The VAMPire, Part I

Phil Schniter

THE OHIO STATE UNIVERSITY

Duke UNIVERSITY

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

Supported in part by NSF grants IIP-1539960 and CCF-1527162.

Les Houches Workshop on
Statistical Physics, Learning, Inference, and Networks—Feb 2017

Overview

Outline

The Linear Regression Problem

In this talk, we consider the following linear regression problem:

- Observations:

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

- Prior:

$$p(\boldsymbol{x};\boldsymbol{\theta}_1) \quad \text{with deterministic unknown parameters } \boldsymbol{\theta}_1.$$

- Likelihood function:

$$\ell(\boldsymbol{x};\theta_2) \triangleq \mathcal{N}(\boldsymbol{y};\boldsymbol{A}\boldsymbol{x},\boldsymbol{I}/\theta_2) \quad \text{with deterministic unknown precision } \theta_2.$$

Goal: infer $\boldsymbol{x}$ and estimate $\boldsymbol{\theta} \triangleq [\boldsymbol{\theta}_1, \theta_2]$ (as needed).

(Sundeep will discuss more complicated priors and likelihoods!)

Outline

Variational Inference

- For the moment, let's suppose that $\boldsymbol{\theta}$ is known.
- We would like to compute the 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 can avoid computing $Z(\boldsymbol{\theta})$ through variational optimization:

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

  but the density constraint keeps the problem difficult.

Expectation Consistent Approximation

- In Expectation consistent approximation (EC)[1], the density constraint is relaxed to moment-matching constraints:

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

  for any affine linear diagonalization operator $\mathrm{diag} : \mathbb{R}^{N\times N} \to \mathbb{R}^N$.

- The stationary points of EC are

$$b_1(\boldsymbol{x}) \propto p(\boldsymbol{x}; \boldsymbol{\theta}_1)\mathcal{N}(\boldsymbol{x}; \boldsymbol{r}_1, \mathbf{1} \oslash \boldsymbol{\gamma}_1)$$
$$b_2(\boldsymbol{x}) \propto \ell(\boldsymbol{x}; \theta_2)\mathcal{N}(\boldsymbol{x}; \boldsymbol{r}_2, \mathbf{1} \oslash \boldsymbol{\gamma}_2)$$
$$q(\boldsymbol{x}) = \mathcal{N}(\boldsymbol{x}; \widehat{\boldsymbol{x}}, \mathbf{1} \oslash \boldsymbol{\eta})$$
$$\text{s.t.} \begin{cases} \mathrm{E}\{\boldsymbol{x}|b_1\} = \mathrm{E}\{\boldsymbol{x}|b_2\} = \widehat{\boldsymbol{x}} \\ \mathrm{diag}(\mathrm{Cov}\{\boldsymbol{x}|b_1\}) = \mathrm{diag}(\mathrm{Cov}\{\boldsymbol{x}|b_2\}) = \boldsymbol{\eta}. \end{cases}$$

  which match the replica prediction of MMSE.[2]

[1] Opper,Winther'04, [2] Kabashima,Vehkaperä'14

An EP Algorithm

An expectation-propagation (EP)[3] algorithm whose fixed points match EC is

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

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

$\boldsymbol{\eta}_1 \leftarrow \mathbf{1} \oslash \operatorname{diag}\left[\operatorname{Cov}\left\{\boldsymbol{x} \,\middle|\, \boldsymbol{r}_1 = \boldsymbol{x} + \mathcal{N}(\mathbf{0}, \operatorname{Diag}(\boldsymbol{\gamma}_1)^{-1})\right\}\right]$	posterior precision
$\widehat{\boldsymbol{x}}_1 \leftarrow \mathrm{E}\left\{\boldsymbol{x} \,\middle|\, \boldsymbol{r}_1 = \boldsymbol{x} + \mathcal{N}(\mathbf{0}, \operatorname{Diag}(\boldsymbol{\gamma}_1)^{-1})\right\}$	denoising
$\boldsymbol{\gamma}_2 \leftarrow \boldsymbol{\eta}_1 - \boldsymbol{\gamma}_1$	extrinsic precision
$\boldsymbol{r}_2 \leftarrow (\widehat{\boldsymbol{x}}_1 \odot \boldsymbol{\eta}_1 - \boldsymbol{r}_1 \odot \boldsymbol{\gamma}_1) \oslash \boldsymbol{\gamma}_2$	Onsager correction
$\boldsymbol{\eta}_2 \leftarrow \mathbf{1} \oslash \operatorname{diag}\left[\left(\theta_2 \boldsymbol{A}^{\mathsf{T}}\boldsymbol{A} + \operatorname{Diag}(\boldsymbol{\gamma}_2)\right)^{-1}\right]$	posterior precision
$\widehat{\boldsymbol{x}}_2 \leftarrow \left(\theta_2 \boldsymbol{A}^{\mathsf{T}}\boldsymbol{A} + \operatorname{Diag}(\boldsymbol{\gamma}_2)\right)^{-1}\left(\theta_2 \boldsymbol{A}^{\mathsf{T}}\boldsymbol{y} + \boldsymbol{\gamma}_2 \odot \boldsymbol{r}_2\right)$	LMMSE estimation
$\boldsymbol{\gamma}_1 \leftarrow \boldsymbol{\eta}_2 - \boldsymbol{\gamma}_2$	extrinsic precision
$\boldsymbol{r}_1 \leftarrow (\widehat{\boldsymbol{x}}_2 \odot \boldsymbol{\eta}_2 - \boldsymbol{r}_2 \odot \boldsymbol{\gamma}_2) \oslash \boldsymbol{\gamma}_1$	Onsager correction

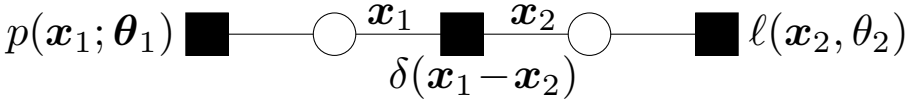
Questions: 1) Does it converge? Can we predict its trajectory?
2) Can we avoid the matrix inverse?

[3] Minka'01

Vector AMP (VAMP) 

- Consider EP with the "uniform diagonalization" $\mathrm{diag}(\boldsymbol{C}) = \mathbf{1}\,\mathrm{tr}(\boldsymbol{C})/N$.
  By precomputing the SVD $\boldsymbol{A} = \boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^{\mathsf{T}}$, we can[4]
  1) avoid matrix inversion and
  2) derive a rigorous state evolution. (Details in Sundeep's talk!)

- We call the algorithm "vector AMP" because, like the AMP algorithm,[5][6]
  it has a rigorous state-evolution and can be derived using an approximation of message passing, now on a vector-valued factor graph:



- Importantly, VAMP is provably convergent under either
  1) iid prior $p(\boldsymbol{x};\boldsymbol{\theta}_1)$ and large, right-rotationally invariant $\boldsymbol{A}$,
  2) strictly log-concave prior $p(\boldsymbol{x};\boldsymbol{\theta}_1)$ and arbitrary $\boldsymbol{A}$.

[4]Rangan,Schniter,Fletcher'16, [5]Donoho,Maleki,Montanari'09, [6]Bayati,Montanari'10

The VAMP Algorithm

For any Lipschitz function $\boldsymbol{g}_1(\cdot;\gamma_1,\boldsymbol{\theta}_1):\mathbb{R}^N\to\mathbb{R}^N\ldots$

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

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

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

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

$$\gamma_2 \leftarrow \eta_1-\gamma_1 \qquad \text{extrinsic precision}$$

$$\boldsymbol{r}_2 \leftarrow (\widehat{\boldsymbol{x}}_1\eta_1-\boldsymbol{r}_1\gamma_1)/\gamma_2 \qquad \text{Onsager correction}$$

$$\eta_2 \leftarrow \frac{1}{N}\sum_{n=1}^{N}\frac{1}{\theta_2 s_n^2+\gamma_2} \qquad \text{posterior precision}$$

$$\widehat{\boldsymbol{x}}_2 \leftarrow \boldsymbol{V}\left(\theta_2\operatorname{Diag}(\boldsymbol{s})^2+\gamma_2\boldsymbol{I}\right)^{-1}\left(\theta_2\operatorname{Diag}(\boldsymbol{s})\boldsymbol{U}^{\mathsf{T}}\boldsymbol{y}+\gamma_2\boldsymbol{V}^{\mathsf{T}}\boldsymbol{r}_2\right) \qquad \text{LMMSE estimation}$$

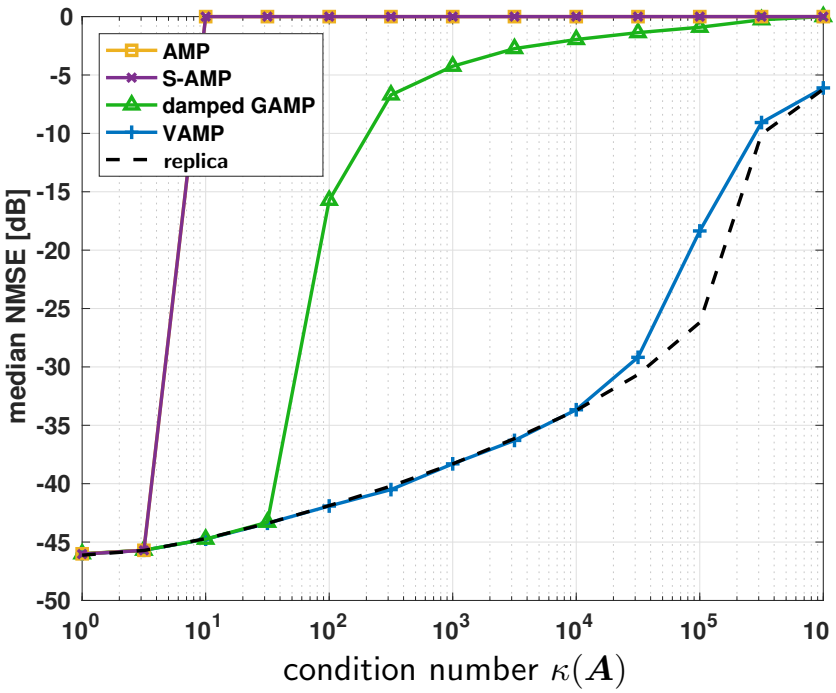
$$\gamma_1 \leftarrow \eta_2-\gamma_2 \qquad \text{extrinsic precision}$$

$$\boldsymbol{r}_1 \leftarrow (\widehat{\boldsymbol{x}}_2\eta_2-\boldsymbol{r}_2\gamma_2)/\gamma_1 \qquad \text{Onsager correction}$$

Complexity is dominated by two matrix-vector multiplies per iteration.

Experiment with Matched Priors

Comparison of several algorithms[7] 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
$\Pr\{X_0 \neq 0\} = 0.1$

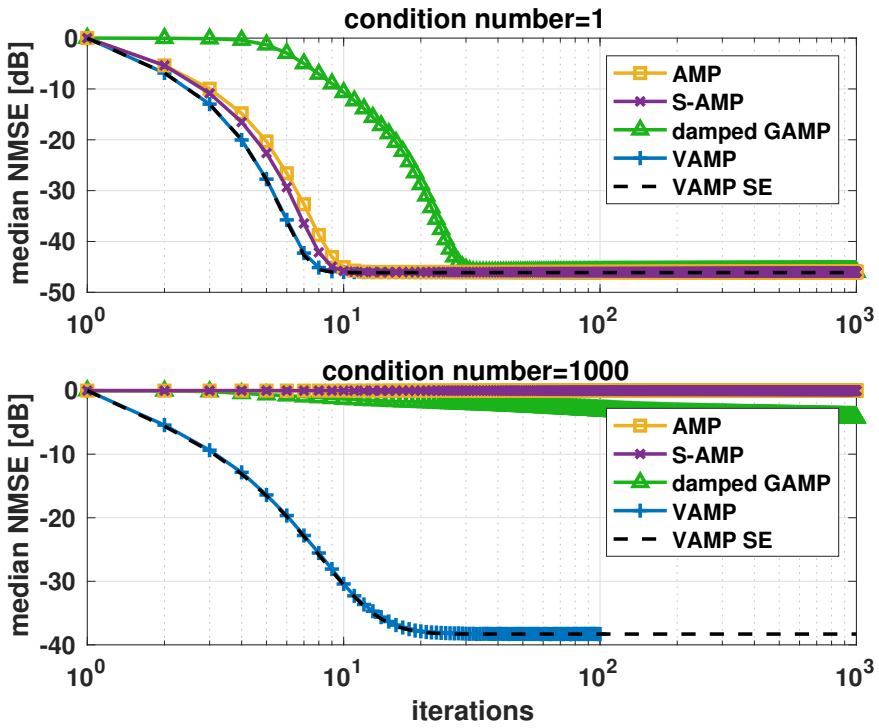
SNR $= 40$dB

VAMP follows replica prediction[8] over a wide range of condition numbers.

[7] S-AMP: Cakmak,Fleury,Winther'14, AD-GAMP: Vila,Schniter,Rangan,Krzakala,Zdeborová'15
[8] Tulino,Caire,Verdú,Shamai'13

Experiment with Matched Priors

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 $= 40$dB

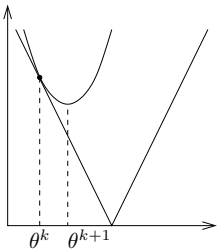
VAMP is fast even when $\boldsymbol{A}$ is ill-conditioned.

Outline

Expectation Maximization

- We now return to the problem of estimating $\boldsymbol{\theta}$.
- The EM algorithm[9] is majorization-minimization approach to ML estimation that iteratively minimizes a tight upper bound on $-\ln p(\boldsymbol{y}|\boldsymbol{\theta})$:

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



- Can also write EM in terms of the Gibbs free energy:[10]

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

- Thus, we can interleave EM and VAMP to solve

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

[9] Dempster,Laird,Rubin'77, [10] Neal,Hinton'98

The EM-VAMP Algorithm

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

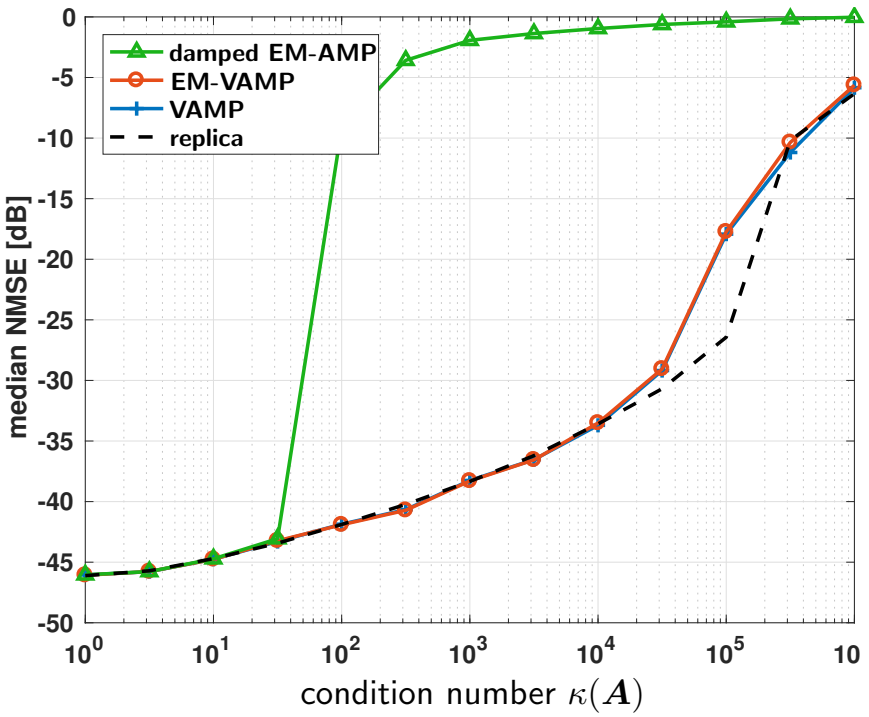
For $k = 1, 2, 3, \ldots$

$\eta_1 \leftarrow \gamma_1 / \big\langle \boldsymbol{g}_1'(\boldsymbol{r}_1; \gamma_1, \widehat{\boldsymbol{\theta}}_1) \big\rangle$	posterior precision
$\widehat{\boldsymbol{x}}_1 \leftarrow \boldsymbol{g}_1(\boldsymbol{r}_1; \gamma_1, \widehat{\boldsymbol{\theta}}_1)$	denoising
$\gamma_2 \leftarrow \eta_1 - \gamma_1$	extrinsic precision
$\boldsymbol{r}_2 \leftarrow (\widehat{\boldsymbol{x}}_1 \eta_1 - \boldsymbol{r}_1 \gamma_1)/\gamma_2$	Onsager correction
$\widehat{\theta}_2 \leftarrow \arg\max_{\theta_2} \mathrm{E}\{\ln \ell(\boldsymbol{x}; \theta_2) \,\vert\, \boldsymbol{r}_2; \gamma_2, \widehat{\theta}_2\}$	EM update
$\eta_2 \leftarrow \gamma_2 / \big\langle \boldsymbol{g}_2'(\boldsymbol{r}_2; \gamma_2, \widehat{\theta}_2) \big\rangle$	posterior precision
$\widehat{\boldsymbol{x}}_2 \leftarrow \boldsymbol{g}_2(\boldsymbol{r}_2; \gamma_2, \widehat{\theta}_2)$	LMMSE estimation
$\gamma_1 \leftarrow \eta_2 - \gamma_2$	extrinsic precision
$\boldsymbol{r}_1 \leftarrow (\widehat{\boldsymbol{x}}_2 \eta_2 - \boldsymbol{r}_2 \gamma_2)/\gamma_1$	Onsager correction
$\widehat{\boldsymbol{\theta}}_1 \leftarrow \arg\max_{\boldsymbol{\theta}_1} \mathrm{E}\{\ln p(\boldsymbol{x}; \boldsymbol{\theta}_1) \,\vert\, \boldsymbol{r}_1; \gamma_1, \widehat{\boldsymbol{\theta}}_1\}$	EM update

Experiments suggest it helps to update $\widehat{\theta}_2$ several times per VAMP iteration.

Experiment with Learned $\boldsymbol{\theta}$

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 $\kappa(\boldsymbol{A})$

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

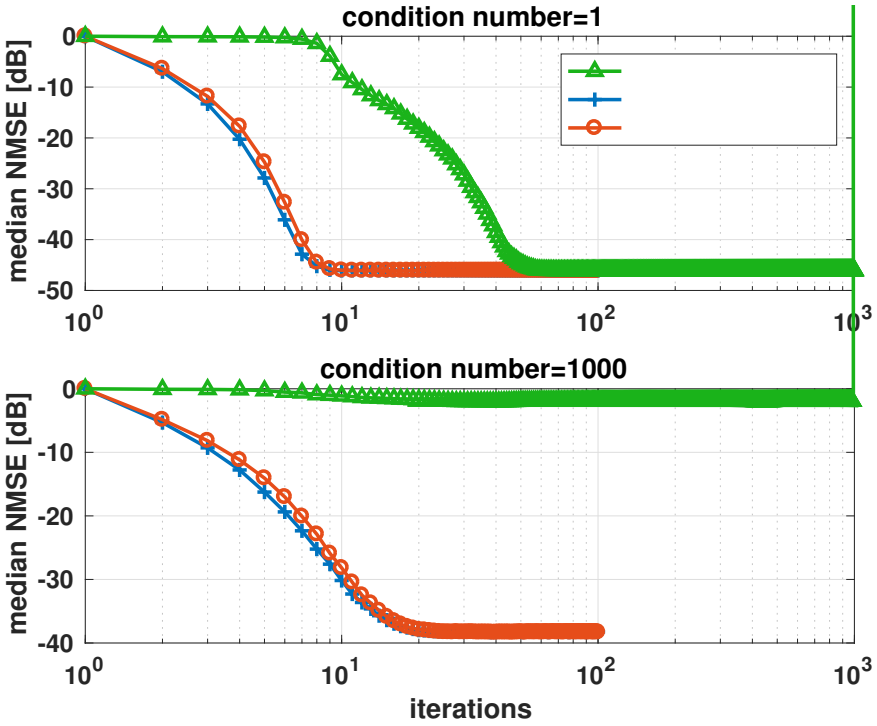
SNR $= 40$dB

EM-VAMP achieves oracle performance at all condition numbers![11]

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

Experiment with Learned $\boldsymbol{\theta}$

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



State Evolution and Consistency

- Like VAMP, EM-VAMP has a rigorous state-evolution when the prior is i.i.d. and $\boldsymbol{A}$ is large and right-rotationally invariant.[12]

- 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 $\boldsymbol{\theta}_1$.[12]

- 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

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

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

More details in Sundeep's talk!

[12] Fletcher,Rangan,Schniter'17

Outline

Damped VAMP

- Practical applications of VAMP may involve matrices $\boldsymbol{A}$ that are *not* large and right-rotationally invariant and/or signals that are *not* i.i.d.!
- Can prove that that a damped version of VAMP converges when

$$\exists c_1, c_2 > 0 \;\; \text{s.t.} \;\; \frac{\gamma}{\gamma + c_1}\boldsymbol{I} \leq \frac{\partial \boldsymbol{g}_1}{\partial \boldsymbol{r}}(\boldsymbol{r}, \gamma) \leq \frac{\gamma}{\gamma + c_2}\boldsymbol{I},$$

  as occurs with MAP inference under strictly log-concave prior $p(\boldsymbol{x}; \boldsymbol{\theta}_1)$.
- To do the damping, we simply replace the $\boldsymbol{r}_1$ update with

$$\boldsymbol{r}_1 \leftarrow \zeta\big[(\widehat{\boldsymbol{x}}_2\eta_2 - \boldsymbol{r}_2\gamma_2)/\gamma_1\big] + (1-\zeta)\boldsymbol{r}_1 \quad \text{for some } \zeta \in (0, 1].$$

- For convergence, it suffices that $\zeta \leq \frac{2\min\{\gamma_1,\gamma_2\}}{\gamma_1+\gamma_2}$. Thus
  - the damping factor can be adapted using γ_1, γ_2,
  - no damping (i.e., $\zeta = 1$) suffices when $\gamma_1 = \gamma_2$.

Connections to ADMM

- Consider the optimization problem

$$\arg\min_{\boldsymbol{x}} f_1(\boldsymbol{x}) + f_2(\boldsymbol{x}) \quad \text{with, e.g.,} \quad \begin{cases} f_1(\boldsymbol{x}) = -\log p(\boldsymbol{x};\boldsymbol{\theta}_1) \\ f_2(\boldsymbol{x}) = \frac{\theta_2}{2}\|\boldsymbol{A}\boldsymbol{x}-\boldsymbol{y}\|^2 \end{cases}$$

  and define the augmented Lagrangian

$$L_\gamma(\boldsymbol{x}_1,\boldsymbol{x}_2,\boldsymbol{s}) = f_1(\boldsymbol{x}_1) + f_2(\boldsymbol{x}_2) + \boldsymbol{s}^{\mathsf{T}}(\boldsymbol{x}_1-\boldsymbol{x}_2) + \frac{\gamma}{2}\|\boldsymbol{x}_1-\boldsymbol{x}_2\|^2.$$

- An ADMM-like algorithm (known as Peaceman-Rachford splitting) is

$$\begin{aligned} \widehat{\boldsymbol{x}}_1 &\leftarrow \arg\min_{\boldsymbol{x}_1} L_\gamma(\boldsymbol{x}_1,\widehat{\boldsymbol{x}}_2,\boldsymbol{s}) \\ \boldsymbol{s} &\leftarrow \boldsymbol{s} + \gamma(\widehat{\boldsymbol{x}}_1 - \widehat{\boldsymbol{x}}_2) \\ \widehat{\boldsymbol{x}}_2 &\leftarrow \arg\min_{\boldsymbol{x}_2} L_\gamma(\widehat{\boldsymbol{x}}_1,\boldsymbol{x}_2,\boldsymbol{s}) \\ \boldsymbol{s} &\leftarrow \boldsymbol{s} + \gamma(\widehat{\boldsymbol{x}}_1 - \widehat{\boldsymbol{x}}_2) \end{aligned}$$

- PRS is guaranteed to converge for strictly convex $f_1(\cdot)$ and $f_2(\cdot)$.

Connections to ADMM

- Now consider VAMP applied to the same MAP optimization problem, but with $\gamma_1 = \gamma_2 \triangleq \gamma$ enforced at each iteration.

- Define $\boldsymbol{s}_i \triangleq \gamma(\widehat{\boldsymbol{x}}_i - \boldsymbol{r}_i)$ for $i = 1, 2$.

- The γ-forced VAMP manifests as[13]

$$\begin{aligned}
\widehat{\boldsymbol{x}}_1 &\leftarrow \arg\min_{\boldsymbol{x}_1} L_\gamma(\boldsymbol{x}_1, \widehat{\boldsymbol{x}}_2, \boldsymbol{s}_1) \\
\boldsymbol{s}_2 &\leftarrow \boldsymbol{s}_1 + \gamma(\widehat{\boldsymbol{x}}_1 - \widehat{\boldsymbol{x}}_2) \\
\widehat{\boldsymbol{x}}_2 &\leftarrow \arg\min_{\boldsymbol{x}_2} L_\gamma(\widehat{\boldsymbol{x}}_1, \boldsymbol{x}_2, \boldsymbol{s}_2) \\
\boldsymbol{s}_1 &\leftarrow \boldsymbol{s}_2 + \gamma(\widehat{\boldsymbol{x}}_1 - \widehat{\boldsymbol{x}}_2)
\end{aligned}$$

  which is equivalent to Peaceman-Rachford splitting.

- This agrees with the earlier result that VAMP needs no damping when $\gamma_1 = \gamma_2$ (for strictly convex f_1, f_2).

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

Outline

Plug-and-play VAMP

- Recall that the nonlinear estimation step in VAMP (or AMP)

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

  can be interpreted as "denoising" the pseudo-measurement $\boldsymbol{r}_1$.

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

- Such denoising procedures can be "plugged into" signal recovery algorithms like ADMM[14], AMP[15], or VAMP[16].

- For AMP and VAMP, the divergence can be approximated using Monte-Carlo:

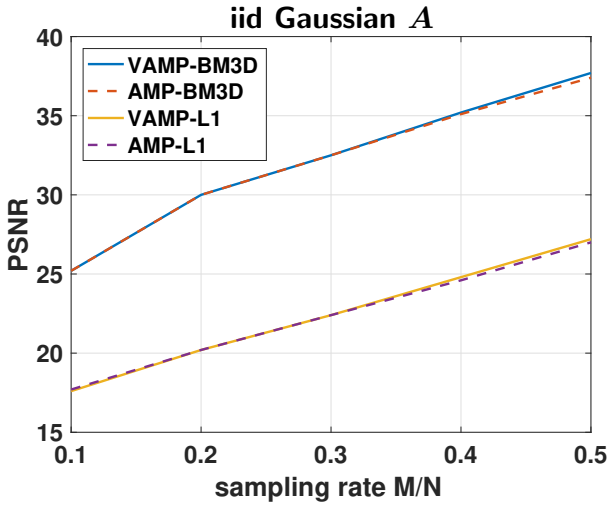
$$\langle \boldsymbol{g}_1'(\boldsymbol{r},\gamma_1)\rangle = \frac{1}{N}\operatorname{tr}\left[\frac{\partial \boldsymbol{g}_1}{\partial \boldsymbol{r}_1}\right] \approx \frac{1}{K}\sum_{k=1}^{K} \frac{\boldsymbol{p}_k^{\mathsf{T}}\left[\boldsymbol{g}_1(\boldsymbol{r}+\epsilon\boldsymbol{p}_k,\gamma_1) - \boldsymbol{g}_1(\boldsymbol{r},\gamma_1)\right]}{N\epsilon}$$

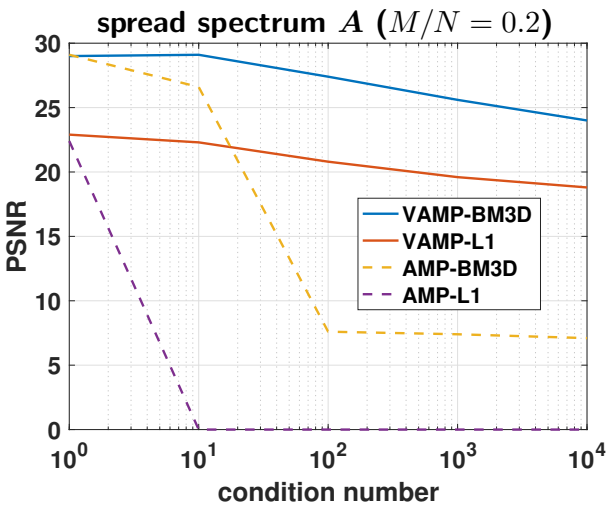
  with random vectors $\boldsymbol{p}_k \in \{\pm 1\}^N$ and small $\epsilon > 0$. Often, $K = 1$ suffices.

[14] Bouman et al'13, [15] Metzler,Maleki,Baraniuk'14, [16] Schniter,Rangan,Fletcher'16

Experiment: Image Recovery with Random Matrices

Plug-and-play versions of VAMP and AMP work similarly when $\boldsymbol{A}$ is i.i.d., but VAMP can handle a larger class of random matrices $\boldsymbol{A}$.





Results above are averaged over 128×128 versions of

lena, *barbara*, *boat*, *fingerprint*, *house*, *peppers*

and 10 random realizations of $\boldsymbol{A}, \boldsymbol{w}$.

Image Recovery with Non-Random Matrices

- Many imaging applications (e.g., MRI) use low-frequency Fourier measurements, in which case $\boldsymbol{A} = \boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^{\mathsf{T}} = \boldsymbol{I}\,[\boldsymbol{I}\ \boldsymbol{0}]\,\boldsymbol{F}$.
- This causes problems for VAMP because the signal correlation structure interacts with $\boldsymbol{F}$ in a way that VAMP is not designed to handle.
- Why? Say $\boldsymbol{x}$ is a natural image, and consider $\boldsymbol{q} = \boldsymbol{V}^{\mathsf{T}}\boldsymbol{x}$.
  - If $\boldsymbol{V}$ is large and Haar, then $\boldsymbol{q}$ will be iid Gaussian.
  - If $\boldsymbol{V}^{\mathsf{T}} = \boldsymbol{F}$, the low-freq entries of $\boldsymbol{q}$ will be much stronger than the others.

VAMP treats $\boldsymbol{V}^{\mathsf{T}}\boldsymbol{x}$ as iid Gaussian and thus diverges when $\boldsymbol{V}^{\mathsf{T}} = \boldsymbol{F}$!

Whitened VAMP  for Image REcovery (VAMPire)

- To apply VAMP with non-random Fourier measurements, we propose to operate on the whitened signal:

$$\boldsymbol{y} = \underbrace{[\boldsymbol{I}\ \boldsymbol{0}]\boldsymbol{F}\boldsymbol{R}_x^{1/2}}_{\boldsymbol{A}}\boldsymbol{s} + \boldsymbol{w} \text{ for } \begin{cases} \boldsymbol{R}_x = \mathrm{E}\{\boldsymbol{x}\boldsymbol{x}^{\mathsf{T}}\} \\ \boldsymbol{s} = \text{whitened signal coefficients} \end{cases}$$

  and perform plug-and-play denoising from the whitened-coefficient space:

$$\widehat{\boldsymbol{s}}_1 = \boldsymbol{g}_1(\boldsymbol{r}_1, \gamma_1) = \boldsymbol{R}_x^{-1/2}\text{denoise}(\boldsymbol{R}_x^{1/2}\boldsymbol{r}_1; \gamma_1 N/\operatorname{tr}(\boldsymbol{R}_x)).$$

- In practice, we approximate $\boldsymbol{R}_x \approx \boldsymbol{W}^{\mathsf{T}} \operatorname{Diag}(\boldsymbol{\tau})^2 \boldsymbol{W}$, where $\boldsymbol{W}$ is a wavelet transform and τ_i^2 specifies the energy of the ith wavelet coefficient (which is easy to predict for natural images).

Whitened VAMP  for Image REcovery (VAMPire)

- The resulting matrix $\boldsymbol{A} = [\boldsymbol{I}\ \boldsymbol{0}]\boldsymbol{F}\boldsymbol{W}\,\mathrm{Diag}(\boldsymbol{\tau})$ does not yield a right singular vector matrix $\boldsymbol{V}$ with a fast multiplication.

- But since $\boldsymbol{A}$ has a fast implementation, the LMMSE stage can be computed via (preconditioned) LSQR:

$$\boldsymbol{g}_2(\boldsymbol{r}_2;\gamma_2) = (\gamma_w \boldsymbol{A}^{\mathsf{T}}\boldsymbol{A} + \gamma_2 \boldsymbol{I})^{-1}(\gamma_w \boldsymbol{A}^{\mathsf{T}}\boldsymbol{y} + \gamma_2 \boldsymbol{r}_2) = \begin{bmatrix}\sqrt{\gamma_w}\boldsymbol{A} \\ \sqrt{\gamma_2}\boldsymbol{I}\end{bmatrix}^{+} \begin{bmatrix}\sqrt{\gamma_w}\boldsymbol{y} \\ \sqrt{\gamma_2}\boldsymbol{r}_2\end{bmatrix}$$

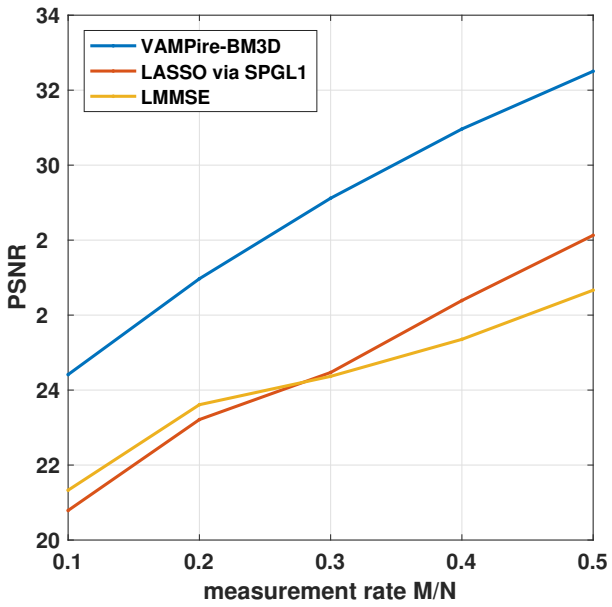
- The divergence $\langle \boldsymbol{g}_2'(\boldsymbol{r}_2;\gamma_2)\rangle$ can be approximated using Monte-Carlo:

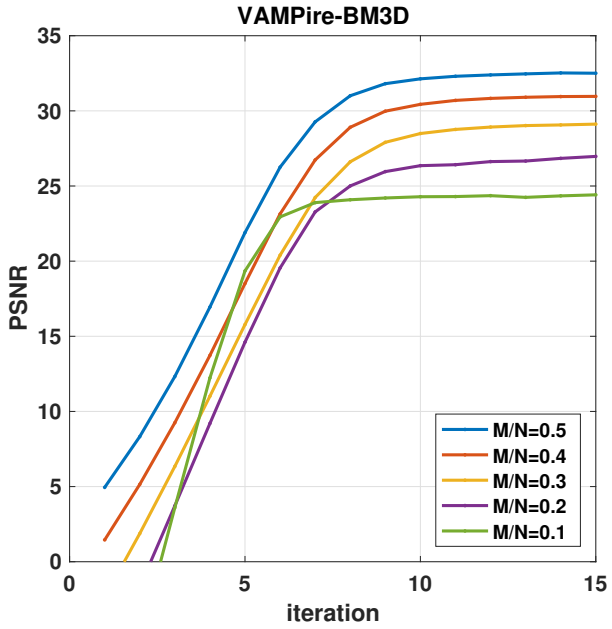
$$\langle \boldsymbol{g}_2' \rangle = \frac{\gamma_2}{N}\,\mathrm{tr}\left[\left(\gamma_w \boldsymbol{A}^{\mathsf{H}}\boldsymbol{A} + \gamma_2 \boldsymbol{I}\right)^{-1}\right] \approx \frac{1}{NK}\sum_{k=1}^{K} \boldsymbol{p}_k \begin{bmatrix}\sqrt{\gamma_w}\boldsymbol{A} \\ \sqrt{\gamma_2}\boldsymbol{I}\end{bmatrix}^{+} \begin{bmatrix}\boldsymbol{0} \\ \sqrt{\gamma_2}\boldsymbol{p}_k\end{bmatrix},$$

  where $\mathrm{E}\{\boldsymbol{p}_k \boldsymbol{p}_k^{\mathsf{H}}\} = \boldsymbol{I}$. Here again, (preconditioned) LSQR can be used.
  In practice, $K = 1$ suffices.

Image Recovery Experiments

- Fourier measurements sampled at M lowest frequencies
- SNR=40dB
- 128×128 images $\{$*lena*, *barbara*, *boat*, *fingerprint*, *house*, *peppers*$\}$
- db1 wavelet decomposition, $D = 2$ levels

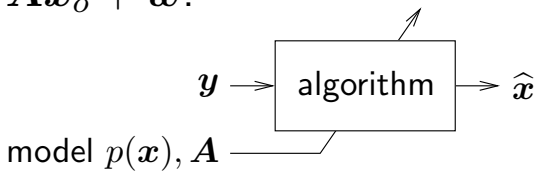




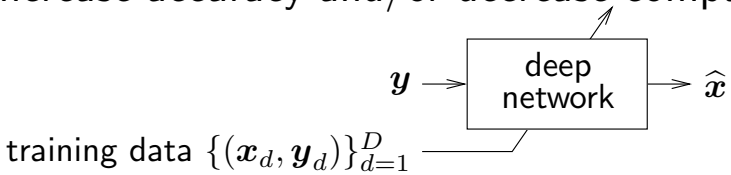
Outline

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?



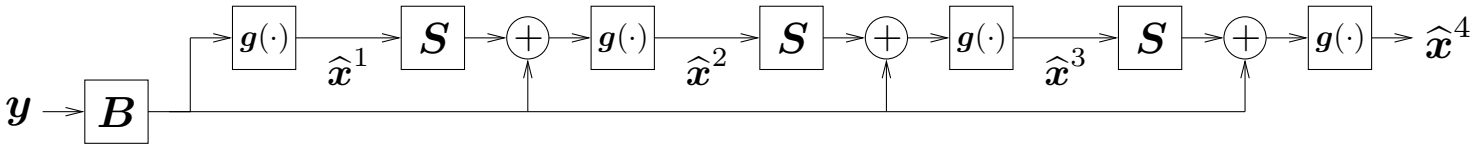
- Are there connections between these approaches?

Unfolding Algorithms into Networks

Consider, e.g., the classical sparse-reconstruction algorithm, ISTA.[17]

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

Gregor & LeCun[18] proposed to "unfold" it into a deep net and "learn" improved parameters using training data, yielding "learned ISTA" (LISTA):

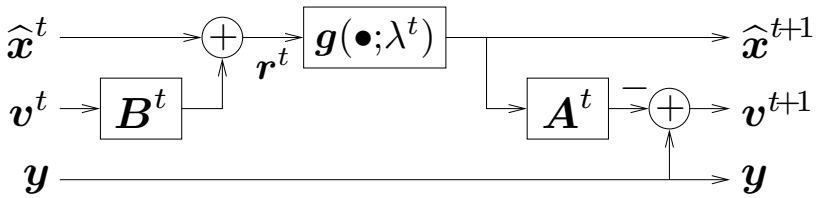


The same "unfolding & learning" idea can be used to improve AMP, yielding "learned AMP" (LAMP).[19]

[17] Daubechies,Defrise,DeMol'04. [18] Gregor,LeCun'10. [19] Borgerding,Schniter'16.

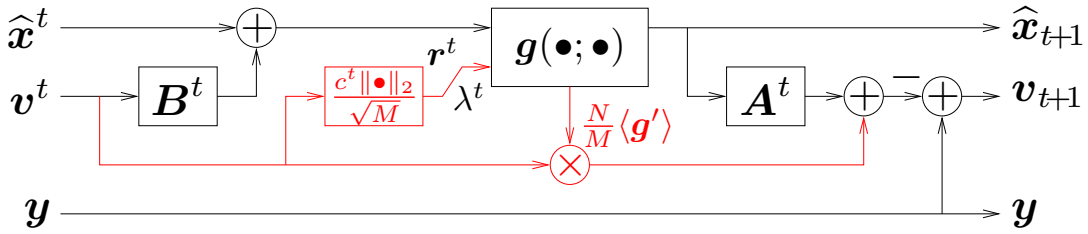
Onsager-Corrected Deep Networks

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:

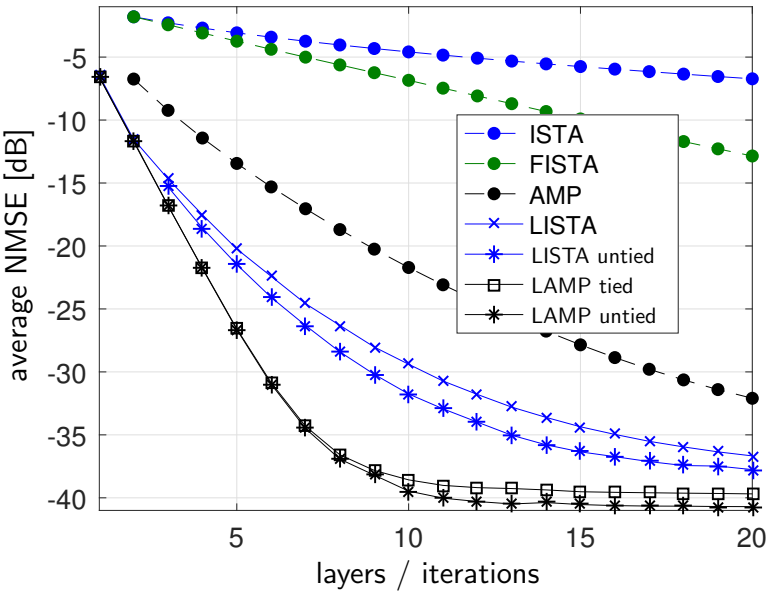


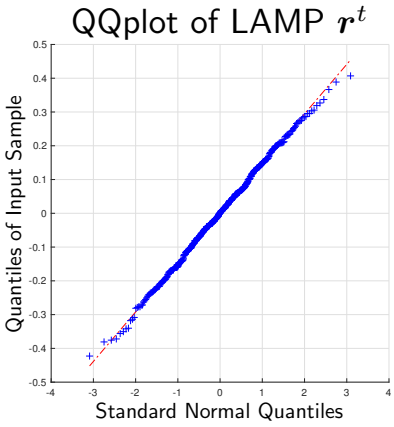
Onsager correction now aims to decouple errors across layers.

LAMP performance with soft-threshold denoising

LISTA beats AMP,FISTA,ISTA
LAMP beats LISTA

in convergence speed and asymptotic MSE.

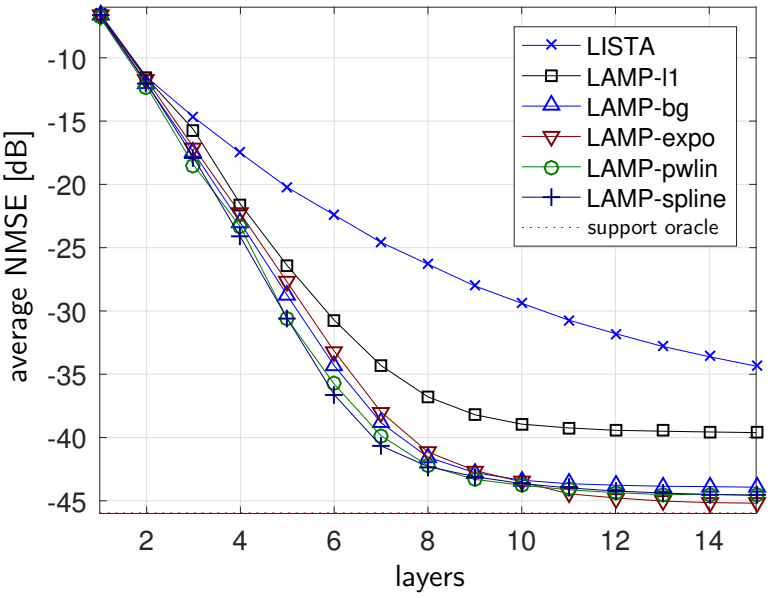




LAMP with more sophisticated scalar denoisers

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

What happens with more sophisticated (learned) denoisers?



Here we learned the parameters of these denoiser families:

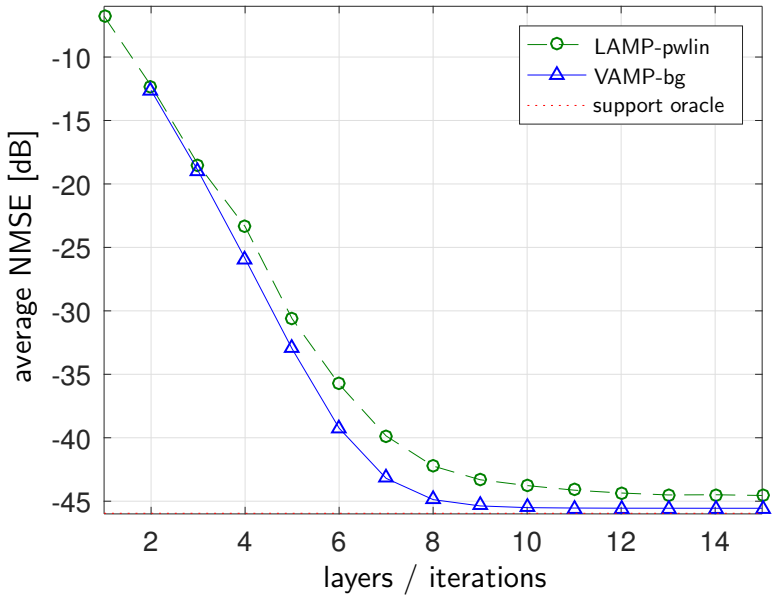
- scaled soft-thresholding
- conditional mean under BG
- Exponential kernel[20]
- Piecewise Linear[20]
- Spline[21]

Big improvement!

[20] Guo,Davies'15. [21] Kamilov,Mansour'16.

LAMP versus VAMP

How does our best Learned AMP compare to (unlearned) VAMP?

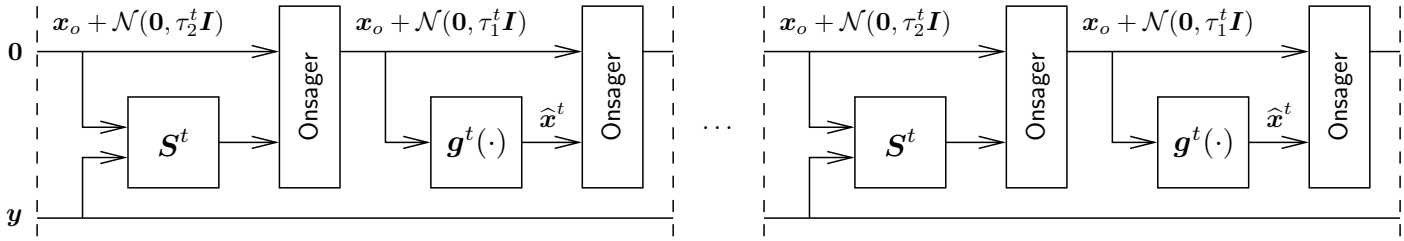


VAMP wins!

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 does not improve matched VAMP!

  VAMP is locally optimal

- 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   

- By merging classical algorithms (EC,EP) with SVDs and modern analysis tools (Bolthausen conditioning trick), we get "VAMP": a fast and robust algorithm with a rigorous SE analysis.
- VAMP is like Peaceman-Rachford splitting with adaptive penalties γ_1, γ_2.
- Can combine with EM methods to handle priors/likelihood with unknown parameters, rigorously.
- Can apply sophisticated denoising algorithms (BM3D) via "plug and play."
- Damping and whitening may be needed in practice, when $\boldsymbol{A}$ is not random and $\boldsymbol{x}$ is not iid.
- Can unroll VAMP into an deep network that is interpretable (and in some cases optimal).
- Lots more to do! (Generalized linear models, multilayer, bilinear models...)

Stay tuned for Sundeep's talk!