Vector Approximate Message Passing

Phil Schniter

Collaborators: Sundeep Rangan (NYU), Alyson Fletcher (UCLA)

Supported in part by NSF grant CCF-1527162.

iTWIST @ Aalborg University — Aug 24, 2016
Standard Linear Regression

Goal: Recover $x_o \in \mathbb{R}^N$ from observations $y = Ax_o + w \in \mathbb{R}^M$

Examples:

- **Compressive Sensing / Medical Imaging:**
  - $y =$ measurements $x_o =$ sparse image/signal representation
  - $w =$ sensor noise $A = \Phi \Psi$, $\Phi$ measurement operator, $\Psi$ basis

- **Wireless communications:**
  - $y =$ received samples $x_o =$ finite-alphabet symbols
  - $w =$ noise & interference $A =$ channel operator

- **Statistics / Machine Learning:**
  - $y =$ experimental outcomes $x_o =$ prediction coefficients
  - $w =$ model error $A =$ feature data
Implicit assumptions used in most of this talk

**Standard linear regression:**

Recover $x_o \in \mathbb{R}^N$ from $y = Ax_o + w \in \mathbb{R}^M$

- $A$ is a known and high dimensional (e.g., $M, N \gtrsim 100$)
- Often $N \gg M$ (more unknowns than observations)
- $w \sim \mathcal{N}(0, \tau_w I)$ (additive white Gaussian noise)
- $x_o$ is “structured” (e.g., sparse, natural image, etc.)
- Quantities are real-valued (but can be easily extended to complex-valued)

Later will describe extension to **generalized linear model**:

Recover $x_o$ from $y \sim p(y|z)$ with hidden $z = Ax_o$. 
Regularized loss minimization

One way to approach this problem is

\[
\hat{x} = \arg \min_x \frac{1}{2} \| y - Ax \|^2 + \lambda f(x)
\]

where

- \( \frac{1}{2} \| y - Ax \|^2 \) is the quadratic loss function
- \( f(x) \) is a suitably chosen regularizer
  - convex \( f(\cdot) \) leads to a convex optimization problem
  - choosing \( f(x) = \| x \|_1 \) yields sparse \( \hat{x} \)
- \( \lambda > 0 \) is a tuning parameter

Bayesian interpretation:

\[
\hat{x} = \text{MAP estimate of } x \text{ under } \begin{cases} 
\text{likelihood} & p(y|x) = \mathcal{N}(y; Ax, \tau_w I) \\
\text{prior} & p(x) \propto \exp \left( -\lambda f(x)/\tau_w \right) 
\end{cases}
\]
Iterative thresholding

One approach to regularized loss minimization:

| Initialize $\hat{x}^0 = 0$
| For $t = 0, 1, 2, \ldots$
| $v^t = y - A\hat{x}^t$ compute residual
| $\hat{x}^{t+1} = g(\hat{x}^t + A^T v^t)$ thresholding

where

$$g(r) = \arg \min_x \frac{1}{2} \| r - x \|_2^2 + \lambda f(x) \triangleq \text{prox}_{\lambda f}(r)$$

$\| A \|_2^2 < 1$ ensures convergence\(^1\) with convex $f(\cdot)$.

For example, $f(x) = \| x \|_1$ gives “soft thresholding”

$$[g(r)]_j = \text{sgn}(r_j) \max\{0, |r_j| - \lambda\}$$

---

\(^1\) Daubechies, Defrise, DeMol—CPAM’04
Approximate Message Passing (AMP)

A modification of iterative thresholding:

initialize $\hat{x}^0 = 0$, $v^{-1} = 0$

for $t = 0, 1, 2, \ldots$

\[
\begin{align*}
v^t &= y - A\hat{x}^t + \frac{N}{M} v^{t-1} \langle g^{t-1}'(\hat{x}^{t-1} + A^T v^{t-1}) \rangle \\
\hat{x}^{t+1} &= g^t(\hat{x}^t + A^T v^t)
\end{align*}
\]

corrected residual

thresholding

where

\[
\langle g'(r) \rangle \triangleq \frac{1}{N} \sum_{j=1}^{N} \frac{\partial g_j(r)}{\partial r_j} \quad \text{“divergence.”}
\]

Note:

- The residual $v^t$ now includes an “Onsager correction.”
- The thresholding $g^t(\cdot)$ can vary with iteration $t$.
- Can be derived using Gaussian & Taylor-series approximations of min-sum belief-propagation / message passing.
AMP vs ISTA (and FISTA)

Typical convergence behavior with i.i.d. Gaussian $A$:

**Experiment:**
- $M = 250, N = 500$
- $\Pr\{x_n \neq 0\} = 0.1$
- SNR = 40dB
- ISTA, FISTA$^2$, AMP all reach same solution: NMSE = -36.8dB
- Convergence to -35dB:
  - ISTA: 2407 iterations
  - FISTA: 174 iterations
  - AMP: 25 iterations

$^2$ Beck, Teboulle–JIS’09
AMP’s denoising property

**Assumption 1**

- $A \in \mathbb{R}^{M \times N}$ is i.i.d. Gaussian
- $M, N \to \infty$ s.t. $\frac{M}{N} = \delta \in (0, \infty)$
- $f(x) = \sum_{j=1}^{N} f(x_j)$ with Lipschitz $f$

Under Assumption 1, something remarkable happens to the input to the thresherer:\(^3\)

$$r^t \triangleq \hat{x}^t + A^T v^t = x_o + \mathcal{N}(0, \tau_r^t I)$$

with $\tau_r^t = \frac{1}{M} \|v^t\|^2 \triangleq \hat{\tau}_r^t$

In other words, $r^t$ is a noisy version of the true signal $x_o$, where the noise is Gaussian with known variance.

---

\(^3\)Bayati, Montanari–TransIT’11
AMP’s state evolution

Define the iteration-\(t\) mean-squared error (MSE)

\[
\mathcal{E}^t = \frac{1}{N} E \{ \| \hat{x}^t - x_o \|^2 \}.
\]

Under Assumption 1, AMP has the following scalar state evolution (SE):

\[
\text{for } t = 0, 1, 2, \ldots
\]

\[
\tau_r^t = \tau_w + \frac{N}{\mu} \mathcal{E}^t
\]

\[
\mathcal{E}^{t+1} = \frac{1}{N} E \{ \| g^t (x_o + \mathcal{N}(0, \tau_r^t I)) - x_o \|^2 \}
\]

The rigorous proof\(^4\) of the SE uses Bolthausen’s conditioning trick from the statistical physics literature.

\(^4\)Bayati, Montanari—TransIT’11
Choice of denoiser in AMP

1) LASSO/BPDN

- Goal: compute \( \hat{x} = \arg \max_x \frac{1}{2} \| y - Ax \|^2 + \lambda \| x \|_1 \).”
- Use \( g^t(r) = \text{soft}(r; \alpha \sqrt{\hat{\tau}^t_r}) \), where \( \alpha \) has a one-to-one map to \( \lambda \).

2) Bayesian MMSE

- Goal: compute/approximate MMSE estimate \( \hat{x} = E\{x|y\} \).
- Suppose \( x_o \sim \text{i.i.d.} \ p(x_j) \) with known \( p(x_j) \).
- Use \( [g^t(r)]_j = E \{ x_j | r_j = x_{o,j} + N(0, \hat{\tau}^t_r) \} \)  ... scalar denoising!
- MMSE is achieved when the SE has a unique fixed point!

The choice of denoiser determines the problem solved by AMP.
3) Non-parametric (or model free) estimation

- Goal: compute MMSE estimate without knowing i.i.d. prior $p(x_j)$.
- Assume scalar GMM($\theta$) with unknown parameters $\theta$.
- Use MMSE scalar estimator for GMM($\theta^t$) at iteration $t$.
- Use EM algorithm to update $\theta^t$. Details given later...

4) Black-Box Denoisers

- Goal: leverage sophisticated off-the-shelf denoisers like BM3D for natural images or BM4D for image sequences.
- Use $g^t(r) = \text{BM3D}(r; \tau_r^t)$.
- Approximate divergence as $\mathbb{E}[g^t(r)] \approx \frac{1}{N} \sum_{j=1}^{N} \frac{g^t_j(r + \epsilon s) - s_j g^t_j(r)}{\epsilon}$
  where $\{s_j\} \sim \text{i.i.d. uniform } \pm 1$. 

---

Metzler, Maleki, Baraniuk–TIT’16
The limitations of AMP

The good:

- For large i.i.d. sub-Gaussian $\mathbf{A}$, AMP performs provably well.\(^6\)
- Finite-sample analysis shows mild degradation with not-so-large i.i.d. Gaussian $\mathbf{A}$.\(^7\)
- Empirical evidence shows good performance in some other cases (e.g., randomly sub-sampled Fourier $\mathbf{A}$ & i.i.d. sparse $\mathbf{x}$)

The bad:

- For general $\mathbf{A}$, AMP can perform poorly

The ugly:

- For general $\mathbf{A}$, AMP may fail to converge!
  - ill-conditioned $\mathbf{A}$
  - non-zero mean $\mathbf{A}$

---

\(^6\) Bayati, Lelarge, Montanari—AAP’15
\(^7\) Rush, Venkataraman—ISIT’16
This talk: Vector AMP

For SLR $y = Ax + w$, the vector AMP algorithm is\(^8\)

for $t = 0, 1, 2, \ldots$

\[
\begin{align*}
\hat{x}_1^t &= g(r_1^t; \gamma_1^t) \quad \text{denoising} \\
\alpha_1^t &= \langle g'(r_1^t; \gamma_1^t) \rangle \quad \text{divergence} \\
r_2^t &= \frac{1}{1 - \alpha_1^t} (\hat{x}_1^t - \alpha_1^t r_1^t) \quad \text{Onsager correction} \\
\gamma_2^t &= \gamma_1^t \frac{1 - \alpha_1^t}{\alpha_1^t} \quad \text{precision of } r_2^t
\end{align*}
\]

\[
\begin{align*}
\hat{x}_2^t &= \left( A^T A / \hat{\tau}_w + \gamma_2^t I \right)^{-1} \left( A^T y / \hat{\tau}_w + \gamma_2^t r_2^t \right) \quad \text{LMMSE} \\
\alpha_2^t &= \frac{\gamma_2^t}{N} \text{Tr} \left[ (A^T A / \hat{\tau}_w + \gamma_2^t I)^{-1} \right] \quad \text{divergence} \\
r_1^{t+1} &= \frac{1}{1 - \alpha_2^t} (\hat{x}_2^t - \alpha_2^t r_2^t) \quad \text{Onsager correction} \\
\gamma_1^{t+1} &= \gamma_2^t \frac{1 - \alpha_2^t}{\alpha_2^t} \quad \text{precision of } r_1^{t+1}
\end{align*}
\]

Note similarities with standard AMP.

\(^8\)Rangan, Schniter, Fletcher--arXiv:1610.03082.
Vector AMP without matrix inverses

Can avoid matrix inverses using an “economy” SVD $A = USV^T$:

for $t = 0, 1, 2, \ldots$

$\hat{x}^t = g(r_1^t; \gamma_1^t)$  
\text{denoising}

$\alpha_1^t = \langle g'(r_1^t; \gamma_1^t) \rangle$  
\text{divergence}

$r_2^t = \frac{1}{1-\alpha_1^t}(\hat{x}^t - \alpha_1^t r_1^t)$  
\text{Onsager}

$\gamma_2^t = \gamma_1^t \frac{1-\alpha_1^t}{\alpha_1^t}$  
\text{precision}

$\alpha_2^t = \frac{1}{N} \sum_j \gamma_2^t / (s_j^2 / \hat{\tau}_w + \gamma_2^t)$  
\text{divergence}

$r_1^{t+1} = r_2^t + \frac{1}{1-\alpha_2^t} V (S^2 + \hat{\tau}_w \gamma_2^t I)^{-1} S (U^T y - SV^T r_2^t)$  
2 matvec

$\gamma_1^{t+1} = \gamma_2^t \frac{1-\alpha_2^t}{\alpha_2^t}$  
\text{precision}

Note economy SVD computable with $O(M^3 + M^2 N)$ operations.
Why call this “Vector AMP”? 

1) Can be derived using an approximation of message passing on a factor graph, now with vector-valued variable nodes.

2) Performance can be rigorously characterized by a state-evolution in the high-dimensional limit of certain random $A$:

$$SVD \ A = U S V^T$$

- $U$ is deterministic
- $S$ is deterministic
- $V$ is uniformly distributed on the group of orthogonal matrices

“$A$ is right-rotationally invariant”
Message-passing derivation of VAMP

- Write joint density as $p(x, y) = p(x)p(y|x) = p(x)\mathcal{N}(y; Ax, \tau_w I)$

$$p(x) \xrightarrow{} x \xrightarrow{} \mathcal{N}(y; Ax, \tau_w I)$$

- Variable splitting: $p(x_1, x_2, y) = p(x_1)\delta(x_1 - x_2)\mathcal{N}(y; Ax_2, \tau_w I)$

$$p(x_1) \xrightarrow{} x_1 \xrightarrow{} x_2 \xrightarrow{} \mathcal{N}(y; Ax_2, \tau_w I)$$

- Perform\(^9\) message-passing with messages approximated as $\mathcal{N}(\mu, \sigma^2 I)$.

An instance of expectation-propagation\(^{10}\) (EP).

---


\(^{10}\) Minka—Dissertation’01
Free-energy derivation of VAMP

- Want to compute posterior density:

\[ p(x|y) = \frac{p(x)\ell(x)}{Z} \]

with

\[
\begin{align*}
    p(x) &= \text{prior} \\
    \ell(x) &= N(y; Ax, \tau w I), \text{likelihood} \\
    Z &= \int p(x)\ell(x)dx, \text{partition fnn}
\end{align*}
\]

but difficult due to high-dimensional integral.

- What if we compute the density via

\[
\text{arg min}_{b(x)} D(b(x)\|p(x|y))
\]

where the KL divergence can be written as

\[
D(b\|p) = D(b\|p) + D(b\|\ell) + H(b) + \text{const,}
\]

Gibbs free energy

thus avoiding the partition function \(Z\). Still difficult...
Free-energy derivation of VAMP (cont.)

- What about splitting the belief $b(\mathbf{x})$:

$$\arg\min_{b_1, b_2} \max_q J(b_1, b_2, q) \text{ s.t. } b_1 = b_2 = q$$

$$J(b_1, b_2, q) = D(b_1 \| p) + D(b_2 \| \ell) + H(q)$$

noting that $D(\cdot \| p)$ is convex and $H(\cdot)$ is concave?

Still difficult due to the pdf constraint...

- So, relax the pdf constraint to moment-matching constraints:

$$b_1 = b_2 = q \quad \rightarrow \quad \begin{cases} E\{\mathbf{x}|b_1\} = E\{\mathbf{x}|b_2\} = E\{\mathbf{x}|q\} \\ \text{Tr}[\text{Cov}\{\mathbf{x}|b_1\}] = \text{Tr}[\text{Cov}\{\mathbf{x}|b_2\}] = \text{Tr}[\text{Cov}\{\mathbf{x}|q\}] \end{cases}$$

An instance of expectation-consistent approximation\textsuperscript{11} (EC).

\textsuperscript{11}Opper,Winther–NIPS'04, Fletcher,Rangan,Schniter–ISIT’16
The stationary points of the EC optimization are

\[
\begin{align*}
    b_1(x) &\propto p(x) \mathcal{N}(x; r_1; I/\gamma_1) \\
    b_2(x) &\propto \ell(x) \mathcal{N}(x; r_2; I/\gamma_2) \\
    q(x) &= \mathcal{N}(x; \hat{x}; I/\eta)
\end{align*}
\]

for parameters \( r_1, \gamma_1, r_2, \gamma_2, \hat{x}, \eta \) that satisfy

\[
\begin{align*}
    \hat{x} &= E\{x|b_1\} = E\{x|b_2\} = E\{x|q\} \\
    1/\eta &= \frac{1}{N} \text{Tr}[\text{Cov}\{x|b_1\}] = \frac{1}{N} \text{Tr}[\text{Cov}\{x|b_2\}] = \frac{1}{N} \text{Tr}[\text{Cov}\{x|q\}].
\end{align*}
\]

Can then construct algorithms whose fixed points coincide with these stationary points (e.g., EC, ADATAP\(^{12}\), S-AMP\(^{13}\)). But convergence is not guaranteed.

---

\(^{12}\) Opper, Winther–NC’00
\(^{13}\) Cacmak, Winter, Fleury–ITW’14
The aforementioned belief-propagation and free-energy derivations are both well known and heuristic (in general). The resulting algorithms may not converge to their fixed points:

- S-AMP diverges with mildly ill-conditioned matrices
- Even if they do converge, the accuracy of the fixed points is unclear:
  - EP generally suboptimal due to approximation of messages
  - EC generally suboptimal due to approximation of constraint

The important question is whether/when a given heuristic can be rigorously analyzed and proven to work well.

AMP rigorous analyzed under large i.i.d. Gaussian $A$ and Bayes optimal under certain combinations of $\{p(x), \ell(x)\}$. 
VAMP state evolution

VAMP has a rigorous SE\textsuperscript{14}

Assuming empirical convergence of \(\{s_j\} \to S\) and \(\{(r_{1,j}, x_{o,j})\} \to (R_1^0, X_o)\) and Lipschitz continuity of \(g\) and \(g'\), the VAMP-SE under \(\hat{\tau}_w = \tau_w\) is as follows:

for \(t = 0, 1, 2, \ldots\)

\[
E_1^t = E \left\{ \left[ g\left( X_o + N(0, \tau_1^t); \gamma_1^t \right) - X_o \right]^2 \right\}
\]

\(\bar{\alpha}_1^t = E \left\{ g'\left( X_o + N(0, \tau_1^t); \gamma_1^t \right) \right\}\)

\[
\gamma_2^t = \gamma_1^t \frac{1 - \bar{\alpha}_1^t}{\bar{\alpha}_1^t}, \quad \tau_2^t = \frac{1}{(1 - \bar{\alpha}_1^t)^2} \left[ E_1^t - \left( \bar{\alpha}_1^t \right)^2 \tau_1^t \right]
\]

\[
E_2^t = E \left\{ \left[ \frac{S^2}{\tau_w} + \gamma_2^t \right]^{-1} \right\}
\]

\(\bar{\alpha}_2^t = \gamma_2^t E \left\{ \left[ \frac{S^2}{\tau_w} + \gamma_2^t \right]^{-1} \right\}\)

\[
\bar{\gamma}_1^{t+1} = \gamma_2^t \frac{1 - \bar{\alpha}_2^t}{\bar{\alpha}_2^t}, \quad \tau_1^{t+1} = \frac{1}{(1 - \bar{\alpha}_2^t)^2} \left[ E_2^t - \left( \bar{\alpha}_2^t \right)^2 \tau_2^t \right]
\]

More complicated expressions for \(E_2^t\) and \(\bar{\alpha}_2^t\) apply when \(\hat{\tau}_w \neq \tau_w\).

\textsuperscript{14} Rangan, Schniter, Fletcher–arXiv:1610.03082
Connections to the replica prediction

- The **replica method** from statistical physics is often used to characterize the average behavior of large disordered systems.

- Although not fully rigorous, replica predictions are usually correct.

- For **SLR under large right-rotationally invariant $A$ and matched priors**, the MMSE $\mathcal{E}_1(\overline{\gamma}_1)$ should satisfy the fixed-point equation\textsuperscript{15}

$$\overline{\gamma}_1 = R_{A^\top A/\tau_w}(-\mathcal{E}_1(\overline{\gamma}_1)),$$

where $R_C(\cdot)$ denotes the $R$-transform of matrix $C$ and

$$\mathcal{E}_1(\overline{\gamma}_1) \triangleq \mathbb{E}\left\{ \left[ g_{\text{mmse}}(X_o + \mathcal{N}(0, 1/\overline{\gamma}_1); \overline{\gamma}_1) - X_o \right]^2 \right\}.$$

- It can be shown that VAMP’s matched SE obeys the above equation.

- Thus, if the replica prediction is correct, then VAMP’s estimates will be MMSE whenever the replica fixed-point equation has a unique solution.

---

\textsuperscript{15} Tulino, Caire, Verdu, Shamai—TIT’13
Experiment with Matched Priors I

Note robustness w.r.t. condition number of $A$. 

$N = 1024$
$M/N = 0.5$

$A = U \text{Diag}(s)V^T$
$U, V$ drawn uniform
$s_n/s_{n-1} = \phi \ \forall n$
$\phi$ determines $\kappa(A)$

$X_0 \sim$ Bernoulli-Gaussian
$\Pr\{X_0 \neq 0\} = 0.1$

$\text{SNR} = 40\text{dB}$
Experiment with Matched Priors II

Note convergence speed relative to (damped) EM-AMP.

\[ N = 1024 \]
\[ M/N = 0.5 \]

\[ A = U \text{Diag}(s)V^T \]
\[ U, V \text{ drawn uniform} \]
\[ s_n/s_{n-1} = \phi \ \forall n \]
\[ \phi \text{ determines } \kappa(A) \]

\[ X_o \sim \text{Bernoulli-Gaussian} \]
\[ \text{Pr}\{X_0 \neq 0\} = 0.1 \]

\[ \text{SNR} = 40\text{dB} \]
Non-parametric (model-free) regression

- So far we considered recovering $x_o$ from
  \[ y = Ax_o + w, \quad x_o \sim p(x), \quad w \sim \mathcal{N}(0, \tau_w I), \]
  when $p(x)$ and $\tau_w$ are known.

- Can we learn $\tau_w$? Yes, through an EM procedure.\(^\text{16}\)
  Can we learn $p(x)$? Yes if $p(x) = \prod_j p(x_j)$.

- Why is $p(x_j)$ learnable with VAMP?
  - Recall that $r^t_1 = x_o + \mathcal{N}(0, \tau^t_1 I)$.
  - Thus $r^t_1$ contains i.i.d. samples of $p(x_j) \ast \mathcal{N}(x_j; 0, \tau^t_1)$.
  - Should be able to deconvolve $p(x_j)$ from the empirical distribution of $r^t_1$.

- A practical method: Model $p(x_j) = \text{GMM}(x_j; \theta_x)$. Learn parameters $\theta_x$ using EM.

\(^{16}\)Fletcher, Schniter–arXiv:1602.08207
EM-VAMP

Recall \[
\begin{align*}
\text{prior } p(x; \theta_x) \\
\text{likelihood } \ell(x; \tau_w)
\end{align*}
\] → Learn parameters \( \theta \triangleq (\theta_x, \tau_w) \).

EM: iterate

\[Q(\theta; \hat{\theta}^k) = \int p(x|y; \hat{\theta}^k) \ln p(x, y; \theta) dx \quad \text{“expectation”}
\]

\[\hat{\theta}^{k+1} = \arg \max_\theta Q(\theta; \hat{\theta}^k) \quad \text{“maximization”}
\]

which uses the posterior \( p(x|y; \hat{\theta}^k) \) in the E step.

With VAMP’s posterior approx, EM is an alternating approach to

\[
\min_{b_1, b_2, \theta} \max_q \left( D(b_1||p(\theta_x)) + D(b_2||\ell(\tau_w)) + H(q) \right)
\]

s.t.

\[
\begin{align*}
E\{x|b_1\} &= E\{x|b_2\} = E\{x|q\} \\
\text{Tr}[\text{Cov}\{x|b_1\}] &= \text{Tr}[\text{Cov}\{x|b_2\}] = \text{Tr}[\text{Cov}\{x|q\}]
\end{align*}
\]

Can make faster by putting \( \theta \) optimization in the inner loop.
Experiment with Learned Parameters I

Learning both $\tau_w$ and $\theta_x$:

\begin{align*}
N &= 1024 \\
M/N &= 0.5
\end{align*}

\[
A = U \text{Diag}(s)V^T
\]

$U, V$ drawn uniform

$s_n/s_{n-1} = \phi \ \forall n$

$\phi$ determines $\kappa(A)$

$X_o \sim \text{Bernoulli-Gaussian}$

$\Pr\{X_0 \neq 0\} = 0.1$

\[
\text{SNR} = 40\text{dB}
\]

EM-VAMP achieves oracle performance at all condition numbers.
Experiment with Learned Parameters II

Learning both $\tau_w$ and $\theta_x$:

- $N = 1024$
- $M/N = 0.5$

$$A = U \text{ Diag}(s)V^T$$
$U$, $V$ drawn uniform
$s_n/s_{n-1} = \phi \forall n$
$\phi$ determines $\kappa(A)$

$$X_0 \sim \text{Bernoulli-Gaussian}$$
$$\Pr\{X_0 \neq 0\} = 0.1$$

$$\text{SNR} = 40\text{dB}$$

EM-VAMP nearly as fast as VAMP and much faster than EM-AMP.
## Noiseless Image Recovery with BM3D

<table>
<thead>
<tr>
<th>%</th>
<th>L1-AMP PSNR</th>
<th>L1-VAMP PSNR</th>
<th>BM3D-AMP PSNR</th>
<th>BM3D-VAMP PSNR</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>10%</td>
<td>17.7 dB</td>
<td>17.6 dB</td>
<td>25.2 dB</td>
<td>25.2 dB</td>
<td>0.5 s</td>
</tr>
<tr>
<td>20%</td>
<td>20.2 dB</td>
<td>20.2 dB</td>
<td>30.0 dB</td>
<td>30.0 dB</td>
<td>1.0 s</td>
</tr>
<tr>
<td>30%</td>
<td>22.4 dB</td>
<td>22.4 dB</td>
<td>32.5 dB</td>
<td>32.5 dB</td>
<td>1.6 s</td>
</tr>
<tr>
<td>40%</td>
<td>24.6 dB</td>
<td>24.8 dB</td>
<td>35.1 dB</td>
<td>35.2 dB</td>
<td>2.3 s</td>
</tr>
<tr>
<td>50%</td>
<td>27.0 dB</td>
<td>27.2 dB</td>
<td>37.4 dB</td>
<td>37.7 dB</td>
<td>3.1 s</td>
</tr>
</tbody>
</table>

Avg results for recovering 128x128 lena, barbara, boat, fingerprint, house, and peppers from $y = Ax_o$ with i.i.d. Gaussian $A$ at various sampling ratios.

All algorithms use 20 iterations and learn the noise variance $\tau_w$.

VAMP slightly outperforms AMP in accuracy and runtime.
Now look a sampling rates $\leq 5\%$.

Goal: recover $128 \times 128$ lena from $y = Ax_o$ with i.i.d. Gaussian $A$ and unknown $\tau_w$.

BM3D-VAMP does much better than BM3D-AMP.
Generalized linear models

- Until now we have considered SLR, $y = Ax_o + w$.
- VAMP can also support the generalized linear model (GLM)
  \[ y \sim p(y|z) \] with hidden $z = Ax_o$

which supports, e.g.,
- $y_i = z_i + w_i$: additive, possibly non-Gaussian noise
- $y_i = \text{sgn}(z_i + w_i)$: binary classification / one-bit sensing
- $y_i = |z_i + w_i|$: phase retrieval in noise
- Poisson $y_i$: photon-limited imaging

**Trick:** $z = Ax \iff \begin{bmatrix} 0 \\ \tilde{z} \end{bmatrix} = \begin{bmatrix} A - I \\ \tilde{A} \end{bmatrix} \begin{bmatrix} x \\ z \end{bmatrix}$
One-bit compressed sensing / Probit regression

Learning both $\tau_w$ and $\theta_x$:

\[ N = 512 \]
\[ M/N = 4 \]

\[ A = U \text{Diag}(s)V^T \]
\[ U, V \text{ drawn uniform} \]
\[ s_n/s_{n-1} = \phi \forall n \]
\[ \phi \text{ determines } \kappa(A) \]

\[ X_o \sim \text{Bernoulli-Gaussian} \]
\[ \text{Pr}\{X_0 \neq 0\} = 1/32 \]

\[ \text{SNR} = 40\text{dB} \]

VAMP and EM-VAMP robust to ill-conditioned $A$. 
One-bit compressed sensing / Probit regression

Learning both $\tau_w$ and $\theta_x$:

\[
A = U \text{Diag}(s)V^T
\]

$U, V$ drawn uniform
\[s_n/s_{n-1} = \phi \ \forall n\]

$\phi$ determines $\kappa(A)$

\[X_o \sim \text{Bernoulli-Gaussian}\]

\[\text{Pr}\{X_0 \neq 0\} = 1/32\]

SNR = 40dB

EM-VAMP mildly slower than VAMP but much faster than damped AMP.
Conclusions

AMP exhibits some remarkable properties

- low cost-per-iteration and relatively few iterations to convergence,
- intermediate estimates of form $r^t = x_0 + \mathcal{N}(0, \tau_t I)$,
- rigorous state evolution,
- easy tuning of prior & likelihood,
- compatibility with plug-in denoisers like BM3D,

but those properties are guaranteed only under large i.i.d. Gaussian $A$.

Vector AMP has the same properties, but for a much larger class of $A$.

Ongoing work: analysis of EM procedure, bilinear extensions, connections with deep learning, various applications...
Thanks for listening!