MRI Image Recovery Using Damped Denoising Vector AMP

Subrata Sarkar, Rizwan Ahmad,
and Phil Schniter

THE OHIO STATE UNIVERSITY

With support from: NSF CCF-1955587 and NIH R01HL135489

ICASSP 2021

Magnetic Resonance Imaging (MRI)

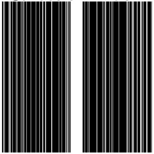
Challenge

- An MRI scan can take more than 45 minutes
- To accelerate MRI, it's common to sample far below the Nyquist rate

Measurement model

- $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}_0 + \boldsymbol{w}$ with $\boldsymbol{A} = \boldsymbol{M}\boldsymbol{F}$
- $\boldsymbol{F} \in \mathbb{C}^{N \times N}$: 2D-DFT matrix
- $\boldsymbol{M} \in \mathbb{R}^{M \times N}$: sampling mask

Cartesian mask



Goal: Recover the unknown image $\boldsymbol{x}_0 \in \mathbb{C}^N$ from noisy k-space measurements $\boldsymbol{y} \in \mathbb{C}^M$ with $M \ll N$

Approach: Plug-and-play recovery using a new algorithm: Damped Denoising Vector-AMP.

Plug-and-Play (PnP) Image Recovery

- The classical approach to image recovery solves the optimization problem

$$\arg\min_{\boldsymbol{x}} \left\{ \frac{\gamma_w}{2} \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|^2 + \phi(\boldsymbol{x}) \right\}$$

where the regularizer $\phi(\cdot)$ penalizes atypical $\boldsymbol{x}$.

- ADMM is a popular algorithm to solve this optimization problem:

$$\begin{aligned} \boldsymbol{x}^{t+1} &= \arg\min_{\boldsymbol{x}} \tfrac{\gamma_w}{2} \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|^2 + \tfrac{\gamma}{2} \|\boldsymbol{x} - \boldsymbol{v}^t + \boldsymbol{u}^t\|^2 \\ \boldsymbol{v}^{t+1} &= \text{prox}_{\gamma^{-1}\phi}(\boldsymbol{x}^{t+1} + \boldsymbol{u}^t) \\ \boldsymbol{u}^{t+1} &= \boldsymbol{u}^t + (\boldsymbol{x}^{t+1} - \boldsymbol{v}^{t+1}), \end{aligned}$$

where $\text{prox}_{\gamma^{-1}\phi}(\boldsymbol{r}) \triangleq \arg\min_{\boldsymbol{x}} \{\phi(\boldsymbol{x}) + \frac{1}{2\gamma} \|\boldsymbol{x} - \boldsymbol{r}\|^2\} \Leftrightarrow$ MAP denoising.

- In PnP-ADMM[1], the prox operator is replaced by a sophisticated image denoiser $\boldsymbol{f}(\cdot)$, like BM3D or DnCNN[2], for much-improved performance.

[1] Venkatakrishnan, Bouman, Wohlberg'13, [2] Zhang, Zuo, Chen, Meng, Zhang'17

Vector Approximate Message Passing (VAMP)

- VAMP[3] is a closely related signal-recovery algorithm.
  - Derived under the assumption of large random $\boldsymbol{A}$.
  - Very similar to ADMM, but adapts the regularization parameter γ.
- When $\boldsymbol{A}$ is right-orthogonally invariant (ROI), i.e., has SVD $\boldsymbol{A} = \boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^{\mathsf{H}}$ with large random unitary $\boldsymbol{V}$:
  - VAMP's macroscopic behavior is rigorously characterized by state-evolution.[3]
  - VAMP converges very quickly, e.g., 5-15 iterations.
  - With MMSE $\boldsymbol{f}$ and unique SE fixed-point, VAMP yields MMSE $\widehat{\boldsymbol{x}}$.
- In "denoising-VAMP" (D-VAMP)[4,5], VAMP's prox step is replaced by a sophisticated image denoiser $\boldsymbol{f}(\cdot)$ like BM3D or DnCNN.
- The MRI measurement matrix $\boldsymbol{A}$ is *not* ROI, and so VAMP may perform poorly or even diverge.

[3] Rangan,Schniter,Fletcher'16, [4] Schniter,Rangan,Fletcher'17, [5] Fletcher,Pandit,Rangan,Sarkar,Sch

Contribution 1: DD-VAMP

- We propose a new D-VAMP damping scheme with improved convergence behavior.
  - Like existing[6] damping schemes, we damp the amplitude quantity $\boldsymbol{r}_2$ and the precision quantity γ_2.
  - But we also damp the Monte-Carlo divergence estimate α_1.
  - And we convert the precision and divergence quantities to amplitudes when damping, and then transform them back.
- We call the algorithm "Damped Denoising VAMP" (DD-VAMP).
- DD-VAMP reduces to D-VAMP when $\theta = 1 = \zeta$.

[6] Rangan,Schniter,Fletcher'16,

The DD-VAMP Algorithm

To recover $\boldsymbol{x}_0$ from $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}_0 + \boldsymbol{w}$ with $\boldsymbol{A} = \boldsymbol{M}\boldsymbol{F}$:

initialize $\boldsymbol{r}_2^0, \gamma_2^0$, $\theta, \zeta \in (0,1]$, $\boldsymbol{q} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$

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

$\boldsymbol{x}_2^t = \boldsymbol{g}(\boldsymbol{r}_2^t; \gamma_2^t)$	linear estimation
$\alpha_2^t = \mathrm{tr}\{\nabla \boldsymbol{g}(\boldsymbol{r}_2^t; \gamma_2^t)\}/N$	divergence
$\boldsymbol{r}_1^t = (\boldsymbol{x}_2^t - \alpha_2^t \boldsymbol{r}_2^t)/(1-\alpha_2^t), \ \gamma_1^t = \gamma_2^t(1-\alpha_2^t)/\alpha_2^t$	Onsager correction
$\boldsymbol{x}_1^t = \boldsymbol{f}(\boldsymbol{r}_1^t; \gamma_1^t)$	denoising
$\overline{\alpha}_1^t = \epsilon^{-1}\boldsymbol{q}^{\mathsf{H}}\left[\boldsymbol{f}(\boldsymbol{r}_1^t + \epsilon \boldsymbol{q}; \gamma_1^t) - \boldsymbol{f}(\boldsymbol{r}_1^t; \gamma_1^t)\right]$	Monte-Carlo divergence
$\alpha_1^t = \left[\theta(\overline{\alpha}_1^t)^{\frac{1}{2}} + (1-\theta)(\alpha_1^{t-1})^{\frac{1}{2}}\right]^2$	damping
$\overline{\boldsymbol{r}}_2^{t+1} = (\boldsymbol{x}_1^t - \alpha_1^t \boldsymbol{r}_1^t)/(1-\alpha_1^t), \ \overline{\gamma}_2^{t+1} = \gamma_1^t(1-\alpha_1^t)/\alpha_1^t$	Onsager correction
$\boldsymbol{r}_2^{t+1} = \zeta \overline{\boldsymbol{r}}_2^{t+1} + (1-\zeta)\boldsymbol{r}_2^t$	damping
$\gamma_2^{t+1} = \left[\zeta(\overline{\gamma}_2^{t+1})^{-\frac{1}{2}} + (1-\zeta)(\gamma_2^t)^{-\frac{1}{2}}\right]^{-2}$	damping

where

$$\boldsymbol{g}(\boldsymbol{r};\gamma) \triangleq \arg\min_{\boldsymbol{x}} \left\{ \frac{\gamma_w}{2}\|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|^2 + \frac{\gamma}{2}\|\boldsymbol{x} - \boldsymbol{r}\|^2 \right\} = \boldsymbol{F}^{\mathsf{H}}(\gamma_w \boldsymbol{M}^{\mathsf{T}}\boldsymbol{M} + \gamma \boldsymbol{I})^{-1}(\gamma \boldsymbol{F}\boldsymbol{r} + \gamma_w \boldsymbol{M}^{\mathsf{T}}\boldsymbol{y})$$

$$\mathrm{tr}\{\nabla \boldsymbol{g}(\boldsymbol{r};\gamma)\}/N = \left((1 - M/N)\gamma_w + \gamma\right)/(\gamma_w + \gamma).$$

Contribution 2: DD-VAMP++

- Empirically, DD-VAMP yields good fixed points.
  - Similar or better than those of PnP-ADMM.
- But the use of damping slows the convergence of DD-VAMP relative to ADMM.
- Importantly, VAMP reduces to the Peaceman-Rachford variant of ADMM (ADMM-PR) when the precisions are fixed, i.e., $\gamma_1^t = \gamma_2^t = \gamma, \forall t$.
- Thus we propose to initialize DD-VAMP using ADMM-PR:
  - Run PnP-ADMM-PR for T_{swi} iterations at precision γ, and then switch to DD-VAMP.
  - Use training data to tune the parameters T_{swi} and γ.
- We call this method "DD-VAMP++."

Experiment: MRI Image Recovery

Goal Recover $\boldsymbol{x}_0 \in \mathbb{C}^N$ from $\boldsymbol{y} = \boldsymbol{M}\boldsymbol{F}\boldsymbol{x}_0 + \boldsymbol{w} \in \mathbb{C}^M$

Experiment Setup

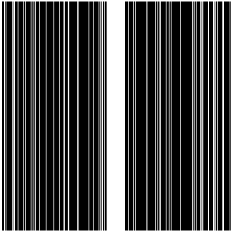
- 128×128 mid-slice, non-fat-suppressed fastMRI[7] knee images.
- Cartesian sampling $\boldsymbol{M}$ with acceleration $N/M = 4$.
- DnCNN[8] denoiser used unless otherwise noted.

Training

- The dataset was randomly split into 30 training and 19 testing images.
- We tuned all algorithmic parameters to minimize NMSE, averaged over iterations $t = 30 \ldots 150$ and medianed over the training images.

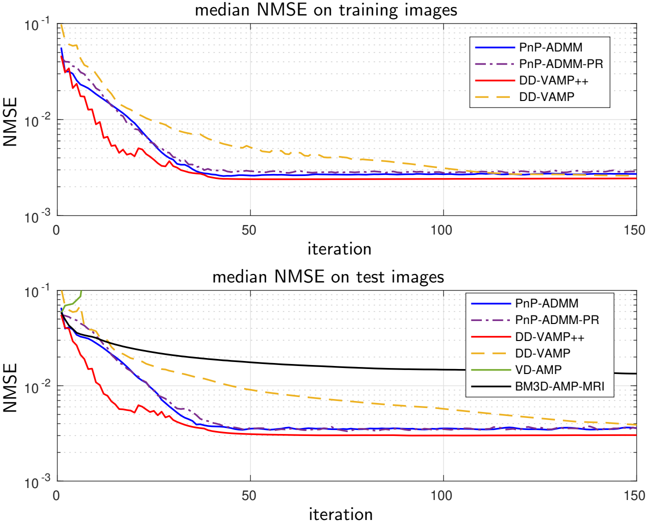
Original

Cartesian Sampling



[7] Zbontar,Knoll,etal'18, [8] Zhang,Zuo,Chen,Meng,Zhang'17

Experiment: MRI Image Recovery

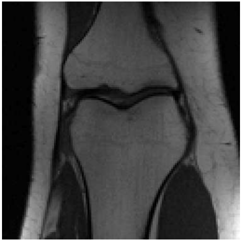


- PnP-ADMM has good fixed points
- DD-VAMP++ converges quickly to better fixed points
- DD-VAMP converges slowly
- BM3D-AMP-MRI[9] has relatively poor fixed points
- VD-AMP[10] diverged

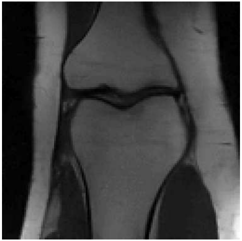
[9] Eksioglu, Tanc–JIS'18, [10] Millard, Hess, Mailhe, Tanner–'20

Experiment: MRI Image Recovery

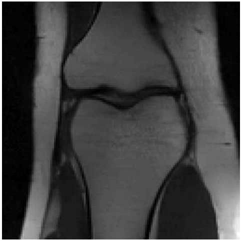
ground truth



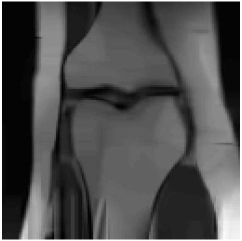
ADMM: -24.443, 0.928



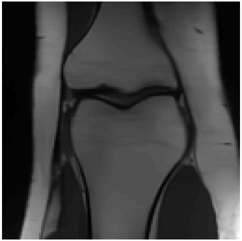
ADMM-PR: -24.694, 0.927



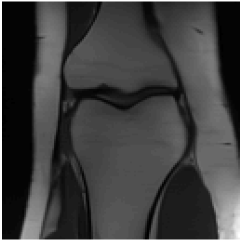
BM3D-AMP-MRI: -19.263, 0.857



DD-VAMP: -25.097, 0.936



DD-VAMP++: **-25.475**, **0.939**



Plot titles report NMSE (dB) and SSIM after 150 iterations

Summary

- Our approach builds on "denoising-VAMP" (D-VAMP), which is an optimal algorithm for large ROI $\boldsymbol{A}$ but fails in practical MRI.
- We propose DD-VAMP and DD-VAMP++ and apply them to MRI image recovery.
    - DD-VAMP uses a novel damping scheme
    - DD-VAMP++ adds an ADMM-PR-based initialization
- Empirical results suggest that both algorithms have better fixed points than PnP-ADMM, and that DD-VAMP++ converges faster than PnP-ADMM.