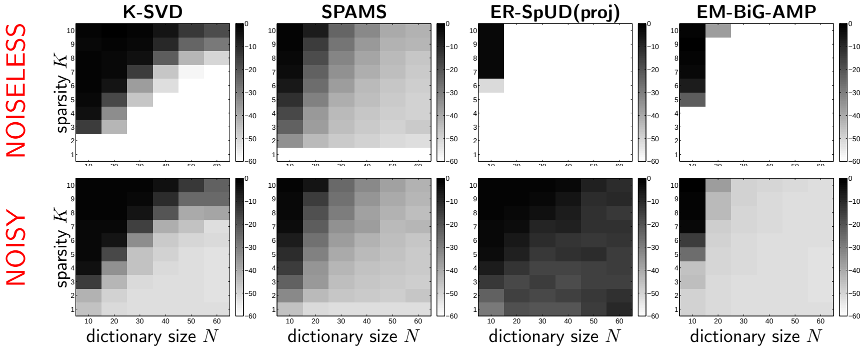
- K-SVD [Aharon/Elad/Bruckstein'06]
  - the standard; a generalization of K-means clustering
- SPAMS [Mairal/Bach/Ponce/Sapiro'10]
  - a highly optimized online approach
- ER-SpUD [Spielman/Wang/Wright'12]
  - the recent breakthrough on provably exact dictionary recovery

to our proposed technique:

- EM-BiG-AMP
  - BiG-AMP under AWGN, $\mathcal{BG}$ signal, and EM-adjusted $\lambda, \mu_{\mathsf{x}}, v_{\mathsf{x}}, v_{\mathsf{w}}$.

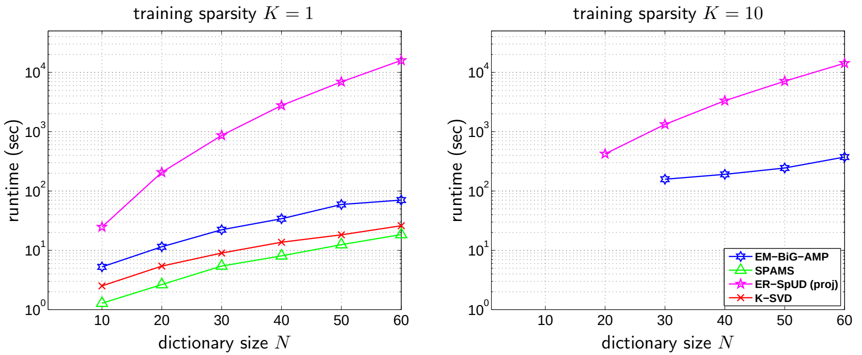
Square Dictionary Recovery: Phase Transitions

Mean NMSE over 10 realizations for recovery of an $N \times N$ dictionary from $L = 5N \log N$ examples with sparsity K:



- Noiseless case: EM-BiG-AMP's phase transition curve is much better than that of K-SVD and SPAMS and almost as good as ER-SpUD(proj)'s.
- Noisy case: EM-BiG-AMP is robust to noise, while ER-SpUD(proj) is not.

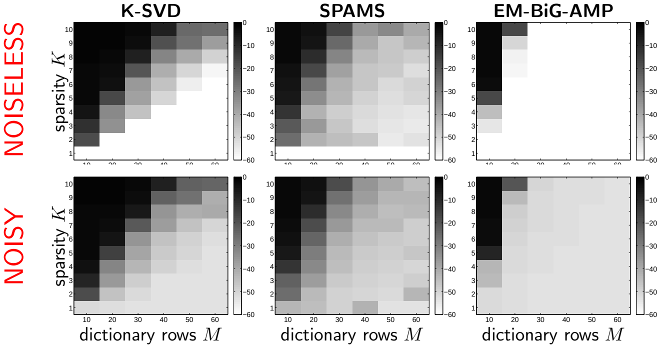
Square Dictionary Recovery: Runtime to NMSE=-60 dB



- BiG-AMP runs within a factor-of-5 from the fastest approach (SPAMS).
- BiG-AMP runs orders-of-magnitude faster than ER-SpUD(proj).

Overcomplete Dictionary Recovery: Phase Transitions

Mean NMSE over 10 realizations for recovery of an $M\times(2M)$ dictionary from $L=5N\log N=10M\log(2M)$ examples with sparsity K:



- Noiseless case: EM-BiG-AMP's phase transition curve is much better than that of K-SVD and SPAMS. Note: ER-SpUD not applicable when $M\neq N$.
- Noisy case: EM-BiG-AMP is again robust to noise.

Hyperspectral Unmixing / Nonnegative Matrix Factorization

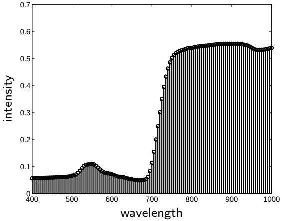
- In Hyperspectral Imaging (HSI), sensors capture M wavelengths per pixel, over a scene of L pixels comprised of N materials.

- We model the received HSI data $\boldsymbol{Y}$ as

$$\boldsymbol{Y} = \boldsymbol{A}\boldsymbol{X} + \boldsymbol{W} \in \mathbb{R}_+^{M \times L},$$

where the nth column of $\boldsymbol{A} \in \mathbb{R}_+^{M \times N}$ is the spectrum of the nth material, the lth column of $\boldsymbol{X} \in \mathbb{R}_+^{N \times L}$ describes the abundance of materials at the lth pixel (and thus must sum to one), and $\boldsymbol{W}$ is additive noise.

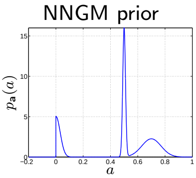
spectrum at one pixel

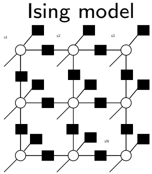


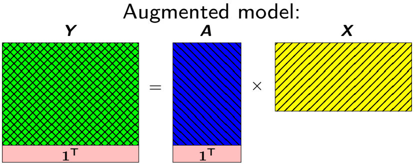
- We then jointly estimate $\boldsymbol{A}$ and $\boldsymbol{X}$ from the noisy observations $\boldsymbol{Y}$.
  - Standard unmixing algs (e.g., VCA [Nascimento'05], FSNMF [Gillis'12]) assume the existence of pure-pixels, which may not occur in practice.
  - Furthermore, they do *not* exploit spectral coherence, spatial coherence, and sparsity, which do occur in practice.
  - Recent Bayesian approaches to unmixing (e.g., SCU [Mittelman'12]) exploit spatial coherence using Dirichlet processes, albeit at very high complexity.

EM-BiG-AMP for HSI Unmixing

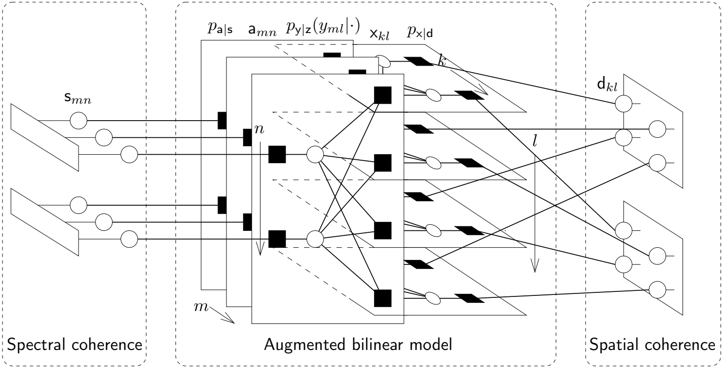
- To enforce non-negativity we place non-negative Gaussian Mixture (NNGM) prior on a_{mn}, and to encourage sparsity a Bernoulli-NNGM prior on x_{nl}.
  - We then use EM to learn the (B)NNGM parameters.
- To exploit spectral coherence we employ a hidden Gauss-Markov chain across each column in $\boldsymbol{A}$, and to exploit spatial coherence we employ an Ising model to capture the support across each row in $\boldsymbol{X}$.
  - We use EM to learn the Gauss-Markov and Ising parameters.
- To enforce the sum-to-one constraint on each column of $\boldsymbol{X}$, we augment both $\boldsymbol{Y}$ and $\boldsymbol{A}$ with a row of random variables with mean one and variance zero.







EM-BiG-AMP for HSI Unmixing



- Inference on the bilinear sub-graph is tackled using the BiG-AMP algorithm.
- Inference on the Gauss-Markov and Ising subgraphs are tackled using standard soft-input/soft-output belief propagation methods.
- Messages are exchanged between the three sub-graphs according to the sum-product algorithm, akin to "turbo" decoding in modern communication receivers [Schniter'10].

Numerical Results: Pure-Pixel Synthetic Data

- Pure pixel abundance maps $\boldsymbol{X}$ of size $L = 50 \times 50$ were generated with $N = 5$ materials residing in equal-sized spatial strips.
- Endmember spectra $\boldsymbol{A}$ were taken from a reflectance library.
- AWGN-corrupted observations had SNR $= 30$ dB.

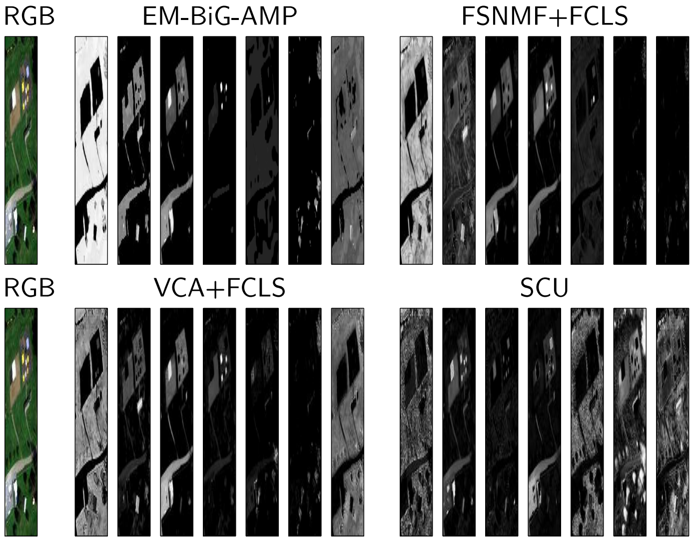
RGB view of data in 2D

- Averaging performance over 10 realizations . . .

	$\hat{\boldsymbol{A}}$ runtime	$\hat{\boldsymbol{X}}$ runtime	Total runtime	NMSE$_{\hat{\boldsymbol{A}}}$	NMSE$_{\hat{\boldsymbol{X}}}$
EM-BiG-AMP	–	–	5.57 sec	-57.4 dB	-108.6 dB
VCA + UCLS	0.05 sec	0.0007 sec	0.05 sec	-39.6 dB	-12.0 dB
VCA + FCLS	0.05 sec	4.08	4.13 sec	-39.6 dB	-30.5 dB
FSNMF + UCLS	0.002 sec	0.0008 sec	0.002 sec	-23.4 dB	-6.8 dB
FSNMF + FCLS	0.002 sec	3.97 sec	3.97 sec	-25.3 dB	-12.5 dB
SCU	–	–	2808 sec	-30.6 dB	-20.5 dB

- EM-BiG-AMP's runtime is comparable to VCA+FCLS and FSNMF+FCLS, and 2-3 orders of magnitude faster than SCU.

Results: SHARE 2012 dataset



- Data consisted of $M = 360$ spectral bands, ranging from 400-2450nm, taken over scene of $L = 150 \times 100$ pixels.
- EM-BiG-AMP gives estimated abundance maps with higher constrast, suggesting better unmixing.
- The lack of ground-truth prevents a quantitative comparison.

Conclusion

- BiG-AMP = approximate message passing for the generalized bilinear model.
- A novel approach to matrix completion, robust PCA, dictionary learning, non-negative matrix factorization, etc.
- Includes mechanisms for adaptive dampling, parameter tuning, model-order selection, non-separable priors.
- Competitive with the best current algorithms for each application.
  - Best phase transitions for MC, RPCA, overcomplete DL.
  - Runtimes not far from the fastest algorithms.
- Currently working on generalizations of BiG-AMP to parameteric models (e.g., Toeplitz matrices), as well as various applications of BiG-AMP.

References

1. J. T. Parker, P. Schniter and V. Cevher, "Bilinear Generalized Approximate Message Passing," *arXiv:i1310:2632*, 2013.
2. J. Vila, P. Schniter, and J. Meola, "Hyperspectral Image Unmixing via Bilinear Generalized Approximate Message Passing," Proc. SPIE, 2013.
3. D.L. Donoho, A. Maleki, and A. Montanari, "Message passing algorithms for compressed sensing: I. Motivation and construction," *ITW*, 2010.
4. S. Rangan, "Generalized approximate message passing for estimation with random linear mixing," *ISIT*, 2011. (See also *arXiv:1010.5141*).
5. P. Schniter and V. Cevher, "Approximate message passing for bilinear models," *SPARS*, 2011.
6. A. Javanmard and A. Montanari, "State evolution for general approximate message passing algorithms, with applications to spatial coupling," *arXiv:1211.5164*, 2012.
7. J. Ziniel and P. Schniter, "Binary classification and feature selection via generalized approximate message passing," 2013.
8. P. Schniter and S. Rangan, "Compressive phase retrieval via generalized approximate message passing," *Allerton*, 2012.
9. S. Rangan, P. Schniter, E. Riegler, A. Fletcher, V. Cevher "Fixed Points of Generalized Approximate Message Passing with Arbitrary Matrices," *ISIT*, 2013 (see also *arXiv:1301.6295*).
10. J. P. Vila and P. Schniter, "Expectation-Maximization Gaussian-Mixture Approximate Message Passing," *IEEE Trans. Signal Process.*, 2013.
11. Z. Wen, W. Yin, and Y. Zhang, "Solving a low-rank factorization model for matrix completion by a nonlinear successive over-relaxation algorithm," *Math. Program. Comput.*, 2012.

References (cont.)

[12] P. Schniter, "Turbo reconstruction of structured sparse signals," *Proc. Conf. Inform. Science & Syst.*, 2010.

[13] Z. Lin, M. Chen, L. Wu, and Y. Ma, "The augmented Lagrange multiplier method for exact recovery of corrupted low-rank matrices," *arXiv:1009.5055*, 2010.

[14] L. Balzano, R. Nowak, and B. Recht, "Online identification and tracking of subspaces from highly incomplete information," *arXiv:1006.4046*, 2010.

[15] S. D. Babacan, M. Luessi, R. Molina, and A. K. Katsaggelos, "Sparse Bayesian methods for low-rank matrix estimation," *IEEE Trans. Signal Process.*, 2012.

[16] J. He, L. Balzano, and J. Lui, "Online robust subspace tracking from partial information," *arXiv:1109.3827*, 2011.

[17] M. Aharon, M. Elad, and A. Bruckstein, "K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Trans. Sig. Process.*, 2006.

[18] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, "Online learning for matrix factorization and sparse coding," *J. Mach. Learn. Res.*, 2010.

[19] D. A. Spielman, H. Wang, and J. Wright, "Exact recovery of sparsely-used dictionaries," *J. Mach. Learn. Res.*, 2012.

[20] J. Nascimento and J. Bioucas-Dias, "Vertex component analysis: A fast algorithm to unmix hyperspectral data," *IEEE Trans. GeoSci. Remote Sens.*, 2005.

[21] N. Gillis and S.A. Vavasis, "Fast and robust recursive algorithms for separable nonnegative matrix factorization," *arXiv:1208.1237*, 2012.

[22] R. Mittelman, N. Dobigeon, and A. Hero, "Hyperspectral image unmixing using a multiresolution sticky HDP," *IEEE Trans. Signal Process.*, 2012.