Compressive Phase Retrieval via Generalized Approximate Message Passing

Philip Schniter

Joint work with Sundeep Rangan

Supported in part by NSF grant CCF-1018368 and DARPA/ONR grant N66001-10-1-4090.

FFT Workshop – 2/23/2013
Phase Retrieval

- **Goal**: Recover signal $x_0 \in \mathbb{C}^n$ from $m$ magnitude-only measurements
  \[ y = |Ax_0 + w|, \]
  where $A \in \mathbb{C}^{m \times n}$ is a known linear transform and $w \in \mathbb{C}^m$ is noise.

- **Motivation**: In many applications, it feasible to measure the intensity, but not the phase, of the Fourier transform of the signal-of-interest:
  - X-ray crystallography,
  - transmission electron microscopy,
  - coherent diffractive imaging,
  - astronomical imaging, etc.

- **Feasibility**: To make the solution to $y = |Ax|$ unique (up to a global phase) w.p.1, $m = 4n - o(n)$ i.i.d Gaussian measurements are necessary [Heinosaari/Mazzarella/Wolf’11] and $m = 4n - 2$ are sufficient [Balan/Casazza/Edidin’06].
Phase Retrieval: Classical Approaches

Most classical approaches are iterative in nature. For example,

- Alternate between...
  - projecting $A\hat{x}$ onto the magnitude constraint $y$, yielding $\hat{z}$,
  - projecting $A^+\hat{z}$ onto an apriori known support set, yielding $\hat{x}$.

However, due to the non-convexity of the first projection, it is easy for such algorithms to get trapped in local minima.
Phase Retrieval: Convex Approaches

Recently, some convex relaxations have been proposed.

– Noting that $y_i^2 = |a_i^H x|^2 = \text{tr}(a_i a_i^H X)$ for $X = xx^H$, pose as “min $X \succeq 0 \text{ rank}(X)$ s.t. $\text{tr}(a_i a_i^H X) = y_i^2$ for $i = 1...m$.” (NP hard!)

Relax to “min $\text{tr}(X)$ s.t. $\text{tr}(a_i a_i^H X) = y_i^2$ for $i = 1...m$,” (convex!) known as PhaseLift [Candes/Strohmer/Voroninski’11].

– Another semidefinite program (with similar performance) known as PhaseCut was proposed in [Waldspurger/D’Aspremont/Mallat’12].

It was recently shown [Candes/Li’12] that

- with very high probability, PhaseLift perfectly recovers an arbitrary $x$ from $m \geq c_0 n$ noiseless measurements, where $c_0$ is a constant,
- and also that PhaseLift can be made robust to noise.
Recall that $m \geq 4n - o(n)$ magnitude measurements are needed for $y = |Ax|$ to have a unique (up to a phase) solution for $x \in \mathbb{C}^n$.

Sometimes we can only afford $m \ll 4n$ magnitude measurements, in which case the problem becomes one of compressive phase retrieval.

For successful compressive phase retrieval (CPR), one needs to leverage additional structure in $x$, such as sparsity.
Compressive Phase Retrieval: Prior Work

- Assuming knowledge of $\|x_0\|_1$, [Moravec/Romberg/Baraniuk’07]
  - appended this constraint onto the classical RAAR algorithm, and
  - used RIP-based arguments to establish that $m \gtrsim k^2 \log(n/k^2)$ magnitude measurements suffice for recovery.

However, the algorithm was prone to local minima and slow convergence. Also, knowledge of $\|x_0\|_1$ is rarely available in practice.

- Taking a convex approach, [Ohlsson/Yang/Dong/Sastry’12] proposed the following generalization of PhaseLift, which they call CPRL:

$$\min_{X \succeq 0} \text{tr}(X) + \lambda \|X\|_1 + \mu \sum_{i=1}^{m} \left| \text{tr}(a_i a_i^H X) - y_i^2 \right|^2,$$

and performed both RIP and mutual coherence analyses. Seems promising...
Bring out the GAMP

Zed: Bring out the Gimp.

Maynard: Gimp’s sleeping.

Zed: Well, I guess you’re gonna have to go wake him up now, won’t you?


We propose a new approach to CPR based on generalized approximate message passing (GAMP).

Numerical results show

- excellent phase transitions,
- excellent NMSE & robustness to noise,
- excellent runtime,

enabling, e.g., practical compressive image retrieval.
For these numerical results we generated random...  

- signals $\mathbf{x}_0$ as $k$-sparse, $n=512$-length, Bernoulli-circular-Gaussian,  
- matrices $\mathbf{A} = \Phi \mathbf{F}$, where $\Phi \in \mathbb{C}^{m \times n}$ is i.i.d circular Gaussian and $\mathbf{F}$ is the $n \times n$ DFT matrix,  
- noise $\mathbf{w}$ as circular white Gaussian (added prior to taking magnitude),

and we monitored the phase-corrected normalized reconstruction MSE

$$NMSE \triangleq \min_{\theta} \frac{||\hat{\mathbf{x}} - e^{i\theta} \mathbf{x}_0||_2^2}{||\mathbf{x}_0||_2^2}.$$
Phase transition

PR-GAMP’s empirical success rate, averaged over 500 realizations, was

\[ \text{prGAMP success@} -40\text{dB, rdft, N=512, snr=100dB, avg=500} \]

where success \( \triangleq \{ \text{NMSE} < 10^{-4} \} \).

Philip Schniter (OSU)
Comparison to phase-oracle GAMP

Comparing the 50%-success contours of PR- and phase-oracle GAMP:

![Graph showing comparison between PR-GAMP and phase-oracle GAMP.]

we see that PR-GAMP requires about $4 \times$ the number of measurements as phase-oracle GAMP. (Very interesting!)
PR-GAMP’s median NMSE, measured over the same 500 realizations, was showing that recovery is very accurate above the phase transition.
Noise Robustness of PR-GAMP

The median NMSE, measured over 2000 realizations:

![Graph showing NMSE vs SNR](image)

shows that PR-GAMP loses about 3 dB at medium-to-high SNR.
**Comparison to CPRL** [Ohlsson/Yang/Dong/Sastry’12]

Empirical success rate (and median runtime) over 100 realizations:

<table>
<thead>
<tr>
<th></th>
<th>((m, n) = (20, 32))</th>
<th>((m, n) = (30, 48))</th>
<th>((m, n) = (40, 64))</th>
</tr>
</thead>
<tbody>
<tr>
<td>(k = 1:)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CPRL</td>
<td>0.96 (4.9 sec)</td>
<td>0.97 (51 sec)</td>
<td>0.99 (291 sec)</td>
</tr>
<tr>
<td>PR-GAMP</td>
<td>0.83 (0.4 sec)</td>
<td>0.94 (0.3 sec)</td>
<td>0.99 (0.3 sec)</td>
</tr>
<tr>
<td>(k = 2:)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CPRL</td>
<td>0.55 (5.8 sec)</td>
<td>0.55 (58 sec)</td>
<td>0.58 (316 sec)</td>
</tr>
<tr>
<td>PR-GAMP</td>
<td>0.72 (0.4 sec)</td>
<td>0.92 (0.3 sec)</td>
<td>1.0 (0.3 sec)</td>
</tr>
</tbody>
</table>

Note:

- CPRL runtime limited us to these **toy problems**.
- CPRL succeeds when sparsity \(k = 1\), but not when \(k \geq 2\). GAMP instead suffers when problem dimensions are very small.
- CPRL’s runtime grows very quickly with problem dimensions! GAMP’s runtime is invariant to the dimension of these toy problems.
Compressive Image Recovery

65536 image pixels, 32768 measurements, 30dB SNR:

original image

PR-GAMP (-29.7dB NMSE)

PR-GAMP runtime: only 11.1 sec.
Compressive Image Recovery: Details

- Measurements were collected using
  \[ A = \begin{bmatrix} B_1 & B_2 \end{bmatrix} \begin{bmatrix} F & F \end{bmatrix} \begin{bmatrix} M_1 \\ M_2 \end{bmatrix} \]
  
  with banded i.i.d-Gaussian \( B_i \) (10 nonzero entries per column), Fourier \( F \), and binary masks \( M_i \).

- Over 100 random measurement & noise realizations, we observed
  - 89% success rate, where “success” meant \( \text{NMSE} < -27 \text{ dB} \), and
  - median runtime of 13.4 sec.
So what’s the approach?

1. Formulate as a **Bayesian inference** problem by assuming
   - \( y_i = |\langle A x \rangle_i + w_i | \quad \forall i \)
   - \( w_i \sim \mathcal{CN}(0, \nu^w) \) i.i.d
   - \( p(x) = \prod_j p_X(x_j) \) for sparsity promoting \( p_X \)

2. Use **GAMP**, a state-of-the-art **loopy belief propagation** method, to approximate the marginal posterior pdfs \( \{p_{X_j|Y(\cdot|y)}\}_{j=1}^n \).
Generalized Approximate Message Passing (GAMP)

- The evolution of GAMP:
  - The original AMP [Donoho/Maleki/Montanari’09] solves the LASSO problem \( \min_x \| y - A x \|_2^2 + \lambda \| x \|_1 \) popular in compressive sensing, i.e., MAP estimation of i.i.d Laplacian signal, thru dense \( A \), in AWGN.
  - The Bayesian AMP [Donoho/Maleki/Montanari’10] extended the above to a generic i.i.d signal prior and MMSE estimation.
  - The generalized AMP [Rangan’10] extended the above to generic i.i.d likelihoods \( p_{Y|Z}(y_i | a_i^H x) \), for both MAP and MMSE inference.

- In the end, GAMP produces a sophisticated iterative thresholding alg, whose complexity is dominated by one application of \( A \) and \( A^H \) per iteration with relatively few iterations (e.g., tens). Very fast!

- Rigorous large-system analyses (under i.i.d sub-Gaussian \( A \)) have established that GAMP follows a state-evolution trajectory whose fixed-points have nice properties [Rangan’10], [Javanmard/Montanari’12].
GAMP Heuristics (Sum-Product)

1. Message from $y_i$ node to $x_j$ node:
   \[
   p_{i \rightarrow j}(x_j) \propto \int \left[ p_{Y|Z}(y_i; \sum_r a_{ir} x_r) \prod_{r \neq j} p_{i \leftarrow r}(x_r) \right] \text{d}x_r
   \]
   \[
   \approx \int p_{Y|Z}(y_i; z_i) \mathcal{N}(z_i; \hat{z}_i(x_j), \nu_i^z(x_j)) \approx \mathcal{CN}
   \]

   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 $\{p_{i \leftarrow j}\}_{i=1}^m$, GAMP employs a Taylor-series approximation of their difference, whose error vanishes as $m \rightarrow \infty$ for dense $A$ (and similar for $\{p_{i \rightarrow j}\}_{j=1}^n$ as $n \rightarrow \infty$).

   Finally, need to compute only $O(m+n)$ messages!
Require: Matrix $A$, sum-prod $\in \{\text{true, false}\}$, initializations $\hat{x}^0, \nu_x^0$

$t = 0$, $\hat{s}^{-1} = 0$, $\forall i,j : S_{ij} = \vert A_{ij} \vert^2$

repeat

$\nu_p^t = S \nu_x^t$, $\hat{p}^t = A \hat{x}^t - \hat{s}^{t-1}.\nu_p^t$ (gradient step)

if sum-prod then

$\forall i : \nu_z^t_i = \text{var}(Z_i \mid y_i)$, $\hat{z}^t_i = E(Z_i \mid y_i)$ for $p_{Z_i \mid Y_i}(z \mid y) \propto p_{Y \mid Z}(y \mid z)CN(z; \hat{p}_i^t, \nu_{p_i}^t)$

else

$\forall i : \nu_z^t_i = \nu_{p_i}^t \text{prox}_{-\nu_{p_i}^t} \log p_{Y \mid Z}(y_i,.) (\hat{p}_i^t)$, $\hat{z}^t_i = \text{prox}_{-\nu_{p_i}^t} \log p_{Y \mid Z}(y_i,.) (\hat{p}_i^t)$,

end if

$\nu_s^t = (1 - \nu_z^t./\nu_p^t)./\nu_p^t$, $\hat{s}^t = (\hat{z}^t - \hat{p}^t)./\nu_p^t$ (dual update)

$\nu_r^t = 1./\left(S^T \nu_s^t\right)$, $\hat{r}^t = \hat{x}^t + \nu_r^t.A^T \hat{s}^t$ (gradient step)

if sum-prod then

$\forall j : \nu_{x_j}^t = \text{var}(X_j \mid \hat{r}_j^t)$, $\hat{z}_j^t = E(X_j \mid \hat{r}_j^t)$ for $p_{X_j \mid R_j}(x \mid r) \propto p_{X}(x)CN(x; r, \nu_{r_j}^t)$

else

$\forall j : \nu_{x_j}^{t+1} = \nu_{r_j}^t \text{prox}_{-\nu_{r_j}^t} \log p_X(.) (\hat{r}_j^t)$, $\hat{x}_{j}^{t+1} = \text{prox}_{-\nu_{r_j}^t} \log p_X(.) (\hat{r}_j^t)$,

end if

$t \leftarrow t + 1$

until Terminated

Note connections to Arrow-Hurwicz, primal-dual, ADMM, proximal FB splitting,...
To apply GAMP to phase retrieval, we need a likelihood function \( p_{Y|Z}(\cdot|\cdot) \) relating the noisy magnitude measurements \( \{y_i\}_{i=1}^{m} \) to the corresponding noiseless transform outputs \( \{z_i\}_{i=1}^{m} \) (recalling that \( z_i \triangleq [Ax]_i \)).

- When \( Z \) and \( W \) are both circular, one can show that
  \[
  Y = |Z + W| \iff Y = e^{j\Theta}(Z + W)\bigg|_{\Theta \sim U[0,2\pi)}
  \]
  in the sense that both models yield the same \( p_{Z|Y}(\cdot|\cdot) \).

- Assuming \( W \sim \mathcal{CN}(0, \nu^w) \), we then margin out \( \Theta \) to obtain
  \[
  p_{Y|Z}(y|z) = \frac{1}{\pi \nu^w} e^{-\frac{(|y| - |z|)^2}{\nu^w}} I_0(\rho) e^{-\rho} \quad \text{for} \quad \rho \triangleq \frac{2|y| |z|}{\nu^w},
  \]
  where \( I_0(\cdot) \) is the 0th-order modified Bessel function of the first kind.

Other models are also possible, e.g., \( Y = |Z| + W \) or \( Y = |Z|^2 + W \).
GAMP for Phase Retrieval: Signal Prior

For compressive phase retrieval, we need a structured signal prior $p_X(\cdot)$.

- **Separable priors** constrain $p_X(x) = \prod_{j=1}^{n} p_X(x_j)$ with, e.g.,
  - sparsity promotion: $p_X(x_j) = \lambda f_X(x_j) + (1-\lambda)\delta(x_j)$
  - real-valuedness: $p_X(x_j)$ supported on $x_j \in \mathbb{R}$
  - non-negativity: $p_X(x_j)$ supported on $x_j \in \mathbb{R}^+ \cup \{0\}$

  and are directly supported by GAMP.

- **Non-separable priors** model structure across $\{x_j\}$, e.g.,
  - structured sparsity: 
    
    $\begin{cases}
    p_X(x) = \sum_{s \in \{0,1\}^n} p_S(s) \prod_{j=1}^{n} p_X|S(x_j|s_j) \\
    p_S(s) = \text{block, Markov field/chain/tree, ...}
    \end{cases}$

  but are not directly supported by GAMP.

- In any case, we want the assumed $p_X(\cdot)$ to match the empirical distribution of the true $\{x_j\}_{j=1}^{n}$, which is apriori unknown.
The basic GAMP algorithm is limited by two major assumptions:
1. separable $p(y|z) = \prod_i p_{Y_i|Z_i}(y_i|z_i)$ and $p(x) = \prod_j p_{X_j}(x_j)$
2. that are well matched to the data.

The EM-turbo-GAMP framework circumvents these limitations by learning [Vila/Schniter’12] possibly non-separable [Schniter’10] priors:
PR-GAMP: Ongoing Work

PR-GAMP is a work-in-progress. Things we are working on include:

- Derivation of the state evolution.
- Automatic learning of signal prior $p_X(\cdot)$ via the EM-GM approach from [Vila/Schniter’12].
- Exploitation of the hidden-Markov-tree support structure of natural images via the turbo approach from [Som/Schniter’10].
- MAP formulation of PR-GAMP.
- Connections to optimization algorithms.
(Compressive) phase retrieval is a longstanding problem that is experiencing a rebirth through compressive sensing and convex relaxation.

We proposed a new approach to CPR based on generalized approximate message passing (GAMP).

Empirical results show an excellent phase transition (4× meas of phase-oracle), excellent noise robustness (∼3 dB worse than phase-oracle), and excellent runtime (many orders of magnitude faster than convex relaxation).

As a practical demonstration, we accurately recovered a 64k-pixel image from 32k noisy measurements in only 11 seconds.
All of these methods are integrated into **GAMPmatlab**: http://sourceforge.net/projects/gampmatlab/

Thanks!
Bibliography