Plug-and-play Image Recovery using Vector AMP

Philip Schniter (Ohio State/Duke)
Sundeep Rangan (NYU)
Alyson Fletcher (UCLA)

BASP Frontiers Workshop 2017
Supported by NSF grant 1527162

Image Recovery

- Our goal is to recover an N-pixel image $\boldsymbol{u}^0$ from $M \ll N$ noisy linear measurements

$$\boldsymbol{y} = \boldsymbol{\Phi}\boldsymbol{u}^0 + \boldsymbol{w} \in \mathbb{C}^M \quad \text{with} \quad \begin{cases} \boldsymbol{u}^0 : \text{true image} \\ \boldsymbol{\Phi} : \text{linear measurement operator} \\ \boldsymbol{w} : \text{white noise of variance } \sigma_w^2. \end{cases}$$

- In the sparsity-based approach [1], one writes

$$\boldsymbol{y} = \boldsymbol{\Phi}\boldsymbol{\Psi}\boldsymbol{c}^0 + \boldsymbol{w} \in \mathbb{C}^M \quad \text{with} \quad \begin{cases} \boldsymbol{c}^0 : \text{wavelet coefficients} \\ \boldsymbol{\Psi} : \text{inverse wavelet transform,} \end{cases}$$

and first recovers a sparse estimate $\widehat{\boldsymbol{c}}$ of $\boldsymbol{c}^0$, then later the image $\widehat{\boldsymbol{u}} = \boldsymbol{\Psi}\widehat{\boldsymbol{c}}$.

- In the plug-and-play approach [2], one repeatedly calls an image denoising algorithm (e.g., BM3D)

$$\widehat{\boldsymbol{u}}_t = \text{denoise}(\boldsymbol{r}_t; \sigma_t) \quad \text{where} \quad \begin{cases} \boldsymbol{r}_t : \text{noisy version of } \boldsymbol{u}^0 \\ \sigma_t^2 : \text{noise variance} \end{cases}$$

inside an iterative reconstruction algorithm like ADMM or AMP.

Approximate Message Passing (AMP)

- For recovery of $\boldsymbol{x}^0$ from $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}^0 + \boldsymbol{w}$, the AMP algorithm [3] is

Input $\boldsymbol{y}, \boldsymbol{A}, \boldsymbol{g}(\cdot; \sigma_t)$ and initialize $\widehat{\boldsymbol{x}}_0 = \boldsymbol{0}$ and $\boldsymbol{v}_{-1} = \boldsymbol{0}$.
For $t = 0, 1, 2, \dots, T-1$,

$$\begin{aligned} \boldsymbol{v}_t &= \boldsymbol{y} - \boldsymbol{A}\widehat{\boldsymbol{x}}_t + \tfrac{N}{M}\alpha_t \boldsymbol{v}_{t-1} && \text{Onsager-corrected residual} \\ \boldsymbol{r}_t &= \widehat{\boldsymbol{x}}_t + \boldsymbol{A}^{\mathsf{H}}\boldsymbol{v}_t && \text{back-projection} \\ \sigma_t^2 &= M^{-1}\|\boldsymbol{v}_t\|^2 && \text{variance update} \\ \widehat{\boldsymbol{x}}_{t+1} &= \boldsymbol{g}(\boldsymbol{r}_t; \sigma_t) && \text{denoising} \\ \alpha_{t+1} &= \langle \boldsymbol{g}'(\boldsymbol{r}_t; \sigma_t) \rangle && \text{divergence} \end{aligned}$$

Return $\widehat{\boldsymbol{x}}_T$

where the divergence is defined as

$$\langle \boldsymbol{g}'(\boldsymbol{r}_t; \sigma_t) \rangle = \frac{1}{N} \operatorname{tr}\left[\frac{\partial \boldsymbol{g}}{\partial \boldsymbol{r}}(\boldsymbol{r}_t; \sigma_t) \right].$$

- With large i.i.d. Gaussian $\boldsymbol{A}$:
  - AMP has a rigorous state-evolution (SE) [4] when $\boldsymbol{g}$ is Lipschitz and separable:

$$[\boldsymbol{g}(\boldsymbol{r}; \sigma)]_j = g(r_j; \sigma) \ \forall j.$$

  - The SE fixed points are "good" in that they match the replica prediction (recently proven correct [5]) under an i.i.d. signal and MMSE scalar denoiser g.
  - Good empirical performance in plug-and-play case, as in Metzler/Maleki/Baraniuk [6].
- With other $\boldsymbol{A}$:
  - AMP can diverge when $\boldsymbol{A}$ is (mildly) mean-perturbed, ill-conditioned, or structured [7].
  - Damping or sequential updating helps convergence over a limited range of $\boldsymbol{A}$.
  - Even if it converges, AMP's fixed points are suboptimal with non-i.i.d. Gaussian $\boldsymbol{A}$.

References

1. A. Chambolle, R. A. DeVore, N. Y. Lee, and B. J. Lucier, "Nonlinear wavelet image processing: Variational problems, compression, and noise removal through wavelet shrinkage," *IEEE Trans. Image Process.*, 1998.
2. S. V. Venkatakrishnan, C. A. Bouman, and B. Wohlberg, "Plug-and-play priors for model based reconstruction," *GlobalSIP* 2013.
3. D.L. Donoho, A. Maleki, and A. Montanari, "Message passing algorithms for compressed sensing," *PNAS* 2009.
4. M. Bayati and A. Montanari, "The dynamics of message passing on dense graphs, with applications to compressed sensing," *IEEE Trans. Info. Thy*, 2011.
5. G. Reeves and H.D. Pfister, "The replica-symmetric prediction for compressed sensing with Gaussian matrices is exact," *ISIT*, 2016.
6. C. A. Metzler, A. Maleki, and R. G. Baraniuk, "From denoising to compressed sensing," *IEEE Trans. Info. Thy*, 2016. (See also arXiv:1406.4175 and http://dsp.rice.edu/software/DAMP-toolbox.)
7. J. Vila, P. Schniter, S. Rangan, F. Krzakala, and L. Zdeborová, "Adaptive damping and mean removal for the generalized approximate message passing algorithm," *ICASSP*, 2015.
8. S. Rangan, P. Schniter, and A. K. Fletcher, "Vector Approximate Message Passing," *arXiv:1610.03082*.
9. A. M. Tulino, G. Caire, S. Verdú, and S. Shamai (Shitz), "Support recovery with sparsely sampled free random matrices," *IEEE Trans. Info. Thy.*, 2013.
10. B. Roman, B. Adcock and A. C. Hansen, "On asymptotic structure in compressed sensing," *arXiv:1406.4178.*

Vector AMP (VAMP)

- For recovery of $\boldsymbol{x}^0$ from $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}^0 + \boldsymbol{w}$, the VAMP algorithm [8] is

Input $\widetilde{\boldsymbol{g}}$ from (1) and denoiser $\boldsymbol{g}$, and initialize $\widetilde{\boldsymbol{r}}_1 = \boldsymbol{0}$ and $\widetilde{\sigma}_1$.
For $t = 1, 2, \dots, T$,

$$\begin{aligned} \widetilde{\boldsymbol{x}}_t &= \widetilde{\boldsymbol{g}}(\widetilde{\boldsymbol{r}}_t; \widetilde{\sigma}_t) && \text{LMMSE estimation} \\ \widetilde{\alpha}_t &= \langle \widetilde{\boldsymbol{g}}'(\widetilde{\boldsymbol{r}}_t; \widetilde{\sigma}_t) \rangle && \text{divergence} \\ \boldsymbol{r}_t &= (\widetilde{\boldsymbol{x}}_t - \widetilde{\alpha}_t \widetilde{\boldsymbol{r}}_t)/(1 - \widetilde{\alpha}_t) && \text{Onsager correction} \\ \sigma_t^2 &= \widetilde{\sigma}_t^2 \widetilde{\alpha}_t/(1 - \widetilde{\alpha}_t) && \text{variance update} \\ \widehat{\boldsymbol{x}}_t &= \boldsymbol{g}(\boldsymbol{r}_t; \sigma_t) && \text{denoising} \\ \alpha_t &= \langle \boldsymbol{g}'(\boldsymbol{r}_t; \sigma_t) \rangle && \text{divergence} \\ \widetilde{\boldsymbol{r}}_{t+1} &= (\widehat{\boldsymbol{x}}_t - \alpha_t \boldsymbol{r}_t)/(1 - \alpha_t) && \text{Onsager correction} \\ \widetilde{\sigma}_{t+1}^2 &= \sigma_t^2 \alpha_t/(1 - \alpha_t) && \text{variance update} \end{aligned}$$

Return $\widehat{\boldsymbol{x}}_T$

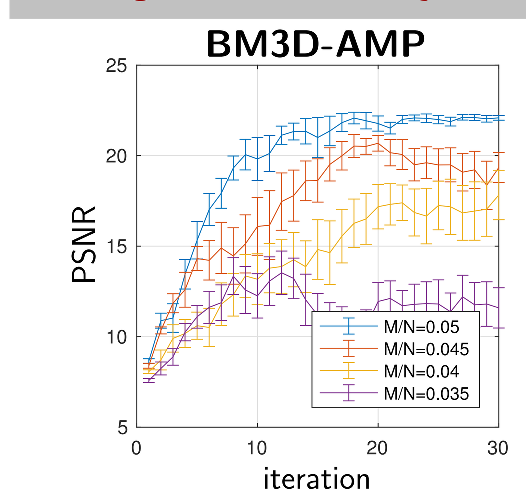
where, given the SVD $\boldsymbol{A} = \boldsymbol{U}\operatorname{Diag}(\boldsymbol{s})\boldsymbol{V}^{\mathsf{H}}$, the LMMSE stage does

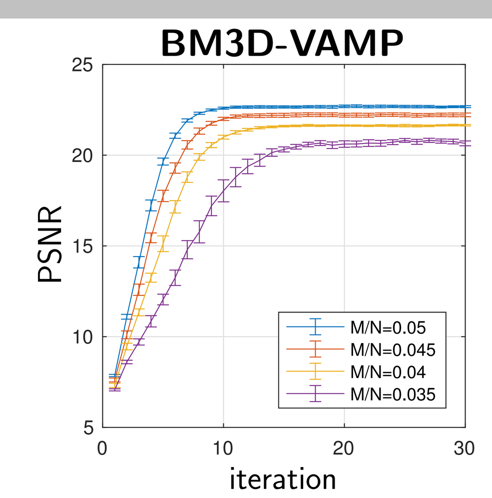
$$\begin{aligned} \widetilde{\boldsymbol{g}}(\widetilde{\boldsymbol{r}}, \widetilde{\sigma}) &= \left(\widetilde{\sigma}^2\boldsymbol{A}^{\mathsf{H}}\boldsymbol{A} + \sigma_w^2\boldsymbol{I}\right)^{-1}\left(\widetilde{\sigma}^2\boldsymbol{A}^{\mathsf{H}}\boldsymbol{y} + \sigma_w^2\widetilde{\boldsymbol{r}}\right) \qquad (1) \\ &= \boldsymbol{V}\left(\widetilde{\sigma}^2\operatorname{Diag}(\boldsymbol{s})^2 + \sigma_w^2\boldsymbol{I}\right)^{-1}\left(\widetilde{\sigma}^2\operatorname{Diag}(\boldsymbol{s})\boldsymbol{U}^{\mathsf{H}}\boldsymbol{y} + \sigma_w^2\boldsymbol{V}^{\mathsf{H}}\widetilde{\boldsymbol{r}}_t\right) \end{aligned}$$

$$\langle \widetilde{\boldsymbol{g}}(\widetilde{\boldsymbol{r}}, \widetilde{\sigma}) \rangle = N^{-1}\operatorname{tr}\left[(\widetilde{\sigma}^2\boldsymbol{A}^{\mathsf{H}}\boldsymbol{A} + \sigma_w^2\boldsymbol{I})^{-1}\right]\sigma_w^2 = \frac{1}{N}\sum_{n=0}^{N-1}\frac{\sigma_w^2}{\widetilde{\sigma}^2 s_n^2 + \sigma_w^2}$$

- We say that $\boldsymbol{A}$ is right-rotationally invariant if a $\boldsymbol{V}$ is a Haar-distributed random matrix (i.e., uniformly distributed on the set of unitary matrices).
  - The other SVD quantities, $\boldsymbol{U}$ and $\boldsymbol{s}$, are deterministic and arbitrary
  - This model includes mean-perturbed and ill-conditioned $\boldsymbol{A}$
- With large, right-rotationally invariant $\boldsymbol{A}$:
  - VAMP has a rigorous state-evolution (SE) [8] when $\boldsymbol{g}$ is Lipschitz and separable.
  - The SE fixed points are "good" in that, with an i.i.d. signal x_j and MMSE scalar denoiser g, they match the replica prediction from [9].
  - Excellent empirical performance in plug-and-play case (see below).
- With structured $\boldsymbol{V}$:
  - VAMP can diverge, especially when $\boldsymbol{V}^{\mathsf{H}}\boldsymbol{x}^0$ is not i.i.d. Gaussian!
  - This occurs, e.g., when $\boldsymbol{V}$ is the DFT matrix and $\boldsymbol{x}^0$ is a natural image!

Image Recovery with i.i.d. Gaussian $\boldsymbol{\Phi}$ (after 20 iterations)





128×128 images

average over:
{*lena, barbara, boat, fingerprint, house, peppers*}

avg over 10 realizations

sampling ratio	10%		20%		30%		40%		50%	
	PSNR	time	PSNR	time	PSNR	time	PSNR	time	PSNR	time
ℓ_1-AMP	17.7	0.7s	20.2	1.5s	22.4	2.3s	24.6	3.3s	27.0	4.7s
ℓ_1-VAMP	17.6	0.8s	20.2	1.4s	22.4	2.0s	24.8	2.6s	27.2	3.4s
BM3D-AMP	25.2	11.3s	30.0	10.0s	32.5	10.1s	35.1	11.0s	37.4	12.3s
BM3D-VAMP	25.2	11.6s	30.0	9.8s	32.5	9.5s	35.2	10.1s	37.7	10.7s

Image Recovery with $\boldsymbol{\Phi} = \operatorname{Diag}(\boldsymbol{s})\boldsymbol{P}\boldsymbol{F}\operatorname{Diag}(\pm 1)$, 10 iterations

Here, $\frac{M}{N} = 0.2$, $\boldsymbol{s}$ logarithmically spaced, $\boldsymbol{P}$ = random permutation, $\boldsymbol{F}$ = DFT2.

condition no.	1		10		10^2		10^3		10^4	
	PSNR	time	PSNR	time	PSNR	time	PSNR	time	PSNR	time
ℓ_1-AMP	22.4	0.03	<0	—	<0	—	<0	—	<0	—
ℓ_1-VAMP	22.9	0.05	22.3	0.05	20.8	0.04	19.6	0.04	18.8	0.04
BM3D-AMP	29.1	4.2s	26.6	4.6s	7.6	—	7.4	—	7.1	—
BM3D-VAMP	29.0	4.2s	29.1	4.1s	27.4	5.1s	25.6	5.2s	24	5.2s

Whitened VAMP for Image Recovery (VAMPire)

- To apply VAMP to image recovery with non-random (e.g., Fourier) $\boldsymbol{\Phi}$, we "whiten" the signal:

$$\boldsymbol{y} = \underbrace{\boldsymbol{\Phi}\boldsymbol{\Psi}\operatorname{Diag}(\boldsymbol{\tau})}_{\boldsymbol{A}}\boldsymbol{s}^0 + \boldsymbol{w} \ \text{for} \ \begin{cases} \boldsymbol{s}^0 : \text{whitened wavelet coefs} \\ \boldsymbol{\tau} : \text{wavelet standard deviations} \end{cases}$$

and use plug-and-play denoising in the whitened-coefficient space:

$$\widehat{\boldsymbol{s}}_t = \boldsymbol{g}(\boldsymbol{r}_t, \sigma_t) = \operatorname{Diag}(\boldsymbol{\tau})^{-1}\boldsymbol{\Psi}^{\mathsf{H}}\text{denoise}\big(\boldsymbol{\Psi}\operatorname{Diag}(\boldsymbol{\tau})\boldsymbol{r}_t; N^{-1/2}\|\boldsymbol{\tau}\|\sigma_t\big).$$

For $\boldsymbol{\Psi}$ we use any orthonormal wavelet transform, and for $\boldsymbol{\tau}$ we assume $+22$dB approximation coefs and -7dB/level detail coefs.

- Since the $\boldsymbol{U}, \boldsymbol{V}$ matrices of the resulting $\boldsymbol{A}$ are no longer fast transforms, we solve (1) approximately via preconditioned LSQR:

$$\widetilde{\boldsymbol{g}}(\widetilde{\boldsymbol{r}}, \widetilde{\sigma}) = \begin{bmatrix} \widetilde{\sigma}\boldsymbol{A} \\ \sigma_w\boldsymbol{I} \end{bmatrix}^{+}\begin{bmatrix} \widetilde{\sigma}\boldsymbol{y} \\ \sigma_w\widetilde{\boldsymbol{r}} \end{bmatrix} = \operatorname{Diag}(\boldsymbol{\tau})^{-1}\begin{bmatrix} \boldsymbol{\Phi}\boldsymbol{\Psi}\widetilde{\sigma}/\sigma_w \\ \operatorname{Diag}(\boldsymbol{\tau})^{-1} \end{bmatrix}^{+}\begin{bmatrix} \boldsymbol{y}\widetilde{\sigma}/\sigma_w \\ \widetilde{\boldsymbol{r}} \end{bmatrix}.$$

- The divergence $\widetilde{\alpha}_t$ is approximated using the Monte-Carlo approximation

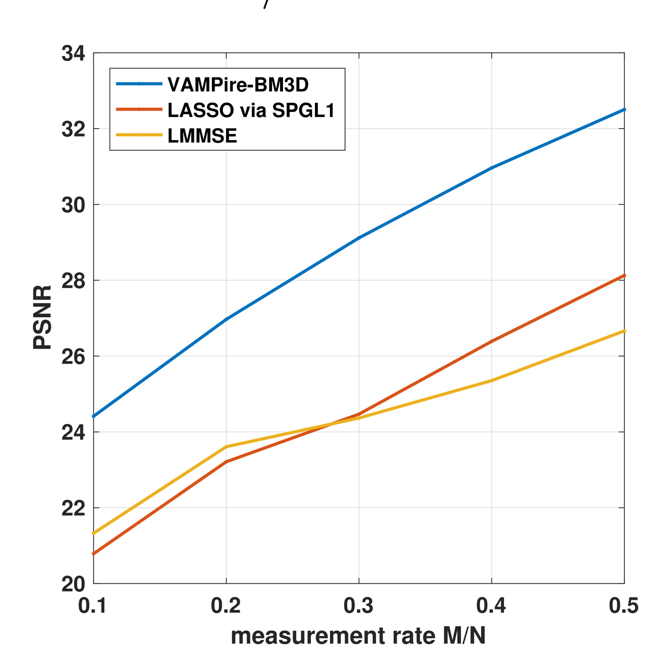
$$\widetilde{\alpha}_t = \frac{\sigma_w^2}{N}\operatorname{tr}\left[(\widetilde{\sigma}_t^2\boldsymbol{A}^{\mathsf{H}}\boldsymbol{A} + \sigma_w^2\boldsymbol{I})^{-1}\right] \approx \frac{1}{NK}\sum_{k=1}^{K}\boldsymbol{p}_k\begin{bmatrix} \widetilde{\sigma}_t\boldsymbol{A} \\ \sigma_w\boldsymbol{I} \end{bmatrix}^{+}\begin{bmatrix} \boldsymbol{0} \\ \sigma_w\boldsymbol{p}_k \end{bmatrix},$$

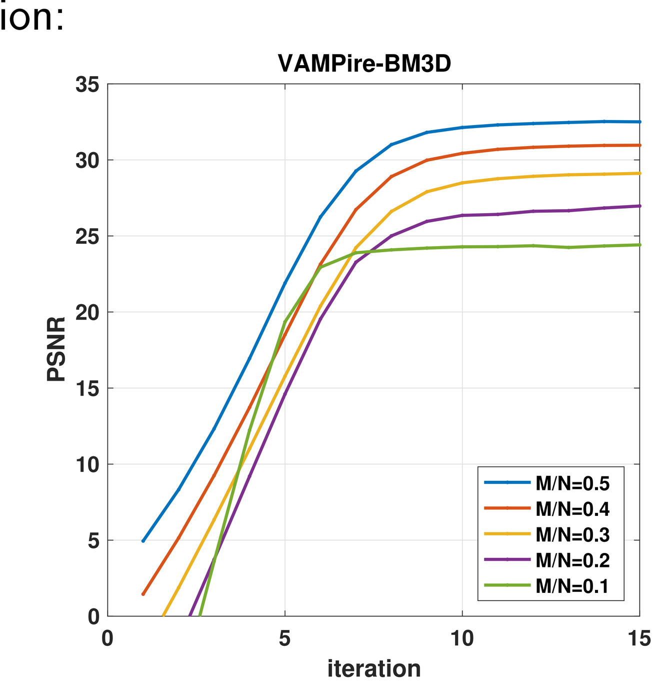
where $\mathrm{E}\{\boldsymbol{p}_k\boldsymbol{p}_k^{\mathsf{H}}\} = \boldsymbol{I}$. Here again, preconditioned LSQR can be used.

- The divergence α_t is approximated a Monte-Carlo approximation inspired by Metzler/Maleki/Baraniuk [6], but different due to the form of $\boldsymbol{g}(\cdot)$.
- Finally, we employ a damping scheme under which VAMP has been proven to converge for any strictly convex $\widetilde{\boldsymbol{g}}$ and $\boldsymbol{g}$.

Image Recovery with Subsampled 2D-Fourier $\boldsymbol{\Phi}$

- Experiment setup:
  - DFT $\boldsymbol{\Phi}$, subsampling pattern from Roman/Adcock/Hansen [10]
  - SNR=40dB
  - 128×128 images {*lena, barbara, boat, fingerprint, house, peppers*}
  - db1 wavelet decomposition, $D = 2$ levels
  - PSNR results averaged over 10 realizations and six images above
- PSNR vs M/N and PSNR vs iteration:





- Example image recovery at $M/N = 0.3$:

- Example image recovery at $M/N = 0.1$: