Expectation-Maximization Gaussian-Mixture Approximate Message Passing

Philip Schniter and Jeremy Vila

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 $x$ from noisy sub-Nyquist measurements

$$y = Ax + w \quad x \in \mathbb{R}^N \quad y, w \in \mathbb{R}^M \quad M < N.$$ where $x$ is $K$-sparse with $K < M$, or compressible.

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

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

$$\min_x \| y - Ax \|_2^2 + \alpha \| x \|_1$$

where $\alpha$ can be tuned in accordance with sparsity and SNR.
Phase Transition Curves (PTC)

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

- Suppose \( \{x_n\} \) are drawn i.i.d.
  \[
p_X(x_n) = \lambda f(x_n) + (1 - \lambda) \delta(x_n)
\]
  with known \( \lambda \triangleq \frac{K}{N} \).

- 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 \( f(\cdot) \).

Motivations

For practical compressive sensing...

- want **minimal MSE**
  - distributions are unknown \( \Rightarrow \) 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 \(\psi\)

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

- **Exploit** the learned priors in near-MMSE signal reconstruction
  - use generalized approximate message passing (GAMP)
Approximate Message Passing (AMP)

- AMP methods infer $x$ from $y = Ax + w$ 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 $A$.
  - 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 $A$.
- AMP is a form of iterative thresholding, requiring only two applications of $A$ per iteration and $\approx 25$ iterations. Very fast!
- Rigorous large-system analyses (under i.i.d Gaussian $A$) have established that (G)AMP follows a state-evolution trajectory with optimal properties [Bayati, Montanari '10], [Rangan '10].
Message from $y_i$ node to $x_j$ node:

$$p_{i \rightarrow j}(x_j) \propto \int_N \mathcal{N}(y_i; \sum_r a_{ir} x_r, \psi) \prod_{r \neq j} p_{i \leftarrow r}(x_r) \mathcal{N}(y_i; z_i, \psi) \mathcal{N}(z_i; \hat{z}_i(x_j), \nu_i^\tilde{z}(x_j)) \sim \mathcal{N}$$

To compute $\hat{z}_i(x_j), \nu_i^\tilde{z}(x_j)$, 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!).

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 $M \rightarrow \infty$ for dense $A$ (and similar for $\{p_{i \leftarrow j}\}_{i=1}^N$ as $N \rightarrow \infty$).

Finally, need to compute only $O(M+N)$ messages!
Expectation-Maximization

- We use expectation-maximization (EM) to learn the signal and noise prior parameters $q \triangleq \{ \lambda, \omega, \theta, \phi, \psi \}$
  - The missing data is chosen to be the signal and noise vectors $(x, w)$.
  - The updates are performed coordinate-wise.
  - For example, updating $\lambda$ at the $i^{th}$ EM iteration involves
    
    \[ (\text{E-step}) \quad Q(\lambda|q^i) = \sum_{n=1}^{N} \mathbb{E} \left\{ \ln p(x_n; \lambda, \omega^i, \theta^i, \phi^i) | y; q^i \right\} \]
    
    \[ (\text{M-step}) \quad \lambda^{i+1} = \mathop{\arg\max}_{\lambda \in (0,1)} Q(\lambda|q^i). \]
  
  The updates of $(\omega, \theta, \phi, \psi)$ 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 $\lambda$ according to the theoretical LASSO PTC.
- initializing the noise and active-signal variances using known energies $\|y\|_2^2, \|A\|_F^2$ and user-supplied SNR$^0$ (which defaults to 20 dB):
  \[
  \psi^0 = \frac{\|y\|_2^2}{(\text{SNR}^0 + 1)M}, \quad (\sigma^2)^0 = \frac{\|y\|_2^2 - M\psi^0}{\lambda^0\|A\|_F^2}
  \]
- fixing $L$ (e.g., $L = 3$) and initializing the GM parameters $(\omega, \theta, \phi)$ as the best fit to a uniform distribution with variance $\sigma^2$.

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):

![Graph showing true and learned signal pdfs](image_url)
Empirical PTCs: Bernoulli-Rademacher ($\pm 1$) signals

- We now evaluate noiseless reconstruction performance via phase-transition curves constructed using $N = 1000$-length signals, i.i.d Gaussian $\mathcal{A}$, 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.
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 ($\pm 1$) 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 $\mathbf{A}$.
- For this, we fixed $N = 1000$, $K = 100$, $\text{SNR} = 25\text{dB}$ 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 Recovery: Bernoulli-Gaussian and Bernoulli signals

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.
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 $M/N$. 

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

Noisy Student-t recovery NMSE.
We fix $M/N = 0.5$, $K/N = 0.1$, $\text{SNR} = 25\text{dB}$, and average 50 trials.

For all $N > 1000$, EM-GM-GAMP has the fastest runtime!

EM-GM-GAMP can also leverage fast operators for $\mathbf{A}$ (e.g., FFT).
Extension to structured sparsity (Justin Ziniel)

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

<table>
<thead>
<tr>
<th>algorithm</th>
<th>$M/N = 1/5$</th>
<th>$M/N = 1/3$</th>
<th>$M/N = 1/2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>EM-GM-AMP</td>
<td>-9.04 dB 8.77 s</td>
<td>-12.72 dB 10.26 s</td>
<td>-17.17 dB 11.92 s</td>
</tr>
<tr>
<td>turbo EM-GM-AMP</td>
<td>-12.34 dB 9.37 s</td>
<td>-16.07 dB 11.05 s</td>
<td>-20.94 dB 12.96 s</td>
</tr>
</tbody>
</table>
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!