Memory AMP and Recent Results on Its Implementation

Brian M. Kurkoski
Joint work with Shunqi Huang and Lei Liu

Japan Advanced Institute of Science and Technology and Zhejiang University

March 13, 2024
BIRS Workshop — Algorithmic Structures for Uncoordinated Communications and Statistical Inference in Exceedingly Large Spaces

Outline

Problem Formulation

- System model:

$$\begin{aligned} \Gamma: &\quad \boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{n}, \\ \Phi: &\quad x_i \sim P_X(x),\ \forall i. \end{aligned}$$

  where $\boldsymbol{A} \in \mathbb{C}^{M\times N}$, $\boldsymbol{n} \sim \mathcal{CN}(0, \sigma^2 \boldsymbol{I}_M)$, and $\boldsymbol{A}, \boldsymbol{y}, \sigma^2, P_X(\cdot)$ are known.
- Assumptions:
  (1) $M, N \to \infty$ with fixed $\delta = M/N$.
  (2) $\boldsymbol{A}$ is right-unitarily-invariant.
  (3) $\boldsymbol{x}$ is IID. For convenience, $\mathrm{E}\{\boldsymbol{x}\} = \boldsymbol{0}$ and $\frac{1}{N}\mathrm{E}\{\|\boldsymbol{x}\|^2\} = 1$.
- Goal: Given $\{\boldsymbol{y}, \boldsymbol{A}, \Gamma, \Phi\}$, find an MMSE estimate of $\boldsymbol{x}$:

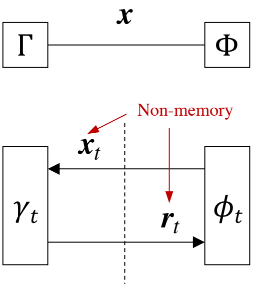
$$MSE \to \mathrm{mmse}\{\boldsymbol{x}|\boldsymbol{y}, \boldsymbol{A}, \Gamma, \Phi\}$$

  For non-Gaussian $\boldsymbol{x}$, without the assumptions of $M, N \to \infty$ and $\boldsymbol{A}$, finding the optimal solution is generally NP-hard.

Approximate Message Passing (AMP)

- AMP-type algorithms:

$$\text{linear estimator (LE)}: \quad \boldsymbol{r}_t = \gamma_t(\boldsymbol{x}_t),$$
$$\text{non}-\text{linear esitmator (NLE)}: \quad \boldsymbol{x}_{t+1} = \phi_t(\boldsymbol{r}_t).$$



- AMP:

$$\text{LE}: \quad \boldsymbol{r}_t = \boldsymbol{x}_t + \boldsymbol{A}^{\mathrm{H}}(\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}_t) + \boldsymbol{r}_t^{\text{Onsager}},$$
$$\text{NLE}: \quad \boldsymbol{x}_{t+1} = \phi(\boldsymbol{r}_t) = \mathrm{E}\{\boldsymbol{x}|\boldsymbol{r}_t\},$$

where $\boldsymbol{r}_t^{\text{Onsager}} = \beta\langle\phi'(\boldsymbol{r}_{t-1})\rangle(\boldsymbol{r}_{t-1} - \boldsymbol{x}_{t-1})$.

✓ Bayes optimal
✓ Low-complexity
× IID $\boldsymbol{A}$ is required

□ D. L. Donoho, A. Maleki, and A. Montanari, "Message-passing algorithms for compressed sensing," in *Proc. Nat. Acad. Sci.*, 2009.

Orthogonal/Vector AMP (OAMP/VAMP)

- OAMP/VAMP:

$$\text{LE}: \ \boldsymbol{r}_t = \boldsymbol{x}_t + \tfrac{1}{\epsilon_t^\gamma}\boldsymbol{A}^{\mathrm{H}}\big(\rho_t\boldsymbol{I} + \boldsymbol{A}\boldsymbol{A}^{\mathrm{H}}\big)^{-1}(\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}_t),$$

$$\text{NLE}: \ \boldsymbol{x}_{t+1} = \tfrac{1}{\epsilon_{t+1}^\phi}\big[\hat{\phi}_t(\boldsymbol{r}_t) + (1-\epsilon_{t+1}^\phi)\boldsymbol{r}_t\big],$$

where ϵ_t^γ and ϵ_t^ϕ are orthogonal parameters.

✓ Bayes optimal (replica)

✓ Unitarily-invariant $\boldsymbol{A}$

× High-complexity

□ J. Ma and L. Ping, "Orthogonal AMP," *IEEE Access*, 2017.

□ S. Rangan, P. Schniter, and A. Fletcher, "Vector approximate message passing," *IEEE Trans. Inf. Theory*, 2019.

Convolutional AMP (CAMP)

- CAMP:

$$\begin{aligned} \text{LE}: \quad & \boldsymbol{r}_t = \boldsymbol{x}_t + \boldsymbol{A}^H(\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}_t) + \boldsymbol{r}_t^{\text{Onsager}}, \\ \text{NLE}: \quad & \boldsymbol{x}_{t+1} = \phi(\boldsymbol{r}_t) = \text{E}\{\boldsymbol{x}|\boldsymbol{r}_t\}, \end{aligned}$$

where $\boldsymbol{r}_t^{\text{Onsager}} = \sum_{\tau=0}^{t-1} \left[\prod_{t'=\tau}^{t-1} \langle \phi'(\boldsymbol{r}_{t'}) \rangle \right] (\theta_{t-\tau} \boldsymbol{A}^{\text{T}} \boldsymbol{A} - g_{t-\tau})(\boldsymbol{r}_\tau - \boldsymbol{x}_\tau)$.

✓ Bayes optimal (replica), if converges

✓ Unitarily-invariant $\boldsymbol{A}$

✓ Low-complexity

× Fails to converge for $\boldsymbol{A}$ with high condition numbers

□ K. Takeuchi, "Bayes-optimal convolutional AMP," *IEEE Trans. Inf. Theory*, 2021.

Memory AMP

- Memory AMP (MAMP):

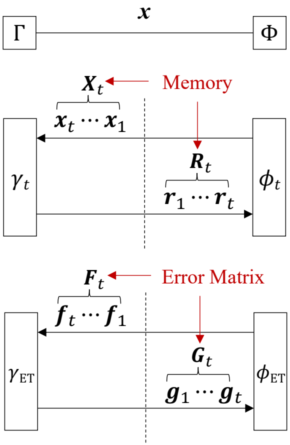
$$\begin{aligned} \text{LE}: \quad & \boldsymbol{r}_t = \gamma_t(\boldsymbol{X}_t) = \boldsymbol{Q}_t\boldsymbol{y} + \textstyle\sum_{i=1}^{t} \boldsymbol{P}_{t,i}\boldsymbol{x}_i, \\ \text{NLE}: \quad & \boldsymbol{x}_{t+1} = \phi_t(\boldsymbol{R}_t)\,, \end{aligned}$$

under orthogonality: $\forall t \geq 1$,

$$\tfrac{1}{N}\boldsymbol{g}_t^{\rm H}\boldsymbol{x} \overset{\rm a.s.}{=} 0, \quad \tfrac{1}{N}\boldsymbol{F}_t^{\rm H}\boldsymbol{g}_t \overset{\rm a.s.}{=} \boldsymbol{0}, \quad \tfrac{1}{N}\boldsymbol{G}_t^{\rm H}\boldsymbol{f}_{t+1} \overset{\rm a.s.}{=} \boldsymbol{0},$$

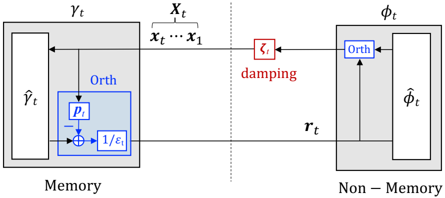
where $\boldsymbol{G}_t$ and $\boldsymbol{F}_t$ are error matrices.

- AMP, OAMP/VAMP, CAMP can be unified under MAMP.



☐ L. Liu, S. Huang, and B. M. Kurkoski, "Memory AMP," *IEEE Trans. Inf. Theory*, 2022.

Bayes-Optimal MAMP (BO-MAMP) — Principle



- The LE has memory and consists of a local estimator $\hat{\gamma}_t$ and orthogonalization.
  1. $\hat{\gamma}_t$ approaches $(\rho_t \boldsymbol{I} + \boldsymbol{A}\boldsymbol{A}^{\mathrm{H}})^{-1}(\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}_t)$ in OAMP/VAMP.
  2. The error of $\boldsymbol{r}_t$ is orthogonal to $\boldsymbol{x}$ and the errors of $\boldsymbol{x}_1, \cdots, \boldsymbol{x}_t$.
- Damping ζ is added. ζ is analytically optimized to guarantee convergence. Also improves convergence speed.
- NLE: Same as OAMP/VAMP and has no memory.

MAMP with Gradient Descent

GD-MAMP:

$$\begin{aligned} \text{LE}: \quad \boldsymbol{u}_t &= \theta_t \boldsymbol{B}\boldsymbol{u}_{t-1} + \xi_t(\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}_t), \\ \boldsymbol{r}_t &= \gamma_t(\boldsymbol{X}_t) = \tfrac{1}{\varepsilon_t^\gamma}\big(\boldsymbol{A}^{\mathrm{H}}\boldsymbol{u}_t + \textstyle\sum_{i=1}^{t} p_{t,i}\boldsymbol{x}_i\big), \\ \text{NLE}: \boldsymbol{x}_{t+1} &= \big[\boldsymbol{x}_1 \cdots \boldsymbol{x}_t \ \phi_t(\boldsymbol{r}_t)\big] \cdot \zeta_{t+1}, \end{aligned}$$

- $\boldsymbol{u}_t$ is an estimate of $(\rho_t \boldsymbol{I} + \boldsymbol{A}\boldsymbol{A}^{\mathrm{H}})^{-1}(\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}_t)$.
- The parameters θ_t and ξ_t are optimized, $p_{t,i}$ and ε_t^γ are chosen to ensure orthogonality. Computation not shown, but we'll more talk about these later.
- ζ_{t+1} is the optimized damping vector
- $\phi_t(\cdot)$ is the same as that in OAMP/VAMP,

Gradient descent (GD) is used to approximate $\frac{\xi_t}{\theta_t}(\rho_t \boldsymbol{I} + \boldsymbol{A}\boldsymbol{A}^{\mathrm{H}})^{-1}(\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}_t)$ by $\boldsymbol{u}_t$.

Why Memory? Intuition 1: Gradient Descent Avoids Matrix Inverse

Want to eliminate matrix inverse in OAMP: $\frac{\xi_t}{\theta_t}(\rho_t \boldsymbol{I} + \boldsymbol{A}\boldsymbol{A}^{\mathrm{T}})^{-1}(\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}_t)$.
Solve $\boldsymbol{W}\boldsymbol{u} = \boldsymbol{b}$ without finding $\boldsymbol{W}^{-1}$.

Intuition 1 Solving $\boldsymbol{u} = \boldsymbol{W}^{-1}\boldsymbol{b}$ is equivalent to

$$\arg\min f(\boldsymbol{u}) = \tfrac{1}{2}\boldsymbol{u}^{\mathrm{T}}\boldsymbol{W}\boldsymbol{u} - \boldsymbol{b}^{\mathrm{T}}\boldsymbol{u}$$

when $\boldsymbol{W}$ is positive definite. Find solution using gradient descent with step:

$$\begin{aligned}\boldsymbol{u}_i &= \boldsymbol{u}_{i-1} - \alpha \nabla f(\boldsymbol{u}_{i-1}) \\ &= \boldsymbol{u}_{i-1} + \alpha(\boldsymbol{b} - \boldsymbol{W}\boldsymbol{u}_{i-1}).\end{aligned}$$

Then $\boldsymbol{u}_i$ approaches the correct value $\boldsymbol{u}$.

Choosing $\alpha = \frac{2}{\lambda_{\max} + \lambda_{\min}}$ is close to optimal. Shown for real-valued case.

ChatGPT: Is it possible to find a matrix inverse using gradient descent?

ChatGPT

While there might be unconventional methods or iterative approaches that can approximate a matrix inverse, gradient descent is not the typical choice for this particular problem. If you need to find the inverse of a matrix, it's recommended to use established linear algebra techniques for accuracy and efficiency.

Why Memory? Intuition 2: Neumann Series for Matrix Inverse

Let $\rho(\boldsymbol{C})$ denote the spectral radius of $\boldsymbol{C}$. If $\rho(\boldsymbol{C}) < 1$, then

$$(\boldsymbol{I} - \boldsymbol{C})^{-1} = \sum_{i=0}^{\infty} \boldsymbol{C}^i.$$

Choose $\boldsymbol{C} = \boldsymbol{I} - \boldsymbol{W}$. When $\rho(\boldsymbol{C}) \geq 1$, let $\boldsymbol{C}' = \boldsymbol{I} - \theta(\boldsymbol{I} - \boldsymbol{C})$, where θ ensures $\rho(\boldsymbol{C}') < 1$. With $\boldsymbol{u}_0 = \boldsymbol{0}$:

$$\boldsymbol{u}_i = \boldsymbol{C}'\boldsymbol{u}_{i-1} + \theta\boldsymbol{b}.$$

Then $\boldsymbol{u}_i$ approaches $\boldsymbol{u} = \boldsymbol{W}^{-1}\boldsymbol{b}$. This iteration is identical to gradient descent.

To accelerate convergence, we can minimize $\rho(\boldsymbol{C}')$ by:

$$\theta = \frac{2}{\lambda_1 + \lambda_2},$$

where λ_1 and λ_2 denote the maximum and minimum eigenvalues of $\boldsymbol{I} - \boldsymbol{C}$.

Overview of AMP-Type Algorithms

Table 1: Overview of AMP-Type Algorithms

Algorithm	Matrix $\boldsymbol{A}$	Convergence	Time complexity	Optimality
AMP	IID	Converges	Low: $\mathcal{O}(MN)$	Bayes-optimal
OAMP/VAMP	Right unitarily invariant	Converges	High: $\mathcal{O}(M^2N)$	Bayes-optimal
CAMP	Right unitarily invariant	Diverges in high condition numbers	Low: $\mathcal{O}(MN)$	Bayes-optimal
GD-MAMP	Right unitarily invariant	Converges	Low: $\mathcal{O}(MN)$	Bayes-optimal

- D. L. Donoho, A. Maleki, and A. Montanari, "Message-passing algorithms for compressed sensing," in *Proc. Nat. Acad. Sci.*, 2009.
- J. Ma and L. Ping, "Orthogonal AMP," *IEEE Access*, 2017.
- S. Rangan, P. Schniter, and A. Fletcher, "Vector approximate message passing," *IEEE Trans. Inf. Theory*, 2019.
- K. Takeuchi, "Bayes-optimal convolutional AMP," *IEEE Trans. Inf. Theory*, 2021.
- L. Liu, S. Huang, and B. M. Kurkoski, "Memory AMP," *IEEE Trans. Inf. Theory*, 2022.

Comparison of AMP-Type Algorithms

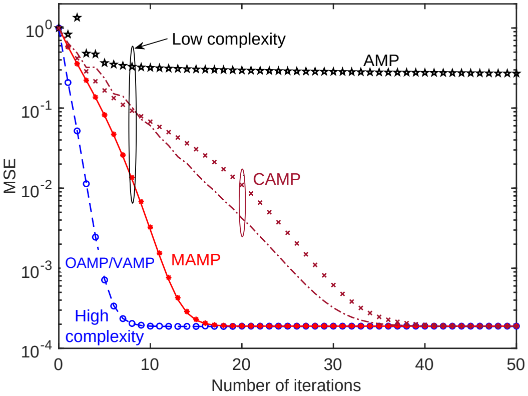


Figure 1: $M = 2^{13}, N = 2^{14}, \kappa(\boldsymbol{A}) = 10, \mathrm{SNR} = 30\mathrm{dB}$

- AMP:
  low complexity
  poor MSE
- OAMP/VAMP:
  fastest convergence
  high complexity
- CAMP:
  low complexity
  slow convergence
  incorrect state evolution
- GD-MAMP:
  low complexity
  fast convergence
  correct state evolution

Outline

1. Preliminaries
   - Problem Formulation
   - AMP, OAMP/VAMP, and CAMP
   - MAMP and GD-MAMP
2. Overflow-Avoiding GD-MAMP
   - Overflow Problem in GD-MAMP
   - OA-GD-MAMP with eigenvalues of $\boldsymbol{A}\boldsymbol{A}^{\mathrm{H}}$
   - OA-GD-MAMP without eigenvalues of $\boldsymbol{A}\boldsymbol{A}^{\mathrm{H}}$
3. Complexity-Reduced GD-MAMP
   - Complexity analysis of GD-MAMP
   - Complexity-Reduced GD-MAMP
4. Conclusion

Overflow Problem in GD-MAMP

- Representation of floating point numbers from the IEEE 754 technical standard:
  - binary32 (single precision): $\pm 1.18 \times 10^{-38} \sim \pm 3.4 \times 10^{38}$
  - binary64 (double precision): $\pm 2.23 \times 10^{-308} \sim \pm 1.8 \times 10^{308}$

  Double-precision is widely used, including in Matlab and Python
- The dreaded NaN will appear for values that are too large.
- Overflow problem: In GD-MAMP, some intermediate variables may increase exponentially, and exceed the maximum value of double precision.

Which intermediate variables cause overflow in GD-MAMP?

- In iteration t, the parameters (1) ξ_t (2) $p_{t,i}$ and (3) $v^{\gamma}_{t,t}$ (variance of $\boldsymbol{r}_t$) require w_t.
- $\lambda^{\dagger} = (\lambda_{\max} + \lambda_{\min})/2$ and $\boldsymbol{B} = \lambda^{\dagger}\boldsymbol{I} - \boldsymbol{A}\boldsymbol{A}^{\mathrm{H}}$, where $\lambda_{\max}, \lambda_{\min}$ denotes the maximum and minimum eigenvalues of $\boldsymbol{A}\boldsymbol{A}^{\mathrm{H}}$.

$$b_k \equiv \frac{1}{N}\mathrm{tr}\{\boldsymbol{B}^k\},$$
$$w_k \equiv \frac{1}{N}\mathrm{tr}\{\boldsymbol{A}^{\mathrm{H}}\boldsymbol{B}^k\boldsymbol{A}\} = \lambda^{\dagger} b_k - b_{k+1}.$$

- b_k can be computed if the eigenvalues of $\boldsymbol{A}\boldsymbol{A}^{\mathrm{H}}$ are known. Otherwise, there are simple methods to approximate $\lambda_{\max}, \lambda_{\min}$ and b_k.
- If $\lambda_{\max} > \lambda_{\min} + 2$, the spectral radius $\rho(\boldsymbol{B}) > 1$, b_{2k} increases exponentially.
- Even w_k may increase exponentially.

Overflow of w_k

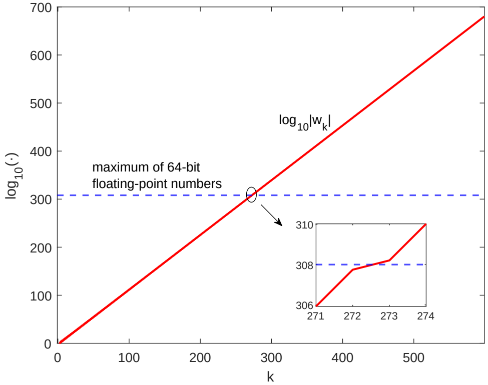


Figure 2: $\log_{10}|w_k|$ versus k

- $w_1, \cdots, w_{2T}$ are required, where T is the maximum number of iterations.
- As shown in Figure 2, $|w_k|$ increases exponentially as k increases.

Overflow Problem in GD-MAMP

- To compute ξ_t, $p_{t,i}$ and $v_{t,t}^{\gamma}$, we need w_k:

$$b_k \equiv \tfrac{1}{N}\mathrm{tr}\{\boldsymbol{B}^k\},$$
$$w_k \equiv \tfrac{1}{N}\mathrm{tr}\{\boldsymbol{A}^{\mathrm{H}}\boldsymbol{B}^k\boldsymbol{A}\} = \lambda^{\dagger} b_k - b_{k+1}.$$

- While w_k increases exponentially, it always appears in the product ϑw_k, which is bounded (i.e. ϑ is small).
- For any $\vartheta \in \mathbb{R}\backslash\{0\}$, the following holds:

$$\vartheta w_k = \frac{\mathrm{sgn}(\vartheta)}{N}\mathbf{1}^{\mathrm{T}}\left[(\lambda^{\dagger}\mathbf{1} - \boldsymbol{\lambda}_{\boldsymbol{B}}) \circ \boldsymbol{s}_{\lambda}^{\circ k} \circ \mathrm{e}^{\circ \log|\vartheta|\mathbf{1} + k\boldsymbol{\lambda}_B^{\log}}\right],$$

where $\boldsymbol{\lambda}_B$ denotes the eigenvalues of $\boldsymbol{B}$, $\boldsymbol{s}_{\lambda} \equiv \mathrm{sgn}(\boldsymbol{\lambda}_B)$, $\boldsymbol{\lambda}_B^{\log} \equiv \log^{\circ}|\boldsymbol{\lambda}_B|$ and $\circ$ is component-wise operation.

Overflow-Avoiding GD-MAMP with Eigenvalues of $\boldsymbol{A}\boldsymbol{A}^{\mathrm{H}}$

- Theorem 1 requires $\mathcal{O}(M)$ computations, GD-MAMP requires to compute $\mathcal{O}(T^3)$ terms involving w_k, where T is the number of iterations. The overall complexity is $\mathcal{O}(MT^3)$. How to reduce the complexity?
- Define

$$\chi_k \equiv \theta_0^k w_k,$$

where $\theta_0 = (\lambda^\dagger + \sigma^2)^{-1} > 0$. We pre-compute $\chi_1, \cdots, \chi_{2T-1}$ before the iterations. Computing ϑw_k can be reduced to a scalar operation:

$$\vartheta w_k = \mathrm{sgn}(\vartheta) \mathrm{e}^{\log|\alpha| - k \log \vartheta} \chi_k.$$

- Pre-computing $\chi_1, \cdots, \chi_{2T-1}$ costs $\mathcal{O}(MT)$, and computing terms involving w_k in iterations costs $\mathcal{O}(T^3)$. The overall complexity is reduced to $\mathcal{O}(MT + T^3)$.

Overflow-Avoiding GD-MAMP with Eigenvalues of $\boldsymbol{A}\boldsymbol{A}^{\mathrm{H}}$

Theorem (2)

For any $k \geq 0$,

$$|\chi_k| \leq \delta(\lambda^\dagger + \theta_0^{-1}),$$

where $\delta = M/N$.

- Theorem 2 shows that χ_k is bounded. In other words, computing χ_k has no risks of overflow.

Overflow-Avoiding GD-MAMP without Eigenvalues of $\boldsymbol{A}\boldsymbol{A}^{\mathrm{H}}$

In large-scale systems, computation of eigenvalues of $\boldsymbol{A}\boldsymbol{A}^{\mathrm{H}}$ may be impractical.

- A method to estimate the maximum and minimum eigenvalue $\lambda_{\max}$ and $\lambda_{\min}$ was given in [LHK22].
- For $k \geq 0$, χ_k can be estimated by

$$\chi_k = \bar{\boldsymbol{h}}_i^{\mathrm{H}} \bar{\boldsymbol{h}}_{k-i}$$

where $i = \lceil k/2 \rceil$ and $\bar{\boldsymbol{h}}_i$ is given by a recursion

$$\bar{\boldsymbol{h}}_i = \theta_0(\lambda^{\dagger}\boldsymbol{I} - \boldsymbol{A}\boldsymbol{A}^{\mathrm{H}})\bar{\boldsymbol{h}}_{i-1}$$

with $\bar{\boldsymbol{h}}_0 = \boldsymbol{A}\boldsymbol{h}_0$, $\boldsymbol{h}_0 \sim \mathcal{N}(\mathbf{0}, \frac{1}{N}\boldsymbol{I}_N)$.

Simulation Results

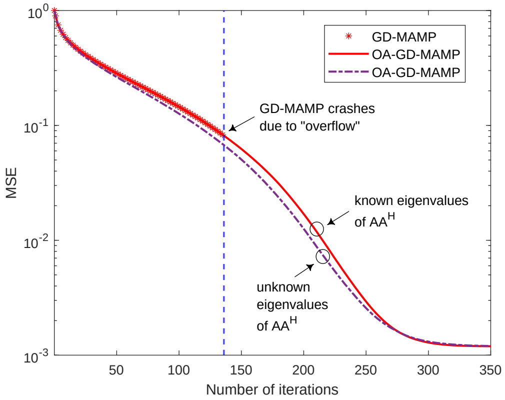


Figure 3: $M = 2^{13}, N = 2^{14}, \kappa(\boldsymbol{A}) = 1000, \mathrm{SNR} = 35\mathrm{dB}$

- When $t > 136$, the unmodified GD-MAMP does not reach the fixed point since b_{137} and w_{137} overflows.
- Both OA-GD-MAMP with and without eigenvalues of $\boldsymbol{A}\boldsymbol{A}^{\mathrm{H}}$ work properly.

Outline

1. Preliminaries
   - Problem Formulation
   - AMP, OAMP/VAMP, and CAMP
   - MAMP and GD-MAMP
2. Overflow-Avoiding GD-MAMP
   - Overflow Problem in GD-MAMP
   - OA-GD-MAMP with eigenvalues of $\boldsymbol{A}\boldsymbol{A}^{\mathrm{H}}$
   - OA-GD-MAMP without eigenvalues of $\boldsymbol{A}\boldsymbol{A}^{\mathrm{H}}$
3. Complexity-Reduced GD-MAMP
   - Complexity analysis of GD-MAMP
   - Complexity-Reduced GD-MAMP
4. Conclusion

Complexity Analysis of GD-MAMP

Let T be the number of iterations. The main complexity of GD-MAMP is $\mathcal{O}(MNT)$, dominated by the number of matrix-vector products, each with $\mathcal{O}(MN)$

$$\begin{aligned}\text{LE}:\quad \boldsymbol{u}_t &= \theta_t \boldsymbol{B}\boldsymbol{u}_{t-1} + \xi_t(\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}_t),\\ \boldsymbol{r}_t &= \tfrac{1}{\varepsilon_t^\gamma}\big(\boldsymbol{A}^{\mathrm{H}}\boldsymbol{u}_t + \textstyle\sum_{i=1}^{t} p_{t,i}\boldsymbol{x}_i\big),\\ \text{NLE}: \boldsymbol{x}_{t+1} &= \big[\boldsymbol{x}_1 \cdots \boldsymbol{x}_t \; \phi_t(\boldsymbol{r}_t)\big]\cdot \zeta_{t+1}.\end{aligned}$$

GD-MAMP requires 4 matrix-vector products per iteration?

- Computing $\boldsymbol{A}\phi_{t-1}(\boldsymbol{r}_{t-1})$ to estimate $v^{\phi}_{t,1}, \cdots, v^{\phi}_{t,t}$ requires one (hidden in ζ_t).
- Computing $\boldsymbol{B}\boldsymbol{u}_{t-1} = (\lambda^{\dagger}\boldsymbol{I} - \boldsymbol{A}\boldsymbol{A}^{\mathrm{H}})\boldsymbol{u}_{t-1}$ requires two.
- Computing $\boldsymbol{A}^{\mathrm{H}}\boldsymbol{u}_t$ requires one.

Easily eliminate one matrix-vector product:

- Two of the products are $\boldsymbol{A}^{\mathrm{H}}\boldsymbol{u}_t$, these only need to be computed once.

GD-MAMP requires 3 matrix-vector products per iteration!

Complexity-Reduced GD-MAMP Using Approximate ξ_t

Finding the damping vector ζ_t nominally requires one matrix-vector product:

1. Compute $\boldsymbol{z}_t = \boldsymbol{y} - \boldsymbol{A}\phi_{t-1}(\boldsymbol{r}_{t-1})$.
2. Estimate $v^{\phi}_{t,1}, \cdots, v^{\phi}_{t,t}$ by using $\boldsymbol{z}_t$, where $v^{\phi}_{t,i}$ denotes the covariance of $\phi_{t-1}(\boldsymbol{r}_{t-1})$ and $\boldsymbol{x}_i$ for $i < t$.
3. Compute ζ_t from the covariance matrix $\boldsymbol{V}^{\phi}_t$ of $\boldsymbol{x}_1, \cdots, \boldsymbol{x}_{t-1}$ and $\phi_{t-1}(\boldsymbol{r}_{t-1})$.

To remove the above matrix-vector product:
(1) We move the damping from the NLE to the LE (details omitted).
(2) ξ_t nominally depends on $\boldsymbol{z}_t$ and $\boldsymbol{V}^{\phi}_t$. But we found that approximating ξ_t gave little to no performance loss:

$$\tilde{\xi}_t = 1/(v^{\phi}_{t,t} + \sigma^2).$$

where $v^{\phi}_{t,t}$ is the variance of $\boldsymbol{x}_t$, given as that in OAMP/VAMP.

- The resulting complexity-reduced GD-MAMP requires only 2 matrix-vector products per iteration.

Simulation Results

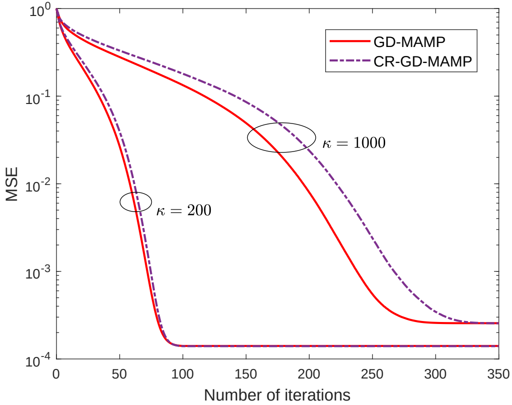


Figure 4: MSE versus number of iterations, $M = 2^{13}, N = 2^{14}, \mathrm{SNR} = 35\mathrm{dB}$

- CR-GD-MAMP requires a few more iterations to converge.

Simulation Results

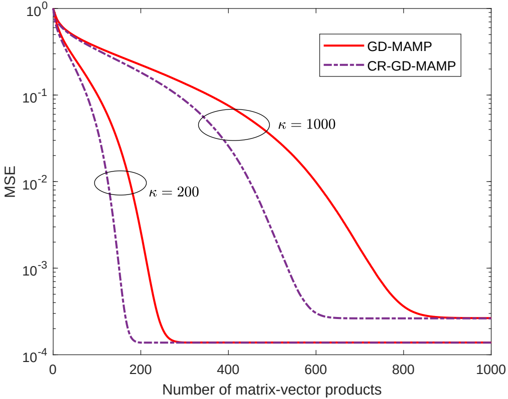


Figure 5: MSE versus number of matrix-vector products, $M = 2^{13}, N = 2^{14}, \mathrm{SNR} = 35\mathrm{dB}$

- CR-GD-MAMP achieves the same MSE as GD-MAMP while requiring about 2/3 matrix-vector products.

Conclusion

GD-MAMP is a memory AMP algorithm that overcomes the weakness of AMP, OAMP/VAMP and CAMP.

(1) To solve the overflow problem, we propose OA-GD-MAMP:

- With known eigenvalues of $\boldsymbol{A}\boldsymbol{A}^{\mathrm{H}}$, OA-GD-MAMP is equivalent to GD-MAMP.
- Otherwise, OA-GD-MAMP can achieve nearly the same performance.

(2) GD-MAMP requires three matrix-vector products per iteration. To reduce it:

- We propose CR-GD-MAMP as a variant of GD-MAMP. It requires only two matrix-vector products per iteration with almost the same convergence speed.