Robust Tensor CUR: Rapid Low-Tucker-Rank Tensor Recovery with Sparse Corruptions

Longxiu Huang

Michigan State University
Department of Computational Mathematics, Science, and Engineering
Department of Mathematics

AWM research symposium 2023

Who?

HanQin Cai
UCF
Statistics

Zehan Chao
UCLA
Math

Deanna Needell
UCLA
Math

Outline

1. Motivation
2. Tensor Decompositions
3. Robust Decompositions

Motivation

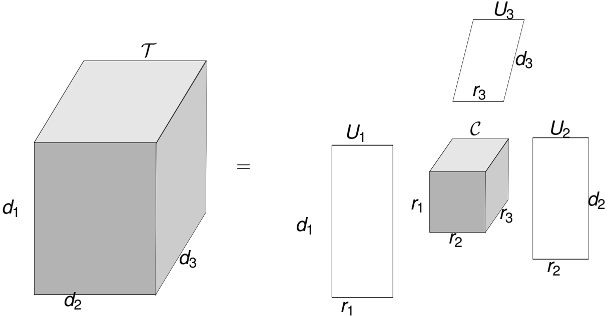
Figure: Background Separation Problem

Basic concepts for tensor

- Let $\mathcal{T} \in \mathbb{R}^{d_1 \times \cdots \times d_n}$. The multilinear rank of $\mathcal{T}$ is $\boldsymbol{r} = (r_1, \cdots, r_n)$, if $\text{rank}(\mathcal{T}_{(i)}) = r_i$ for $i = 1, \cdots, n$.
- Mode-i product: Let $\mathcal{C} \in \mathbb{R}^{r_1 \times \cdots \times r_n}$ and $U_i \in \mathbb{R}^{d_i \times r_i}$, then the multiplication between $\mathcal{C}$ on its i^{th} mode with U_i is denoted as $\mathcal{X} = \mathcal{C} \times_i U_i$ with

$$\mathcal{X}_{j_1, \cdots, j_{i-1}, k, j_{i+1}, \cdots, j_n} = \sum_{s=1}^{d_i} \mathcal{C}_{j_1, \cdots, j_{i-1}, s, j_{i+1}, \cdots, j_n} U_i(k, s).$$

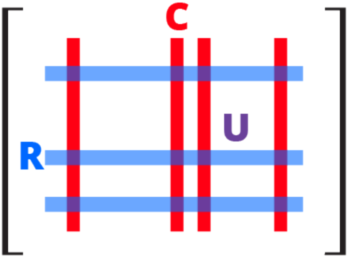
- Illustration for $\mathcal{T} = \mathcal{C} \times_1 U_1 \times_2 U_2 \times_3 U_3$



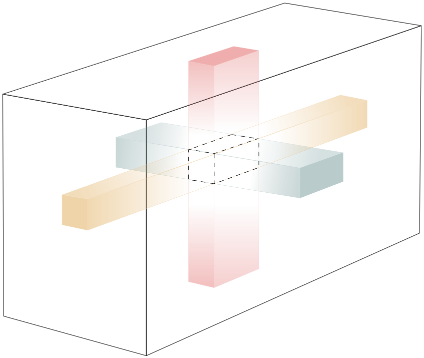
Tensor CUR Decompositions

Motivation

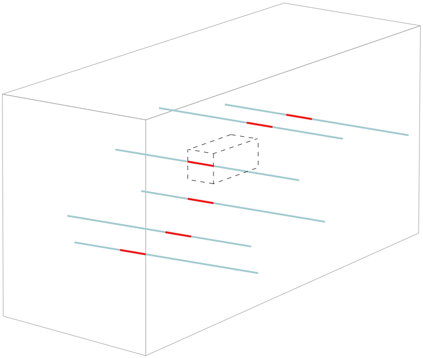
Let $A \in \mathbb{R}^{d_1 \times d_2}$ with CUR decomposition of $A = CU^\dagger R$. Then
$A = CU^\dagger R = CU^\dagger UU^\dagger R = U \times_1 (CU^\dagger) \times_2 (R^T(U^T)^\dagger)$.



Tensor CUR Decompositions[3]



Chidori CUR Decomposition[4]



Fiber CUR Decomposition

[3] Cai–Hamm–H–Needell, Mode-wise Tensor Decompositions: Multi-dimensional Generalizations of CUR Decompositions, JMLR, 2021

[4] Thanks to Dustin Mixon for this name from chidori joint game!

Characterizations of Tensor CUR Decompositions

Theorem (Cai–Hamm–H–Needell, 2021)

(Chidori CUR) Let $\mathcal{A} \in \mathbb{R}^{d_1 \times \cdots \times d_n}$ *with* $\operatorname{rank}(\mathcal{A}) = (r_1, \ldots, r_n)$*. Let* $I_i \subseteq [d_i]$*. Set* $\mathcal{R} = \mathcal{A}(I_1, \cdots, I_n)$*,* $C_i = \mathcal{A}_{(i)}(:, J_i := \otimes_{j \neq i} I_j)$ *and* $U_i = C_i(I_i, :)$*. Then the following are equivalent:*

1. $\operatorname{rank}(U_i) = r_i$,
2. $\mathcal{A} = \underbrace{\mathcal{R} \times_1 (C_1 U_1^\dagger) \times_2 \cdots \times_n (C_n U_n^\dagger)}_{CUR}$,
3. $\operatorname{rank}(\mathcal{R}) = (r_1, \cdots, r_n)$,
4. $\operatorname{rank}(\mathcal{A}_{(i)}(I_i, :)) = r_i$ *for all* $i \in [n]$.

Moreover, if the above statements hold, then $\mathcal{A} = \mathcal{A} \times_{i=1}^{n} (C_i C_i^\dagger)$*.*

Characterizations of Tensor CUR Decompositions

Theorem (Cai–Hamm–H–Needell, 2021)

(Fiber CUR): Let $\mathcal{A} \in \mathbb{R}^{d_1 \times \cdots \times d_n}$ *with* $\text{rank}(\mathcal{A}) = (r_1, \ldots, r_n)$. *Let* $I_i \subseteq [d_i]$ *and* $J_i \subseteq [\prod_{j \neq i} d_j]$. *Set* $\mathcal{R} = \mathcal{A}(I_1, \cdots, I_n)$, $C_i = \mathcal{A}_{(i)}(:, J_i)$ *and* $U_i = C_i(I_i, :)$. *Then the following statements are equivalent*

1. $\text{rank}(U_i) = r_i$,
2. $\underbrace{\mathcal{A} = \mathcal{R} \times_1 (C_1 U_1^\dagger) \times_2 \cdots \times_n (C_n U_n^\dagger)}_{CUR}$,
3. $\text{rank}(C_i) = r_i$ *for all* $i \in [n]$ *and* $\text{rank}(\mathcal{R}) = (r_1, \cdots, r_n)$,
4. $\text{rank}(C_i) = r_i$ *and* $\text{rank}(\mathcal{A}_{(i)}(I_i, :)) = r_i$ *for all* $i \in [n]$.

Robust Decompositions

Observe: $\mathcal{D} = \mathcal{L} + \mathcal{S} \in \mathbb{R}^{d \times \cdots \times d}$, where $\mathcal{L}$ is low rank data, and $\mathcal{S}$ is sparse (but arbitrary magnitude) noise

Return: Estimate of $\mathcal{L}$

Robust Decompositions

Observe: $\mathcal{D} = \mathcal{L} + \mathcal{S} \in \mathbb{R}^{d\times\cdots\times d}$, where $\mathcal{L}$ is low rank data, and $\mathcal{S}$ is sparse (but arbitrary magnitude) noise

Return: Estimate of $\mathcal{L}$

Assumptions:

- (Incoherence) $\mathcal{L} = \mathcal{C} \times_{i=1}^{n} V_i$ (HOSVD):

$$\mu_i(\mathcal{L}) := \max_{j_i} \frac{d}{r} \left\| V_i^T e_{j_i} \right\|_2^2 \leq \mu.$$

Robust Decompositions

Observe: $\mathcal{D} = \mathcal{L} + \mathcal{S} \in \mathbb{R}^{d \times \cdots \times d}$, where $\mathcal{L}$ is low rank data, and $\mathcal{S}$ is sparse (but arbitrary magnitude) noise

Return: Estimate of $\mathcal{L}$

Assumptions:

- (Incoherence) $\mathcal{L} = \mathcal{C} \times_{i=1}^{n} V_i$ (HOSVD):

$$\mu_i(\mathcal{L}) := \max_{j_i} \frac{d}{r} \left\| V_i^T e_{j_i} \right\|_2^2 \leq \mu.$$

- (Sparsity)

$$\max_{k_i} \left\| \mathcal{S} \times_i e_{k_i}^{(i)} \right\|_0 \leq \alpha d^{n-1}.$$

Robust Decompositions

When $n = 2$, it is also termed Robust PCA.

- Using CUR within iterative Robust PCA (Cai-Hamm-H-Li-Wang, 2021[1])

[1] Cai–Hamm–H–Li–Wang, Rapid Robust Principal Component Analysis: CUR Accelerated Inexact Low Rank Estimation, IEEE-SPL, 2021

Iterated Robust CUR (IRCUR) – (SVD→CUR)

Algorithm 1 **I**terated **R**obust **CUR** for RPCA (IRCUR)

Input: $\boldsymbol{D}$: observed corrupted data matrix; r: rank; ε: target precision level; ζ_0: initial thresholding value; γ: thresholding decay parameter; $|\mathcal{I}|, |\mathcal{J}|$: sampling number of rows and columns.
Uniformly sample row indices $\mathcal{I}$ and column indices $\mathcal{J}$.
$\boldsymbol{L}_0 = \boldsymbol{0}, \quad \boldsymbol{S}_0 = \boldsymbol{0}, \quad k = 0$
while $e_k > \varepsilon$ (e_k is defined as in (6)) **do**
  (Optional) Resample indices $\mathcal{I}$ and $\mathcal{J}$
  // Phase I: updating sparse component
  $\zeta_{k+1} = \gamma^k \zeta_0$
  $[\boldsymbol{S}_{k+1}]_{:,\mathcal{J}} = \mathcal{T}_{\zeta_{k+1}}[\boldsymbol{D} - \boldsymbol{L}_k]_{:,\mathcal{J}}$
  $[\boldsymbol{S}_{k+1}]_{\mathcal{I},:} = \mathcal{T}_{\zeta_{k+1}}[\boldsymbol{D} - \boldsymbol{L}_k]_{\mathcal{I},:}$
  // Phase II: updating low rank component
  $\boldsymbol{C}_{k+1} = [\boldsymbol{D} - \boldsymbol{S}_{k+1}]_{:,\mathcal{J}}$
  $\boldsymbol{U}_{k+1} = \mathcal{H}_r([\boldsymbol{D} - \boldsymbol{S}_{k+1}]_{\mathcal{I},\mathcal{J}})$
  $\boldsymbol{R}_{k+1} = [\boldsymbol{D} - \boldsymbol{S}_{k+1}]_{\mathcal{I},:}$
  $\boldsymbol{L}_{k+1} = \boldsymbol{C}_{k+1}\boldsymbol{U}_{k+1}^{\dagger}\boldsymbol{R}_{k+1}$ // Do not compute this step
  $k = k + 1$
end while
Output: $\boldsymbol{C}_k, \boldsymbol{U}_k, \boldsymbol{R}_k$: CUR decomposition of $\boldsymbol{L}$.

Remark

Reduce RPCA's complexity from $\mathcal{O}(rd^2)$ flops to $\mathcal{O}(r^2 d \log^2 d)$ flops per iteration.

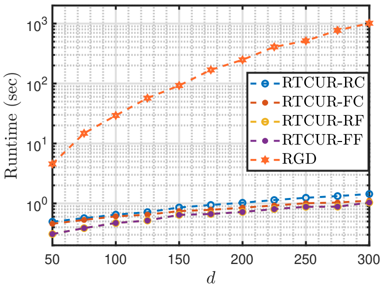
Robust Tensor Decomposition: generalized IRCUR[2]

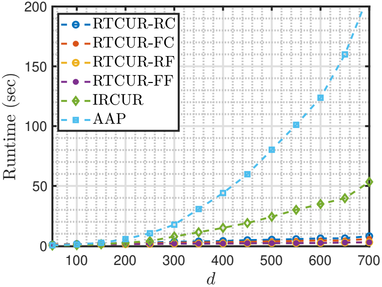
Algorithm 1 Robust Tensor CUR (RTCUR)

Input: $\mathcal{X} = \mathcal{L}_\star + \mathcal{S}_\star \in \mathbb{R}^{d_1 \times \cdots \times d_n}$: observed tensor; $(r_1, \cdots, r_n)$: underlying multilinear rank of $\mathcal{L}_\star$; ε: targeted precision; $\zeta^{(0)}, \gamma$: thresholding parameters; $\{|I_i|\}_{i=1}^n, \{|J_i|\}_{i=1}^n$: cardinalities for sample indices.
Initialization: $\mathcal{L}^{(0)} = \mathbf{0}, \mathcal{S}^{(0)} = \mathbf{0}, k = 0$
Uniformly sample the indices $\{I_i\}_{i=1}^n, \{J_i\}_{i=1}^n$
while $e^{(k)} > \varepsilon$ **do** // $e^{(k)}$ is defined in (14)
  (Optional) Resample the indices $\{I_i\}_{i=1}^n, \{J_i\}_{i=1}^n$
  // Step (I): Updating $\mathcal{S}$
  $\zeta^{(k+1)} = \gamma \cdot \zeta^{(k)}$
  $\mathcal{S}^{(k+1)} = \mathrm{HT}_{\zeta^{(k+1)}}(\mathcal{X} - \mathcal{L}^{(k)})$
  // Step (II): Updating $\mathcal{L}$
  $\mathcal{R}^{(k+1)} = (\mathcal{X} - \mathcal{S}^{(k+1)})(I_1, \cdots, I_n)$
  for $i = 1, \cdots, n$ **do**
    $C_i^{(k+1)} = (\mathcal{X} - \mathcal{S}^{(k+1)})_{(i)}(:, J_i)$
    $U_i^{(k+1)} = \mathrm{SVD}_{r_i}(C_i^{(k+1)}(I_i, :))$
  end for
  $\mathcal{L}^{(k+1)} = \mathcal{R}^{(k+1)} \times_{i=1}^n C_i^{(k+1)} \left(U_i^{(k+1)}\right)^\dagger$
  $k = k + 1$
end while
Output: $\mathcal{R}^{(k)}, C_i^{(k)}, U_i^{(k)}$ for $i = 1, \cdots, n$: the estimates of the tensor Fiber CUR decomposition of $\mathcal{L}_\star$.

[2] Cai–Chao–H–Needell, Robust Tensor CUR Decompositions: Rapid Low-Tucker-Rank Tensor Recovery with Sparse Corruption, SIIMS, 2023

Runtime vs. Dimension





Runtime vs. Relative Error

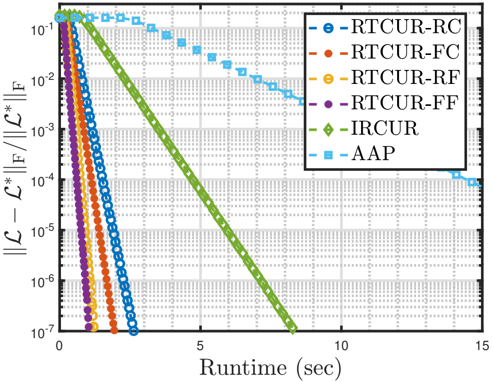
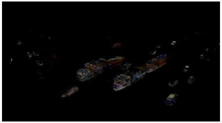


Figure: Runtime vs. relative error comparison: 3-mode tensor with $d = 500$ and multilinear rank $(3, 3, 3)$.

Original

RTCUR-F

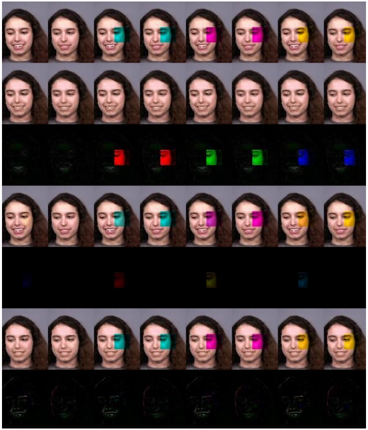
ADMM



AAP

Runtime (sec)	RTCUR-F	ADMM	AAP
	6.15	1099.3	97.85

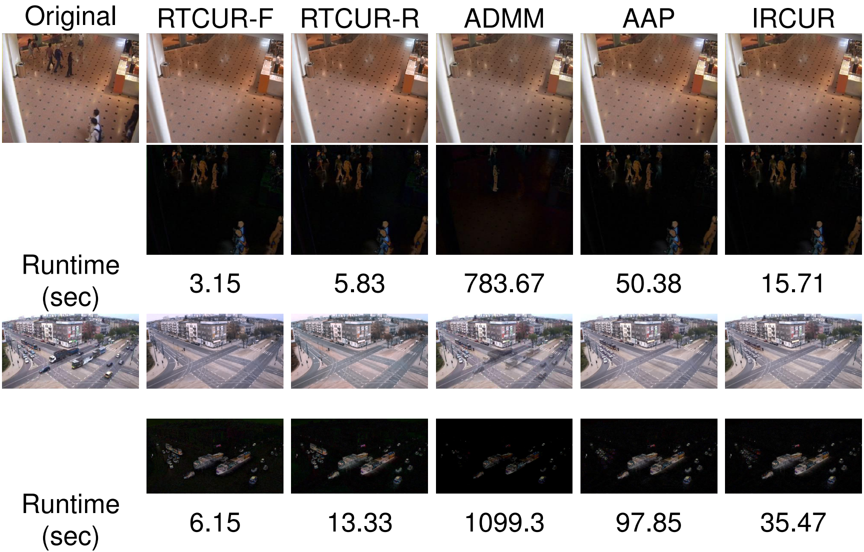
Face Modeling



1st row: the corrupted faces;
2nd and 3rd rows: output from **RTCUR-F**;
4th and 5th rows: output from from **ADMM**;
6th and 7th rows: output from **AAP**;

Thanks for listening!

Questions?


Original
RTCUR-F
RTCUR-R
ADMM
AAP
IRCUR
Runtime (sec)
3.15
5.83
783.67
50.38
15.71
Runtime (sec)
6.15
13.33
1099.3
97.85
35.47