A Block-Randomized Stochastic Method with Importance Sampling for CP Tensor Decomposition[1]

Yajie Yu

Chongqing University, Chongqing, P.R. China

CQSIAM2023, 13RD May 2023

[1] A joint work with Hanyu Li
E-mail: zqyu@cqu.equ.cn

Presentation Outline

A Tensor is an Multi-Way Array

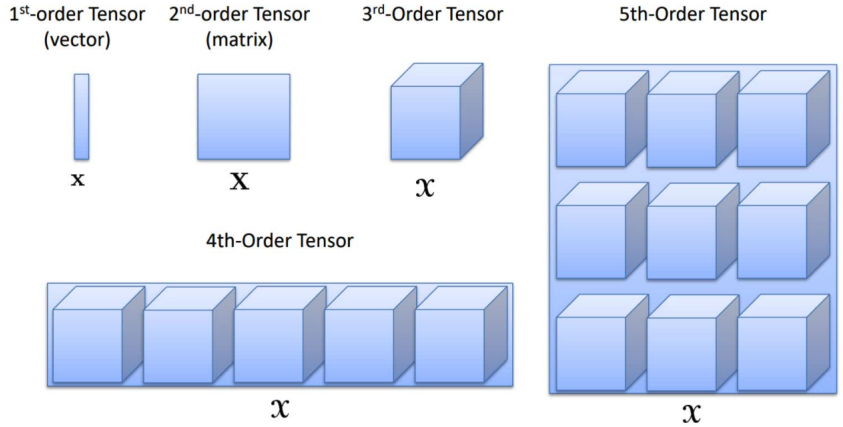
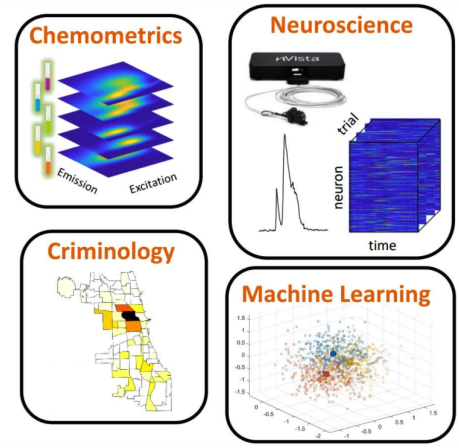


Figure: Graphical representation of multiway array (tensor) data.

Tensors Come From Many Applications

Tensor Decomposition Finds Patterns in Massive Data (Unsupervised Learning)



- Chemometrics: Emission x Excitation x Samples (Fluorescence Spectroscopy)
- Neuroscience: Neuron x Time x Trial
- Criminology: Day x Hour x Location x Crime (Chicago Crime Reports)
- Machine Learning: Multivariate Gaussian Mixture Models Higher-Order Moments
- Transportation: Pickup x Dropoff x Time (Taxis)
- Sports: Player x Statistic x Season (Basketball)
- Cyber-Traffic: IP x IP x Port x Time
- Social Network: Person x Person x Time x Interaction-Type
- Signal Processing: Sensor x Frequency x Time
- Trending Co-occurrence: Term A x Term B x Time

CP Decomposition Approximates an N-th Order Tensor as a sum of R Rank-One Tensors

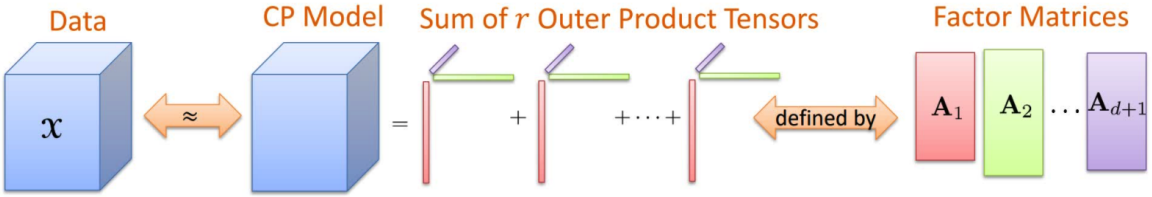


Figure: Graphical representation of CP decomposition.

- $\mathcal{X} \approx \tilde{\mathcal{X}} = \sum_{r=1}^{R} \boldsymbol{a}_r^{(1)} \circ \boldsymbol{a}_r^{(2)} \circ \cdots \circ \boldsymbol{a}_r^{(N)} = [[\mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \cdots, \mathbf{A}^{(N)}]]$;
- $\mathcal{O}(NIR)$ parameters: is linear to the tensor order N.
- high compression ability;
- Uniqueness: Under modest conditions, CP is unique up to permutation and scaling.

Prototypical CP Least Squares Subproblem is "Tall and Skinny"

Structure of Khatri-Rao Product (KRP): Hadamard Combinations of Rows of Inputs

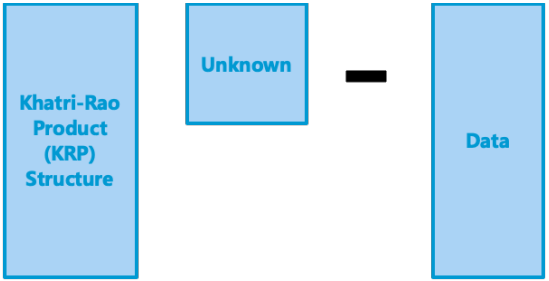


Figure: CP-ALS: $\min_{\mathbf{A}\in\mathbb{R}^{I_n\times R}} \|\mathbf{Z}^{(n)}\mathbf{A}^\intercal - \mathbf{X}_{(n)}^\intercal\|_F^2$.

- CP-ALS[2]: Solve N such subproblems per outer iteration;

[2] Tamara G. Kolda and Brett W. Bader. "Tensor Decompositions and Applications". In: *SIAM Rev.* 51.3 (2009), pp. 455–500. DOI: 10.1137/07070111X.

Randomized CP-ALS by Make Full Use of the KRP Struture

Efficient Sampling/Sketching Without Forming KRP

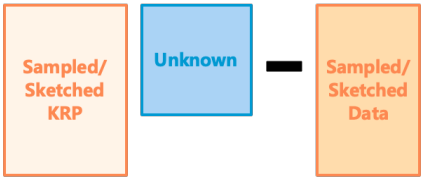


Figure: CPRAND or CP-ARLS-LEV: $\min_{\mathbf{A}\in\mathbb{R}^{I_n\times R}} \|\mathbf{S}\mathbf{Z}^{(n)}\mathbf{A}^\intercal - \mathbf{S}\mathbf{X}_{(n)}^\intercal\|_F^2$.

- CPRAND[3]: Solve N smaller subproblems per outer iteration with uniform sampling and KFJLT;
- CP-ARLS-LEV[4]: Solve N smaller subproblems per outer iteration with leverage-based sampling.

[3] Casey Battaglino, Grey Ballard, and Tamara G. Kolda. "A Practical Randomized CP Tensor Decomposition". In: *SIAM J. Matrix Anal. Appl.* 39.2 (2018), pp. 876–901. DOI: 10.1137/17M1112303.

[4] Brett W. Larsen and Tamara G. Kolda. "Practical Leverage-Based Sampling for Low-Rank Tensor Decomposition". In: *SIAM J. Matrix Anal. Appl.* 43.3 (2022), pp. 1488–1517. DOI: 10.1137/21M1441754.

Another Popular Approach for Computing CP Decomposition: Direct/All-at-Once Optimization

$$\min_{\{\mathbf{A}^{(n)}\in\mathbb{R}^{I_n\times R}\}_{n=1}^{N}} f(\mathbf{A}^{(1)},\cdots,\mathbf{A}^{(N)}), \text{ where } f(\mathbf{A}^{(1)},\cdots,\mathbf{A}^{(N)}) = \frac{1}{2}\|\mathcal{X}-[\![\mathbf{A}^{(1)},\mathbf{A}^{(2)},\cdots,\mathbf{A}^{(N)}]\!]\|_F^2$$

- BrasCPD[5]: A block-randomized stochastic method which is a judicious combination of randomized block coordinate descent (BCD) and stochastic proximal gradient (SPG); uniform sampling;
- mBrasCPD[6]: momentum accelerated version;
- iBrasCPD[7]:inertial accelerated version;
- DS-MVR[8] : momentum acceleration and the variance reduction technique.
- Others: [AKD11; ADK11; XY13; PTC13; VD16; HF20; Van21]

[5] Xiao Fu et al. "Block-Randomized Stochastic Proximal Gradient for Low-Rank Tensor Factorization". In: *IEEE Trans. Signal Process.* 68 (2020), pp. 2170–2185. DOI: 10.1109/TSP.2020.2982321..

[6] Qingsong Wang, Chunfeng Cui, and Deren Han. "A Momentum Block-Randomized Stochastic Algorithm for Low-Rank Tensor CP Decomposition". In: *Pac. J. Optim.* 17.3 (2021), pp. 433–452.

[7] Qingsong Wang et al. "Inertial accelerated SGD algorithms for solving large-scale lower-rank tensor CP decomposition problems". In: *J. Comput. Appl. Math.* 423 (2023), p. 114948. DOI: 10.1016/j.cam.2022.114948.

[8] Qingsong Wang, Chunfeng Cui, and Deren Han. "Accelerated Doubly Stochastic Gradient Descent for Tensor CP Decomposition". In: *J. Optim. Theory Appl.* (2023), pp. 1–40. DOI: 10.1007/s10957-023-02193-5.

Presentation Outline

Motivations

The full gradient:

$$\mathbf{G}_{(t)}^{(n)} = \mathbf{A}_{(t)}^{(n)}(\mathbf{Z}^{(n)})^{\intercal}\mathbf{Z}^{(n)} - \mathbf{X}_{(n)}\mathbf{Z}^{(n)}$$

- Combine BrasCPD[9] with empirical importance sampling using efficient sampling technique mentioned above;
1. Leverage-based sampling[10];
2. Euclidean-based sampling[11];
- The theoretical optimal sampling probability distribution is given based on the idea of variance reduction.

[9] Xiao Fu et al. "Block-Randomized Stochastic Proximal Gradient for Low-Rank Tensor Factorization". In: *IEEE Trans. Signal Process.* 68 (2020), pp. 2170–2185. DOI: 10.1109/TSP.2020.2982321..

[10] Brett W. Larsen and Tamara G. Kolda. "Practical Leverage-Based Sampling for Low-Rank Tensor Decomposition". In: *SIAM J. Matrix Anal. Appl.* 43.3 (2022), pp. 1488–1517. DOI: 10.1137/21M1441754.

[11] Deanna Needell, Nathan Srebro, and Rachel Ward. "Stochastic Gradient Descent, Weighted Sampling, and the Randomized Kaczmarz Algorithm". In: *Math. Program.* 155.1 (2016), pp. 549–573. DOI: 10.1007/s10107-015-0864-7.

Exploit KRP Structure to Bound Leverage Scores: Use Leverage Scores of Factor Matrices for Sampling Probabilities

Leverage Scores Key to Limiting Samples(But too Expensive to Compute)

Lemma 2.1 (Leverage Score Bounds for KRP Matrix[12])

For matrices $\mathbf{A}^{(k)} \in \mathbb{R}^{I_k \times R}$ with $I_k > R$ for $k = 1, \cdots, K$, let ℓ_{i_k} be the leverage score of the i_k-th row of $\mathbf{A}^{(k)}$. Then, for the KRP matrix $\mathbf{Z} = \mathbf{A}^{(1)} \odot \mathbf{A}^{(2)} \odot \cdots \odot \mathbf{A}^{(K)}$, the leverage score $\ell_{i_1,\cdots,i_K}$ of its j-th row corresponding to $\{i_1, \cdots, i_K\}$ satisfies

$$\ell_{i_1,\cdots,i_K} \leq \prod_{k=1}^{K} \ell_{i_k}.$$

- The leveraged-based sampling probability for the j-th row of the above KRP matrix $\mathbf{Z}$ can be set to be

$$p_j = \frac{1}{R^K} \prod_{k=1}^{K} \ell_{i_k}. \tag{2.1}$$

[12] Dehua Cheng et al. "SPALS: Fast Alternating Least Squares via Implicit Leverage Scores Sampling". In: *Proceedings of the 30th International Conference on Neural Information Processing Systems*. Barcelona Spain: Curran Associates Inc., 2016, pp. 721–729.

Efficient Sampling Without Forming KRP

Lemma 2.2 (Sampling KRP Efficiently[13])

Let $\mathbf{A}^{(k)} \in \mathbb{R}^{I_k \times R}$ *for* $k = 1, \cdots, K$, *and* $\boldsymbol{\ell}(\mathbf{A}^{(k)})$ *be the vector of leverage scores for* $\mathbf{A}^{(k)}$. *Let*

$$i_k \sim \textsc{multinomial}(\boldsymbol{\ell}(\mathbf{A}^{(k)})/R) \ \text{ for } \ k = 1, \cdots, K.$$

Then, the probability of selecting the multi-index $\{i_1, \cdots, i_K\}$ *is* p_j *in* (2.1).

[13] Brett W. Larsen and Tamara G. Kolda. "Practical Leverage-Based Sampling for Low-Rank Tensor Decomposition". In: *SIAM J. Matrix Anal. Appl.* 43.3 (2022), pp. 1488–1517. DOI: 10.1137/21M1441754.

Use Squared Euclidean Norms of Factor Matrices for Sampling Probabilities

Lemma 2.3

Let $\mathbf{A}^{(k)} \in \mathbb{R}^{I_k \times R}$ with $I_k > R$ for $k = 1, \cdots, K$. For the KRP matrix $\mathbf{Z} = \mathbf{A}^{(1)} \odot \mathbf{A}^{(2)} \odot \cdots \odot \mathbf{A}^{(K)}$, the squared Euclidean norms of its j-th row corresponding to $\{i_1, \cdots, i_K\}$ satisfies

$$\|\mathbf{Z}(j,:)\|_2^2 \leq \prod_{k=1}^{K} \|\mathbf{A}^{(k)}(i_k,:)\|_2^2.$$

- The Euclidean-based sampling probability for the j-th row of the above KRP matrix $\mathbf{Z}$ can be set to be

$$p_j = \prod_{k=1}^{K} \frac{\|\mathbf{A}^{(k)}(i_k,:)\|_2^2}{\|\mathbf{A}^{(k)}\|_F^2}. \tag{2.2}$$

Euclidean-Based Sampling: Efficient Sampling Without Forming KRP

Lemma 2.4

Let $\mathbf{A}^{(k)} \in \mathbb{R}^{I_k \times R}$ for $k = 1, \cdots, K$, and $\boldsymbol{p}_k$ be the Euclidean-based probability distribution for $\mathbf{A}^{(k)}$ with $\beta = 1$, i.e., $\boldsymbol{p}_k(i) = \|\mathbf{A}^{(k)}(i,:)\|_2^2 / \|\mathbf{A}^{(k)}\|_F^2$ with $i \in [I_k]$. Let

$$i_k \sim \text{MULTINOMIAL}(\boldsymbol{p}_k) \ \text{ for } \ k = 1, \cdots, K.$$

Then, the probability of selecting the multi-index $\{i_1, \cdots, i_K\}$ is p_j in (2.2).

Efficient Importance Sampling Without Forming KRP

Algorithm 1 SKRP-ST with importance sampling (SKRP-ST-I)[14]

function $[\mathbf{Z}_{\mathcal{F}_n}^{(n)}, \mathbf{X}_{(n)}^{\mathcal{F}_n}, \boldsymbol{p}_{\mathcal{F}_n}]$ = SKRP-ST-I($n, |\mathcal{F}_n|, \{\boldsymbol{p}_k\}_{k=1,k\neq n}^{N}, \{\mathbf{A}^{(k)}\}_{k=1,k\neq n}^{N}$)
  for $k = 1, \cdots, n-1, n+1, \cdots, N$ **do**
    $\boldsymbol{idx}(:,k) = \textsc{Randsample}(I_k, |\mathcal{F}_n|, true, \boldsymbol{p}_k)$
  end for
  $[\mathbf{Z}_{\mathcal{F}_n}^{(n)}, \mathbf{X}_{(n)}^{\mathcal{F}_n}]$ = SKRP-ST [15]($n, \boldsymbol{idx}, \{\mathbf{A}^{(k)}\}_{k=1,k\neq n}^{N}$)
  $\boldsymbol{p}_{\mathcal{F}_n} \leftarrow$ (8.2) with leverage/Euclidean-based probability distribution
  return $\mathbf{Z}_{\mathcal{F}_n}^{(n)}, \mathbf{X}_{(n)}^{\mathcal{F}_n}, \boldsymbol{p}_{\mathcal{F}_n}$
end function

[14] Brett W. Larsen and Tamara G. Kolda. "Practical Leverage-Based Sampling for Low-Rank Tensor Decomposition". In: *SIAM J. Matrix Anal. Appl.* 43.3 (2022), pp. 1488–1517. DOI: 10.1137/21M1441754.

[15] SKRP-ST: Efficient uniform sampling without forming KRP in [BBK18]; see the Appendix for detailed algorithms.

Importance Sampled Rows by Probability of Selection to Eliminate Bias

- **Stochastic Gradient**:

$$\mathbf{G}_{(t)}^{(n)} = \frac{1}{|\mathcal{F}_n|J_n}\left(\mathbf{A}_{(t)}^{(n)}(\mathbf{D}(\mathbf{Z}_{\mathcal{F}_n}^{(n)})^\intercal)\mathbf{Z}_{\mathcal{F}_n}^{(n)} - \mathbf{D}\mathbf{X}_{(n)}^{\mathcal{F}_n}\mathbf{Z}_{\mathcal{F}_n}^{(n)}\right), \tag{2.3}$$

  where $\mathbf{D} = \mathrm{diag}[\frac{1}{p_{j_1}}, \cdots, \frac{1}{p_{j_{|\mathcal{F}_n|}}}]$ is from sampled leverage- or Euclidean- based probability distribution.
- Define $\xi_{(t)} \in \{1, \cdots, N\}$ and $\zeta_{(t)} \subseteq \{1, \cdots, J_{\xi_{(t)}}\}$ as the random variables (r.v.s) responsible for selecting the mode and fibers in the t-th iteration, respectively.

Theorem 2.5 (Unbiased Gradient)

Denote $\mathcal{B}_{(t)}$ as the filtration generated by the r.v.s

$$\{\xi_{(0)}, \zeta_{(0)}, \xi_{(1)}, \zeta_{(1)}, \cdots, \xi_{(t-1)}, \zeta_{(t-1)}\}$$

such that the t-th iteration $\boldsymbol{\theta}_{(t)}$ is determined conditioned on $\mathcal{B}_{(t)}$. Then the stochastic gradient in (2.3) *is the unbiased estimate of the full gradient with respect to (w.r.t.) $\mathbf{A}^{(\xi_{(t)})}$, i.e.,*

$$\mathbb{E}_{\zeta_{(t)}}\left[\mathbf{G}_{(t)}^{(\xi_{(t)})} \mid \mathcal{B}_{(t)}, \xi_{(t)}\right] = \nabla_{\mathbf{A}^{(\xi_{(t)})}} f(\boldsymbol{\theta}_{(t)}).$$

A Doubly Randomized Computational Framework for Large-Scale CP Decomposition

1. Sampling a mode n from all modes $\{1, \cdots, N\}$ of the tensor;
2. Forming sampled $\mathbf{Z}^{(n)}$ of this mode by sampling latent factors;
3. Sampling some fibers of this mode;
4. Updating the corresponding latent factor via stochastic gradient operations.

- **B**lock-**Ra**ndomized **W**eighted **S**GD for **CPD** (BrawsCPD)
  - Update $\mathbf{A}^{(n)}_{(t+1)} \leftarrow \mathbf{A}^{(n)}_{(t)} - \alpha_t \mathbf{G}^{(n)}_{(t)}$, $\mathbf{A}^{(n')}_{(t+1)} \leftarrow \mathbf{A}^{(n')}_{(t)}$ for $n' \neq n$.
- **Ada**grad version of block-randomized **W**eighted SGD for **CPD** (AdawCPD)
  - Update step size: $[\boldsymbol{\eta}^{(n)}_{(t)}]_{i,r} \leftarrow \frac{\eta}{\left(b+\sum_{t'=1}^{t}[\mathbf{G}^{(n)}_{(t)}]^2_{i,r}\right)^{1/2}}, \ \ i \in [I_n], \ r \in [R].$
  - Update $\mathbf{A}^{(n)}$: $\mathbf{A}^{(n)}_{(t+1)} \leftarrow \mathbf{A}^{(n)}_{(t)} - \boldsymbol{\eta}^{(n)}_{(t)} \circledast \mathbf{G}^{(n)}_{(t)}$, $\mathbf{A}^{(n')}_{(t+1)} \leftarrow \mathbf{A}^{(n')}_{(t)}$ for $n' \neq n$

There Exists a Subsequence of the Solution Sequence that Vonverges to a Stationary Point in Expectation

Theorem 2.6 (Convergence Properties)

Suppose that the step size schedule follows the Robbins-Monro rule: $\sum_{t=0}^{\infty} \alpha^t = \infty$ *and* $\sum_{t=0}^{\infty} (\alpha^t)^2 < \infty$*, and the updates* $\mathbf{A}_{(t)}^{(n)}$ *are bounded for all* n *and* t*. The solution sequence produced by* `BrawsCPD` *satisfies*

$$\liminf_{t \to \infty} \mathbb{E}\left[\|\nabla f(\boldsymbol{\theta}_{(t)})\|_F^2\right] = 0,$$

and the solution sequence produced by `AdawCPD` *satisfies*

$$\Pr\left(\liminf_{t \to \infty} \|\nabla f(\boldsymbol{\theta}_{(t)})\|_F^2 = 0\right) = 1.$$

Optimal Sampling Probability Distribution in the Sense of Minimizing Variance

Theorem 2.7 (Optimal Sampling Probability Distribution)

In the setting of Theorem 2.5, suppose that $\boldsymbol{p} \in \mathbb{R}^{J_n}$ *is any probability distribution and* $\mathbf{R}_{(t)}^{(n)} = \mathbf{A}_{(t)}^{(n)}(\mathbf{Z}_{(t)}^{(n)})^{\intercal} - \mathbf{X}_{(n)}$. *Then if* $\boldsymbol{p}$ *is as*

$$p_i = \frac{\|\mathbf{R}_{(t)}^{(\xi_{(t)})}(:,i)\|_2 \|\mathbf{Z}_{(t)}^{(\xi_{(t)})}(i,:)\|_2}{\sum_{i'=1}^{J_n} \|\mathbf{R}_{(t)}^{(\xi_{(t)})}(:,i')\|_2 \|\mathbf{Z}_{(t)}^{(\xi_{(t)})}(i',:)\|_2}, \ i = 1, \cdots, J_n,$$

$\mathbb{E}_{\zeta_{(t)}}\left[\|\mathbf{G}_{(t)}^{(\xi_{(t)})} - \nabla_{\mathbf{A}^{(\xi_{(t)})}} f(\boldsymbol{\theta}_{(t)})\|_F^2 \mid \mathcal{B}_{(t)}, \xi_{(t)}\right]$ *achieves its minimum as*

$$\frac{1}{|\mathcal{F}_n|}\left(\sum_{j_f=1}^{J_n} \|\mathbf{R}_{(t)}^{(\xi_{(t)})}(:,j_f)\|_2 \|\mathbf{Z}_{(t)}^{(\xi_{(t)})}(j_f,:)\|_2\right)^2 - \frac{1}{|\mathcal{F}_n|}\|\nabla_{\mathbf{A}^{(\xi_{(t)})}} f(\boldsymbol{\theta}_{(t)})\|_F^2.$$

Presentation Outline

Setup

- Data generation[16]
  - Generating by factor matrices with parameters R_{true}, $noise$, $spread$ and $magnitude$.
  1. Generating factor matrices with $spread$ (how many non-zeros elements are added to each of these first three columns) and $magnitude$ (those non-zero elements are chosen).
  2. $\mathcal{X}_{true} = [\![\mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \mathbf{A}^{(3)}]\!]$.
  3. Adding noise: $\mathcal{X} = \mathcal{X}_{true} + noise \left(\frac{\|\mathcal{X}_{true}\|}{\|\mathcal{N}\|} \right) \mathcal{N}$.
- Experimental setup
  - Checking convergence: $Tol = \frac{\|[\![\hat{\mathbf{A}}^{(1)}, \hat{\mathbf{A}}^{(2)}, \hat{\mathbf{A}}^{(3)}]\!] - \mathcal{X}\|_F^2}{\|\mathcal{X}\|_F^2}$.
  - Number of trials: 10.
  - Environmental: 2.3 GHz 8-Core Intel Core i9 CPU with 16 GB 2400 MHz DDR4 memory.

[16] Brett W. Larsen and Tamara G. Kolda. "Practical Leverage-Based Sampling for Low-Rank Tensor Decomposition". In: *SIAM J. Matrix Anal. Appl.* 43.3 (2022), pp. 1488–1517. DOI: 10.1137/21M1441754.

Different Tensor Size

Table: Performance of the algorithms with $Tol = 10^{-5}$, $|\mathcal{F}_n| = 18$, the target rank $R = 10$, $noise = 0$, and random initialization for different tensors generated by $I \times 10$ factor matrices with different I.

Algorithms		$I = 100$ $spread = 15$, $magnitude = 24$	$I = 200$ $spread = 30$, $magnitude = 30$	$I = 300$ $spread = 45$, $magnitude = 36$	$I = 400$ $spread = 60$, $magnitude = 42$	$I = 500$ $spread = 75$, $magnitude = 48$
AdaCPD [Fu+20]	Iterations	5962.7	4206.3	3105.8	3271.9	4229.5
	Seconds	23.997029	131.46287	472.43042	1241.3702	3224.5938
EAdawCPD	Iterations	2242.1	572.2	390.3	391.1	488.6
	Seconds	9.1444463	18.08937	58.656213	150.04822	372.96764
LAdawCPD	Iterations	2394.9	577.8	429.9	483.2	607.3
	Seconds	10.634388	18.25461	64.193687	184.35716	463.1262

- More numerical results see the Appendix:

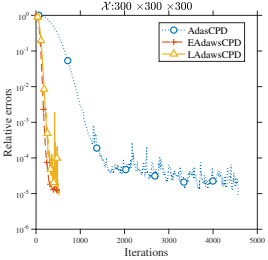
1. Different targer rank;
2. Different noise;
3. Nonnegative constraint;
4. Real data.

Real Data: Performance

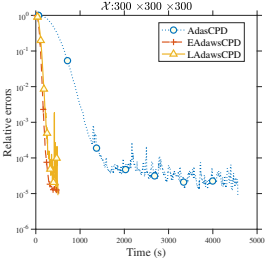
Table: Performance of algorithms on real datasets.

Algorithms		SalinasA. ($83 \times 86 \times 224$)	Indian Pines ($145 \times 145 \times 220$)	Pavia Uni. ($610 \times 340 \times 103$)
		$R = 10$, $\lvert\mathcal{F}_n\rvert = 20$	$R = 10$, $\lvert\mathcal{F}_n\rvert = 20$	$R = 100$, $\lvert\mathcal{F}_n\rvert = 20$
AdaCPD [Fu+20]	Tol	0.00697493	0.00782241	0.02831117
	Seconds	288.535306	653.276364	4387.12147
EAdawCPD	Tol	0.00611473	0.00725646	0.02792415
	Seconds	291.374795	659.02316	4450.21809
LAdawCPD	Tol	0.00644397	0.00741898	0.02796972
	Seconds	295.962095	663.648648	4559.80737

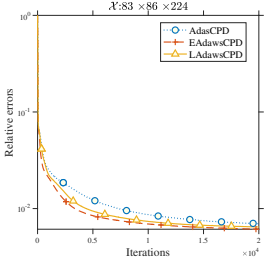
Visualisation of Experimental Results



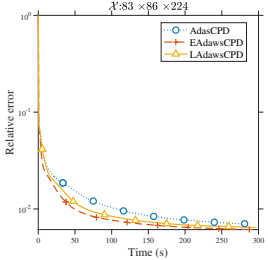
(a) Number of iterations v.s. Relative errors: Sythetic $I = 300$



(b) Time v.s. Relative errors: Sythetic $I = 300$



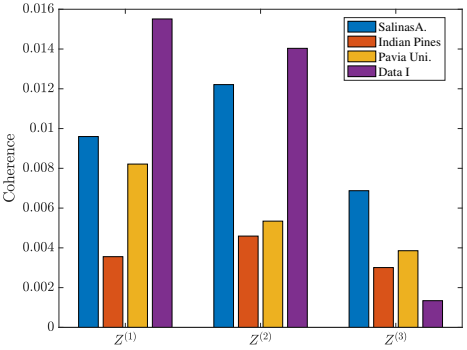
(c) Number of iterations v.s. Relative errors: SalinasA



(d) Time v.s. Relative errors: SalinasA

Figure: (a)-(b) Output by the algorithms with $Tol = 10^{-5}$, $|\mathcal{F}_n| = 18$, $R = 10$, $noise = 0$, and random initialization for the tensor with $I = 300$, $R_{true} = 10$, $spread = 45$, and $magnitude = 36$; (c)-(d) Output by the algorithms with $|\mathcal{F}_n| = 20$ and $R = 10$ for SalinasA.

Some Interpretations of the Experimental Results



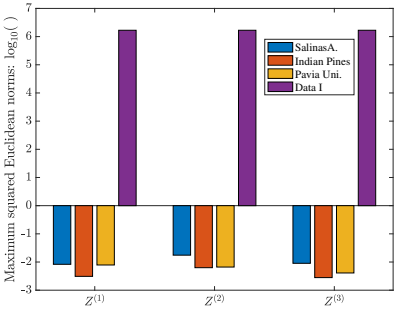
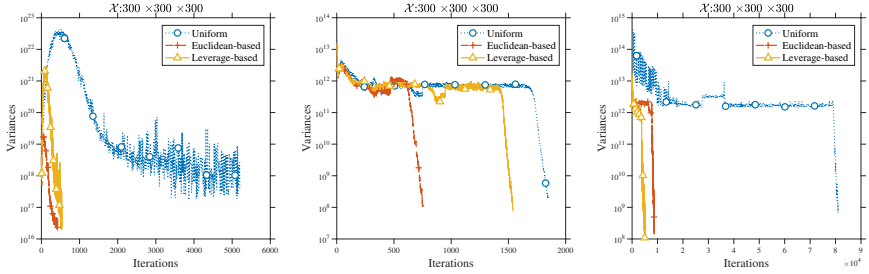


Figure: **Left**: Coherence for four different tensors.; **Right**: Maximum squared Euclidean norms for four different tensors.

Compare Variances of Different Data

- **Data I**: Previous data.
- **Data II**: Generated by three factor matrices of size 300×10 with independent standard Gaussian entries.
- **Data III**: The same as **Data II** except that one entry of each factor matrices is chosen uniformly and set to 20.



(a) Data I **(b)** Data II **(c)** Data III

Figure: Comparison of variances of stochastic gradients for different tensors.

Presentation Outline

Conclusions

- Conclusions
  - We propose a block-randomized gradient descent method with importance sampling for CP decomposition based on two empirical sampling strategies.
  - We provide a theoretical optimal sampling probability contribution.
  - Numerical experiments are provided to test the proposed methods.
- Future works
  - Momentum version[17].
  - Variance reduction technique[18].
  - Scaled technique[19].

[17] Qingsong Wang, Chunfeng Cui, and Deren Han. "A Momentum Block-Randomized Stochastic Algorithm for Low-Rank Tensor CP Decomposition". In: *Pac. J. Optim.* 17.3 (2021), pp. 433–452.

[18] Qingsong Wang, Chunfeng Cui, and Deren Han. "Accelerated Doubly Stochastic Gradient Descent for Tensor CP Decomposition". In: *J. Optim. Theory Appl.* (2023), pp. 1–40. DOI: 10.1007/s10957-023-02193-5.

[19] Tian Tong, Cong Ma, and Yuejie Chi. "Accelerating Ill-Conditioned Low-Rank Matrix Estimation via Scaled Gradient Descent.". In: *J. Mach. Learn. Res.* 22.150 (2021), pp. 1–63.

Thanks!

Y.J. Yu and H.Y. Li. 'A Block-Randomized Stochastic Method with Importance Sampling for CP Tensor Decomposition"on arXiv.

References I

[ADK11] Evrim Acar, Daniel M Dunlavy, and Tamara G Kolda. "A scalable optimization approach for fitting canonical tensor decompositions". In: *J. Chemom.* 25.2 (2011), pp. 67–86.

[AKD11] Evrim Acar, Tamara G Kolda, and Daniel M Dunlavy. *All-at-once optimization for coupled matrix and tensor factorizations*. arXiv:1105.3422. 2011.

[BBK18] Casey Battaglino, Grey Ballard, and Tamara G. Kolda. "A Practical Randomized CP Tensor Decomposition". In: *SIAM J. Matrix Anal. Appl.* 39.2 (2018), pp. 876–901. DOI: 10.1137/17M1112303.

[Che+16] Dehua Cheng et al. "SPALS: Fast Alternating Least Squares via Implicit Leverage Scores Sampling". In: *Proceedings of the 30th International Conference on Neural Information Processing Systems*. Barcelona Spain: Curran Associates Inc., 2016, pp. 721–729.

References II

[Dri+12] Petros Drineas et al. “Fast Approximation of Matrix Coherence and Statistical Leverage”. In: *J. Mach. Learn. Res.* 13.1 (2012), pp. 3475–3506.

[Fu+20] Xiao Fu et al. “Block-Randomized Stochastic Proximal Gradient for Low-Rank Tensor Factorization”. In: *IEEE Trans. Signal Process.* 68 (2020), pp. 2170–2185. DOI: 10.1109/TSP.2020.2982321..

[HF20] Kejun Huang and Xiao Fu. “Low-Complexity Levenberg-Marquardt Algorithm for Tensor Canonical Polyadic Decomposition”. In: *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2020, pp. 3922–3926.

[KB09] Tamara G. Kolda and Brett W. Bader. “Tensor Decompositions and Applications”. In: *SIAM Rev.* 51.3 (2009), pp. 455–500. DOI: 10.1137/07070111X.

References III

[LK22] Brett W. Larsen and Tamara G. Kolda. “Practical Leverage-Based Sampling for Low-Rank Tensor Decomposition”. In: *SIAM J. Matrix Anal. Appl.* 43.3 (2022), pp. 1488–1517. DOI: 10.1137/21M1441754.

[MMY15] Ping Ma, Michael Mahoney, and Bin Yu. “A statistical Perspective on algorithmic leveraging”. In: *J. Mach. Learn. Res.* 16.27 (2015), pp. 861–911. DOI: 10.1002/cem.1335.

[NSW16] Deanna Needell, Nathan Srebro, and Rachel Ward. “Stochastic Gradient Descent, Weighted Sampling, and the Randomized Kaczmarz Algorithm”. In: *Math. Program.* 155.1 (2016), pp. 549–573. DOI: 10.1007/s10107-015-0864-7.

[PTC13] Anh-Huy Phan, Petr Tichavský, and Andrzej Cichocki. “Low Complexity Damped Gauss–Newton Algorithms for CANDECOMP/PARAFAC”. In: *SIAM J. Matrix Anal. Appl.* 34.1 (2013), pp. 126–147. DOI: 10.1137/100808034.

References IV

[TMC21] Tian Tong, Cong Ma, and Yuejie Chi. “Accelerating Ill-Conditioned Low-Rank Matrix Estimation via Scaled Gradient Descent.”. In: *J. Mach. Learn. Res.* 22.150 (2021), pp. 1–63.

[Van21] Michiel Vandecappelle. “Numerical Algorithms for Tensor Decompositions”. PhD thesis. Arenberg Doctoral School, 2021.

[VD16] Nico Vervliet and Lieven De Lathauwer. “A Randomized Block Sampling Approach to Canonical Polyadic Decomposition of Large-Scale Tensors”. In: *IEEE J. Sel. Topics Signal Process.* 10.2 (2016), pp. 284–295. DOI: 10.1109/JSTSP.2015.2503260.

[Wan+23] Qingsong Wang et al. “Inertial accelerated SGD algorithms for solving large-scale lower-rank tensor CP decomposition problems”. In: *J. Comput. Appl. Math.* 423 (2023), p. 114948. DOI: 10.1016/j.cam.2022.114948.

References V

[WCH21] Qingsong Wang, Chunfeng Cui, and Deren Han. "A Momentum Block-Randomized Stochastic Algorithm for Low-Rank Tensor CP Decomposition". In: *Pac. J. Optim.* 17.3 (2021), pp. 433–452.

[WCH23] Qingsong Wang, Chunfeng Cui, and Deren Han. "Accelerated Doubly Stochastic Gradient Descent for Tensor CP Decomposition". In: *J. Optim. Theory Appl.* (2023), pp. 1–40. DOI: 10.1007/s10957-023-02193-5.

[Woo14] David P. Woodruff. "Sketching as a Tool for Numerical Linear Algebra". In: *Found. Trends Theor. Comput. Sci.* 10.1–2 (2014), pp. 1–157. DOI: 10.1561/0400000060.

[XY13] Yangyang Xu and Wotao Yin. "A Block Coordinate Descent Method for Regularized Multiconvex Optimization with Applications to Nonnegative Tensor Factorization and Completion". In: *SIAM J. Imaging Sci.* 6.3 (2013), pp. 1758–1789. DOI: 10.1137/120887795.

Presentation Outline

Algorithm 2 CP-ALS[20]

function $[\boldsymbol{\lambda}, \{\mathbf{A}^{(n)}\}_{n=1}^{N}]$ = CP-ALS($\mathcal{X}$, R, $\{\mathbf{A}_{(0)}^{(n)}\}_{n=1}^{N}$)

▷ N-way tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$;

▷ rank R, initialization $\{\mathbf{A}_{(0)}^{(n)}\}_{n=1}^{N}$

  repeat

    for $n = 1, \cdots, N$ **do**

      $\mathbf{V} \leftarrow \mathbf{A}^{(1)\intercal}\mathbf{A}^{(1)} * \cdots * \mathbf{A}^{(n-1)\intercal}\mathbf{A}^{(n-1)} * \mathbf{A}^{(n+1)\intercal}\mathbf{A}^{(n+1)} * \cdots * \mathbf{A}^{(N)\intercal}\mathbf{A}^{(n)}$

      $\mathbf{A}^{(n)} \leftarrow \mathbf{X}_{(n)}(\mathbf{A}^{(N)} \odot \cdots \odot \mathbf{A}^{(n+1)} \odot \mathbf{A}^{(n-1)} \odot \cdots \odot \mathbf{A}^{(1)})\mathbf{V}^{\dagger}$

      normalize columns of $\mathbf{A}^{(n)}$ (storing norms as $\boldsymbol{\lambda}$)

    end for

  until some stopping criterion is reached

  return $\boldsymbol{\lambda}, \{\mathbf{A}^{(n)}\}_{n=1}^{N}$

end function

[20] Tamara G. Kolda and Brett W. Bader. "Tensor Decompositions and Applications". In: *SIAM Rev.* 51.3 (2009), pp. 455–500. DOI: 10.1137/07070111X.

Algorithm 3 SKRP and Sampled tensor fibers (SKRP-ST)[21]

function $[\mathbf{Z}_{\mathcal{F}_n}^{(n)}, \mathbf{X}_{(n)}^{\mathcal{F}_n}]$ = SKRP-ST(n, $\boldsymbol{idx}$, $\{\mathbf{A}^{(k)}\}_{k=1,k\neq n}^{N}$)
  $\mathbf{Z}_{\mathcal{F}_n}^{(n)} \leftarrow \mathbf{1}$ ▷ $\mathbf{1} \in \mathbb{R}^{|\mathcal{F}_n| \times R}$ a matrix with all elements 1
  for $m = 1, \cdots, n-1, n+1, \cdots, N$ **do**
    $\mathbf{A}_{|\mathcal{F}_n|}^{(m)} \leftarrow \mathbf{A}^{(m)}(\boldsymbol{idx}(:, m), :)$
    $\mathbf{Z}_{\mathcal{F}_n}^{(n)} \leftarrow \mathbf{Z}_{\mathcal{F}_n}^{(n)} \circledast \mathbf{A}_{|\mathcal{F}_n|}^{(m)}$
  end for
  $\mathcal{X}^{\mathcal{F}_n} \leftarrow \mathcal{X}(\boldsymbol{idx}(:, 1), \cdots, \mathcal{X}(\boldsymbol{idx}(:, n-1), :, \mathcal{X}(\boldsymbol{idx}(:, n+1), \cdots, \mathcal{X}(\boldsymbol{idx}(:, N))$
  $\mathbf{X}_{(n)}^{\mathcal{F}_n} \leftarrow$ UNFOLDING($\mathcal{X}^{\mathcal{F}_n}, n$)
  return $\mathbf{Z}_{\mathcal{F}_n}^{(n)}, \mathbf{X}_{(n)}^{\mathcal{F}_n}$
end function

[21] Casey Battaglino, Grey Ballard, and Tamara G. Kolda. "A Practical Randomized CP Tensor Decomposition". In: *SIAM J. Matrix Anal. Appl.* 39.2 (2018), pp. 876–901. DOI: 10.1137/17M1112303.

Algorithm 4 CPRAND[22]

function $[\boldsymbol{\lambda}, \{\mathbf{A}^{(n)}\}_{n=1}^{N}]$ = CPRAND($\mathcal{X}$, R, $\{\mathbf{A}_{(0)}^{(n)}\}_{n=1}^{N}$, $|\mathcal{F}_n|$)

▷ N-way tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$;

▷ rank R, sample size $|\mathcal{F}_n|$,initialization $\{\mathbf{A}_{(0)}^{(n)}\}_{n=1}^{N}$

  repeat

    for $n = 1, \cdots, N$ **do**

      get $\boldsymbol{idx}$ using uniform sampling

      $[\mathbf{Z}_{\mathcal{F}_n}^{(n)}, \mathbf{X}_{(n)}^{\mathcal{F}_n}]$ = SKRP-ST(n, $\boldsymbol{idx}$, $\{\mathbf{A}^{(k)}\}_{k=1,k\neq n}^{N}$)

      $\mathbf{A}^{(n)} \leftarrow \arg\min_{\mathbf{A}} \|\mathbf{Z}_{\mathcal{F}_n}^{(n)}\mathbf{A}^{\intercal} - \mathbf{X}_{(n)}^{\mathcal{F}_n}\|_F$

    end for

  until some stopping criterion is reached

  return $\{\mathbf{A}^{(n)}\}_{n=1}^{N}$

end function

[22] Casey Battaglino, Grey Ballard, and Tamara G. Kolda. "A Practical Randomized CP Tensor Decomposition". In: *SIAM J. Matrix Anal. Appl.* 39.2 (2018), pp. 876–901. DOI: 10.1137/17M1112303.

Algorithm 5 CP-ALS with leveraged-based sampling[23]

function $[\boldsymbol{\lambda}, \{\mathbf{A}^{(n)}\}_{n=1}^{N}]$ = CPRAND($\mathcal{X}$, R, $\{\mathbf{A}_{(0)}^{(n)}\}_{n=1}^{N}$, $|\mathcal{F}_n|$)

▷ N-way tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$;

▷ rank R, sample size $|\mathcal{F}_n|$,initialization $\{\mathbf{A}_{(0)}^{(n)}\}_{n=1}^{N}$

Compute the leverage scores of factor matrices $\{\mathbf{A}^{(0)}\}_{n=1}^{N}$ and get the probability distributions

repeat

for $n = 1, \cdots, N$ **do**

get $\boldsymbol{idx}$ using leverage-based sampling with the probability distributions

$[\mathbf{Z}_{\mathcal{F}_n}^{(n)}, \mathbf{X}_{(n)}^{\mathcal{F}_n}]$ = SKRP-ST(n, $\boldsymbol{idx}$, $\{\mathbf{A}^{(k)}\}_{k=1,k\neq n}^{N}$)

$\mathbf{A}^{(n)} \leftarrow \arg\min_{\mathbf{A}} \|\mathbf{Z}_{\mathcal{F}_n}^{(n)}\mathbf{A}^{\intercal} - \mathbf{X}_{(n)}^{\mathcal{F}_n}\|_F$

Recompute the leverage scores of $\mathbf{A}^{(n)}$ and according probability distributions

end for

until some stopping criterion is reached

return $\{\mathbf{A}^{(n)}\}_{n=1}^{N}$

end function

[23] Brett W. Larsen and Tamara G. Kolda. "Practical Leverage-Based Sampling for Low-Rank Tensor Decomposition". In: *SIAM J. Matrix Anal. Appl.* 43.3 (2022), pp. 1488–1517. DOI: 10.1137/21M1441754.

Algorithm 6 BrasCPD[24]

function $\{\mathbf{A}^{(n)}\}_{n=1}^{N}$ = BrasCPD($\mathcal{X}$, R, $|\mathcal{F}_n|$, $\{\mathbf{A}_{(0)}^{(n)}\}_{n=1}^{N}$, $\{\alpha^t\}$) ▷ N-way tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$;
▷ rank R, sample size $|\mathcal{F}_n|$; initialization $\{\mathbf{A}_{(0)}^{(n)}\}_{n=1}^{N}$, step size $\{\alpha^t\}_{t=0,1,\cdots}$
  $t \leftarrow 0$
  repeat
    Uniformly sample n from $\{1, \cdots, N\}$
    Sample $\mathcal{F}_n$ uniformly from $\{1, \cdots, J_n\}$
    Compute $\mathbf{G}_{(t)}^{(n)} = \frac{1}{|\mathcal{F}_n|}\left(\mathbf{A}_{(t)}^{(n)}(\mathbf{Z}_{\mathcal{F}_n}^{(n)})^{\intercal}\mathbf{Z}_{\mathcal{F}_n}^{(n)} - \mathbf{X}_{(n)}^{\mathcal{F}_n}\mathbf{Z}_{\mathcal{F}_n}^{(n)}\right)$
    Update $\mathbf{A}_{(t+1)}^{(n)} \leftarrow \mathbf{A}_{(t)}^{(n)} - \alpha_t \mathbf{G}_{(t)}^{(n)}$, and $\mathbf{A}_{(t+1)}^{(n')} \leftarrow \mathbf{A}_{(t)}^{(n')}$ for $n' \neq n$
    $t \leftarrow t + 1$
  until some stopping criterion is reached
  return $\{\mathbf{A}^{(n)}\}_{n=1}^{N}$
end function

[24] Xiao Fu et al. "Block-Randomized Stochastic Proximal Gradient for Low-Rank Tensor Factorization". In: *IEEE Trans. Signal Process.* 68 (2020), pp. 2170–2185. DOI: 10.1109/TSP.2020.2982321..

Algorithm 7 BrawsCPD

function $\{\mathbf{A}^{(n)}\}_{n=1}^{N}$= BrawsCPD($\mathcal{X}, R, |\mathcal{F}_n|, \{\mathbf{A}_{(0)}^{(n)}\}_{n=1}^{N}, \{\alpha_t\}$)
  $t \leftarrow 0$
  repeat
    Uniformly sample n from $\{1, \cdots, N\}$
    $[\mathbf{Z}_{\mathcal{F}_n}^{(n)}, \mathbf{X}_{(n)}^{\mathcal{F}_n}, \boldsymbol{p}_{\mathcal{F}_n}]$ = SKRP-ST-I($n, |\mathcal{F}_n|, \{\boldsymbol{p}_k\}_{k=1,k\neq n}^{N}, \{\mathbf{A}^{(N)}\}_{k=1,k\neq n}^{N}$)
    Form the stochastic gradient $\mathbf{G}_{(t)}^{(n)} \leftarrow \frac{1}{|\mathcal{F}_n|J_n}\left(\mathbf{A}_{(t)}^{(n)}(\mathbf{D}(\mathbf{Z}_{\mathcal{F}_n}^{(n)})^{\intercal})\mathbf{Z}_{\mathcal{F}_n}^{(n)} - \mathbf{D}\mathbf{X}_{(n)}^{\mathcal{F}_n}\mathbf{Z}_{\mathcal{F}_n}^{(n)}\right)$
    Update $\mathbf{A}_{(t+1)}^{(n)} \leftarrow \mathbf{A}_{(t)}^{(n)} - \alpha_t\mathbf{G}_{(t)}^{(n)}$, $\mathbf{A}_{(t+1)}^{(n')} \leftarrow \mathbf{A}_{(t)}^{(n')}$ for $n' \neq n$
    $t \leftarrow t + 1$
  until some stopping criterion is reached
  return $\{\mathbf{A}^{(n)}\}_{n=1}^{N}$
end function

Algorithm 8 AdawCPD

function $\{\mathbf{A}^{(n)}\}_{n=1}^{N}$ = AdawCPD($\mathcal{X}, R, |\mathcal{F}_n|, \{\mathbf{A}_{(0)}^{(n)}\}_{n=1}^{N}$)
  $t \leftarrow 0$
  repeat
    Uniformly sample n from $\{1, \cdots, N\}$
    $[\mathbf{Z}_{\mathcal{F}_n}^{(n)}, \mathbf{X}_{(n)}^{\mathcal{F}_n}, \boldsymbol{p}_{\mathcal{F}_n}]$ = SKRP-ST-I($n, |\mathcal{F}_n|, \{\boldsymbol{p}_k\}_{k=1,k\neq n}^{N}, \{\mathbf{A}^{(N)}\}_{k=1,k\neq n}^{N}$)
    Form the stochastic gradient $\mathbf{G}_{(t)}^{(n)} \leftarrow \frac{1}{|\mathcal{F}_n| J_n} \left(\mathbf{A}_{(t)}^{(n)} (\mathbf{D}(\mathbf{Z}_{\mathcal{F}_n}^{(n)})^{\intercal}) \mathbf{Z}_{\mathcal{F}_n}^{(n)} - \mathbf{D}\mathbf{X}_{(n)}^{\mathcal{F}_n} \mathbf{Z}_{\mathcal{F}_n}^{(n)} \right)$
    Determine the step size $\boldsymbol{\eta}_{(t)}^{(n)} \leftarrow \frac{\eta}{\left(b + \sum_{t'=1}^{t} [\mathbf{G}_{(t)}^{(n)}]_{i,r}^2\right)^{1/2}}$
    Update $\mathbf{A}_{(t+1)}^{(n)} \leftarrow \mathbf{A}_{(t)}^{(n)} - \alpha_t \mathbf{G}_{(t)}^{(n)}$, $\mathbf{A}_{(t+1)}^{(n')} \leftarrow \mathbf{A}_{(t)}^{(n')}$ for $n' \neq n$
    $t \leftarrow t + 1$
  until some stopping criterion is reached
  return $\{\mathbf{A}^{(n)}\}_{n=1}^{N}$
end function

Presentation Outline

Different tensor size

Table: Performance of the algorithms with $Tol = 10^{-5}$, $|\mathcal{F}_n| = 18$, the target rank $R = 10$, $noise = 0$, and random initialization for different tensors generated by $I \times 10$ factor matrices with different I.

Algorithms		$I = 100$	$I = 200$	$I = 300$	$I = 400$	$I = 500$
		$spread = 15$, $magnitude = 24$	$spread = 30$, $magnitude = 30$	$spread = 45$, $magnitude = 36$	$spread = 60$, $magnitude = 42$	$spread = 75$, $magnitude = 48$
AdasCPD [Fu+20]	Iterations	5962.7	4206.3	3105.8	3271.9	4229.5
	Seconds	23.997029	131.46287	472.43042	1241.3702	3224.5938
EAdawsCPD	Iterations	2242.1	572.2	390.3	391.1	488.6
	Seconds	9.1444463	18.08937	58.656213	150.04822	372.96764
LAdawsCPD	Iterations	2394.9	577.8	429.9	483.2	607.3
	Seconds	10.634388	18.25461	64.193687	184.35716	463.1262

Different Tensor Size

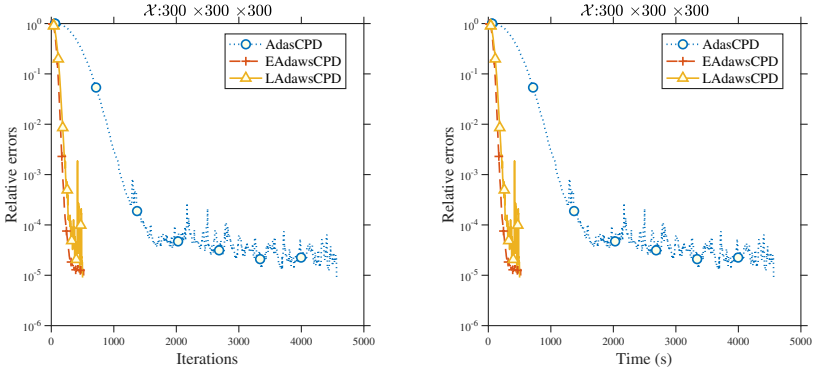


Figure: Number of iterations v.s. Relative errors and Time v.s. Relative errors output by the algorithms with $Tol = 10^{-5}$, $|\mathcal{F}_n| = 18$, $R = 10$, $noise = 0$, and random initialization for the tensor with $I = 300$, $R_{true} = 10$, $spread = 45$, and $magnitude = 36$.

Different Target Rank

Table: Performance of the algorithms with $Tol = 10^{-5}$, $|\mathcal{F}_n| = 18$, different target ranks R, $noise = 0$, and random initialization for the tensor generated by 300×10 factor matrices with $spread = 45$ and $magnitude = 36$.

Algorithms		$R = 5$	$R = 10$	$R = 15$	$R = 20$
AdaCPD [Fu+20]	Iterations	3059.1	3105.8	4823.7	7094.8
	Seconds	480.47423	472.43042	767.12695	1125.8787
EAdawCPD	Iterations	474	390.3	406.7	824.9
	Seconds	73.963123	58.656213	65.151505	132.01017
LAdawCPD	Iterations	535.7	429.9	567.1	827.1
	Seconds	83.791824	64.193687	90.722473	132.73588

Different Noise

Table: Performance of the algorithms with $Tol = 10^{-5}$, $|\mathcal{F}_n| = 18$, the target rank $R = 10$, and random initialization for different tensors generated by 300×10 factor matrices with $spread = 45$, $magnitude = 36$, and different noises.

Algorithms		$noise = 0$	$noise = 0.01$	$noise = 0.1$	$noise = 1$
AdaCPD [Fu+20]	Iterations	3105.8	3211.7	3024.9	3511.2
	Seconds	472.430421	505.928736	481.988384	553.33599
EAdawCPD	Iterations	390.3	360.5	381.6	414.6
	Seconds	58.6562125	56.748149	60.5743992	65.3003044
LAdawCPD	Iterations	429.9	410	439	452.1
	Seconds	64.1936869	64.6965127	70.8384762	71.6683267

Nonnegative Constraint

Table: Performance of the algorithms with $Tol = 10^{-5}$, $|\mathcal{F}_n| = 18$, the target rank $R = 10$, $noise = 0$, and random initialization for different tensors generated by $I \times 10$ factor matrices with different I under nonnegative constraint.

Algorithms		$I = 100$ $spread = 15$, $magnitude = 24$	$I = 200$ $spread = 30$, $magnitude = 30$	$I = 300$ $spread = 45$, $magnitude = 36$	$I = 400$ $spread = 60$, $magnitude = 42$	$I = 500$ $spread = 75$, $magnitude = 48$
AdaCPD [Fu+20]	Iterations	500.1	910.7	1424.5	2203.7	3372.8
	Seconds	2.0256353	28.981344	225.79166	737.83326	2550.5717
EAdawCPD	Iterations	156.6	190.6	250.6	319	394.2
	Seconds	0.6437757	6.0335162	40.041681	110.00088	299.55425
LAdawCPD	Iterations	153.7	205.9	271.5	337.5	437.6
	Seconds	0.6892614	6.6271954	43.308729	111.84547	331.55795

Nonnegative Constraint

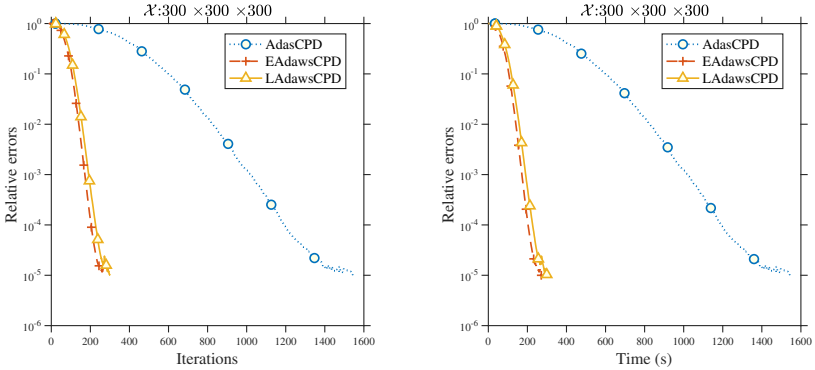


Figure: Number of iterations v.s. Relative errors and Time v.s. Relative errors output by the algorithms with $Tol = 10^{-5}$, $|\mathcal{F}_n| = 18$, $R = 10$, $noise = 0$, and random initialization for the tensor with $I = 300$, $R_{true} = 10$, $spread = 45$, $magnitude = 36$, and $\mathbf{A}^{(n)} \geq \mathbf{0}$.

Real data

Table: Size and Type of Real Datasets.

Dataset	Size	Type
SalinasA.	$83 \times 86 \times 224$	Hyperspectral
Indian Pines	$145 \times 145 \times 220$	Hyperspectral
Pavia Uni.	$610 \times 340 \times 103$	Hyperspectral

Real Data: Performance

Table: Performance of algorithms on real datasets.

Algorithms		SalinasA. $R = 10$, $\lvert\mathcal{F}_n\rvert = 20$	Indian Pines $R = 10$, $\lvert\mathcal{F}_n\rvert = 20$	Pavia Uni. $R = 100$, $\lvert\mathcal{F}_n\rvert = 20$
AdaCPD [Fu+20]	Tol	0.00697493	0.00782241	0.02831117
	Seconds	288.535306	653.276364	4387.12147
EAdawCPD	Tol	0.00611473	0.00725646	0.02792415
	Seconds	291.374795	659.02316	4450.21809
LAdawCPD	Tol	0.00644397	0.00741898	0.02796972
	Seconds	295.962095	663.648648	4559.80737

Real Data: Performance

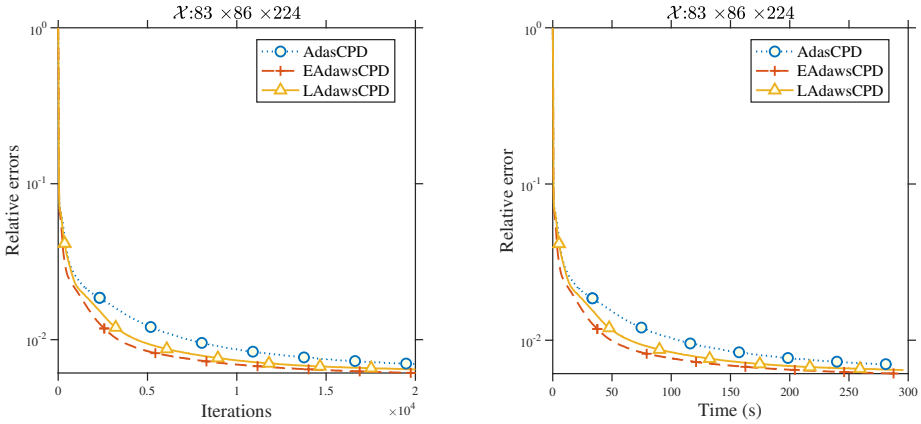
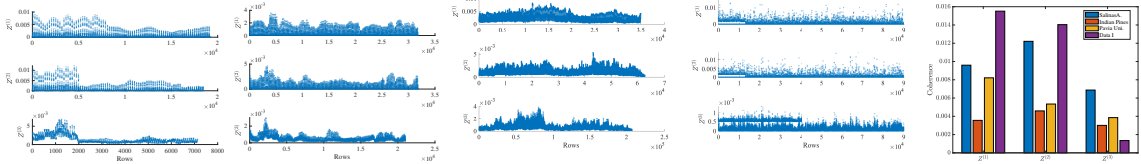


Figure: Number of iterations v.s. Relative errors and Time v.s. Relative errors output by the algorithms with $|\mathcal{F}_n| = 20$ and $R = 10$ for SalinasA..

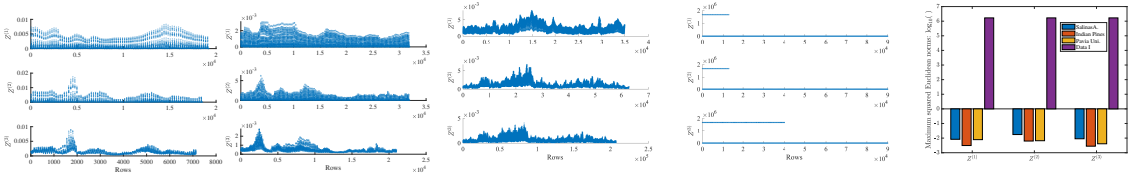
More Interpretations of the Experimental Results



(a) SalinasA. **(b)** Indian Pines **(c)** Pivia Uni. **(d)** Data I **(e)** Bar chart

Figure: (a)-(d) Leverage scores for four different tensors. (e) Coherence for four different tensors.

More Interpretations of the Experimental Results



(a) SalinasA. **(b)** Indian Pines **(c)** Pivia Uni. **(d)** Data I **(e)** Bar chart

Figure: (a)-(d) Squared Euclidean norms for four different tensors. (e) Maximum squared Euclidean norms for four different tensors.

Presentation Outline

Definition 8.1 (Leverage Scores [Dri+12])

Let $\mathbf{A} \in \mathbb{R}^{m \times n}$ with $m > n$, and let $\mathbf{Q} \in \mathbb{R}^{m \times n}$ be any orthogonal basis for the column space of $\mathbf{A}$. The *leverage score* of the i-th row of $\mathbf{A}$ is given by

$$\ell_i(\mathbf{A}) = \|\mathbf{Q}(i,:)\|_2^2.$$

Definition 8.2 (Leveraged-based Probability Distribution [Woo14])

Let $\mathbf{A} \in \mathbb{R}^{m \times n}$ with $m > n$. We say a probability distribution $\boldsymbol{p} = [p_1, \cdots, p_m]^\intercal$ is a *leveraged-based probability distribution* for $\mathbf{A}$ if $p_i \geq \beta \frac{\ell_i(\mathbf{A})}{n}$ with $0 < \beta \leq 1$ and $i \in [m]$.

Definition 8.3 (Euclidean-based Probability Distribution)

Let $\mathbf{A} \in \mathbb{R}^{m \times n}$ with $m > n$. We say a probability distribution $\boldsymbol{p} = [p_1, \cdots, p_m]^\intercal$ is an *Euclidean-based probability distribution* for $\mathbf{A}$ if $p_i \geq \beta \|\mathbf{A}(i,:)\|_2^2 / \|\mathbf{A}\|_F^2$ with $0 < \beta \leq 1$ and $i \in [m]$.

Mini-Batches and Their Sampling Probabilities

1. First sample $|\mathcal{F}_n|$ rows from each $\{\mathbf{A}^{(k)}\}_{k=1,k\neq n}^{N}$ using the leveraged-based probability distribution for $\mathbf{A}^{(k)}$, i.e., $\boldsymbol{p}_k = \frac{\boldsymbol{\ell}(\mathbf{A}^{(k)})}{R}$. Thus, we can get the $\boldsymbol{idx}$:

$$\begin{bmatrix} \{i_1^{(j_1)} & \cdots & i_{n-1}^{(j_1)} & i_{n+1}^{(j_1)} & \cdots & i_N^{(j_1)}\} \\ \{i_1^{(j_2)} & \cdots & i_{n-1}^{(j_2)} & i_{n+1}^{(j_2)} & \cdots & i_N^{(j_2)}\} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \{i_1^{(j_{|\mathcal{F}_n|})} & \cdots & i_{n-1}^{(j_{|\mathcal{F}_n|})} & i_{n+1}^{(j_{|\mathcal{F}_n|})} & \cdots & i_N^{(j_{|\mathcal{F}_n|})}\} \end{bmatrix}. \tag{8.1}$$

2. Based on (8.1), we can obtain the index set $\mathcal{F}_n = \{j_1, j_2, \cdots, j_{|\mathcal{F}_n|}\}$ of the sampled $|\mathcal{F}_n|$ rows of $\mathbf{Z}^{(n)}$, and the corresponding sampling probabilities

$$\boldsymbol{p}_{\mathcal{F}_n} = [p_{j_1}, \cdots, p_{j_{|\mathcal{F}_n|}}]^{\mathsf{T}}, \tag{8.2}$$

where $p_{j_f} = \frac{\bar{\ell}_{j_f}(\mathbf{Z}^{(n)})}{R^{N-1}}$ and $\bar{\ell}_{j_f}(\mathbf{Z}^{(n)}) = \prod_{k=1,k\neq n}^{N} \ell_{i_k^{(j_f)}}(\mathbf{A}^{(k)})$.

- With the above $\boldsymbol{idx}$, we can find $\mathbf{Z}_{\mathcal{F}_n}^{(n)}$ and $\mathbf{X}_{(n)}^{\mathcal{F}_n}$ using efficient sampling without forming KRP, and using (8.2) to get corresponding probabilities.

Variance of Stochastic Gradient

Theorem 8.4

In the setting of Theorem 2.5, suppose that $\boldsymbol{p} \in \mathbb{R}^{J_n}$ *is any probability distribution proposed in previous, and* $\mathbf{R}_{(t)}^{(n)} = \mathbf{A}_{(t)}^{(n)}(\mathbf{Z}_{(t)}^{(n)})^{\intercal} - \mathbf{X}_{(n)}$. *Then*

$$\mathbb{E}_{\zeta_{(t)}}\left[\|\mathbf{G}_{(t)}^{(\xi_{(t)})} - \nabla_{\mathbf{A}^{(\xi_{(t)})}} f(\boldsymbol{\theta}_{(t)})\|_F^2 \mid \mathcal{B}_{(t)}, \xi_{(t)}\right]$$
$$= \frac{1}{|\mathcal{F}_n|}\sum_{j_f=1}^{J_n}\frac{1}{p_{j_f}}\|\mathbf{R}_{(t)}^{(\xi_{(t)})}(:, j_f)\|_2^2\|\mathbf{Z}_{(t)}^{(\xi_{(t)})}(j_f, :)\|_2^2 - \frac{1}{|\mathcal{F}_n|}\|\nabla_{\mathbf{A}^{(\xi_{(t)})}} f(\boldsymbol{\theta}_{(t)})\|_F^2.$$

Variance of Stochastic Gradient

Table: Variances for different probability distributions

Probability distributions	$\mathbb{E}_{\zeta_{(t)}}\left[\|\mathbf{G}_{(t)}^{(\xi_{(t)})} - \nabla_{\mathbf{A}^{(\xi_{(t)})}} f(\theta_{(t)})\|_F^2 \mid \mathcal{B}_{(t)}, \xi_{(t)}\right]$
Uniform[25]	$\frac{J_n}{\lvert\mathcal{F}_n\rvert}\sum_{j_f=1}^{J_n}\|\mathbf{R}_{(t)}^{(\xi_{(t)})}(:,j_f)\|_2^2\|\mathbf{Z}_{(t)}^{(\xi_{(t)})}(j_f,:)\|_2^2 - \frac{1}{\lvert\mathcal{F}_n\rvert}\|\nabla_{\mathbf{A}^{(\xi_{(t)})}} f(\boldsymbol{\theta}_{(t)})\|_F^2$
Leverage-based	$\frac{R^{N-1}}{\lvert\mathcal{F}_n\rvert}\sum_{j_f=1}^{J_n}\frac{\|\mathbf{R}_{(t)}^{(\xi_{(t)})}(:,j_f)\|_2^2\|\mathbf{Z}_{(t)}^{(\xi_{(t)})}(j_f,:)\|_2^2}{\prod_{k\neq\xi_{(t)}}\ell_{i_k^{(j_f)}}(\mathbf{A}^{(k)})} - \frac{1}{\lvert\mathcal{F}_n\rvert}\|\nabla_{\mathbf{A}^{(\xi_{(t)})}} f(\boldsymbol{\theta}_{(t)})\|_F^2$
Euclidean-based	$\frac{\prod_{k\neq\xi_{(t)}}\|\mathbf{A}^{(k)}\|_F^2}{\lvert\mathcal{F}_n\rvert}\sum_{j_f=1}^{J_n}\frac{\|\mathbf{R}_{(t)}^{(\xi_{(t)})}(:,j_f)\|_2^2\|\mathbf{Z}_{(t)}^{(\xi_{(t)})}(j_f,:)\|_2^2}{\prod_{k\neq\xi_{(t)}}\|\mathbf{A}^{(k)}(i_k^{(j_f)},:)\|_2^2} - \frac{1}{\lvert\mathcal{F}_n\rvert}\|\nabla_{\mathbf{A}^{(\xi_{(t)})}} f(\boldsymbol{\theta}_{(t)})\|_F^2$

[25] Ping Ma, Michael Mahoney, and Bin Yu. "A statistical Perspective on algorithmic leveraging". In: *J. Mach. Learn. Res.* 16.27 (2015), pp. 861–911. DOI: 10.1002/cem.1335.

Complexities

Table: Comparison of leading order computational complexities ($m > R$ and allow $|\mathcal{F}_n| \ll R$)

Method	Complexity without initialization
CP-ALS	$\mathcal{O}\left(\#it \cdot NRI^N\right)$
CPRAND-MIX	$\mathcal{O}\left(I^N \log(I^N) + \#it \cdot mNIR\right)$
CP-ALS-LEV	$\mathcal{O}\left(NIR^2 + \#it \cdot mNIR\right)$
`BrasCPD`	$\mathcal{O}\left(\#it \cdot IR\lvert\mathcal{F}_n\rvert\right)$
`AdaCPD`	$\mathcal{O}\left(\#it \cdot IR\lvert\mathcal{F}_n\rvert\right)$
`EBrawCPD`	$\mathcal{O}\left(IR + \#it \cdot IR\lvert\mathcal{F}_n\rvert\right)$
`LBrawCPD`	$\mathcal{O}\left(NIR^2 + \#it \cdot IR\lvert\mathcal{F}_n\rvert\right)$
`EAdawCPD`	$\mathcal{O}\left(IR + \#it \cdot IR\lvert\mathcal{F}_n\rvert\right)$
`LAdawCPD`	$\mathcal{O}\left(NIR^2 + \#it \cdot IR\lvert\mathcal{F}_n\rvert\right)$