Sandia National Laboratories

Exceptional service in the national interest

RECENT IMPROVEMENTS IN CP POISSON TENSOR ALGORITHMS

Jeremy M. Myers[1,2], Daniel M. Dunlavy[1]

August 21, 2023

[01211] Generalized and non-Gaussian Tensor Decompositions

10th International Congress on Industrial and Applied Mathematics (ICIAM 2023)

Tokyo, Japan and Virtual

[1]Sandia National Laboratories, Albuquerque, NM and Livermore, CA
[2]College of William and Mary, Williamsburg, VA

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia LLC, a wholly owned subsidiary of Honeywell International Inc. for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

SAND2023-07881C

MOTIVATING STOCHASTIC + DETERMINISTIC TENSOR ALGORITHMS

Recent trend in theoretical computer science & numerical linear algebra (and elsewhere!):

- Use randomization to solve very large, hard problems
  - data mining, information science, compression, scientific computing
- Often faster with equivalent levels of error
- Examples: low-rank matrix decompositions, streaming, regression, linear systems [1]

Typical approach: use stochasticity for a fast approximation and determinism for refinement to yield effective algorithms with theoretical guarantees.

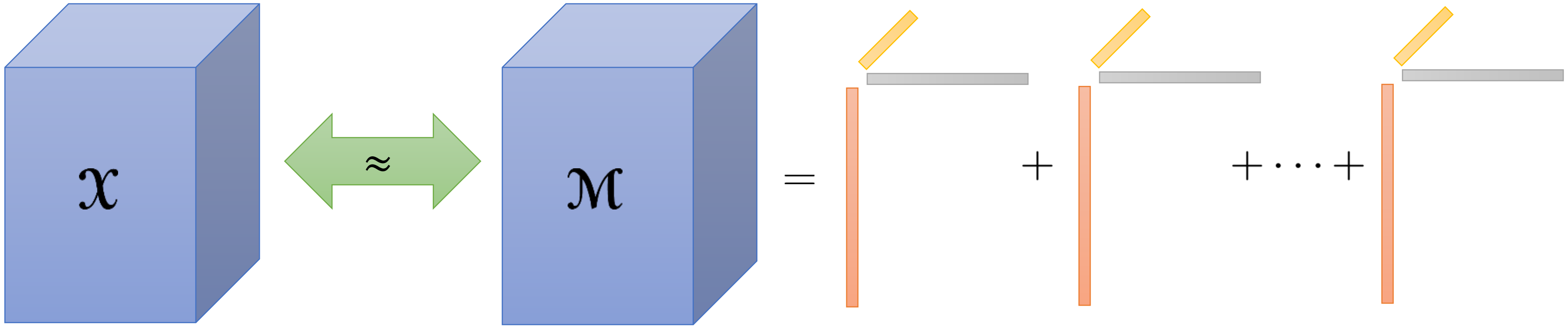
How can we extend the existing approaches to **low-rank tensor decompositions**?

[1] Martinsson and Tropp, Randomized Numerical Linear Algebra: Foundations & Algorithms, *Acta Numerica*, 2020.

LOW-RANK TENSOR APPROXIMATION

Canonical polyadic decomposition (CPD)

Data CP model Sum of r outer product tensors



$$\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_d} \qquad \mathcal{M} = [\![\mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_d]\!] \in \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_d}$$

- $\mathcal{X}$ is the data tensor in d dimensions or modes.
- $\mathcal{M}$ is the model tensor.
- $\mathbf{A}_k$ is an $n_k \times r$ factor matrix.
- $\mathbf{i} = (i_1, i_2, \ldots, i_d)$ is a multi-index

Low-rank CPD

- Assume $\operatorname{rank}(\mathcal{X}) = r$.
- Typically choose $r \ll \min\{n_1, n_2, \ldots, n_d\}$.

Poisson CPD

$$\mathcal{X}_{\mathbf{i}} \sim \operatorname{Poisson}(\mathcal{M}_{\mathbf{i}})$$

LOW-RANK TENSOR APPROXIMATION

Poisson tensor maximum likelihood estimation

Statistical method to compute low-rank Poisson CPD

$$\min_{\mathcal{M}} f_{\mathcal{X}}(\mathcal{M}) = \min \sum_{\mathbf{i}} m_{\mathbf{i}} - x_{\mathbf{i}} \log(m_{\mathbf{i}})$$

$$\text{where } a_{i_1}^{(1)} a_{i_2}^{(2)} \dots a_{i_d}^{(d)} = m_{\mathbf{i}} \text{ are the optimization variables}$$

- This a **nonlinear, nonconvex** optimization problem.
- The **maximum likelihood estimator (MLE)** corresponds to the **global optimizer** $\mathcal{M}^*$ for this problem.
- The typical approach is to *flatten* or *unfold* the tensors into matrices and use **local** methods.
  - Stochastic: Generalized Canonical Polyadic (GCP) tensor decomposition [2, 3]
  - Deterministic: Canonical Polyadic Alternating Poisson Regression (CPAPR) [4]

[2] Hong, Kolda, and Duersch, Generalized Canonical Polyadic Tensor Decomposition, *SIAM Review*, 2020
[3] Kolda and Hong, Stochastic Gradients for Large-Scale Tensor Decomposition, *SIAM Journal on Mathematics of Data Science*, 2020
[4] Chi and Kolda, On Tensors, Sparsity, and Nonnegative Factorizations, *SIAM Journal on Matrix Analysis and Applications*, 2012

OUR NEW STOCHASTIC + DETERMINISTIC TENSOR ALGORITHMS

How can current local methods be leveraged together to improve likelihood of finding the MLE/global optimizer?

Proposed methods

Hybrid GCP-CPAPR

- Inspired by Simulated Annealing.
- Improves probability of convergence to global optimizer and reduces cost compared to standalone methods.

Restarted CPAPR with SVDrop

- Uses novel heuristic to avoid suboptimal solutions w.r.t. global optimizer.
- Saves computation by restarting when the iterates are detected to be headed to a suboptimal solution.

HYBRID GCP-CPAPR

Hybrid GCP-CPAPR (HybridGC) intuition

1. Use stochastic optimization to compute a fast approximate solution.
2. Use deterministic optimization to refine approximate solution.

```
**Algorithm** $\textsc{HybridGC}(\text{tensor } \boldsymbol{\mathscr{X}}, \text{ rank } r, \text{ initial guess } \boldsymbol{\mathscr{M}}_0)$
    $\boldsymbol{\mathscr{M}}_1 \leftarrow \mathrm{GCP}(\boldsymbol{\mathscr{X}}, r, \boldsymbol{\mathscr{M}}_0)$
    $\boldsymbol{\mathscr{M}}_2 \leftarrow \mathrm{CPAPR}(\boldsymbol{\mathscr{X}}, r, \boldsymbol{\mathscr{M}}_1)$
    **return** model tensor $\widehat{\boldsymbol{\mathscr{M}}} = \boldsymbol{\mathscr{M}}_2$ as estimate to $\boldsymbol{\mathscr{M}}^*$
```

NUMERICAL EXPERIMENTS WITH HYBRID GCP-CPAPR

Methodology

1. Generate N random starting points.
2. Compute decompositions with CPAPR & GCP separately.
3. HybridGC step: refine GCP decompositions with CPAPR.
4. Analyze average behavior of our experiments.

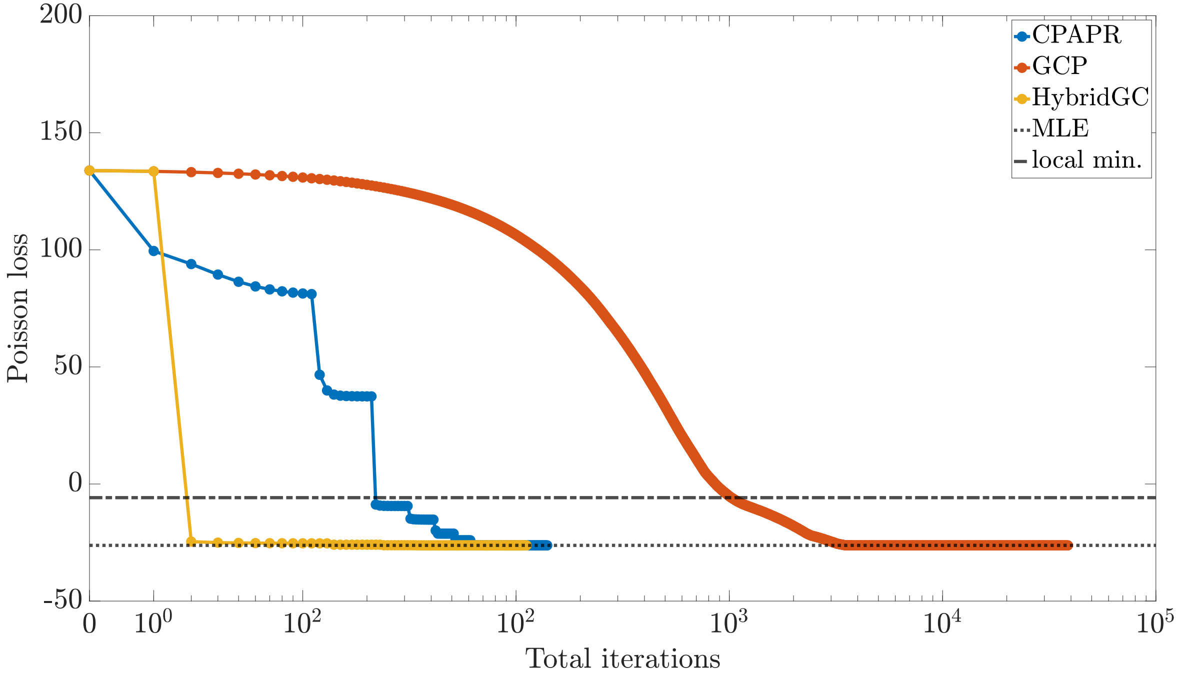
Datasets

1. **Small**: 4 x 6 x 8, 17 nonzeros, $r = 3$, $N > 110\text{k}$
2. **Large**: 1k x 1k x 1k, 98k nonzeros, $r = 20$, $N = 100$

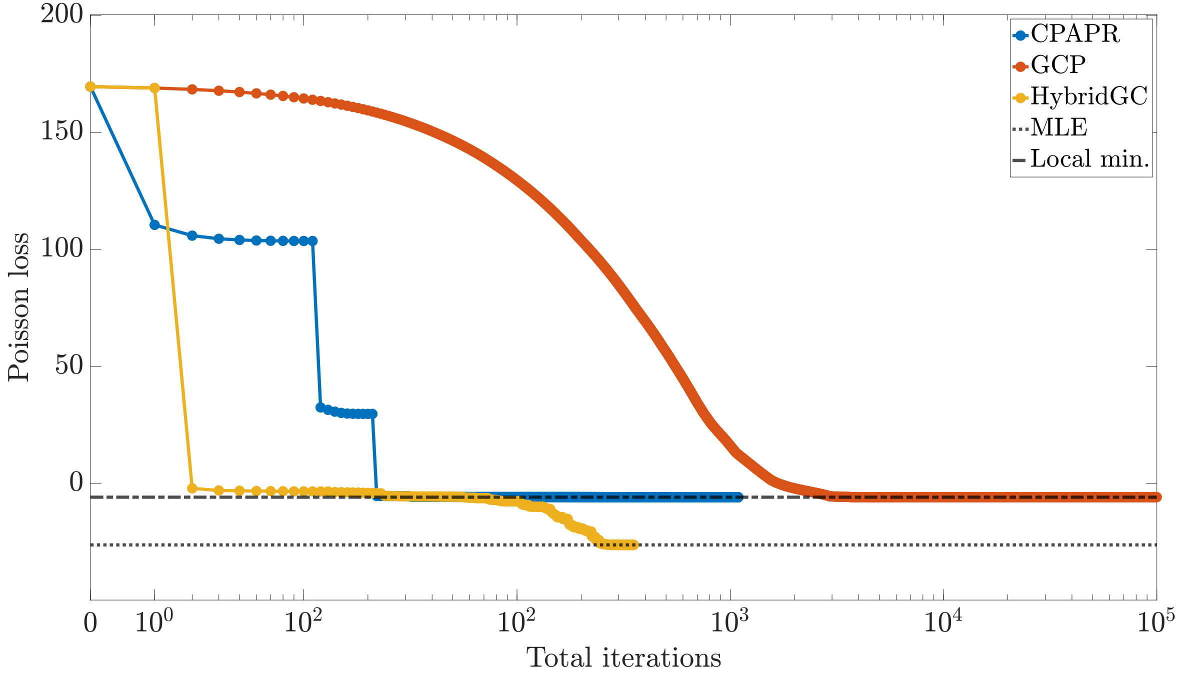
Error measures

- Based on loss function values.
- Probability estimate of finding MLE/global optimizer.
- Spectral properties of unfolded tensor.

HYBRID GCP-CPAPR RESULTS: OPTIMIZATION VARIABLES VIEW



Ex. 1



Ex. 2

PROBABILITY OF FINDING MAXIMUM LIKELIHOOD ESTIMATOR (MLE)

Small dataset ($N > 110K$)

ϵ	CPAPR	GCP	HybridGC
10^{-1}	0.963	0.963	**0.967**
10^{-2}	0.963	0.963	**0.967**
10^{-3}	0.963	0.879	**0.967**
10^{-4}	0.963	0.003	**0.967**

Large dataset ($N = 100$)

ϵ	CPAPR	GCP	HybridGC
10^{-1}	1.00	1.00	1.00
10^{-2}	**0.46**	0.04	**0.46**
10^{-3}	0.03	0.00	**0.17**
10^{-4}	0.00	0.00	**0.01**

Relative distance from MLE

$$\epsilon = \frac{|f_{\mathcal{X}}(\widehat{\mathcal{M}}) - f_{\mathcal{X}}(\mathcal{M}^*)|}{|f_{\mathcal{X}}(\mathcal{M}^*)|}$$

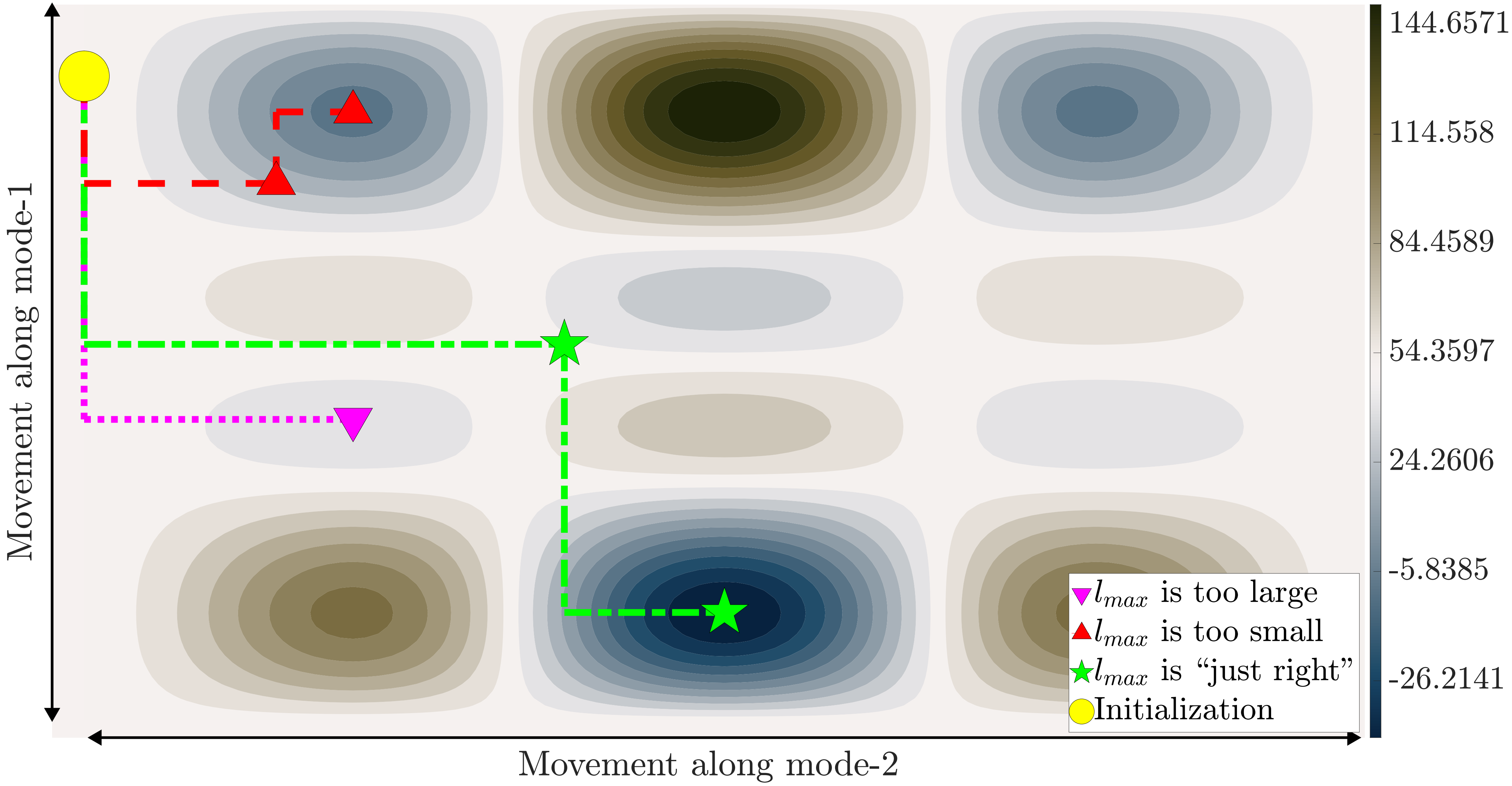
For small choices of ϵ, HybridGC is the most likely to estimate MLE/global optimizer.

OUR FUNDAMENTAL (YET SIMPLE) QUESTION

Why and when do these methods fail?

We'll try to answer this for CPAPR.

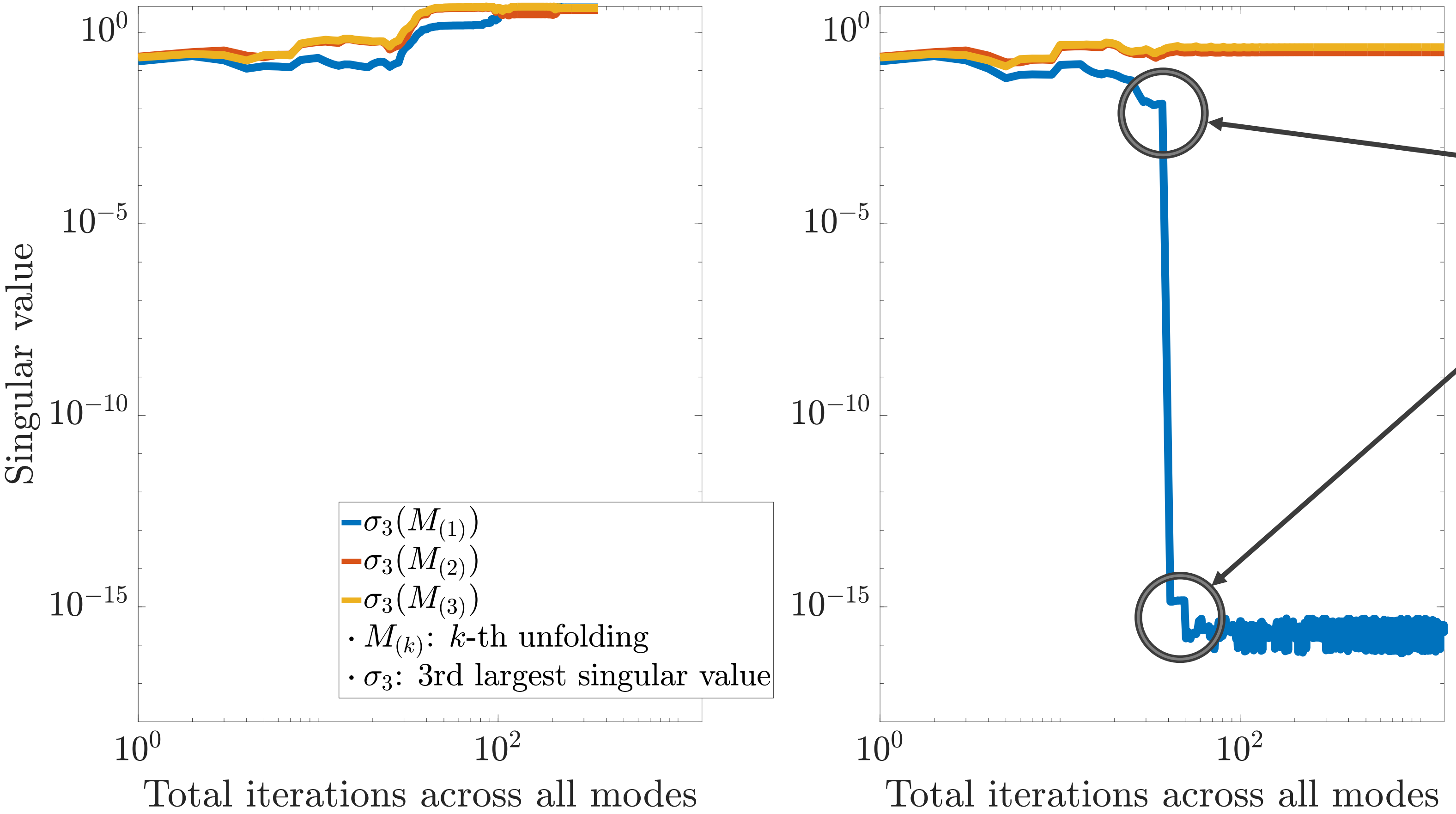
CONVERGENCE DEPENDS ON NUMBER OF STEPS IN SEARCH DIRECTION



CHALLENGING BEHAVIOR ON SMALL DATASET

"Optimal" # of inner iterations:
leads to MLE

Too many inner iterations:
leads to other local minimizer



Spectral property

The ratio of successive singular values may be a useful heuristic.

Related work

- Two-factor degeneracies (2FD)[5]
- Heuristic to detect 2FD[6]

[5] Kruskal, Harshman, and Lundy, How 3-MFA data can cause degenerate parafac solutions, among other relationships, in *Multiway Data Analysis*, 1989

[6] Mitchell and Burdick, Slowly converging parafac sequences: Swamps and two-factor degeneracies, *Journal of Chemometrics*, 1994

RESTARTED CPAPR WITH SVDROP (OVERVIEW)

Choose the following parameters:

- k_{max}: Maximum number of outer iterations
- l_{max} : Maximum number of inner iterations
- j: Compute spectral properties every $j \leq l_{max}$ inner iterations
- γ: Maximum threshold of spectral properties for acceptable search path (e.g., $\gamma = 10^6$)

While (not converged), compute a rank-R decomposition with CPAPR:

1. Every step, update current model.
2. Every j steps, compute spectral properties of current model.
3. If (spectral properties) $< \gamma$, checkpoint and continue.
4. Otherwise, choose a new initial guess and **restart**.

RESTARTED CPAPR WITH SVDROP (DETAILED)

Choose the following parameters:

- k_{max}: Maximum number of outer iterations
- l_{max} : Maximum number of inner iterations
- j: Compute spectral properties every $j \leq l_{max}$ inner iterations
- γ: Maximum threshold of spectral properties for acceptable search path (e.g., $\gamma = 10^6$)

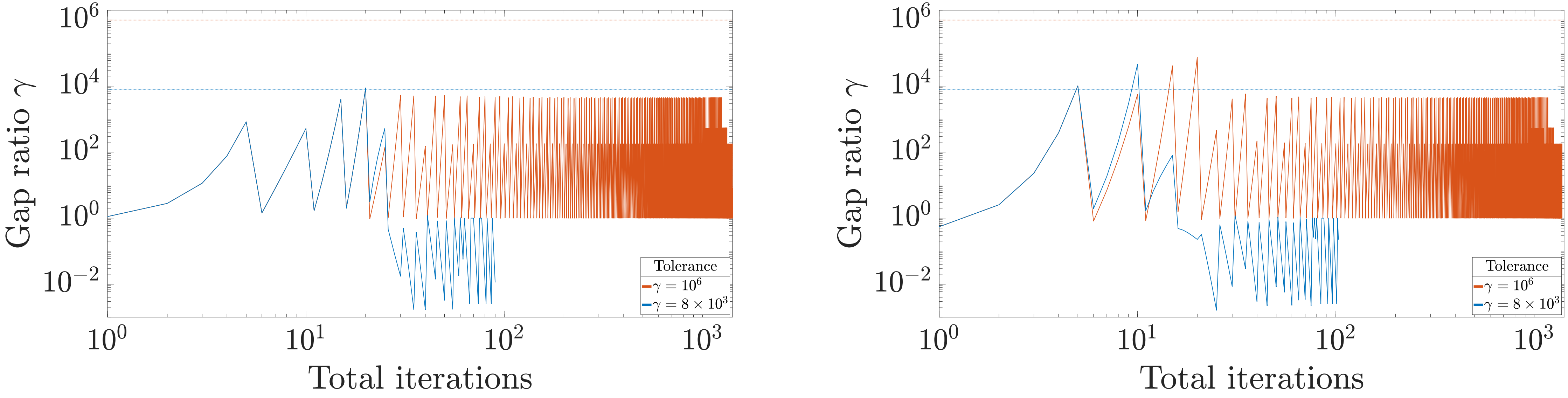
1. Choose an initial guess
2. While not converged, compute a rank-R decomposition with CPAPR:
   a. At the i-th iteration in mode-k, compute the R-th largest singular value $\sigma_{(k)}[R]^{(i)}$.
   b. Proceed for j iterations.
   c. At the $(i+j)$-th iteration in mode-k, compute the R-th largest singular value $\sigma_{(k)}[R]^{(i+j)}$.
   d. If $\sigma_{(k)}[R]^{(i)}/\sigma_{(k)}[R]^{(i+j)} < \gamma$, set $\sigma_{(k)}[R]^{(i)} \leftarrow \sigma_{(k)}[R]^{(i+j)}$ and continue.
   e. Otherwise, **restart**: go to 1.

CONVERGENCE AND PERFORMANCE RESULTS

Probability of convergence to MLE vs. local minimizer with SVDrop; $\gamma = 10^6$, $\epsilon = 10^{-4}$, $N = 4051$

		SVDrop inner iterations τ										
Converged	Minimizer	0	1	2	3	4	5	6	7	8	9	10
Yes	MLE	0	4024	**4049**	4035	4028	4029	3906	3970	3983	3990	3998
Yes	Other KKT point	3905	0	0	0	0	0	102	43	31	24	20
No	-	146	27	**2**	16	23	22	43	38	37	37	33

Sensitivity of SVDrop to γ ($\tau = 2$)



CONCLUSIONS

- SVDrop has the highest likelihood of finding MLE in our experiments.
- The method can be prohibitively expensive when it does fail, but this is rare.

FUTURE WORK

- It's unclear how sensitive SVDrop is to the complex interplay parameters.
- Experiments on **Small** dataset are very limited – do they generalize?
- Are low-accuracy singular values useful?

Contact: {jermyer, dmdunla}@sandia.gov
Paper (to be updated soon): https://arxiv.org/abs/2207.14341