Recent Advances in Approximate Message Passing

Phil Schniter

The Ohio State University

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

Supported in part by NSF grants IIP-1539960, and CCF-1716388.

London Symposium on Information Theory— May 30, 2019

Overview

1. Linear Regression and AMP
2. Vector AMP (VAMP)
3. VAMP for Optimization
4. Variational Interpretation and EM-VAMP
5. Plug-and-play VAMP
6. VAMP as a Deep Neural Network

Outline

The Linear Regression Problem

Consider the following linear regression problem:

Recover $\boldsymbol{x}_o$ from

$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}_o + \boldsymbol{w} \quad \text{with} \quad \begin{cases} \boldsymbol{x}_o \in \mathbb{R}^N & \text{unknown signal} \\ \boldsymbol{A} \in \mathbb{R}^{M \times N} & \text{known linear operator} \\ \boldsymbol{w} \in \mathbb{R}^M & \text{white Gaussian noise.} \end{cases}$$

Typical methodologies:

1. Optimization (or MAP estimation):

$$\widehat{\boldsymbol{x}} = \arg\min_{\boldsymbol{x}} \left\{ \frac{\theta_2}{2} \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}\|_2^2 + R(\boldsymbol{x}; \boldsymbol{\theta}_1) \right\}$$

2. Approximate MMSE:

$$\widehat{\boldsymbol{x}} \approx \mathrm{E}\{\boldsymbol{x}|\boldsymbol{y}\} \quad \text{for} \quad \boldsymbol{x} \sim p(\boldsymbol{x}; \boldsymbol{\theta}_1), \quad \boldsymbol{y}|\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{A}\boldsymbol{x}, \boldsymbol{I}/\theta_2)$$

3. Plug-and-play: iteratively apply a denoising algorithm like BM3D

4. Train a deep network to recover $\boldsymbol{x}_o$ from $\boldsymbol{y}$.

The AMP Methodology

- All of the aforementioned methodologies can be addressed using the Approximate Message Passing (AMP) framework.[1]
- AMP tackles these problems via iterative denoising.
- Each method defines the denoiser $\boldsymbol{g}(\cdot;\gamma,\boldsymbol{\theta}_1) : \mathbb{R}^N \to \mathbb{R}^N$ differently:
  - Optimization: $\boldsymbol{g}(\boldsymbol{r};\gamma,\boldsymbol{\theta}_1) = \arg\min_{\boldsymbol{x}}\{R(\boldsymbol{x};\boldsymbol{\theta}_1) + \frac{\gamma}{2}\|\boldsymbol{x}-\boldsymbol{r}\|_2^2\} \triangleq$ "$\text{prox}_{R/\gamma}(\boldsymbol{r})$"
  - MMSE: $\boldsymbol{g}(\boldsymbol{r};\gamma,\boldsymbol{\theta}_1) = \mathrm{E}\left\{\boldsymbol{x} \,\middle|\, \boldsymbol{r} = \boldsymbol{x} + \mathcal{N}(\boldsymbol{0},\boldsymbol{I}/\gamma)\right\}$
  - Plug-and-play:[2] $\boldsymbol{g}(\boldsymbol{r};\gamma,\boldsymbol{\theta}_1) = \text{BM3D}(\boldsymbol{r}, 1/\gamma)$
  - Deep network: $\boldsymbol{g}(\boldsymbol{r};\gamma,\boldsymbol{\theta}_1)$ is learned from training data.

[1] Donoho,Maleki,Montanari'09, [2] Venkatakrishnan,Bouman,Wohlberg'13

The Original AMP Algorithm

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

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

$\boldsymbol{v}^t = \boldsymbol{y} - \boldsymbol{A}\widehat{\boldsymbol{x}}^t + \frac{N}{M}\boldsymbol{v}^{t-1}\langle \boldsymbol{g}'(\widehat{\boldsymbol{x}}^{t-1} + A^\mathsf{T}\widehat{\boldsymbol{v}}^{t-1}, \gamma^{t-1})\rangle$ corrected residual

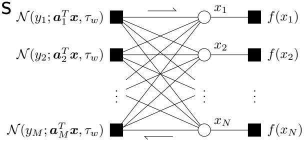
$\widehat{\boldsymbol{x}}^{t+1} = \boldsymbol{g}(\widehat{\boldsymbol{x}}^t + \boldsymbol{A}^\mathsf{T}\boldsymbol{v}^t; \gamma^t)$ denoising

where

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

Note:

- Can be recognized as iterative thresholding plus “Onsager correction.”
- Can be derived using Gaussian & Taylor-series approximations of belief-propagation.



AMP's Denoising Property

Assumption 1

- $\boldsymbol{A} \in \mathbb{R}^{M \times N}$ is i.i.d. sub-Gaussian
- $M, N \to \infty$ s.t. $\frac{M}{N} \to \delta \in (0, \infty)$ ... "large-system limit"
- $[\boldsymbol{g}(\boldsymbol{r})]_j = g(r_j)$ with Lipschitz $g(\cdot)$ ... "separable denoising"

Under Assumption 1, the elements of the denoiser's input $\boldsymbol{r}^t \triangleq \widehat{\boldsymbol{x}}^t + \boldsymbol{A}^\mathsf{T} \boldsymbol{v}^t$ obey[3,4]

$$\boxed{r_j^t = x_{o,j} + \mathcal{N}(0, \tau_r^t)}$$

- That is, $\boldsymbol{r}^t$ is a Gaussian-noise corrupted version of the true signal $\boldsymbol{x}_o$.
- It is now clear why $\boldsymbol{g}(\cdot)$ is called a "denoiser."

Furthermore, the noise variance can be consistently estimated:

$$\widehat{\tau}_r^t \triangleq \frac{1}{M} \|\boldsymbol{v}^t\|^2 \quad \longrightarrow \quad \tau_r^t \quad \text{under Assumption 1.}$$

[3] Bayati,Montanari'11, [4] Bayati,Lelarge,Montanari'15

AMP's State Evolution

- Assume that the measurements $\boldsymbol{y}$ were generated via

$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}_o + \mathcal{N}(\boldsymbol{0}, \tau_w \boldsymbol{I})$$

  where $\boldsymbol{x}_o$ empirically converges to some random variable X_o as $N \to \infty$.

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

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

- Then, under Assumption 1, AMP obeys the following scalar state evolution:

$$\boxed{\begin{aligned} &\text{for } t = 0, 1, 2, \ldots \\ &\qquad \tau_r^t = \tau_w + \tfrac{N}{M}\mathcal{E}^t \\ &\qquad \mathcal{E}^{t+1} = \mathrm{E}\left\{\left[g(X_o + \mathcal{N}(0, \tau_r^t); \gamma^t) - X_o\right]^2\right\} \end{aligned}}$$

Bayes Optimality of AMP

- Now suppose that Assumption 1 holds, and that
$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}_o + \mathcal{N}(\boldsymbol{0}, \tau_w \boldsymbol{I}),$$
where the elements of $\boldsymbol{x}_o$ are i.i.d. draws of some random variable X_o.

- Suppose also that $g(\cdot)$ is the MMSE denoiser, i.e.,
$$g(R; \gamma^t) = \mathrm{E}\left\{X_o \,\middle|\, R = X_o + \mathcal{N}(0, 1/\gamma^t)\right\} \text{ with } \gamma^t = 1/\tau_r^t.$$

- Then, if the state evolution has a unique fixed point, $\widehat{\boldsymbol{x}}^t$ converges to the MMSE estimate[5] of $\boldsymbol{x}_o$ as $t \to \infty$.

[5] Bayati,Montanari'11

AMP: The good, the bad, and the ugly

The good:

- With large[6] i.i.d. sub-Gaussian $\boldsymbol{A}$, AMP is rigorously characterized by a scalar state-evolution whose fixed points, when unique, are Bayes optimal.
- Empirically, AMP behaves well with many other "sufficiently random" $\boldsymbol{A}$ (e.g., randomly sub-sampled Fourier $\boldsymbol{A}$ & i.i.d. sparse $\boldsymbol{x}$).

The bad:

- With general $\boldsymbol{A}$, AMP gives no guarantees.

The ugly:

- With some $\boldsymbol{A}$, AMP may fail to converge! (e.g., ill-conditioned or non-zero-mean $\boldsymbol{A}$)

[6] Rush,Venkataramanan'16

Outline

Vector AMP (VAMP) 

- VAMP is similar to AMP, but it supports a larger class of random matrices.

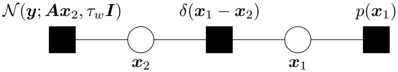
- As before, the goal is to recover $\boldsymbol{x}_o$ from $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}_o + \mathcal{N}(\boldsymbol{0}, \tau_w \boldsymbol{I})$.

- VAMP yields a precise analysis for right-orthogonally invariant $\boldsymbol{A}$:

$$\text{svd}(\boldsymbol{A}) = \boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^\mathsf{T} \quad \text{for} \quad \begin{cases} \boldsymbol{U}\text{: deterministic orthogonal} \\ \boldsymbol{S}\text{: deterministic diagonal} \\ \boldsymbol{V}\text{: ``Haar;'' uniform on set of orthogonal matrices} \end{cases}$$

  of which i.i.d. Gaussian is a special case.

- Can be derived as a form of expectation propagation (EP).



VAMP: The Algorithm

Take SVD $\boldsymbol{A} = \boldsymbol{U}\,\mathrm{Diag}(\boldsymbol{s})\boldsymbol{V}^\mathsf{T}$, choose $\zeta \in (0,1]$ and Lipschitz $\boldsymbol{g}_1(\cdot;\gamma_1,\boldsymbol{\theta}_1) : \mathbb{R}^N \to \mathbb{R}^N$.

Initialize $\boldsymbol{r}_1, \gamma_1$.

For $k = 1, 2, 3, \ldots$

$\widehat{\boldsymbol{x}}_1 \leftarrow \boldsymbol{g}_1(\boldsymbol{r}_1;\gamma_1,\boldsymbol{\theta}_1)$ — denoising of $\boldsymbol{r}_1 = \boldsymbol{x}_o + \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}/\gamma_1)$

$$\eta_1 \leftarrow \gamma_1 N / \operatorname{tr}\left[\frac{\partial \boldsymbol{g}_1(\boldsymbol{r}_1;\gamma_1,\boldsymbol{\theta}_1)}{\partial \boldsymbol{r}_1}\right]$$

$\boldsymbol{r}_2 \leftarrow (\eta_1 \widehat{\boldsymbol{x}}_1 - \gamma_1 \boldsymbol{r}_1)/(\eta_1 - \gamma_1)$ — Onsager correction

$\gamma_2 \leftarrow \eta_1 - \gamma_1$

$\widehat{\boldsymbol{x}}_2 \leftarrow \boldsymbol{g}_2(\boldsymbol{r}_2;\gamma_2,\theta_2)$ — LMMSE estimate $\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{r}_2, \boldsymbol{I}/\gamma_2)$ from $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}/\theta_2)$

$$\eta_2 \leftarrow \gamma_2 N / \operatorname{tr}\left[\frac{\partial \boldsymbol{g}_2(\boldsymbol{r}_2;\gamma_2,\theta_2)}{\partial \boldsymbol{r}_2}\right]$$

$\boldsymbol{r}_1 \leftarrow \zeta(\eta_2 \widehat{\boldsymbol{x}}_2 - \gamma_2 \boldsymbol{r}_2)/(\eta_2 - \gamma_2) + (1-\zeta)\boldsymbol{r}_1$ — Onsager correction

$\gamma_1 \leftarrow \zeta(\eta_2 - \gamma_2) + (1-\zeta)\gamma_1$ — damping

where
$$\begin{aligned} \boldsymbol{g}_2(\boldsymbol{r}_2;\gamma_2,\theta_2) &= \boldsymbol{V}\big(\theta_2\,\mathrm{Diag}(\boldsymbol{s})^2 + \gamma_2 \boldsymbol{I}\big)^{-1}\big(\theta_2\,\mathrm{Diag}(\boldsymbol{s})\boldsymbol{U}^\mathsf{T}\boldsymbol{y} + \gamma_2 \boldsymbol{V}^\mathsf{T}\boldsymbol{r}_2\big) \\ \eta_2 &= \tfrac{1}{N}\textstyle\sum_{n=1}^N (\theta_2 s_n^2 + \gamma_2)^{-1} \end{aligned}$$

two mat-vec mults per iteration!

VAMP's Denoising Property

Assumption 2

- $\boldsymbol{A} \in \mathbb{R}^{M \times N}$ is right-orthogonally invariant
- $M, N \to \infty$ s.t. $\frac{M}{N} \to \delta \in (0, \infty)$ . . . "large-system limit"
- $[\boldsymbol{g}_1(\boldsymbol{r})]_j = g_1(r_j)$ with Lipschitz $g_1(\cdot)$ . . . "separable denoising"

Under Assumption 2, the elements of the denoiser's input $\boldsymbol{r}_1^t$ obey[7]

$$\boxed{r_{1,j}^t = x_{o,j} + \mathcal{N}(0, \tau_1^t)}$$

- That is, $\boldsymbol{r}_1^t$ is a Gaussian-noise corrupted version of the true signal $\boldsymbol{x}_o$.
- Here too, we can interpret $\boldsymbol{g}_1(\cdot)$ as a "denoiser."

[7] Rangan,Schniter,Fletcher'16

VAMP's State Evolution

Assume empirical convergence of $\{s_j\} \to S$ and $\{(r_{1,j}^0, x_{o,j})\} \to (R_1^0, X_o)$, and define

$$\mathcal{E}_i^t \triangleq \tfrac{1}{N} \mathrm{E}\left\{\|\widehat{\boldsymbol{x}}_i^t - \boldsymbol{x}_o\|^2\right\} \text{ for } i = 1, 2.$$

Then under Assumption 2, the VAMP obeys the following state-evolution:

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

$$\mathcal{E}_1^t = \mathrm{E}\left\{\left[g\big(X_o + \mathcal{N}(0, \tau_1^t); \gamma_1^t\big) - X_o\right]^2\right\} \qquad \text{MSE}$$

$$\alpha_1^t = \mathrm{E}\left\{g'(X_o + \mathcal{N}(0, \tau_1^t); \gamma_1^t)\right\} \qquad \text{divergence}$$

$$\gamma_2^t = \gamma_1^t \tfrac{1-\alpha_1^t}{\alpha_1^t}, \quad \tau_2^t = \tfrac{1}{(1-\alpha_1^t)^2}\left[\mathcal{E}_1^t - \left(\alpha_1^t\right)^2 \tau_1^t\right]$$

$$\mathcal{E}_2^t = \mathrm{E}\left\{\left[S^2/\tau_w + \gamma_2^t\right]^{-1}\right\} \qquad \text{MSE}$$

$$\alpha_2^t = \gamma_2^t \,\mathrm{E}\left\{\left[S^2/\tau_w + \gamma_2^t\right]^{-1}\right\} \qquad \text{divergence}$$

$$\gamma_1^{t+1} = \gamma_2^t \tfrac{1-\alpha_2^t}{\alpha_2^t}, \quad \tau_1^{t+1} = \tfrac{1}{(1-\alpha_2^t)^2}\left[\mathcal{E}_2^t - \left(\alpha_2^t\right)^2 \tau_2^t\right]$$

Note: Above assumes $\boldsymbol{g}_2(\cdot)$ uses matched noise variance $\theta_2 = 1/\tau_w$.
If not, there are more complicated expressions for $\mathcal{E}_2^t$ and α_2^t.

Bayes Optimality of VAMP

- Now suppose that Assumption 2 holds, and that

$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}_o + \mathcal{N}(\boldsymbol{0}, \tau_w \boldsymbol{I}),$$

  where the elements of $\boldsymbol{x}_o$ are i.i.d. draws of some random variable X_o.

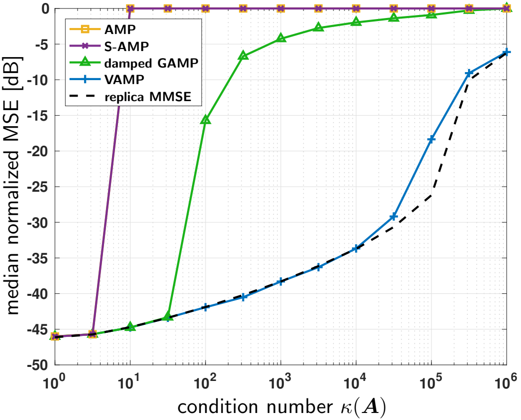
- Suppose also that $g_1(\cdot)$ is the MMSE denoiser, i.e.,

$$g_1(R_1; \gamma_1^t) = \mathrm{E}\left\{X_o \,\middle|\, R_1 = X_o + \mathcal{N}(0, 1/\gamma_1^t)\right\} \text{ with } \gamma_1^t = 1/\tau_1^t.$$

- Then, if the state evolution has a unique fixed point, the MSE of $\widehat{\boldsymbol{x}}_1^t$ converges to the replica prediction of the MMSE as $t \to \infty$.
  - For right-orthogonally invariant $\boldsymbol{A}$, the replica prediction was derived by Tulino/Caire/Verdu/Shamai in 2013. It is conjectured to be correct.
  - For the special case of i.i.d. Gaussian $\boldsymbol{A}$, it was proven to be correct by Reeves/Pfister, and by Barbier/Dia/Macris/Krzakala, both in 2016.

Experiment with MMSE Denoising

Comparison of several algorithms[8] with MMSE denoising.



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

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

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

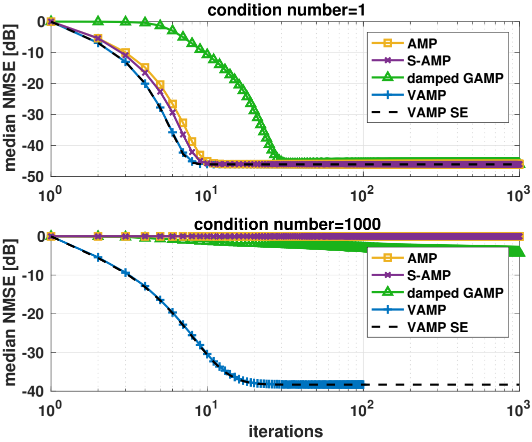
SNR $= 40$dB

VAMP achieves MMSE over a wide range of condition numbers.

[8] S-AMP: Cakmak,Fleury,Winther'14, damped GAMP: Vila,Schniter,Rangan,Krzakala,Zdeborová'15

Experiment with MMSE Denoising

Comparison of several algorithms with priors matched to data.



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

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

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

SNR$=$ 40dB

VAMP is fast even when $\boldsymbol{A}$ is ill-conditioned.

Outline

VAMP for Optimization

- Consider the optimization problem

$$\widehat{\boldsymbol{x}} = \arg\min_{\boldsymbol{x}} \left\{ \tfrac{\theta_2}{2} \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}\|^2 + R(\boldsymbol{x}; \boldsymbol{\theta}_1) \right\}$$

where $R(\cdot)$ is strictly convex.

- If we choose the denoiser

$$\boldsymbol{g}_1(\boldsymbol{r}; \gamma, \boldsymbol{\theta}_1) = \arg\min_{\boldsymbol{x}} \left\{ R(\boldsymbol{x}; \boldsymbol{\theta}_1) + \frac{\gamma}{2} \|\boldsymbol{x} - \boldsymbol{r}\|^2 \right\} = \mathsf{prox}_{R/\gamma}(\boldsymbol{r})$$

and the damping parameter

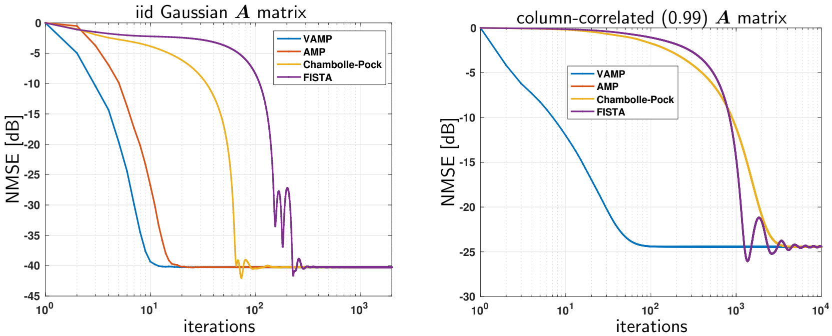
$$\zeta \leq \frac{2\min\{\gamma_1, \gamma_2\}}{\gamma_1 + \gamma_2},$$

then a double-loop version of VAMP converges[9] to the solution for *any* $\boldsymbol{A}$.

- Furthermore, if the γ_1 and γ_2 variables are fixed over the iterations, then VAMP reduces to the Peaceman-Rachford variant of ADMM.

[9] Fletcher, Sahraee, Rangan, Schniter'16

Example of VAMP applied to the LASSO Problem



Solving LASSO to reconstruct 40-sparse $\boldsymbol{x} \in \mathbb{R}^{1000}$ from noisy $\boldsymbol{y} \in \mathbb{R}^{400}$.

$$\widehat{\boldsymbol{x}} = \arg\min_{\boldsymbol{x}} \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|_2^2 + \lambda\|\boldsymbol{x}\|_1.$$

Outline

Interpretation as Variational Inference

- Ideally, we would like to compute the exact posterior density

$$p(\boldsymbol{x}|\boldsymbol{y}) = \frac{p(\boldsymbol{x};\boldsymbol{\theta}_1)\ell(\boldsymbol{x};\theta_2)}{Z(\boldsymbol{\theta})} \quad \text{for} \quad Z(\boldsymbol{\theta}) \triangleq \int p(\boldsymbol{x};\boldsymbol{\theta}_1)\ell(\boldsymbol{x};\theta_2)\,\mathrm{d}\boldsymbol{x},$$

but the high-dimensional integral in $Z(\boldsymbol{\theta})$ is difficult to compute.

- We might try to circumvent $Z(\boldsymbol{\theta})$ through variational optimization:

$$\begin{aligned}
p(\boldsymbol{x}|\boldsymbol{y}) &= \arg\min_b D\big(b(\boldsymbol{x})\big\|p(\boldsymbol{x}|\boldsymbol{y})\big) \text{ where } D(\cdot\|\cdot) \text{ is KL divergence} \\
&= \arg\min_b \underbrace{D\big(b(\boldsymbol{x})\big\|p(\boldsymbol{x};\boldsymbol{\theta}_1)\big) + D\big(b(\boldsymbol{x})\big\|\ell(\boldsymbol{x};\theta_2)\big) + H\big(b(\boldsymbol{x})\big)}_{\text{Gibbs free energy}} \\
&= \arg\min_{b_1,b_2,q} \underbrace{D\big(b_1(\boldsymbol{x})\big\|p(\boldsymbol{x};\boldsymbol{\theta}_1)\big) + D\big(b_2(\boldsymbol{x})\big\|\ell(\boldsymbol{x};\theta_2)\big) + H\big(q(\boldsymbol{x})\big)}_{\triangleq J_{\text{Gibbs}}(b_1,b_2,q;\boldsymbol{\theta})} \\
&\quad \text{s.t. } b_1 = b_2 = q,
\end{aligned}$$

but the density constraint keeps the problem difficult.

Expectation Consistent Approximation

- In expectation-consistent approximation (EC)[10], the density constraint is relaxed to moment-matching constraints:

$$p(\boldsymbol{x}|\boldsymbol{y}) \approx \arg\min_{b_1,b_2,q} J_{\mathsf{Gibbs}}(b_1,b_2,q;\boldsymbol{\theta})$$
$$\text{s.t. } \begin{cases} \mathrm{E}\{\boldsymbol{x}|b_1\} = \mathrm{E}\{\boldsymbol{x}|b_2\} = \mathrm{E}\{\boldsymbol{x}|q\} \\ \mathrm{tr}(\mathrm{Cov}\{\boldsymbol{x}|b_1\}) = \mathrm{tr}(\mathrm{Cov}\{\boldsymbol{x}|b_2\}) = \mathrm{tr}(\mathrm{Cov}\{\boldsymbol{x}|q\}). \end{cases}$$

- The stationary points of EC are the densities

$$\begin{aligned} b_1(\boldsymbol{x}) &\propto p(\boldsymbol{x};\boldsymbol{\theta}_1)\mathcal{N}(\boldsymbol{x};\boldsymbol{r}_1,\boldsymbol{I}/\gamma_1) \\ b_2(\boldsymbol{x}) &\propto \ell(\boldsymbol{x};\theta_2)\mathcal{N}(\boldsymbol{x};\boldsymbol{r}_2,\boldsymbol{I}/\gamma_2) \\ q(\boldsymbol{x}) &= \mathcal{N}(\boldsymbol{x};\widehat{\boldsymbol{x}},\boldsymbol{I}/\eta) \end{aligned} \quad \text{s.t. } \begin{cases} \mathrm{E}\{\boldsymbol{x}|b_1\} = \mathrm{E}\{\boldsymbol{x}|b_2\} = \widehat{\boldsymbol{x}} \\ \mathrm{tr}(\mathrm{Cov}\{\boldsymbol{x}|b_1\}) = \mathrm{tr}(\mathrm{Cov}\{\boldsymbol{x}|b_2\}) = N/\eta, \end{cases}$$

- VAMP iteratively solves for the quantities $\boldsymbol{r}_1, \gamma_1, \boldsymbol{r}_2, \gamma_2, \widehat{\boldsymbol{x}}, \eta$ above.
  - In this setting, VAMP is simply an instance of expectation propagation (EP).

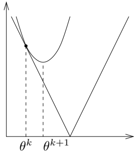
[10] Opper,Winther'04,

Expectation Maximization

- What if the hyperparameters $\boldsymbol{\theta}$ of the prior & likelihood are unknown?.
- The EM algorithm[11] is majorization-minimization approach to ML estimation that iteratively minimizes a tight upper bound on $-\ln p(\boldsymbol{y}|\boldsymbol{\theta})$:

$$\widehat{\boldsymbol{\theta}}^{k+1} = \arg\min_{\boldsymbol{\theta}} \Big\{ -\ln p(\boldsymbol{y}|\boldsymbol{\theta}) + \underbrace{D\big(b^k(\boldsymbol{x}) \big\| p(\boldsymbol{x}|\boldsymbol{y};\boldsymbol{\theta})\big)}_{\geq 0} \Big\}$$

$$\text{with} \quad b^k(\boldsymbol{x}) = p(\boldsymbol{x}|\boldsymbol{y};\widehat{\boldsymbol{\theta}}^k)$$



- EM can also be written in terms of the Gibbs free energy:[12]

$$\widehat{\boldsymbol{\theta}}^{k+1} = \arg\min_{\boldsymbol{\theta}} \underbrace{D\big(b^k(\boldsymbol{x}) \big\| p(\boldsymbol{x};\boldsymbol{\theta}_1)\big) + D\big(b^k(\boldsymbol{x}) \big\| \ell(\boldsymbol{x};\theta_2)\big) + H\big(b^k(\boldsymbol{x})\big)}_{J_{\mathsf{Gibbs}}(b^k, b^k, b^k; \boldsymbol{\theta})}$$

- Thus, we can interleave EM and VAMP to solve

$$\min_{\boldsymbol{\theta}} \min_{b_1, b_2, q} J_{\mathsf{Gibbs}}(b_1, b_2, q; \boldsymbol{\theta}) \text{ s.t. } \begin{cases} \mathrm{E}\{\boldsymbol{x}|b_1\} = \mathrm{E}\{\boldsymbol{x}|b_2\} = \mathrm{E}\{\boldsymbol{x}|q\} \\ \mathrm{tr}[\mathrm{Cov}\{\boldsymbol{x}|b_1\}] = \mathrm{tr}[\mathrm{Cov}\{\boldsymbol{x}|b_2\}] = \mathrm{tr}[\mathrm{Cov}\{\boldsymbol{x}|q\}]. \end{cases}$$

[11] Dempster,Laird,Rubin'77, [12] Neal,Hinton'98

The EM-VAMP Algorithm

Input conditional-mean $\boldsymbol{g}_1(\cdot)$ and $\boldsymbol{g}_2(\cdot)$, and initialize $\boldsymbol{r}_1, \gamma_1, \widehat{\boldsymbol{\theta}}_1, \widehat{\theta}_2$.

For $k = 1, 2, 3, \ldots$

$$\widehat{\boldsymbol{x}}_1 \leftarrow \boldsymbol{g}_1(\boldsymbol{r}_1; \gamma_1, \widehat{\boldsymbol{\theta}}_1) \qquad \text{MMSE estimation}$$

$$\eta_1 \leftarrow \gamma_1 N / \operatorname{tr}\left[\partial \boldsymbol{g}_1(\boldsymbol{r}_1; \gamma_1, \widehat{\boldsymbol{\theta}}_1)/\partial \boldsymbol{r}_1\right]$$

$$\boldsymbol{r}_2 \leftarrow (\eta_1 \widehat{\boldsymbol{x}}_1 - \gamma_1 \boldsymbol{r}_1)/(\eta_1 - \gamma_1)$$

$$\gamma_2 \leftarrow \eta_1 - \gamma_1$$

$$\widehat{\theta}_2 \leftarrow \arg\max_{\theta_2} \mathrm{E}\{\ln \ell(\boldsymbol{x}; \theta_2) \,|\, \boldsymbol{r}_2; \gamma_2, \widehat{\theta}_2\} \qquad \text{EM update}$$

$$\widehat{\boldsymbol{x}}_2 \leftarrow \boldsymbol{g}_2(\boldsymbol{r}_2; \gamma_2, \widehat{\theta}_2) \qquad \text{LMMSE estimation}$$

$$\eta_2 \leftarrow \gamma_2 N / \operatorname{tr}\left[\partial \boldsymbol{g}_2(\boldsymbol{r}_2; \gamma_2, \widehat{\theta}_2)/\partial \boldsymbol{r}_2\right]$$

$$\boldsymbol{r}_1 \leftarrow \zeta(\eta_2 \widehat{\boldsymbol{x}}_2 - \gamma_2 \boldsymbol{r}_2)/(\eta_2 - \gamma_2) + (1-\zeta)\boldsymbol{r}_1$$

$$\gamma_1 \leftarrow \zeta(\eta_2 - \gamma_2) + (1 - \zeta)\gamma_1$$

$$\widehat{\boldsymbol{\theta}}_1 \leftarrow \arg\max_{\boldsymbol{\theta}_1} \mathrm{E}\{\ln p(\boldsymbol{x}; \boldsymbol{\theta}_1) \,|\, \boldsymbol{r}_1; \gamma_1, \widehat{\boldsymbol{\theta}}_1\} \qquad \text{EM update}$$

State Evolution and Consistency

- EM-VAMP has a rigorous state-evolution[13] when the prior is i.i.d. and $\boldsymbol{A}$ is large and right-orthogonally invariant.
- Furthermore, a variant known as "adaptive VAMP" can be shown to yield consistent parameter estimates with an i.i.d. prior in the exponential-family or with finite-cardinality $\boldsymbol{\theta}_1$.[13]
- Essentially, adaptive VAMP replaces the EM update

$$\widehat{\boldsymbol{\theta}}_1 \leftarrow \arg\max_{\boldsymbol{\theta}_1} \mathrm{E}\{\ln p(\boldsymbol{x};\boldsymbol{\theta}_1) \,|\, \boldsymbol{r}_1, \gamma_1, \widehat{\boldsymbol{\theta}}_1\}$$

with

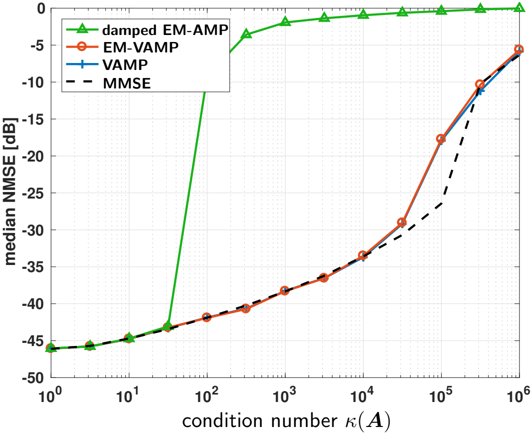
$$(\widehat{\boldsymbol{\theta}}_1, \widehat{\gamma}_1) \leftarrow \arg\max_{(\boldsymbol{\theta}_1,\gamma_1)} \mathrm{E}\{\ln p(\boldsymbol{x};\boldsymbol{\theta}_1) \,|\, \boldsymbol{r}_1, \gamma_1, \widehat{\boldsymbol{\theta}}_1\},$$

which re-estimates the precision γ_1. (And similar for θ_2, γ_2.)

[13] Fletcher,Rangan,Schniter'17

Experiment with Unknown Hyperparameters $\boldsymbol{\theta}$

Learning both noise precision θ_2 and BG mean/variance/sparsity $\boldsymbol{\theta}_1$:



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

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

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

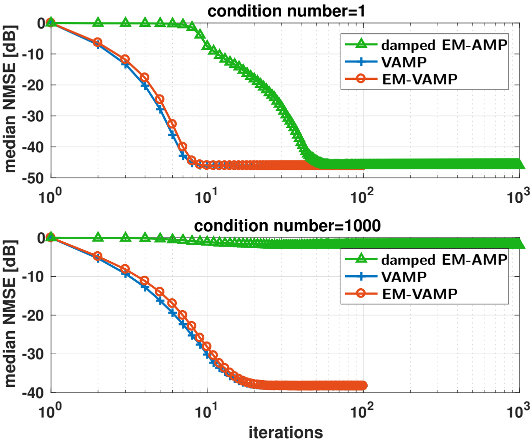
SNR $= 40$dB

EM-VAMP achieves oracle performance at all condition numbers![14]

[14] EM-AMP proposed in Vila,Schniter'11 and Krzakala,Mézard,Sausset,Sun,Zdeborová'12

Experiment with Unknown Hyperparameters $\boldsymbol{\theta}$

Learning both noise precision θ_2 and BG mean/variance/sparsity $\boldsymbol{\theta}_1$:



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

$\boldsymbol{A} = \boldsymbol{U}\,\mathrm{Diag}(\boldsymbol{s})\boldsymbol{V}^{\mathsf{T}}$
$\boldsymbol{U}, \boldsymbol{V} \sim$ Haar
$s_n/s_{n-1} = \phi\ \forall n$
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 damped EM-GAMP.

Outline

Plug-and-play VAMP

- Recall the scalar denoising step of VAMP (or AMP):

$$\widehat{\boldsymbol{x}}_1 \leftarrow \boldsymbol{g}_1(\boldsymbol{r}_1;\gamma_1) \quad \text{where } \boldsymbol{r}_1 = \boldsymbol{x}_o + \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}/\gamma_1)$$

- For certain signal classes (e.g., images), very sophisticated *non-scalar* denoising procedures have been developed (e.g., BM3D, DnCNN).

- Such denoising procedures can be "plugged into" signal recovery algorithms like ADMM, AMP[15], VAMP. Divergence can be approximated via

$$\frac{1}{N}\operatorname{tr}\left[\frac{\partial \boldsymbol{g}_1}{\partial \boldsymbol{r}_1}\right] \approx \frac{1}{K}\sum_{k=1}^{K} \frac{\boldsymbol{p}_k^{\mathsf{T}}\left[\boldsymbol{g}_1(\boldsymbol{r}+\epsilon\boldsymbol{p}_k,\gamma_1) - \boldsymbol{g}_1(\boldsymbol{r},\gamma_1)\right]}{N\epsilon}$$

  with random vectors $\boldsymbol{p}_k \in \{\pm 1\}^N$ and small $\epsilon > 0$. Empirically, $K=1$ suffices.

- Rigoruous state-evolutions established for plug-and-play AMP[16] and VAMP.[17]

[15] Metzler,Maleki,Baraniuk'14, [16] Berthier,Montanari,Nguyen'17, [17] Fletcher,Rangan,Sarkar,Schniter'18

Bilinear estimation via Lifting

- As we now describe, non-scalar denoising facilitates bilinear recovery.
- Say the goal is to recover $\boldsymbol{b} = [b_1, \ldots, b_L]^{\mathsf{T}}$ and $\boldsymbol{c}$ from measurements

$$\boldsymbol{y} = \left(\sum_{l=1}^{L} b_l \boldsymbol{\Phi}_l \right) \boldsymbol{c} + \boldsymbol{w}$$

  where $\{\boldsymbol{\Phi}_l\}$ are known. This arises in calibration problems.

- We can "lift"[18] this bilinear problem to the linear problem

$$\boldsymbol{y} = \underbrace{[\boldsymbol{\Phi}_1 \quad \boldsymbol{\Phi}_2 \quad \cdots \quad \boldsymbol{\Phi}_L]}_{\boldsymbol{A}} \underbrace{\mathrm{vec}(\boldsymbol{c}\boldsymbol{b}^{\mathsf{T}})}_{\boldsymbol{x}} + \boldsymbol{w}$$

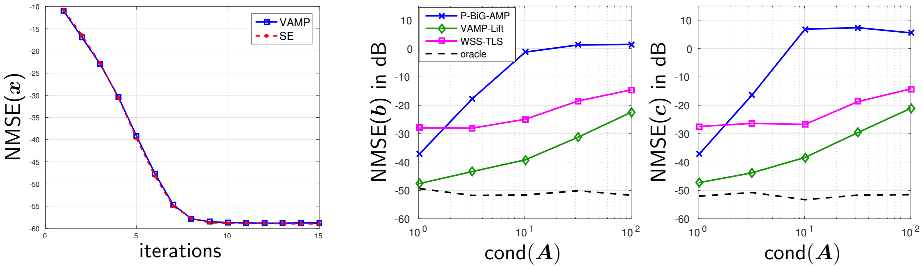
  and apply VAMP with an appropriate denoiser.

[18] Candes,Strohmer,Voroninski'13, Ahmed,Recht,Romberg'14

Experiment: Compressed Sensing with Matrix Uncertainty

Goal: Recover[19] $\boldsymbol{b}$ and sparse $\boldsymbol{c}$ from $\boldsymbol{y} = \left(\sum_{l=1}^{L} b_l \boldsymbol{\Phi}_l\right)\boldsymbol{c} + \boldsymbol{w} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{w}$.

State Evolution: NMSE versus condition number of $\boldsymbol{A}$:

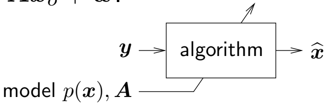


[19] WSS-TLS is from Zhu,Leus,Giannakis'11, P-BiG-AMP is from Parker,Schniter'16.

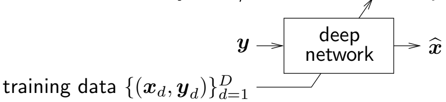
Outline

Deep learning for sparse reconstruction

- Until now we've focused on designing algorithms to recover $\boldsymbol{x}_o \sim p(\boldsymbol{x})$ from measurements $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}_o + \boldsymbol{w}$.



- What about training deep networks to predict $\boldsymbol{x}_o$ from $\boldsymbol{y}$?
  Can we increase accuracy and/or decrease computation?



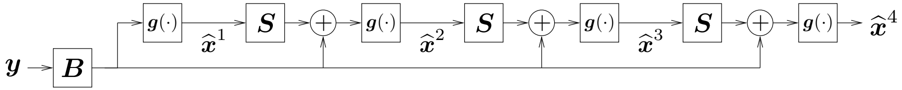
- Are there connections between these approaches?

Unfolding Algorithms into Networks

Consider, e.g., the classical sparse-reconstruction algorithm, ISTA.[20]

$$\boxed{\begin{aligned} \boldsymbol{v}^t &= \boldsymbol{y} - \boldsymbol{A}\widehat{\boldsymbol{x}}^t \\ \widehat{\boldsymbol{x}}^{t+1} &= \boldsymbol{g}(\widehat{\boldsymbol{x}}^t + \boldsymbol{A}^\mathsf{T}\boldsymbol{v}^t) \end{aligned}} \quad \Leftrightarrow \quad \boxed{\widehat{\boldsymbol{x}}^{t+1} = \boldsymbol{g}(\boldsymbol{S}\widehat{\boldsymbol{x}}^t + \boldsymbol{B}\boldsymbol{y}) \text{ with } \begin{array}{l} \boldsymbol{S} \triangleq \boldsymbol{I} - \boldsymbol{A}^\mathsf{T}\boldsymbol{A} \\ \boldsymbol{B} \triangleq \boldsymbol{A}^\mathsf{T} \end{array}}$$

Gregor & LeCun[21] proposed to "unfold" it into a deep net and "learn" improved parameters using training data, yielding "learned ISTA" (LISTA):

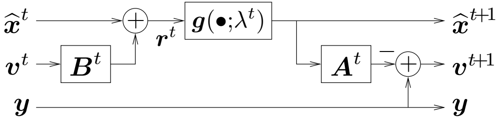


The same "unfolding & learning" idea can be used to improve AMP, yielding "learned AMP" (LAMP).[22]

[20] Daubechies,Defrise,DeMol'04. [21] Gregor,LeCun'10. [22] Borgerding,Schniter'16.

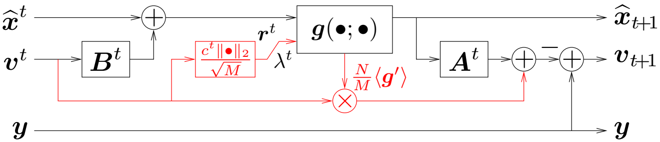
Onsager-Corrected Deep Networks

t^{th} LISTA layer:



to exploit low-rank $\boldsymbol{B}^t\boldsymbol{A}^t$ in linear stage $\boldsymbol{S}^t = \boldsymbol{I} - \boldsymbol{B}^t\boldsymbol{A}^t$.

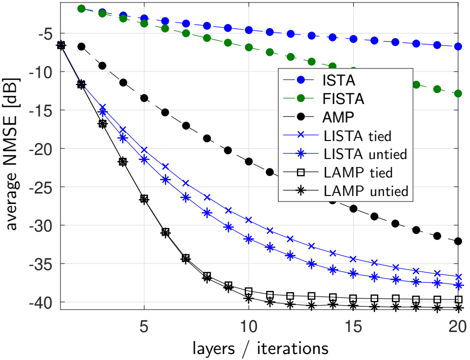
t^{th} LAMP layer:

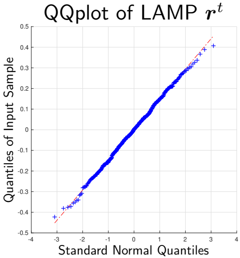


Onsager correction now aims to decouple errors across layers.

LAMP performance with soft-threshold denoising

LISTA beats AMP,FISTA,ISTA
LAMP beats LISTA
in convergence speed and asymptotic MSE.

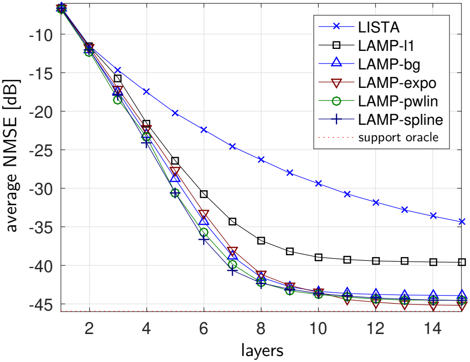




LAMP beyond soft-thresholding

So far, we used soft-thresholding to isolate the effects of Onsager correction.

What happens with more sophisticated (learned) denoisers?



Here we learned the parameters of these denoiser families:

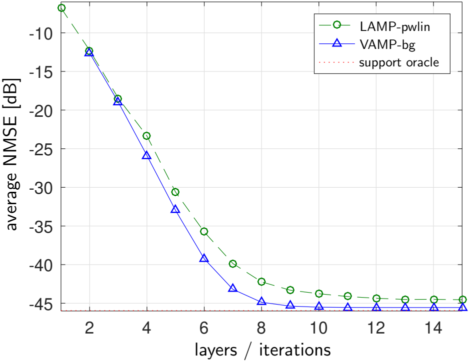
- scaled soft-thresholding
- conditional mean under BG
- Exponential kernel[23]
- Piecewise Linear[23]
- Spline[24]

Big improvement!

[23] Guo,Davies'15. [24] Kamilov,Mansour'16.

LAMP  versus VAMP 

How does our best Learned AMP compare to MMSE VAMP?

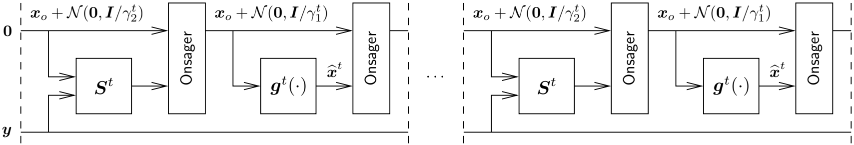


VAMP wins!

So what about "learned VAMP"?

Learned VAMP

- Suppose we unfold VAMP and learn (via backprop) the parameters $\{\boldsymbol{S}^t, \boldsymbol{g}^t\}_{t=1}^T$ that minimize the training MSE.



- Remarkably, backpropagation learns the parameters prescribed by VAMP!

  Theory explains the deep network!

- Onsager correction decouples the design of $\{\boldsymbol{S}^t, \boldsymbol{g}^t(\cdot)\}_{t=1}^T$:

  Layer-wise optimal $\boldsymbol{S}^t, \boldsymbol{g}^t(\cdot)$ $\Rightarrow$ Network optimal $\{\boldsymbol{S}^t, \boldsymbol{g}^t(\cdot)\}_{t=1}^T$

Conclusions

- VAMP is a computationally efficient algorithm for linear regression.
- For inference under large, right orthogonally-invariant $\boldsymbol{A}$, VAMP has a rigorous state evolution whose fixed-points, when unique, match the replica prediction of the MMSE.
- For convex optimization problems, VAMP is provably convergent for any $\boldsymbol{A}$.
- VAMP can be combined with EM to handle priors/likelihood with unknown parameters, again with a rigorous state evolution.
- VAMP supports nonseparable (i.e., "plug-in") denoisers, with a rigorous state evolution.
- Can unfold VAMP into an interpretable deep network.
- Not discussed: GLMs, multilayer approaches, bilinear approaches.