Vector Approximate Message Passing

Phil Schniter

THE OHIO STATE UNIVERSITY Duke UNIVERSITY iiD

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 $\boldsymbol{x}_o \in \mathbb{R}^N$ from observations $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}_o + \boldsymbol{w} \in \mathbb{R}^M$

Examples:

- Compressive Sensing / Medical Imaging:
  $\boldsymbol{y} =$ measurements $\quad \boldsymbol{x}_o =$ sparse image/signal representation
  $\boldsymbol{w} =$ sensor noise $\quad \boldsymbol{A} = \boldsymbol{\Phi\Psi}$, $\boldsymbol{\Phi}$ measurement operator, $\boldsymbol{\Psi}$ basis

- Wireless communications:
  $\boldsymbol{y} =$ received samples $\quad \boldsymbol{x}_o =$ finite-alphabet symbols
  $\boldsymbol{w} =$ noise & interference $\quad \boldsymbol{A} =$ channel operator

- Statistics / Machine Learning:
  $\boldsymbol{y} =$ experimental outcomes $\quad \boldsymbol{x}_o =$ prediction coefficients
  $\boldsymbol{w} =$ model error $\quad \boldsymbol{A} =$ feature data

Implicit assumptions used in most of this talk

Standard linear regression:

Recover $\boldsymbol{x}_o \in \mathbb{R}^N$ from $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}_o + \boldsymbol{w} \in \mathbb{R}^M$

- $\boldsymbol{A}$ is a known and high dimensional (e.g., $M, N \gtrsim 100$)
- often $N \gg M$ (more unknowns than observations)
- $\boldsymbol{w} \sim \mathcal{N}(\mathbf{0}, \tau_w \boldsymbol{I})$ (additive white Gaussian noise)
- $\boldsymbol{x}_o$ 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 $\boldsymbol{x}_o$ from $\boldsymbol{y} \sim p(\boldsymbol{y}|\boldsymbol{z})$ with hidden $\boldsymbol{z} = \boldsymbol{A}\boldsymbol{x}_o$.

Regularized loss minimization

One way to approach this problem is

$$\boxed{\widehat{\boldsymbol{x}} = \arg\min_{\boldsymbol{x}} \tfrac{1}{2}\|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|^2 + \lambda f(\boldsymbol{x})}$$

where

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

Bayesian interpretation:

$$\widehat{\boldsymbol{x}} = \text{MAP estimate of } \boldsymbol{x} \text{ under } \begin{cases} \text{likelihood } p(\boldsymbol{y}|\boldsymbol{x}) = \mathcal{N}(\boldsymbol{y}; \boldsymbol{A}\boldsymbol{x}, \tau_w \boldsymbol{I}) \\ \text{prior } p(\boldsymbol{x}) \propto \exp\left(-\lambda f(\boldsymbol{x})/\tau_w\right) \end{cases}$$

Iterative thresholding

One approach to regularized loss minimization:

$$
\begin{array}{ll}
\text{initialize } \widehat{\boldsymbol{x}}^0 = \mathbf{0} & \\
\text{for } t = 0, 1, 2, \dots & \\
\quad \boldsymbol{v}^t = \boldsymbol{y} - \boldsymbol{A}\widehat{\boldsymbol{x}}^t & \text{compute residual} \\
\quad \widehat{\boldsymbol{x}}^{t+1} = \boldsymbol{g}(\widehat{\boldsymbol{x}}^t + \boldsymbol{A}^\mathsf{T}\boldsymbol{v}^t) & \text{thresholding}
\end{array}
$$

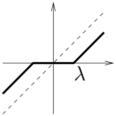
where

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

$\|\boldsymbol{A}\|_2^2 < 1$ ensures convergence[1] with convex $f(\cdot)$.

For example, $f(\boldsymbol{x}) = \|\boldsymbol{x}\|_1$ gives "soft thresholding"

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



[1] Daubechies,Defrise,DeMol–CPAM'04

Approximate Message Passing (AMP)

A modification of iterative thresholding:

initialize $\widehat{\boldsymbol{x}}^0 = \mathbf{0}$, $\boldsymbol{v}^{-1} = \mathbf{0}$

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

$$\boldsymbol{v}^t = \boldsymbol{y} - \boldsymbol{A}\widehat{\boldsymbol{x}}^t + \tfrac{N}{M}\boldsymbol{v}^{t-1}\big\langle \boldsymbol{g}^{t-1\prime}(\widehat{\boldsymbol{x}}^{t-1} + A^{\mathsf{T}}\widehat{\boldsymbol{v}}^{t-1})\big\rangle \quad \text{corrected residual}$$

$$\widehat{\boldsymbol{x}}^{t+1} = \boldsymbol{g}^t(\widehat{\boldsymbol{x}}^t + \boldsymbol{A}^{\mathsf{T}}\boldsymbol{v}^t) \quad \text{thresholding}$$

where

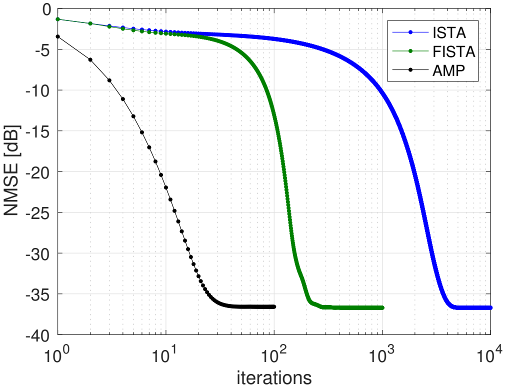
$$\big\langle \boldsymbol{g}'(\boldsymbol{r})\big\rangle \triangleq \tfrac{1}{N}\textstyle\sum_{j=1}^N \frac{\partial g_j(\boldsymbol{r})}{\partial r_j} \quad \text{“divergence.”}$$

Note:

- The residual $\boldsymbol{v}^t$ now includes an “Onsager correction.”
- The thresholding $\boldsymbol{g}^t(\cdot)$ 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 $\boldsymbol{A}$:



Experiment:

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

[2] Beck,Teboulle–JIS'09

AMP's denoising property

Assumption 1

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

Under Assumption 1, something remarkable happens to the input to the thresholder:[3]

$$\boldsymbol{r}^t \triangleq \widehat{\boldsymbol{x}}^t + \boldsymbol{A}^\mathsf{T} \boldsymbol{v}^t = \boldsymbol{x}_o + \mathcal{N}(\boldsymbol{0}, \tau_r^t \boldsymbol{I})$$

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

In other words, $\boldsymbol{r}^t$ is a noisy version of the true signal $\boldsymbol{x}_o$, 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)

$$\mathcal{E}^t = \frac{1}{N}\,\mathrm{E}\left\{\|\widehat{\boldsymbol{x}}^t - \boldsymbol{x}_o\|^2\right\}.$$

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

$$\boxed{\begin{aligned} &\text{for } t = 0, 1, 2, \ldots \\ &\qquad \tau_r^t = \tau_w + \tfrac{N}{M}\mathcal{E}^t \\ &\qquad \mathcal{E}^{t+1} = \tfrac{1}{N}\,\mathrm{E}\left\{\left\|\boldsymbol{g}^t\big(\boldsymbol{x}_o + \mathcal{N}(\boldsymbol{0}, \tau_r^t \boldsymbol{I})\big) - \boldsymbol{x}_o\right\|^2\right\} \end{aligned}}$$

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 "$\widehat{\boldsymbol{x}} = \arg\max_{\boldsymbol{x}} \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|^2 + \lambda\|\boldsymbol{x}\|_1$."
- Use $\boldsymbol{g}^t(\boldsymbol{r}) = \text{soft}\big(\boldsymbol{r}; \alpha\sqrt{\widehat{\tau}_r^t}\big)$, where α has a one-to-one map to λ.

2) Bayesian MMSE

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

The choice of denoiser determines the problem solved by AMP.

Choice of denoiser in AMP (cont.)

3) Non-parametric (or model free) estimation

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

4) Black-Box Denoisers[5]

- Goal: leverage sophisticated off-the-shelf denoisers like BM3D for natural images or BM4D for image sequences.
- Use $\boldsymbol{g}^t(\boldsymbol{r}) = \text{BM3D}(\boldsymbol{r}; \tau_r^t)$.
- Approximate divergence as $\langle \boldsymbol{g}^{t\prime}(\boldsymbol{r}) \rangle \approx \dfrac{1}{N} \sum_{j=1}^{N} \dfrac{g_j^t(\boldsymbol{r} + \epsilon \boldsymbol{s}) - s_j g_j^t(\boldsymbol{r})}{\epsilon}$ where $\{s_j\} \sim$ i.i.d. uniform ± 1.

[5] Metzler,Maleki,Baraniuk–TIT'16

The limitations of AMP

The good:

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

The bad:

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

The ugly:

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

[6] Bayati,Lelarge,Montanari–AAP'15

[7] Rush,Venkataraman–ISIT'16

This talk: Vector AMP

For SLR $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{w}$, the vector AMP algorithm is[8]

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

$$\widehat{\boldsymbol{x}}_1^t = \boldsymbol{g}(\boldsymbol{r}_1^t; \gamma_1^t) \qquad \text{denoising}$$

$$\alpha_1^t = \langle \boldsymbol{g}'(\boldsymbol{r}_1^t; \gamma_1^t) \rangle \qquad \text{divergence}$$

$$\boldsymbol{r}_2^t = \tfrac{1}{1-\alpha_1^t}\left(\widehat{\boldsymbol{x}}_1^t - \alpha_1^t \boldsymbol{r}_1^t\right) \qquad \text{Onsager correction}$$

$$\gamma_2^t = \gamma_1^t \tfrac{1-\alpha_1^t}{\alpha_1^t} \qquad \text{precision of } \boldsymbol{r}_2^t$$

$$\widehat{\boldsymbol{x}}_2^t = \left(\boldsymbol{A}^\mathsf{T}\boldsymbol{A}/\widehat{\tau}_w + \gamma_2^t \boldsymbol{I}\right)^{-1}\left(\boldsymbol{A}^\mathsf{T}\boldsymbol{y}/\widehat{\tau}_w + \gamma_2^t \boldsymbol{r}_2^t\right) \qquad \text{LMMSE}$$

$$\alpha_2^t = \tfrac{\gamma_2^t}{N} \operatorname{Tr}\left[\left(\boldsymbol{A}^\mathsf{T}\boldsymbol{A}/\widehat{\tau}_w + \gamma_2^t \boldsymbol{I}\right)^{-1}\right] \qquad \text{divergence}$$

$$\boldsymbol{r}_1^{t+1} = \tfrac{1}{1-\alpha_2^t}\left(\widehat{\boldsymbol{x}}_2^t - \alpha_2^t \boldsymbol{r}_2^t\right) \qquad \text{Onsager correction}$$

$$\gamma_1^{t+1} = \gamma_2^t \tfrac{1-\alpha_2^t}{\alpha_2^t} \qquad \text{precision of } \boldsymbol{r}_1^{t+1}$$

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 $\boldsymbol{A} = \boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^{\mathsf{T}}$:

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

$\widehat{\boldsymbol{x}}^t = \boldsymbol{g}(\boldsymbol{r}_1^t; \gamma_1^t)$	denoising
$\alpha_1^t = \langle \boldsymbol{g}'(\boldsymbol{r}_1^t; \gamma_1^t) \rangle$	divergence
$\boldsymbol{r}_2^t = \frac{1}{1-\alpha_1^t}\big(\widehat{\boldsymbol{x}}^t - \alpha_1^t \boldsymbol{r}_1^t\big)$	Onsager
$\gamma_2^t = \gamma_1^t \frac{1-\alpha_1^t}{\alpha_1^t}$	precision
$\alpha_2^t = \frac{1}{N}\sum_j \gamma_2^t / \big(s_j^2/\widehat{\tau}_w + \gamma_2^t\big)$	divergence
$\boldsymbol{r}_1^{t+1} = \boldsymbol{r}_2^t + \frac{1}{1-\alpha_2^t}\boldsymbol{V}\big(\boldsymbol{S}^2 + \widehat{\tau}_w \gamma_2^t \boldsymbol{I}\big)^{-1}\boldsymbol{S}\big(\boldsymbol{U}^{\mathsf{T}}\boldsymbol{y} - \boldsymbol{S}\boldsymbol{V}^{\mathsf{T}}\boldsymbol{r}_2^t\big)$	2 matvec
$\gamma_1^{t+1} = \gamma_2^t \frac{1-\alpha_2^t}{\alpha_2^t}$	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 $\boldsymbol{A}$:

SVD $\boldsymbol{A} = \boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^\mathsf{T}$

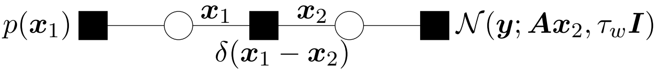
- $\boldsymbol{U}$ *is deterministic*
- $\boldsymbol{S}$ *is deterministic*
- $\boldsymbol{V}$ *is uniformly distributed on the group of orthogonal matrices*

"$\boldsymbol{A}$ *is right-rotationally invariant*"

Message-passing derivation of VAMP

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

- Variable splitting: $p(\boldsymbol{x}_1,\boldsymbol{x}_2,\boldsymbol{y}) = p(\boldsymbol{x}_1)\delta(\boldsymbol{x}_1-\boldsymbol{x}_2)\mathcal{N}(\boldsymbol{y};\boldsymbol{A}\boldsymbol{x}_2,\tau_w\boldsymbol{I})$



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

  An instance of expectation-propagation[10] (EP).

[9] Rangan,Schniter,Fletcher–arXiv:1610.03082.

[10] Minka–Dissertation'01

Free-energy derivation of VAMP

- Want to compute posterior density:

$$p(\boldsymbol{x}|\boldsymbol{y}) = \frac{p(\boldsymbol{x})\ell(\boldsymbol{x})}{Z} \text{ with } \begin{cases} p(\boldsymbol{x}) = \text{prior} \\ \ell(\boldsymbol{x}) = N(\boldsymbol{y}; \boldsymbol{A}\boldsymbol{x}, \tau_w \boldsymbol{I}), \text{ likelihood} \\ Z = \int p(\boldsymbol{x})\ell(\boldsymbol{x})\mathrm{d}\boldsymbol{x}, \text{ partition fxn} \end{cases}$$

  but difficult due to high-dimensional integral.

- What if we compute the density via

$$\arg\min_{b(\boldsymbol{x})} D\big(b(\boldsymbol{x}) \big\| p(\boldsymbol{x}|\boldsymbol{y})\big)$$

  where the KL divergence can be written as

$$D\big(b\big\|p\big) = \underbrace{D\big(b\big\|p\big) + D\big(b\big\|\ell\big) + H\big(b\big)}_{\text{Gibbs free energy}} + \text{const},$$

  thus avoiding the partition function Z. Still difficult...

Free-energy derivation of VAMP (cont.)

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

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

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

  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:

$$b_1 = b_2 = q \quad \longrightarrow \quad \begin{cases} \mathrm{E}\{\boldsymbol{x}|b_1\} = \mathrm{E}\{\boldsymbol{x}|b_2\} = \mathrm{E}\{\boldsymbol{x}|q\} \\ \mathrm{Tr}[\mathrm{Cov}\{\boldsymbol{x}|b_1\}] = \mathrm{Tr}[\mathrm{Cov}\{\boldsymbol{x}|b_2\}] = \mathrm{Tr}[\mathrm{Cov}\{\boldsymbol{x}|q\}] \end{cases}$$

  An instance of expectation-consistent approximation[11] *(EC).*

[11] Opper,Winther–NIPS'04, Fletcher,Rangan,Schniter–ISIT'16

Free-energy derivation of VAMP (cont.)

- The stationary points of the EC optimization are

$$\begin{aligned} b_1(\boldsymbol{x}) &\propto p(\boldsymbol{x})\mathcal{N}(\boldsymbol{x};\boldsymbol{r}_1;\boldsymbol{I}/\gamma_1) \\ b_2(\boldsymbol{x}) &\propto \ell(\boldsymbol{x})\mathcal{N}(\boldsymbol{x};\boldsymbol{r}_2;\boldsymbol{I}/\gamma_2) \\ q(\boldsymbol{x}) &= \mathcal{N}(\boldsymbol{x};\widehat{\boldsymbol{x}};\boldsymbol{I}/\eta) \end{aligned}$$

  for parameters $\boldsymbol{r}_1, \gamma_1, \boldsymbol{r}_2, \gamma_2, \widehat{\boldsymbol{x}}, \eta$ that satisfy

$$\begin{aligned} \widehat{\boldsymbol{x}} &= \mathrm{E}\{\boldsymbol{x}|b_1\} = \mathrm{E}\{\boldsymbol{x}|b_2\} = \mathrm{E}\{\boldsymbol{x}|q\} \\ 1/\eta &= \tfrac{1}{N}\operatorname{Tr}[\operatorname{Cov}\{\boldsymbol{x}|b_1\}] = \tfrac{1}{N}\operatorname{Tr}[\operatorname{Cov}\{\boldsymbol{x}|b_2\}] = \tfrac{1}{N}\operatorname{Tr}[\operatorname{Cov}\{\boldsymbol{x}|q\}]. \end{aligned}$$

- 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

Putting things in perspective

- 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 $\boldsymbol{A}$ and Bayes optimal under certain combinations of $\{p(\boldsymbol{x}), \ell(\boldsymbol{x})\}$.

VAMP state evolution

VAMP has a rigorous SE[14]

Assuming empirical convergence of $\{s_j\} \to S$ and $\{(r^0_{1,j}, x_{o,j})\} \to (R^0_1, X_o)$ and Lipschitz continuity of g and g', the VAMP-SE under $\widehat{\tau}_w = \tau_w$ is as follows:

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

$$\mathcal{E}^t_1 = \mathrm{E}\left\{\left[g\big(X_o + \mathcal{N}(0, \tau^t_1); \overline{\gamma}^t_1\big) - X_o\right]^2\right\} \qquad \text{MSE}$$

$$\overline{\alpha}^t_1 = \mathrm{E}\left\{g'(X_o + \mathcal{N}(0, \tau^t_1); \overline{\gamma}^t_1)\right\} \qquad \text{divergence}$$

$$\overline{\gamma}^t_2 = \overline{\gamma}^t_1 \tfrac{1-\overline{\alpha}^t_1}{\overline{\alpha}^t_1}, \quad \tau^t_2 = \tfrac{1}{(1-\overline{\alpha}^t_1)^2}\left[\mathcal{E}^t_1 - \left(\overline{\alpha}^t_1\right)^2 \tau^t_1\right]$$

$$\mathcal{E}^t_2 = \mathrm{E}\left\{\left[S^2/\tau_w + \overline{\gamma}^t_2\right]^{-1}\right\} \qquad \text{MSE}$$

$$\overline{\alpha}^t_2 = \overline{\gamma}^t_2\, \mathrm{E}\left\{\left[S^2/\tau_w + \overline{\gamma}^t_2\right]^{-1}\right\} \qquad \text{divergence}$$

$$\overline{\gamma}^{t+1}_1 = \overline{\gamma}^t_2 \tfrac{1-\overline{\alpha}^t_2}{\overline{\alpha}^t_2}, \quad \tau^{t+1}_1 = \tfrac{1}{(1-\overline{\alpha}^t_2)^2}\left[\mathcal{E}^t_2 - \left(\overline{\alpha}^t_2\right)^2 \tau^t_2\right]$$

More complicated expressions for $\mathcal{E}^t_2$ and $\overline{\alpha}^t_2$ apply when $\widehat{\tau}_w \neq \tau_w$.

[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 $\boldsymbol{A}$ and matched priors,

  The MMSE $\mathcal{E}_1(\overline{\gamma}_1)$ should satisfy the fixed-point equation[15]

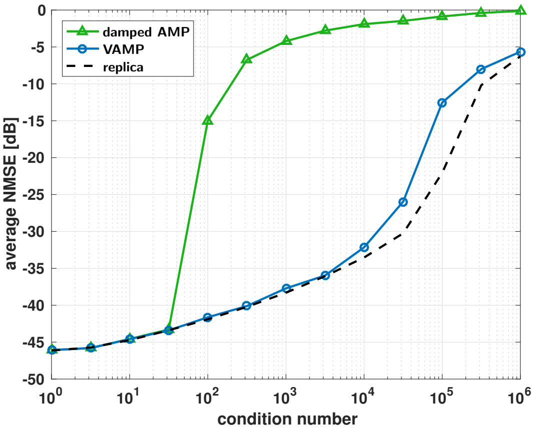
  $$\overline{\gamma}_1 = R_{\boldsymbol{A}^\mathsf{T}\boldsymbol{A}/\tau_w}(-\mathcal{E}_1(\overline{\gamma}_1)),$$

  where $\boldsymbol{R}_{\boldsymbol{C}}(\cdot)$ denotes the R-transform of matrix $\boldsymbol{C}$ and
  $\mathcal{E}_1(\overline{\gamma}_1) \triangleq \mathrm{E}\left\{\left[g_{mmse}\big(X_o + \mathcal{N}(0, 1/\overline{\gamma}_1); \overline{\gamma}_1\big) - X_o\right]^2\right\}.$

- 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.

[15] Tulino,Caire,Verdu,Shamai–TIT'13

Experiment with Matched Priors I



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

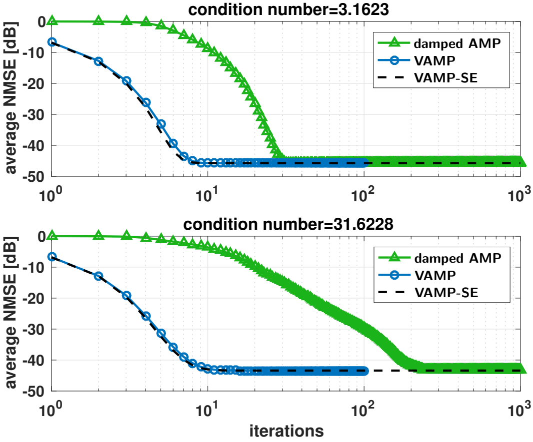
$\boldsymbol{A} = \boldsymbol{U}\,\mathrm{Diag}(\boldsymbol{s})V^{\mathsf{T}}$
$\boldsymbol{U}, \boldsymbol{V}$ drawn uniform
$s_n/s_{n-1} = \phi\ \forall n$
ϕ determines $\kappa(\boldsymbol{A})$

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

SNR $= 40$dB

Note robustness w.r.t. condition number of $\boldsymbol{A}$.

Experiment with Matched Priors II



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

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

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

SNR $= 40$dB

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

Non-parametric (model-free) regression

- So far we considered recovering $\boldsymbol{x}_o$ from

$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}_o + \boldsymbol{w}, \quad \boldsymbol{x}_o \sim p(\boldsymbol{x}), \quad \boldsymbol{w} \sim \mathcal{N}(\mathbf{0}, \tau_w \boldsymbol{I}),$$

  when $p(\boldsymbol{x})$ and τ_w are known.

- Can we learn τ_w? Yes, through an EM procedure.[16]
  Can we learn $p(\boldsymbol{x})$? Yes if $p(\boldsymbol{x}) = \prod_j p(x_j)$.

- Why is $p(x_j)$ learnable with VAMP?
  - Recall that $\boldsymbol{r}_1^t = \boldsymbol{x}_o + \mathcal{N}(\mathbf{0}, \tau_1^t \boldsymbol{I})$.
  - Thus $\boldsymbol{r}_1^t$ contains i.i.d. samples of $p(x_j) * \mathcal{N}(x_j; 0, \tau_1^t)$.
  - Should be able to deconvolve $p(x_j)$ from the empirical distribution of $\boldsymbol{r}_1^t$.

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

[16] Fletcher, Schniter–arXiv:1602.08207

EM-VAMP

- Recall $\begin{cases} \text{prior } p(\boldsymbol{x};\boldsymbol{\theta}_x) \\ \text{likelihood } \ell(\boldsymbol{x};\tau_w) \end{cases} \rightarrow$ Learn parameters $\boldsymbol{\theta} \triangleq (\boldsymbol{\theta}_x, \tau_w)$.
- EM: iterate

  $$\begin{aligned} Q\big(\boldsymbol{\theta};\widehat{\boldsymbol{\theta}}^k\big) &= \textstyle\int p\big(\boldsymbol{x}|\boldsymbol{y};\widehat{\boldsymbol{\theta}}^k\big) \ln p(\boldsymbol{x},\boldsymbol{y};\boldsymbol{\theta}) \mathrm{d}\boldsymbol{x} && \text{“expectation”} \\ \widehat{\boldsymbol{\theta}}^{k+1} &= \arg\max_{\boldsymbol{\theta}} Q(\boldsymbol{\theta};\widehat{\boldsymbol{\theta}}^k) && \text{“maximization”} \end{aligned}$$

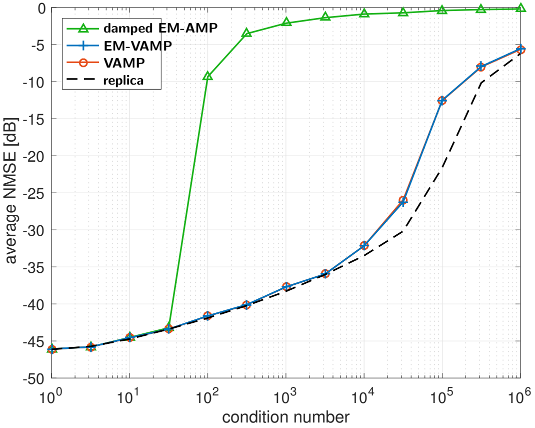
  which uses the posterior $p\big(\boldsymbol{x}|\boldsymbol{y};\widehat{\boldsymbol{\theta}}^k\big)$ in the E step.
- With VAMP's posterior approx, EM is an alternating approach to

  $$\min_{b_1,b_2,\boldsymbol{\theta}} \max_q D\big(b_1\big\|p(\boldsymbol{\theta}_x)\big) + D\big(b_2\big\|\ell(\tau_w)\big) + H(q)$$

  $$\text{s.t. } \begin{cases} \mathrm{E}\{\boldsymbol{x}|b_1\} = \mathrm{E}\{\boldsymbol{x}|b_2\} = \mathrm{E}\{\boldsymbol{x}|q\} \\ \mathrm{Tr}[\mathrm{Cov}\{\boldsymbol{x}|b_1\}] = \mathrm{Tr}[\mathrm{Cov}\{\boldsymbol{x}|b_2\}] = \mathrm{Tr}[\mathrm{Cov}\{\boldsymbol{x}|q\}] \end{cases}$$
- Can make faster by putting $\boldsymbol{\theta}$ optimization in the inner loop.

Experiment with Learned Parameters I

Learning both τ_w and $\boldsymbol{\theta}_x$:



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

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

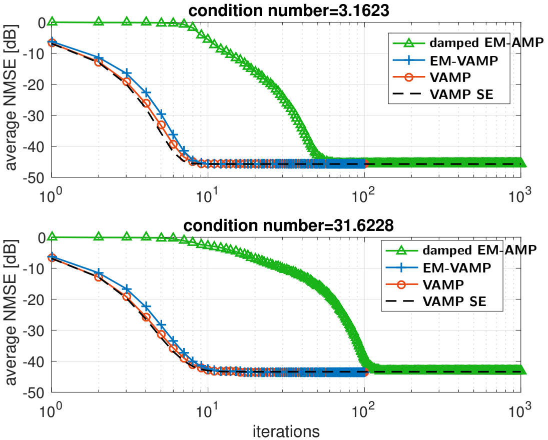
X_o ~Bernoulli-Gaussian
$\Pr\{X_0 \neq 0\} = 0.1$

SNR $= 40$dB

EM-VAMP achieves oracle performance at all condition numbers.

Experiment with Learned Parameters II

Learning both τ_w and $\boldsymbol{\theta}_x$:



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

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

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

SNR $= 40$dB

EM-VAMP nearly as fast as VAMP and much faster than EM-AMP.

Noiseless Image Recovery with BM3D

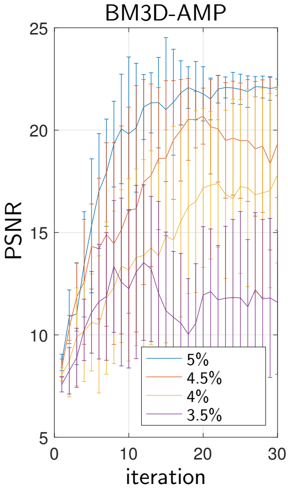
		PSNR	time
10%	L1-AMP	17.7 dB	0.5 s
	L1-VAMP	17.6 dB	0.5 s
	BM3D-AMP	**25.2 dB**	**10.1 s**
	BM3D-VAMP	**25.2 dB**	10.4 s
20%	L1-AMP	20.2 dB	1.0 s
	L1-VAMP	20.2 dB	0.9 s
	BM3D-AMP	**30.0 dB**	8.8 s
	BM3D-VAMP	**30.0 dB**	**8.5 s**
30%	L1-AMP	22.4 dB	1.6 s
	L1-VAMP	22.4 dB	1.4 s
	BM3D-AMP	**32.5 dB**	8.6 s
	BM3D-VAMP	**32.5 dB**	**8.2 s**
40%	L1-AMP	24.6 dB	2.3 s
	L1-VAMP	24.8 dB	1.8 s
	BM3D-AMP	35.1 dB	9.1 s
	BM3D-VAMP	**35.2 dB**	**8.5 s**
50%	L1-AMP	27.0 dB	3.1 s
	L1-VAMP	27.2 dB	2.3 s
	BM3D-AMP	37.4 dB	9.8 s
	BM3D-VAMP	**37.7 dB**	**8.8 s**

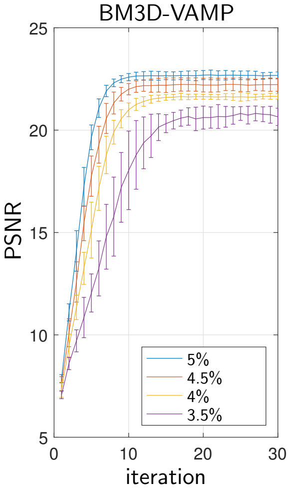
Avg results for recovering 128x128 lena, barbara, boat, fingerprint, house, and peppers from $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}_o$ with i.i.d. Gaussian $\boldsymbol{A}$ at various sampling ratios.

All algorithms use 20 iterations and learn the noise variance τ_w.

VAMP slightly outperforms AMP in accuracy and runtime.

Noiseless Image Recovery with BM3D (cont.)





Now look a sampling rates $\leq 5\%$.

Goal: recover 128x128 lena from $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}_o$ with i.i.d. Gaussian $\boldsymbol{A}$ and unknown τ_w.

BM3D-VAMP does much better than BM3D-AMP.

Generalized linear models

- Until now we have considered SLR, $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}_o + \boldsymbol{w}$.

- VAMP can also support the generalized linear model (GLM)

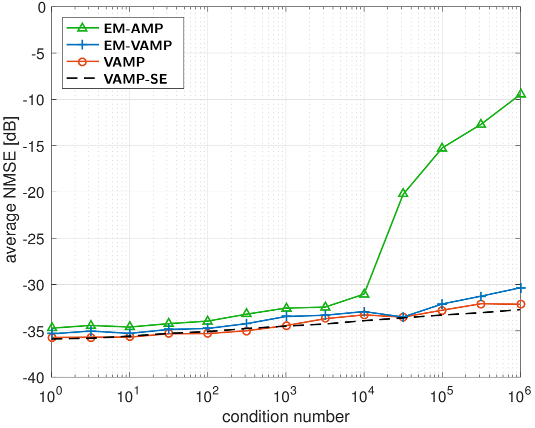
$$\boldsymbol{y} \sim p(\boldsymbol{y}|\boldsymbol{z}) \text{ with hidden } \boldsymbol{z} = \boldsymbol{A}\boldsymbol{x}_o$$

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

- Trick: $\boldsymbol{z} = \boldsymbol{A}\boldsymbol{x} \quad \Leftrightarrow \quad \underbrace{\mathbf{0}}_{\widetilde{\boldsymbol{z}}} = \underbrace{[\boldsymbol{A} \; -\boldsymbol{I}]}_{\widetilde{\boldsymbol{A}}} \underbrace{\begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{z} \end{bmatrix}}_{\widetilde{\boldsymbol{x}}}$

One-bit compressed sensing / Probit regression

Learning both τ_w and $\boldsymbol{\theta}_x$:



$N = 512$
$M/N = 4$

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

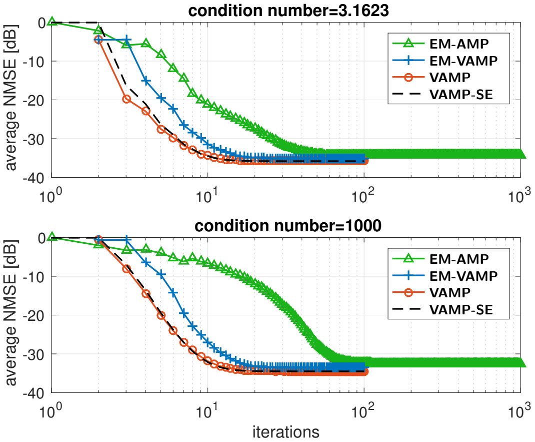
X_o ~Bernoulli-Gaussian
$\Pr\{X_0 \neq 0\} = 1/32$

SNR $= 40\text{dB}$

VAMP and EM-VAMP robust to ill-conditioned $\boldsymbol{A}$.

One-bit compressed sensing / Probit regression

Learning both τ_w and $\boldsymbol{\theta}_x$:



$N = 512$
$M/N = 4$

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

X_o ∼Bernoulli-Gaussian
$\Pr\{X_0 \neq 0\} = 1/32$

SNR $= 40$dB

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 $\boldsymbol{r}^t = \boldsymbol{x}_o + \mathcal{N}(\mathbf{0}, \tau_r^t \boldsymbol{I})$,
- 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 $\boldsymbol{A}$.

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

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

Thanks for listening!