OpAMP: Linear Operator Approximate Message Passing

Riccardo Rossetti
Duke

Bobak Nazer
BU

Galen Reeves
Duke

BIRS Workshop
Algorithmic Structures for Uncoordinated Communications and Statistical Inference in Exceedingly Large Spaces

March 13, 2024

Power Method (i.e., Power Iteration)

- Let $M \in \mathbb{R}^{n \times n}$ be a symmetric matrix.

Power Method (i.e., Power Iteration)

- Let $M \in \mathbb{R}^{n \times n}$ be a symmetric matrix.
- Say we want to estimate the eigenvector $v_1 \in \mathbb{R}^n$ corresponding to the largest magnitude eigenvalue λ_1.

Power Method (i.e., Power Iteration)

- Let $M \in \mathbb{R}^{n\times n}$ be a symmetric matrix.
- Say we want to estimate the eigenvector $v_1 \in \mathbb{R}^n$ corresponding to the largest magnitude eigenvalue λ_1.

Power Method:

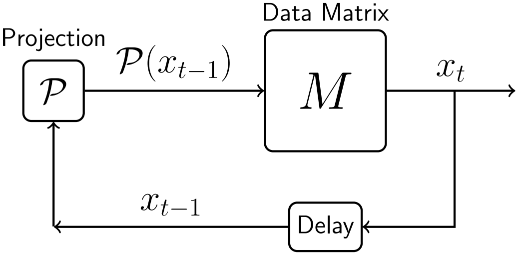
$$x_t = M\hat{v}_{t-1} \qquad \hat{v}_t = \frac{x_t}{\|x_t\|}$$

Power Method (i.e., Power Iteration)

- Let $M \in \mathbb{R}^{n\times n}$ be a symmetric matrix.
- Say we want to estimate the eigenvector $v_1 \in \mathbb{R}^n$ corresponding to the largest magnitude eigenvalue λ_1.

Power Method:

$$x_t = M\hat{v}_{t-1} \qquad \hat{v}_t = \frac{x_t}{\|x_t\|}$$

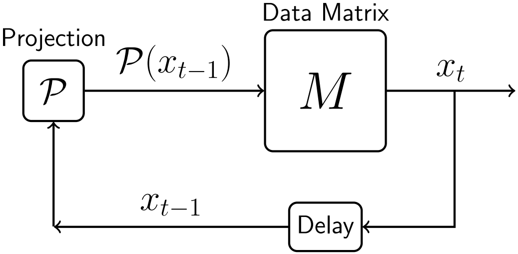


Power Method (i.e., Power Iteration)

- Let $M \in \mathbb{R}^{n \times n}$ be a symmetric matrix.
- Say we want to estimate the eigenvector $v_1 \in \mathbb{R}^n$ corresponding to the largest magnitude eigenvalue λ_1.

Power Method:

$$x_t = M\hat{v}_{t-1} \qquad \hat{v}_t = \frac{x_t}{\|x_t\|}$$



- Classical error bound depends on the spectral gap, vanishing like $\left(\frac{\lambda_2}{\lambda_1}\right)^t$.

Distributed Power Method

- What about a "distributed" power method for very large matrices?

Distributed Power Method

- What about a "distributed" power method for very large matrices?
- Partition rows of the data matrix into J equally-sized submatrices: $M = \begin{bmatrix} M_1 \\ M_2 \\ \vdots \\ M_J \end{bmatrix}$

Distributed Power Method

- What about a "distributed" power method for very large matrices?
- Partition rows of the data matrix into J equally-sized submatrices: $M = \begin{bmatrix} M_1 \\ M_2 \\ \vdots \\ M_J \end{bmatrix}$
- Give each submatrix to a server.

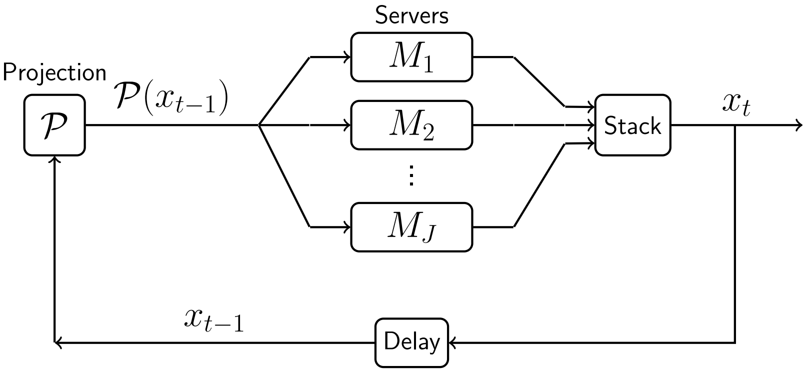
Distributed Power Method

- What about a "distributed" power method for very large matrices?
- Partition rows of the data matrix into J equally-sized submatrices: $M = \begin{bmatrix} M_1 \\ M_2 \\ \vdots \\ M_J \end{bmatrix}$
- Give each submatrix to a server.

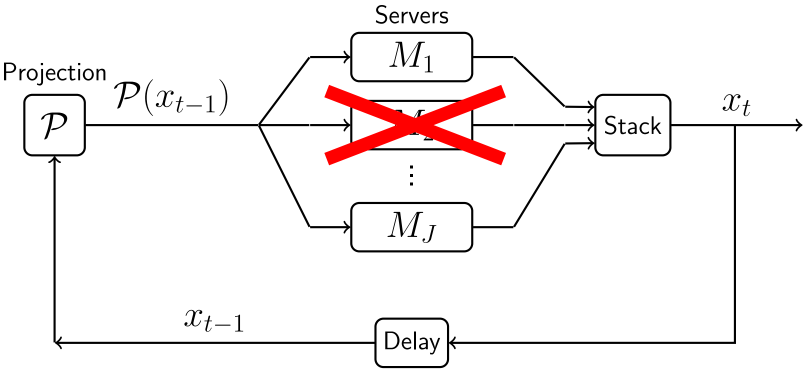
Distributed Power Method:

$$x_t = \begin{bmatrix} x_{t,1} \\ x_{t,2} \\ \vdots \\ x_{t,J} \end{bmatrix} \qquad \begin{aligned} x_{t,1} &= M_1 \hat{v}_{t-1} \\ x_{t,2} &= M_2 \hat{v}_{t-1} \\ &\vdots \\ x_{t,J} &= M_J \hat{v}_{t-1} \end{aligned} \qquad \hat{v}_t = \frac{x_t}{\|x_t\|}$$

Distributed Power Method

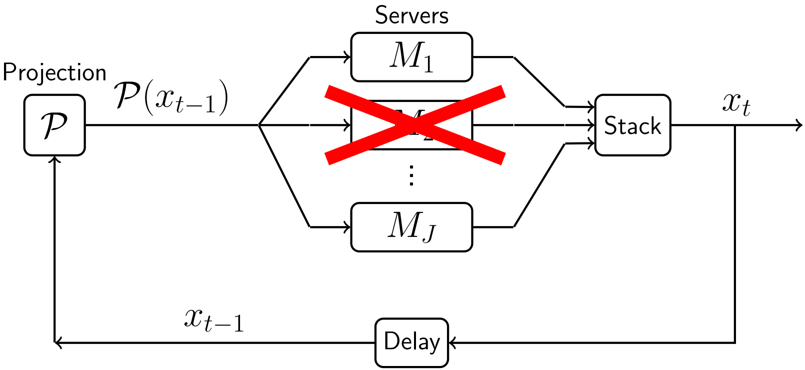


Distributed Power Method



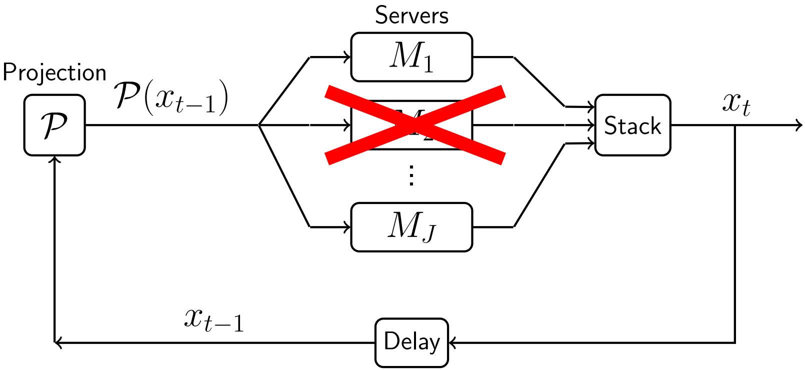
- Stragglers: What if one or more servers do not respond by the deadline?

Distributed Power Method



- Stragglers: What if one or more servers do not respond by the deadline?
- Coded Computing: coding for matrix multiplication with erasures.
  Dutta et al. 2016, Lee et al. 2017, Yu et al. 2017 and many more.

Distributed Power Method



- Stragglers: What if one or more servers do not respond by the deadline?
- Coded Computing: coding for matrix multiplication with erasures.
  Dutta et al. 2016, Lee et al. 2017, Yu et al. 2017 and many more.
- Can we just ignore the missing computations? (We are just refining an estimate.)

Running Example: Spiked Matrix Estimation

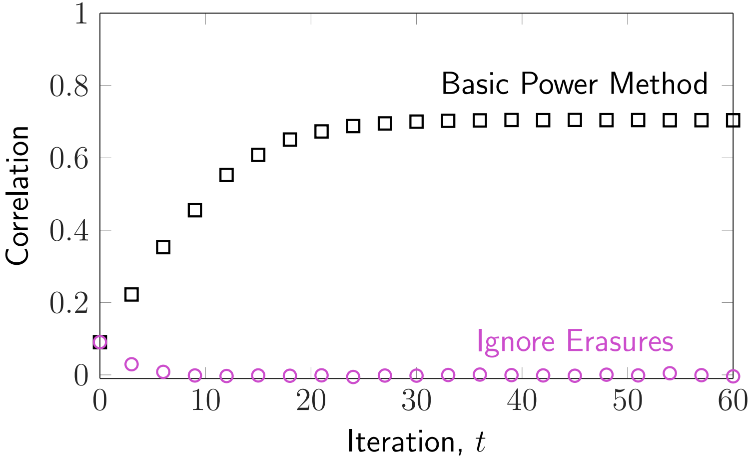
- Throughout the talk, we will evaluate our theorems and numerical experiments for the following spiked matrix model:

$$M = \frac{\lambda}{n}\theta\theta^\top + Z$$

  where $\theta \in \mathbb{R}^n$ is the spike and the noise Z is $\mathsf{GOE}(n)$.

- Goal: Estimate θ with the highest possible correlation $\frac{1}{n}\langle \theta, \hat{\theta} \rangle$.
- Recall that $Z \sim \mathsf{GOE}(n)$ means
  - $Z \in \mathbb{R}^{n \times n}$ is symmetric,
  - independent $\mathsf{N}(0, 1/n)$ entries above the diagonal,
  - independent $\mathsf{N}(0, 2/n)$ entries on the diagonal.
- This is primarily for direct comparison with prior AMP literature.
- Our theory holds more generally.

Distributed Power Method: Ignoring Erasures



- $M = \frac{\lambda}{n}\theta\theta^\top + Z$
- $Z \sim \mathrm{GOE}(n)$
- $\lambda = \sqrt{2}$
- $n = 5000$
- $\theta \sim \mathrm{Unif}(\{\pm 1\}^n)$

- Row erasures are i.i.d. Bernoulli(0.9).
- Setting the missing entries to zero does not work.

Distributed Power Method: Projection Matrix Framework

- Concisely summarize erasures via $\delta_t \in \{0,1\}^n$

$$\delta_{t,i} = \begin{cases} 0 & i^{\text{th}} \text{ row of } M \text{ is erased at iteration t} \\ 1 & \text{otherwise} \end{cases}$$

Distributed Power Method: Projection Matrix Framework

- Concisely summarize erasures via $\delta_t \in \{0,1\}^n$

$$\delta_{t,i} = \begin{cases} 0 & i^{\text{th}} \text{ row of } M \text{ is erased at iteration t} \\ 1 & \text{otherwise} \end{cases}$$

Ignoring Erasures:

$$x_t = \delta_t \circ M\hat{\theta}_{t-1} \qquad \hat{\theta}_t = \frac{\sqrt{n}}{\|x_t\|} x_t$$

Distributed Power Method: Projection Matrix Framework

- Concisely summarize erasures via $\delta_t \in \{0,1\}^n$

$$\delta_{t,i} = \begin{cases} 0 & i^{\text{th}} \text{ row of } M \text{ is erased at iteration t} \\ 1 & \text{otherwise} \end{cases}$$

Ignoring Erasures:

$$x_t = \delta_t \circ M\hat{\theta}_{t-1} \qquad \hat{\theta}_t = \frac{\sqrt{n}}{\|x_t\|} x_t$$

- Why not retain the values of the previous iterate in erased coordinates?

Distributed Power Method: Projection Matrix Framework

- Concisely summarize erasures via $\delta_t \in \{0,1\}^n$

$$\delta_{t,i} = \begin{cases} 0 & i^{\text{th}} \text{ row of } M \text{ is erased at iteration t} \\ 1 & \text{otherwise} \end{cases}$$

Ignoring Erasures:

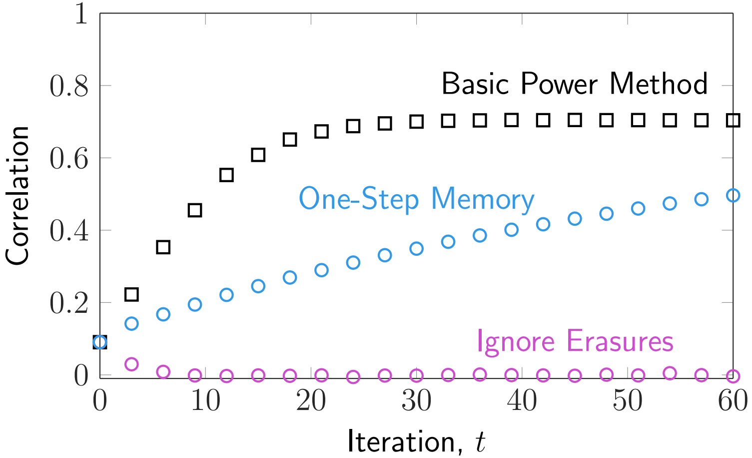
$$x_t = \delta_t \circ M\hat{\theta}_{t-1} \qquad \hat{\theta}_t = \frac{\sqrt{n}}{\|x_t\|}x_t$$

- Why not retain the values of the previous iterate in erased coordinates?

One-Step Memory:

$$x_t = \delta_t \circ M\hat{\theta}_{t-1} + (1-\delta_t) \circ x_{t-1} \qquad \hat{\theta}_t = \frac{\sqrt{n}}{\|x_t\|}x_t$$

Distributed Power Method: One-Step Memory



- $M = \frac{\lambda}{n}\theta\theta^\top + Z$
- $Z \sim \mathsf{GOE}(n)$
- $\lambda = \sqrt{2}$
- $n = 5000$
- $\theta \sim \mathrm{Unif}(\{\pm 1\}^n)$
- $\delta_{t,i}$ i.i.d. Bernoulli(0.1)

- Keeping the past iterate in erased coordinates is much better.

Related Work: Power Method and Subspace Tracking

- Many variations on this problem have been considered in the literature.
- An incomplete sampling:
  - Noisy Power Method [**Hardt and Price 2014, Balcan et al. 2016, Xu and Li 2022**]
  - Coordinate-wise Power Method [**Lei et al. 2016**]
  - Power Method with Momentum [**Xu et al. 2018**]
  - Adaptive Power Method [**Shin et al. 2023**]
  - Distributed Streaming PCA [**Raja and Bajwa 2020**]
  - Communication-Efficient Distributed SVD [**Li et al. 2021**]
  - Oja's Method [**Oja 1982, Oja and Karhunen 1985**]
  - Subspace Tracking with Missing Data [**Balzano et al. 2018, Wang et al. 2018**]
- This Talk: Approximate Message Passing (AMP) perspective on erasures.
  - Per-iteration performance guarantees via coupling to a Gaussian process.
  - (Ultimately) simple correction terms.
  - More efficient computation?

Approximate Message Passing (AMP)

Basic AMP:

$$x_t = M f_t(x_{t-1}) - b_t f_{t-1}(x_{t-2})$$

- Data Matrix: $M \in \mathbb{R}^{n \times n}$
- Denoising Functions: $f_t : \mathbb{R}^n \to \mathbb{R}^n$
- Debiasing Coefficient: $b_t \in \mathbb{R}$

- Early work on AMP was motivated by compressed sensing [**Donoho et al. 2009, Bayati and Montanari 2011, Javanmard and Montanari 2013**].
- Many other applications to regression, matrix estimation, channel coding, massive random access, etc. See recent survey [**Feng et al. 2022**].
- Most work has focused on separable denoisers, we follow the framework of [**Berthier et al. 2020**] that allows non-separable denoisers.

Approximate Message Passing (AMP)

Basic AMP:

$$x_t = Z f_t(x_{t-1}) - b_t f_{t-1}(x_{t-2})$$

- (Centered) Data Matrix: $Z \in \mathbb{R}^{n\times n}$
- Denoising Functions: $f_t : \mathbb{R}^n \to \mathbb{R}^n$
- Debiasing Coefficient: $b_t \in \mathbb{R}$

- Deterministic Initialization: $f_0 \in \mathbb{R}^n$
- State Evolution: $\{y_t\}$, a zero-mean Gaussian process with covariance

$$\begin{aligned}\mathrm{Cov}(y_0) &= \frac{1}{n}\|f_0\|^2\,\mathrm{I}_n\\ \mathrm{Cov}(y_s, y_t) &= \frac{1}{n}\mathbb{E}[\langle f_s(y_{s-1}), f_t(y_{t-1})\rangle]\,\mathrm{I}_n\ , \quad 0 \le s \le t.\end{aligned}$$

- **Assumption 1:** Each $f_t : \mathbb{R}^n \to \mathbb{R}^n$ is L-Lipschitz continuous and satisfies $\frac{1}{\sqrt{n}}\|f_t(0)\| \le C$ where C, L are positive numbers that do not depend on n.

Approximate Message Passing (AMP)

Basic AMP:

$$x_t = Z f_t(x_{t-1}) - b_t f_{t-1}(x_{t-2})$$

- (Centered) Data Matrix: $Z \in \mathbb{R}^{n \times n}$
- Denoising Functions: $f_t : \mathbb{R}^n \to \mathbb{R}^n$
- Debiasing Coefficient: $b_t \in \mathbb{R}$

Theorem

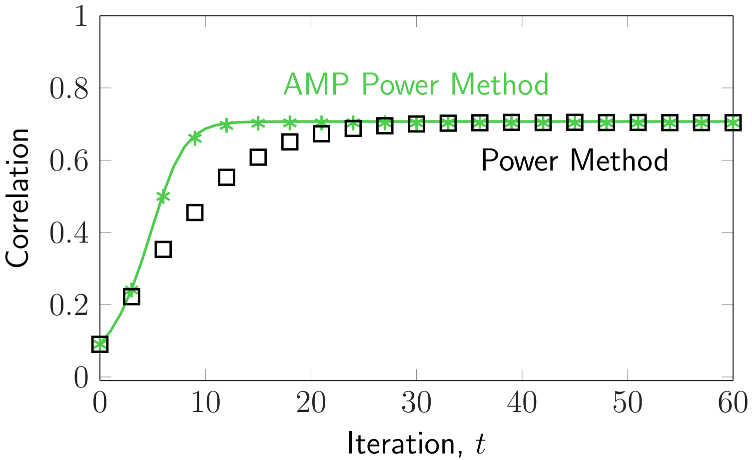
Suppose Assumption 1 holds, $Z \sim \mathsf{GOE}(n)$, and $b_t = \frac{1}{n}\operatorname{tr}(\mathbb{E}[\mathrm{D} f_t(y_{t-1})])$ Then, for any fixed number of iterations T, there exists a sequence (in n) of couplings between $x_{\leq T}$ and $y_{\leq T}$ such that $\frac{\|x_{\leq T} - y_{\leq T}\|}{\sqrt{n}} \xrightarrow[n\to\infty]{\text{pr}} 0.$

AMP Power Method:

$$x_t = M\hat{\theta}_{t-1} - \hat{\theta}_{t-1} \qquad \hat{\theta}_t = \frac{\sqrt{n}}{\|x_t\|} x_t$$

- State evolution provides a "single-letter" characterization of the performance at each iteration.

AMP Power Method



- $M = \frac{\lambda}{n}\theta\theta^\top + Z$
- $Z \sim \mathsf{GOE}(n)$
- $\lambda = \sqrt{2}$
- $n = 5000$
- $\theta \sim \mathrm{Unif}(\{\pm 1\}^n)$
- mark $=$ empirical
- line $=$ state evolution

- AMP correction term leads to empirical speedup.
- Accurate predictions from state evolution (SE).

AMP Power Method with i.i.d. Erasures

- We would like to make the AMP Power Method resilient to erasures.

AMP Power Method with i.i.d. Erasures

- We would like to make the AMP Power Method resilient to erasures.

AMP Power Method with Erasures???

$$x_t = \delta_t \circ M\hat{\theta}_{t-1} + (1-\delta_t) \circ \hat{\theta}_{t-1} \; - \; \text{correction} \qquad \hat{\theta}_t = \frac{\sqrt{n}}{\|x_t\|} x_t$$

AMP Power Method with i.i.d. Erasures

- We would like to make the AMP Power Method resilient to erasures.
- Recall that the δ_t captures the erasure pattern.

AMP Power Method with Erasures???

$$x_t = \delta_t \circ M\hat{\theta}_{t-1} + (1 - \delta_t) \circ \hat{\theta}_{t-1} \; - \; \text{correction} \qquad \hat{\theta}_t = \frac{\sqrt{n}}{\|x_t\|} x_t$$

AMP Power Method with i.i.d. Erasures

- We would like to make the AMP Power Method resilient to erasures.
- Recall that the δ_t captures the erasure pattern.
- Can only compute $\delta_t \circ M f_t(x_{t-1})$ rather than $M f_t(x_{t-1})$.

AMP Power Method with Erasures???

$$x_t = \delta_t \circ M\hat{\theta}_{t-1} + (1-\delta_t) \circ \hat{\theta}_{t-1} \; - \; \text{correction} \qquad \hat{\theta}_t = \frac{\sqrt{n}}{\|x_t\|} x_t$$

AMP Power Method with i.i.d. Erasures

- We would like to make the AMP Power Method resilient to erasures.
- Recall that the δ_t captures the erasure pattern.
- Can only compute $\delta_t \circ M f_t(x_{t-1})$ rather than $M f_t(x_{t-1})$.
- Can we find a good correction term that establishes a rigorous state evolution?

AMP Power Method with Erasures???

$$x_t = \delta_t \circ M\hat{\theta}_{t-1} + (1-\delta_t) \circ \hat{\theta}_{t-1} \ - \ \text{correction} \qquad \hat{\theta}_t = \frac{\sqrt{n}}{\|x_t\|} x_t$$

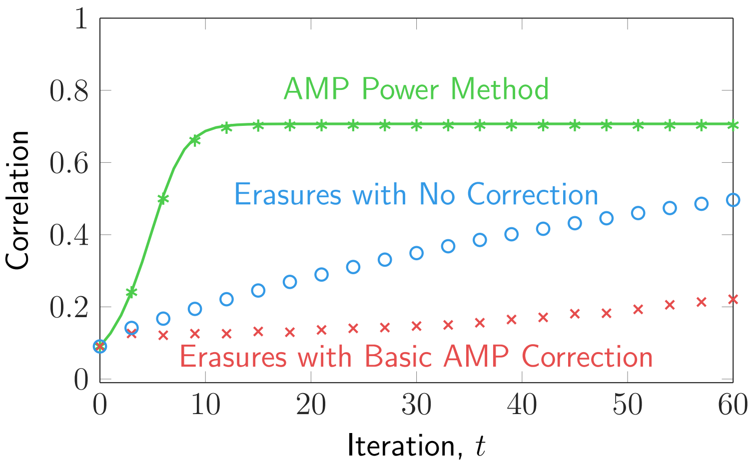
AMP Power Method with i.i.d. Erasures

- We would like to make the AMP Power Method resilient to erasures.
- Recall that the δ_t captures the erasure pattern.
- Can only compute $\delta_t \circ M f_t(x_{t-1})$ rather than $M f_t(x_{t-1})$.
- Can we find a good correction term that establishes a rigorous state evolution?
- Naïve Approach: Just reuse the Basic AMP correction term.

AMP Power Method with Erasures???

$$x_t = \delta_t \circ M\hat{\theta}_{t-1} + (1-\delta_t) \circ \hat{\theta}_{t-1} \; - \; \delta_t \circ \hat{\theta}_{t-1} \qquad \hat{\theta}_t = \frac{\sqrt{n}}{\|x_t\|} x_t$$

AMP Power Method with i.i.d. Erasures



- $M = \frac{\lambda}{n}\theta\theta^\top + Z$
- $Z \sim \mathsf{GOE}(n)$
- $\lambda = \sqrt{2}$
- $n = 5000$
- $\theta \sim \mathrm{Unif}(\{\pm 1\}^n)$
- $\delta_{t,i}$ i.i.d. Bernoulli(0.1)
- mark $=$ empirical
- line $=$ state evolution

- Standard correction term is not helpful: slow convergence and no state evolution.
- How can we derive the correction term? Generalize.

Linear Operator Approximate Message Passing (OpAMP)

OpAMP with Full Memory:

$$x_t = \mathcal{L}_t(Z) f_t(x_0, \ldots, x_{t-1}) - \sum_{s<t} B_{ts} f_s(x_0, \ldots, x_{s-1})$$

- (Centered) Data Matrix: $Z \in \mathbb{R}^{n \times n}$
- Linear Operators: $\mathcal{L}_t(Z) = \sum_{k=1}^{K} L_{tk} Z R_{tk}$ (non-unique decomposition)
- Denoising Functions: $f_t : \mathbb{R}^{n \times t} \to \mathbb{R}^n$
- Matrix-Valued Debiasing Coefficients: $B_{ts} \in \mathbb{R}^{n \times n}$
- **Assumption 1:** Each $f_t : \mathbb{R}^{n \times t} \to \mathbb{R}^n$ is L-Lipschitz continuous and satisfies $\frac{1}{\sqrt{n}} \| f_t(0) \| \leq C$ where C, L are positive numbers that do not depend on n.
- **Assumption 2:** $\|R_{tk}\|_{op}, \|L_{tk}\|_{op} \leq C'$ for all $t, k \in \mathbb{N}_0$ where C', K are positive numbers that do not depend on n.

Linear Operator Approximate Message Passing (OpAMP)

- State Evolution: $\{y_t\}$, a zero-mean Gaussian process with covariance

$$\mathsf{Cov}(y_s, y_t) = \sum_{l,k=1}^{K} \frac{1}{n}\mathbb{E}[\langle R_{sl} f_s(y_{<s}), R_{tk} f_t(y_{<t})\rangle] L_{sl} L_{tk}^\top$$

Theorem

Suppose Assumptions 1 and 2 hold, $Z \sim \mathsf{GOE}(n)$*, and*

$$B_{ts} = \sum_{k,l=1}^{K} \frac{1}{n}\operatorname{tr}(R_{tk}\mathbb{E}[\mathsf{D}_s f_t(y_{<t})] L_{sl})\, L_{tk} R_{sl}$$

Then, for any fixed number of iterations T*, there exists a sequence (in* n*) of couplings between* $x_{\leq T}$ *and* $y_{\leq T}$ *such that* $\frac{\|x_{\leq T} - y_{\leq T}\|}{\sqrt{n}} \xrightarrow[n\to\infty]{\mathrm{pr}} 0.$

OpAMP Proof Sketch: “Lifted” Recursion

- Define a doubly-indexed, full-memory AMP recursion:

$$w_{tk} = Z g_{tk}(w_{<t}) - \sum_{s<t} \sum_{l=1}^{K} c_{tksl}\, g_{sl}(w_{<s}) \ , \quad \text{for } k = 1, \dots, K$$

$$g_{tk}(w_{<t}) = R_{tk} f_t\Big(\sum_{k=1}^{K} L_{0k} w_{0k}, \dots, \sum_{k=1}^{K} L_{t-1,k} w_{t-1,k} \Big) \ .$$

- Can show $x_t = \sum_{k=1}^{K} L_{tk} w_{tk}$.
- State evolution u_{tk} for w_{tk} using full-memory AMP [**Gerbelot and Berthier 2023**].
- Obtain state evolution $y_t = \sum_{k=1}^{K} L_{tk} u_{tk}$ for x_t.

Projection AMP

Projection AMP:

$$x_t = \Pi_t\Big(Z f_t(x_{t-1}) - \sum_{s<t} b_{ts} f_s(x_{s-1})\Big) + \Pi_t^{\perp} x_{t-1}$$

- Projection Matrices: Π_t (not necessarily diagonal, nor commuting)
- Scalar Debiasing Coefficients: $b_{ts} \in \mathbb{R}^{n\times n}$
- Define $C_{ts} = \begin{cases} \mathrm{I}, & s = t \\ \Pi_t^{\perp}\Pi_{t-1}^{\perp}\cdots\Pi_{s+2}^{\perp}\Pi_{s+1}^{\perp} & 0 \le s < t \end{cases}$

Theorem

Suppose Assumption 1 holds, $Z \sim \mathrm{GOE}(n)$*, and* $b_{ts} = \frac{1}{n}\mathrm{tr}(\mathbb{E}[\mathrm{D} f_t(y_{t-1})]C_{t-1,s}\Pi_s)$
Then, for any fixed number of iterations T*, there exists a sequence (in* n*) of couplings between* $x_{\le T}$ *and* $y_{\le T}$ *such that* $\frac{\|x_{\le T} - y_{\le T}\|}{\sqrt{n}} \xrightarrow[n\to\infty]{\mathrm{pr}} 0.$

OpAMP Power Method: i.i.d. Erasures

OpAMP Power Method:

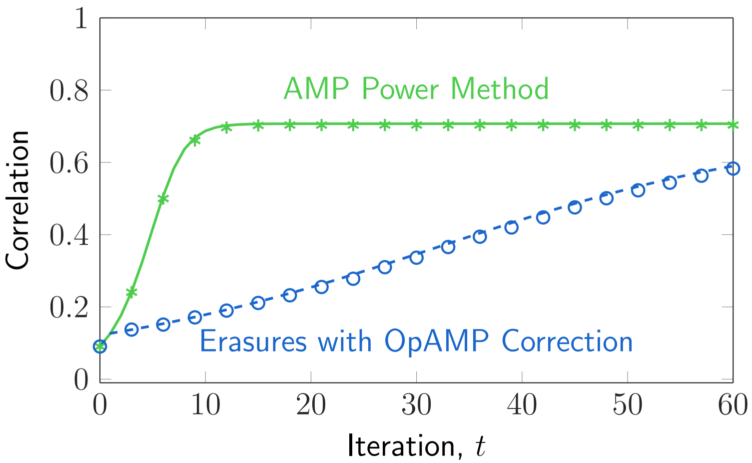
$$x_t = \delta_t \circ \left(M x_{t-1} - \sum_{s<t} b_{ts} \hat{\theta}_s \right) + (1 - \delta_t) \circ x_{t-1} \qquad \hat{\theta}_t = \frac{\sqrt{n}}{\|x_t\|} x_t$$

- If δ_t is elementwise i.i.d. Bernoulli(γ), then we can establish a state evolution by setting the debiasing coefficients to $b_{ts} = \frac{\sqrt{n}}{\|x_{t-1}\|} p_t(s)$ where

$$p_t(s) = \begin{cases} \gamma(1-\gamma)^{t-s-1} & \text{if } s = 1, 2, \ldots, t-1 \\ (1-\gamma)^{t-1} & \text{if } s = 0 \end{cases}$$

- State evolution has a simple form.

OpAMP Power Method: i.i.d. Erasures



- $M = \frac{\lambda}{n}\theta\theta^\top + Z$
- $Z \sim \text{GOE}(n)$
- $\lambda = \sqrt{2}$
- $n = 5000$
- $\theta \sim \text{Unif}(\{\pm 1\}^n)$
- $\delta_{t,i}$ i.i.d. Bernoulli(0.1)
- mark $=$ empirical
- line $=$ state evolution

- With the OpAMP correction term, we can establish a rigorous state evolution.
- Attains the same fixed point as the AMP power method.

OpAMP Power Method: Round Robin Updates

OpAMP Power Method:

$$x_t = \delta_t \circ \left(M x_{t-1} - \sum_{s<t} b_{ts} \hat{\theta}_s \right) + (1 - \delta_t) \circ x_{t-1} \qquad \hat{\theta}_t = \frac{\sqrt{n}}{\|x_t\|} x_t$$

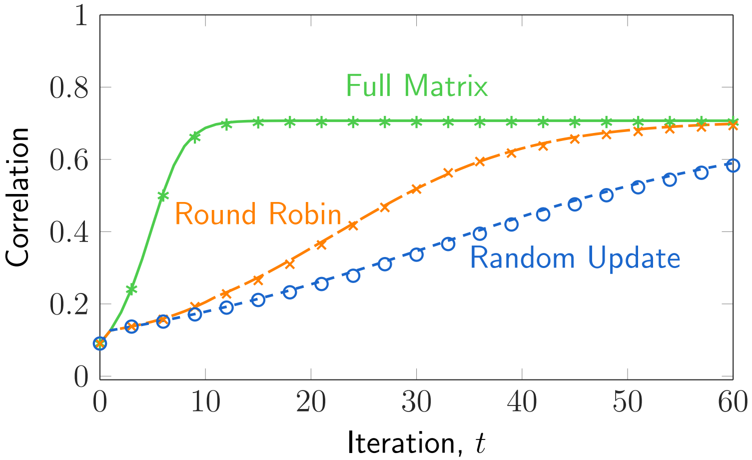
- Consider now the setting where we deliberately only apply a subblock of the data matrix M at each iteration, to reduce the computational load.

- We partition the row indices $\{1, \ldots, n\}$ into J equally-sized subsets $\mathcal{A}_0, \ldots, \mathcal{A}_{J-1}$.

$$\delta_{t,i} = \begin{cases} 1 & i \in \mathcal{A}_{(t \bmod J)} \\ 0 & \text{otherwise} \end{cases}$$

- Debiasing coefficients and state evolution have a simple form.

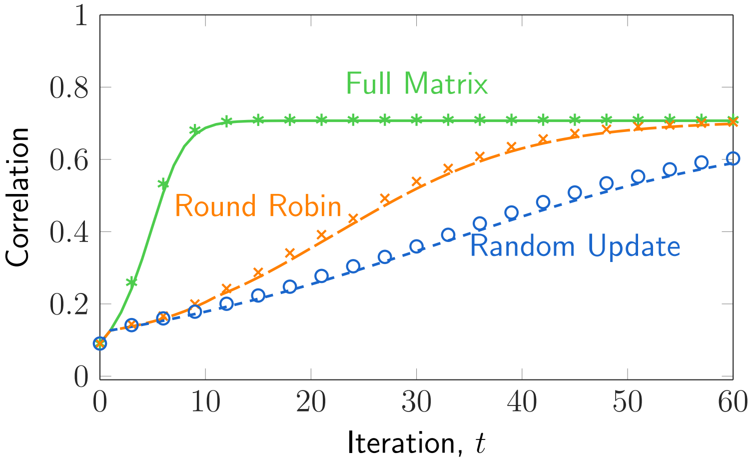
OpAMP Power Method: Round Robin Updates



- $M = \frac{\lambda}{n}\theta\theta^\top + Z$
- $Z \sim \mathsf{GOE}(n)$
- $\lambda = \sqrt{2}$
- $n = 5000$
- $\theta \sim \mathrm{Unif}(\{\pm 1\}^n)$
- mark = empirical
- line = state evolution

- Round Robin: Update 0.1 rows per iteration according to a schedule.
- Noticeable speedup compared to stochastic erasures.

OpAMP Power Method: Subgaussian Matrices



- $M = \frac{\lambda}{n}\theta\theta^\top + Z$
- Z symmetric from i.i.d. Rademacher
- $\lambda = \sqrt{2}$
- $n = 5000$
- $\theta \sim \mathrm{Unif}(\{\pm 1\}^n)$
- mark = empirical
- line = GOE state evolution

- Empirical performance very similar.
- GOE(n) state evolution no longer a theoretical guarantee.

OpAMP Power Method: Efficient Computation

OpAMP Power Method:

$$x_t = \delta_t \circ \left(M x_{t-1} - \sum_{s<t} b_{ts} \hat{\theta}_s \right) + (1 - \delta_t) \circ x_{t-1} \qquad \hat{\theta}_t = \frac{\sqrt{n}}{\|x_t\|} x_t$$

- So far, we have plotted the performance with respect to iteration t.

OpAMP Power Method: Efficient Computation

OpAMP Power Method:

$$x_t = \delta_t \circ \left(M x_{t-1} - \sum_{s<t} b_{ts} \hat{\theta}_s \right) + (1 - \delta_t) \circ x_{t-1} \qquad \hat{\theta}_t = \frac{\sqrt{n}}{\|x_t\|} x_t$$

- So far, we have plotted the performance with respect to iteration t.
- Alternatively, we could plot the performance with respect to the amount of large-scale computation. (Ignores normalization steps, etc.)

OpAMP Power Method: Efficient Computation

OpAMP Power Method:

$$x_t = \delta_t \circ \left(M x_{t-1} - \sum_{s<t} b_{ts} \hat{\theta}_s \right) + (1 - \delta_t) \circ x_{t-1} \qquad \hat{\theta}_t = \frac{\sqrt{n}}{\|x_t\|} x_t$$

- So far, we have plotted the performance with respect to iteration t.
- Alternatively, we could plot the performance with respect to the amount of large-scale computation. (Ignores normalization steps, etc.)
- For instance, we could track the total number of $n \times n$ matrix multiplications.

OpAMP Power Method: Efficient Computation

OpAMP Power Method:

$$x_t = \delta_t \circ \left(M x_{t-1} - \sum_{s<t} b_{ts} \hat{\theta}_s \right) + (1 - \delta_t) \circ x_{t-1} \qquad \hat{\theta}_t = \frac{\sqrt{n}}{\|x_t\|} x_t$$

- So far, we have plotted the performance with respect to iteration t.
- Alternatively, we could plot the performance with respect to the amount of large-scale computation. (Ignores normalization steps, etc.)
- For instance, we could track the total number of $n \times n$ matrix multiplications.
    - Full Matrix: Standard AMP that applies the full matrix.

OpAMP Power Method: Efficient Computation

OpAMP Power Method:

$$x_t = \delta_t \circ \left(M x_{t-1} - \sum_{s<t} b_{ts} \hat{\theta}_s \right) + (1 - \delta_t) \circ x_{t-1} \qquad \hat{\theta}_t = \frac{\sqrt{n}}{\|x_t\|} x_t$$

- So far, we have plotted the performance with respect to iteration t.
- Alternatively, we could plot the performance with respect to the amount of large-scale computation. (Ignores normalization steps, etc.)
- For instance, we could track the total number of $n \times n$ matrix multiplications.
  - Full Matrix: Standard AMP that applies the full matrix.
  - Round Robin: Cycle through fixed subsets of rows of M.

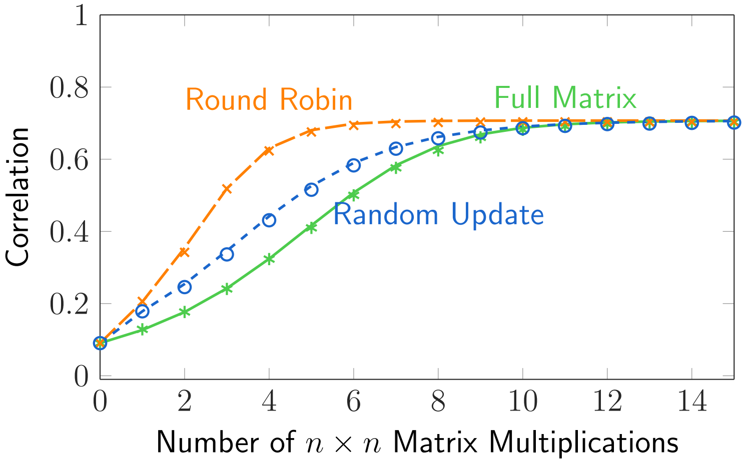
OpAMP Power Method: Efficient Computation

OpAMP Power Method:

$$x_t = \delta_t \circ \left(M x_{t-1} - \sum_{s<t} b_{ts} \hat{\theta}_s \right) + (1 - \delta_t) \circ x_{t-1} \qquad \hat{\theta}_t = \frac{\sqrt{n}}{\|x_t\|} x_t$$

- So far, we have plotted the performance with respect to iteration t.
- Alternatively, we could plot the performance with respect to the amount of large-scale computation. (Ignores normalization steps, etc.)
- For instance, we could track the total number of $n \times n$ matrix multiplications.
  - Full Matrix: Standard AMP that applies the full matrix.
  - Round Robin: Cycle through fixed subsets of rows of M.
  - Random Update: Randomly selected rows of M applied.

OpAMP Power Method: Efficient Computation



- $M = \dfrac{\lambda}{n}\theta\theta^\top + Z$
- $Z \sim \mathsf{GOE}(n)$
- $\lambda = \sqrt{2}$
- $n = 5000$
- $\theta \sim \mathrm{Unif}(\{\pm 1\}^n)$
- mark = empirical
- line = state evolution

- Round Robin: Uses fewer matrix multiplications to converge.
- Random Update: Sometimes uses fewer matrix multiplications to converge.

Conclusions

- AMP perspective on the distributed power method with erasures.
  - Simple state evolution and scalar debiasing coefficients.
  - Same fixed point as no-erasure setting.
  - Computational speedup for partial updates.
  - Can also consider other denoisers, e.g., Bayes.
- Theoretical results established by first generalizing to linear operator AMP, which may be useful in other settings.
- Some follow-up questions:
  - Orthogonal ensembles?
  - Adaptive updates?
  - 1st-order methods with erasures?
  - Noise instead of erasures?
  - Connection to SGD speedup?
- Acknowledgments: Thanks to Nicholas Sacco and Viveck Cadambe for valuable discussions on the power method with erasures.