Expectation-Maximization Gaussian-Mixture Approximate Message Passing

Philip Schniter and Jeremy Vila

THE OHIO STATE UNIVERSITY

Supported in part by NSF-I/UCRC grant IIP-0968910, by NSF grant CCF-1018368, and by DARPA/ONR grant N66001-10-1-4090.

CISS @ Princeton – 3/23/12

Compressive Sensing

- Goal: recover signal $\boldsymbol{x}$ from noisy sub-Nyquist measurements

$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{w} \quad \boldsymbol{x} \in \mathbb{R}^N \quad \boldsymbol{y}, \boldsymbol{w} \in \mathbb{R}^M \quad M < N.$$

  where $\boldsymbol{x}$ is K-sparse with $K < M$, or compressible.

- With sufficient sparsity and appropriate conditions on the mixing matrix $\boldsymbol{A}$ (e.g. RIP, nullspace), accurate recovery of $\boldsymbol{x}$ is possible using polynomial-complexity algorithms.

- A common approach (LASSO) is to solve the convex problem

$$\min_{\boldsymbol{x}} \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|_2^2 + \alpha \|\boldsymbol{x}\|_1$$

  where α can be tuned in accordance with sparsity and SNR.

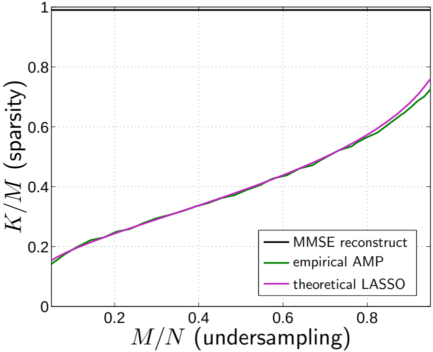
Phase Transition Curves (PTC)

- The PTC identifies ratios $(\frac{M}{N}, \frac{K}{M})$ for which perfect noiseless recovery of K-sparse $\boldsymbol{x}$ occurs (as $M, N, K \to \infty$ under i.i.d Gaussian $\boldsymbol{A}$).
- Suppose $\{x_n\}$ are drawn i.i.d.

$$\boxed{p_X(x_n) = \lambda f(x_n) + (1-\lambda)\delta(x_n)}$$

with known $\lambda \triangleq K/N$.

- LASSO's PTC is invariant to $f(\cdot)$. Thus, LASSO is robust in the face of unknown $f(\cdot)$.
- MMSE-reconstruction's PTC is far better than Lasso's, but requires knowing $f(\cdot)$.



Wu and Verdú, "Optimal phase transitions in compressed sensing," arXiv Nov. 2011.

Motivations

For practical compressive sensing...

- want minimal MSE
  - distributions are unknown ⇒ can't formulate MMSE estimator
  - but there is hope:
    various algs seen to outperform Lasso for specific signal classes
  - really, we want a universal algorithm: good for all signal classes
- want fast runtime
  - especially for large signal-length N (i.e., scalable).
- want to avoid algorithmic tuning parameters,
  - who has the patience to tweak yet another CS algorithm!

Proposed Approach: "EM-GM-GAMP"

- Model the signal and noise using flexible distributions:
  - i.i.d Bernoulli Gaussian-mixture (GM) signal

$$p(x_n) = \lambda \sum_{l=1}^{L} \omega_l \, \mathcal{N}(x_n; \theta_l, \phi_l) + (1-\lambda)\delta(x_n) \quad \forall n$$

  - i.i.d Gaussian noise with variance ψ

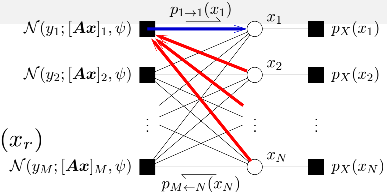
- Learn the prior parameters $\boldsymbol{q} \triangleq \{\lambda, \omega_l, \theta_l, \phi_l, \psi\}_{l=1}^{L}$
  - treat as deterministic and use expectation-maximization (EM)

- Exploit the learned priors in near-MMSE signal reconstruction
  - use generalized approximate message passing (GAMP)

Approximate Message Passing (AMP)

- AMP methods infer $\boldsymbol{x}$ from $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{w}$ using loopy belief propagation with carefully constructed approximations.
  - The original AMP [Donoho, Maleki, Montanari '09] solves the LASSO problem (i.e., Laplacian MAP) assuming i.i.d matrix $\boldsymbol{A}$.
  - The Bayesian AMP [Donoho, Maleki, Montanari '10] framework tackles MMSE inference under generic signal priors.
  - The generalized AMP [Rangan '10] framework tackles MAP or MMSE inference under generic signal & noise priors and generic $\boldsymbol{A}$.
- AMP is a form of iterative thresholding, requiring only two applications of $\boldsymbol{A}$ per iteration and ≈ 25 iterations. Very fast!
- Rigorous large-system analyses (under i.i.d Gaussian $\boldsymbol{A}$) have established that (G)AMP follows a state-evolution trajectory with optimal properties [Bayati, Montanari '10], [Rangan '10].

AMP Heuristics (Sum-Product)



1. Message from y_i node to x_j node:

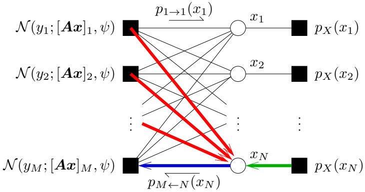
$$p_{i\to j}(x_j) \propto \int_{\{x_r\}_{r\neq j}} \mathcal{N}\big(y_i; \overbrace{\textstyle\sum_r a_{ir}x_r}^{\approx\,\mathcal{N}\text{ via CLT}}, \psi\big) \textstyle\prod_{r\neq j} p_{i\leftarrow r}(x_r)$$

$$\approx \int_{z_i} \mathcal{N}(y_i; z_i, \psi)\, \mathcal{N}\big(z_i; \hat{z}_i(x_j), \nu_i^z(x_j)\big) \sim \mathcal{N}$$

To compute $\hat{z}_i(x_j), \nu_i^z(x_j)$, the means and variances of $\{p_{i\leftarrow r}\}_{r\neq j}$ suffice, thus Gaussian message passing!

Remaining problem: we have $2MN$ messages to compute (too many!).

2. Exploiting similarity among the messages $\{p_{i\leftarrow j}\}_{i=1}^M$, AMP employs a Taylor-series approximation of their difference whose error vanishes as $M\to\infty$ for dense $\boldsymbol{A}$ (and similar for $\{p_{i\leftarrow j}\}_{i=1}^N$ as $N\to\infty$). Finally, need to compute only $\mathcal{O}(M+N)$ messages!



Expectation-Maximization

- We use expectation-maximization (EM) to learn the signal and noise prior parameters $\boldsymbol{q} \triangleq \{\lambda, \boldsymbol{\omega}, \boldsymbol{\theta}, \boldsymbol{\phi}, \psi\}$
  - The missing data is chosen to be the signal and noise vectors $(\boldsymbol{x}, \boldsymbol{w})$.
  - The updates are performed coordinate-wise.
  - For example, updating λ at the i^{th} EM iteration involves

$$\text{(E-step)} \quad Q(\lambda|\boldsymbol{q}^i) = \sum_{n=1}^{N} \mathrm{E}\left\{ \ln p(x_n; \lambda, \boldsymbol{\omega}^i, \boldsymbol{\theta}^i, \boldsymbol{\phi}^i) \middle| \boldsymbol{y}; \boldsymbol{q}^i \right\}$$

$$\text{(M-step)} \quad \lambda^{i+1} = \arg\max_{\lambda \in (0,1)} Q(\lambda|\boldsymbol{q}^i).$$

  The updates of $(\boldsymbol{\omega}, \boldsymbol{\theta}, \boldsymbol{\phi}, \psi)$ are similar (details in paper).

- All quantities needed for the EM updates are provided by GAMP!

Parameter Initialization

Initialization matters; EM can get stuck in a local max. We suggest...

- initializing the sparsity λ according to the theoretical LASSO PTC.
- initializing the noise and active-signal variances using known energies $\|\boldsymbol{y}\|_2^2, \|\boldsymbol{A}\|_F^2$ and user-supplied SNR^0 (which defaults to 20 dB):

$$\psi^0 = \frac{\|\boldsymbol{y}\|_2^2}{(\text{SNR}^0 + 1)M}, \quad (\sigma^2)^0 = \frac{\|\boldsymbol{y}\|_2^2 - M\psi^0}{\lambda^0\|\boldsymbol{A}\|_F^2}$$

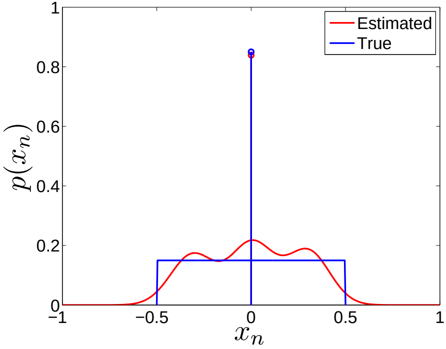
- fixing L (e.g., $L = 3$) and initializing the GM parameters $(\boldsymbol{\omega}, \boldsymbol{\theta}, \boldsymbol{\phi})$ as the best fit to a uniform distribution with variance σ^2.

We have also developed

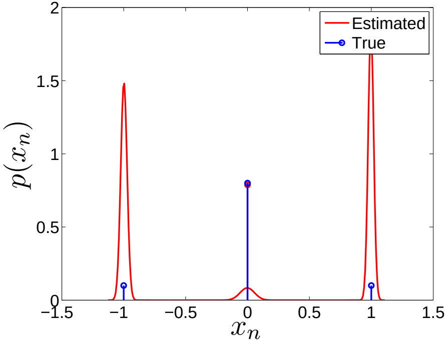
- a "splitting" mode that adds one GM component at a time.
- a "heavy tailed" mode that forces zero-mean GM components.

Examples of Learned Signal-pdfs

The following shows the Gaussian-mixture pdf learned by EM-GM-GAMP when the true active-signal pdf was uniform (left) and ± 1 (right):



True and learned signal pdfs

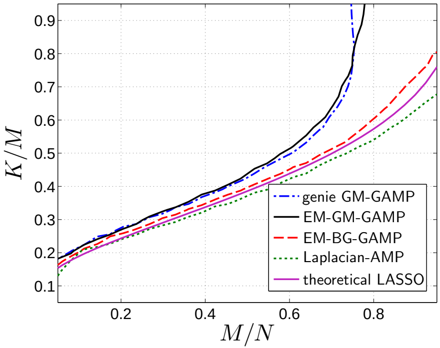


True and learned signal pdfs

Empirical PTCs: Bernoulli-Rademacher (± 1) signals

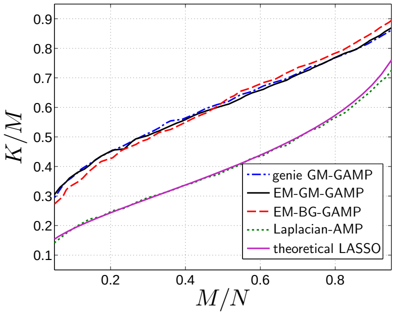
- We now evaluate noiseless reconstruction performance via phase-transition curves constructed using $N = 1000$-length signals, i.i.d Gaussian $\boldsymbol{A}$, and 100 realizations.

- We see EM-GM-GAMP performing significantly better than LASSO for this signal class.
- We also see EM-GM-GAMP performing nearly as well as GM-GAMP under genie-aided parameter settings.

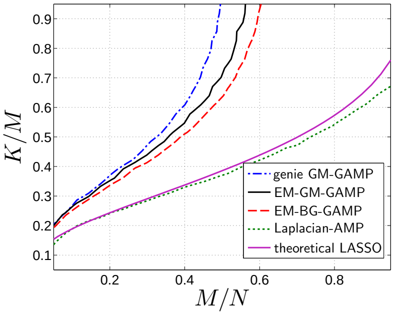


Empirical noiseless Bernoulli-Rademacher PTCs

PTCs for Bernoulli-Gaussian and Bernoulli signals



Empirical noiseless Bernoulli-Gaussian PTCs



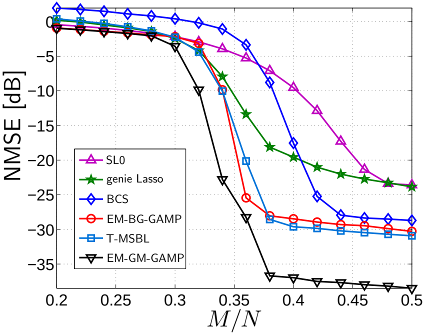
Empirical noiseless Bernoulli PTCs

For these signals, we see EM-GM-GAMP performing. . .

- significantly better than LASSO,
- nearly as well as genie-aided GM-GAMP,
- on par with our previous "EM-BG-GAMP" algorithm.

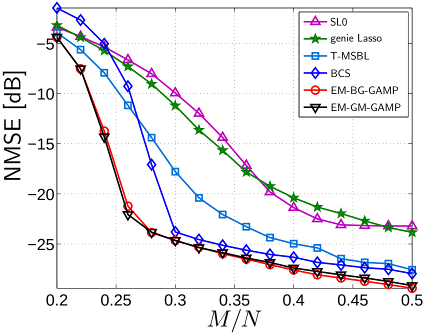
Noisy Recovery: Bernoulli-Rademacher (± 1) signals

- We now compare the normalized MSE of EM-GM-GAMP to several state-of-the-art algorithms (SL0, T-MSBL, BCS, Lasso via SPGL1) for the task of noisy signal recovery under i.i.d Gaussian $\boldsymbol{A}$.
- For this, we fixed $N = 1000$, $K = 100$, SNR $= 25$dB and varied M.
- For these Bernoulli-Rademacher signals, we see EM-GM-GAMP outperforming the other algorithms for all undersampling ratios M/N.
- Notice that our previous EM-BG-GAMP algorithm cannot accurately model the Bernoulli-Rademacher prior.

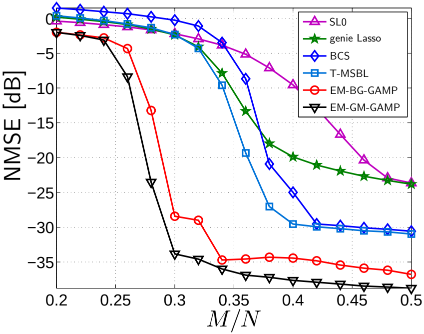


Noisy Bernoulli-Rademacher recovery NMSE.

Noisy Recovery: Bernoulli-Gaussian and Bernoulli signals



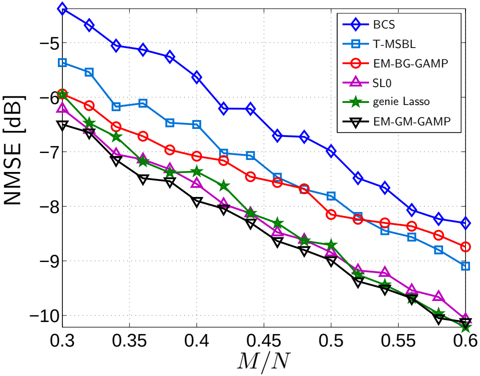
Noisy Bernoulli-Gaussian recovery NMSE.



Noisy Bernoulli recovery NMSE.

- For Bernoulli-Gaussian and Bernoulli signals, EM-GM-GAMP again dominates the other algorithms.
- We attribute the excellent performance of EM-GM-GAMP to its ability to learn and exploit the true signal prior.

Noisy Recovery of Heavy-tailed (Student's-t) signals

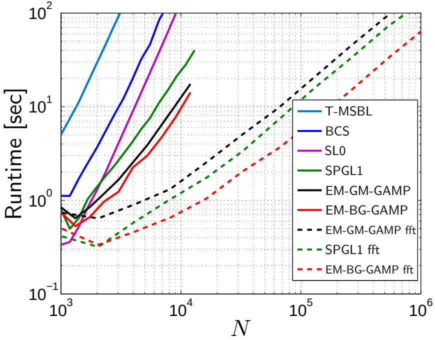


Noisy Student-t recovery NMSE.

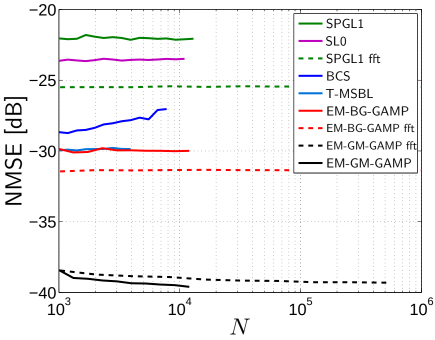
- Algorithm rankings on heavy-tailed signals are often the reverse of those for sparse signals!
- In its "heavy tailed" mode, EM-GM-GAMP performs on par with the best algorithms for all M/N.

Runtime versus signal-length N

- We fix $M/N = 0.5$, $K/N = 0.1$, SNR $= 25$dB, and average 50 trials.



Noisy Bernoulli-Rademacher recovery time.

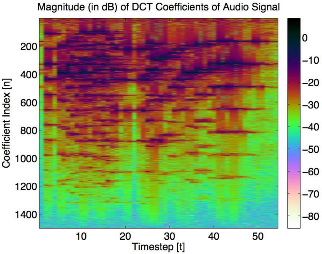


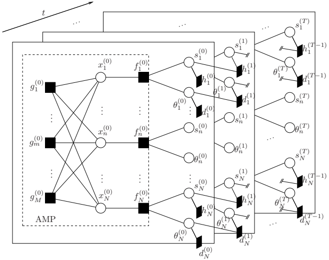
Noisy Bernoulli-Rademacher recovery NMSE.

- For all $N > 1000$, EM-GM-GAMP has the fastest runtime!
- EM-GM-GAMP can also leverage fast operators for $\boldsymbol{A}$ (e.g., FFT).

Extension to structured sparsity (Justin Ziniel)

- Recovery of an audio signal sparsified via DCT $\boldsymbol{\Psi}$ and compressively sampled via i.i.d Gaussian $\boldsymbol{\Phi}$ (so that $\boldsymbol{A} = \boldsymbol{\Phi\Psi}$).
- Exploit persistence of support across time via discrete Markov chains and turbo AMP.





algorithm	$M/N = 1/5$		$M/N = 1/3$		$M/N = 1/2$	
EM-GM-AMP	-9.04 dB	8.77 s	-12.72 dB	10.26 s	-17.17 dB	11.92 s
turbo EM-GM-AMP	-12.34 dB	9.37 s	-16.07 dB	11.05 s	-20.94 dB	12.96 s

Conclusions

- We proposed a sparse reconstruction alg that uses EM to learn GM-signal and AWGN-noise priors, and that uses GAMP to exploit these priors for near-MMSE signal recovery.
- Advantages of EM-GM-GAMP:
  - State-of-the-art NMSE performance for all tested signal types.
  - State-of-the-art complexity for signals of length $N \gtrsim 1000$.
  - Minimal tuning: choose between "sparse" or "heavy-tailed" modes.
- Ongoing related work:
  - Theoretical performance guarantees of EM-GM-GAMP.
  - Extension to non-Gaussian noise.
  - Universal learning/exploitation of structured sparsity.
  - Extensions to matrix completion, dictionary learning, robust PCA.

Matlab code is available at
`http://ece.osu.edu/~vilaj/EMGMAMP/EMGMAMP.html`

Thanks!