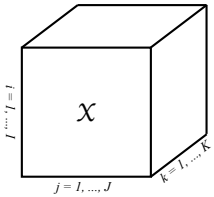
Computing Sparse Tensor Decompositions Using Dimension Trees

Introduction and Motivation

What is a tensor?

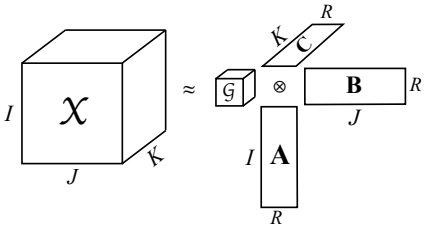
- A vector is a 1-dimensional tensor.
- A matrix is a 2-dimensional tensor.
- A tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_d}$ has d dimensions.



$\mathcal{X} \in \mathbb{R}^{I \times J \times K}$

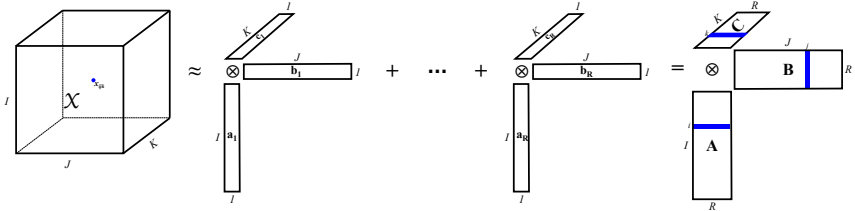
We are interested in the case when $\mathcal{X}$ is **sparse** and of **low rank**.

Tensor Decompositions



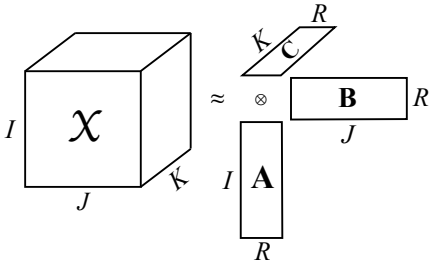
- Tucker decomposition
  - provides a **rank-$(R_1, \ldots, R_N)$ approximation** of a tensor.
  - consists of **a core tensor** $\mathcal{G} \in \mathbb{R}^{R_1 \times \cdots \times R_N}$ and N **factor matrices** having $R_1, \ldots, R_N$ columns.
- CP decomposition
  - provides a **rank-R approximation** of a tensor.
  - Core tensor disappears (it is diagonal).

Applications



- TOPHITS model for performing web-link analysis (Kolda & Bader,'05)
- Analysis of user-group structures using chatroom data (Acar et al., '05)
- Link prediction in temporal graphs (Dunlavy et al., '11)
- Context-aware recommender systems (Shi et al., 12)
- Data compression (Kolda et al., '16)
- Signal processing, computer vision, chemometrics, etc.

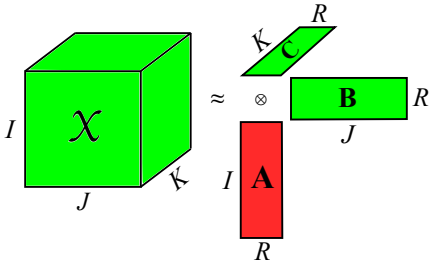
Computing CP Decomposition



Algorithm CP-ALS for 3rd order tensors

Input: $\mathcal{X}$: tensor
R: The rank of approximation
Output: CP decomposition $[\![\lambda; \mathbf{A}, \mathbf{B}, \mathbf{C}]\!]$
repeat
$\hat{\mathbf{A}} \leftarrow \mathbf{X}_{(1)}(\mathbf{C} \odot \mathbf{B})$
$\mathbf{A} \leftarrow \hat{\mathbf{A}}(\mathbf{B}^T\mathbf{B} * \mathbf{C}^T\mathbf{C})^\dagger$
$\hat{\mathbf{B}} \leftarrow \mathbf{X}_{(2)}(\mathbf{A} \odot \mathbf{B})$
$\mathbf{B} \leftarrow \hat{\mathbf{B}}(\mathbf{A}^T\mathbf{A} * \mathbf{C}^T\mathbf{C})^\dagger$
$\hat{\mathbf{C}} \leftarrow \mathbf{X}_{(1)}(\mathbf{B} \odot \mathbf{C})$
$\mathbf{C} \leftarrow \hat{\mathbf{C}}(\mathbf{A}^T\mathbf{A} * \mathbf{B}^T\mathbf{B})^\dagger$
until no improvement or max iterations achieved

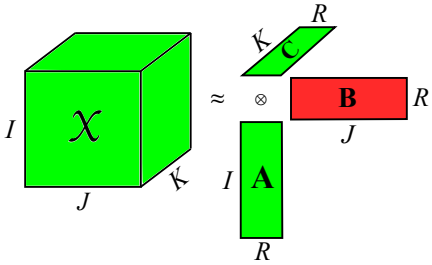
Computing CP Decomposition



Algorithm CP-ALS for 3rd order tensors

Input: $\mathcal{X}$: tensor
R: The rank of approximation
Output: CP decomposition $[\![\lambda; \mathbf{A}, \mathbf{B}, \mathbf{C}]\!]$
repeat
$\hat{\mathbf{A}} \leftarrow \mathbf{X}_{(1)}(\mathbf{C} \odot \mathbf{B})$
$\mathbf{A} \leftarrow \hat{\mathbf{A}}(\mathbf{B}^T\mathbf{B} * \mathbf{C}^T\mathbf{C})^\dagger$
$\hat{\mathbf{B}} \leftarrow \mathbf{X}_{(2)}(\mathbf{A} \odot \mathbf{B})$
$\mathbf{B} \leftarrow \hat{\mathbf{B}}(\mathbf{A}^T\mathbf{A} * \mathbf{C}^T\mathbf{C})^\dagger$
$\hat{\mathbf{C}} \leftarrow \mathbf{X}_{(1)}(\mathbf{B} \odot \mathbf{C})$
$\mathbf{C} \leftarrow \hat{\mathbf{C}}(\mathbf{A}^T\mathbf{A} * \mathbf{B}^T\mathbf{B})^\dagger$
until no improvement or max iterations achieved

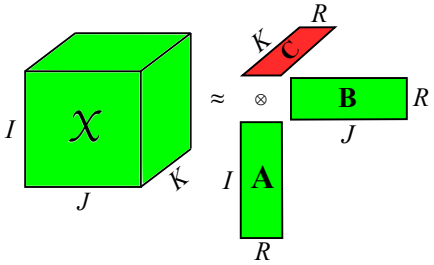
Computing CP Decomposition



Algorithm CP-ALS for 3rd order tensors

Input: $\mathcal{X}$: tensor
R: The rank of approximation
Output: CP decomposition $[\![\lambda; \mathbf{A}, \mathbf{B}, \mathbf{C}]\!]$
repeat
$\hat{\mathbf{A}} \leftarrow \mathbf{X}_{(1)}(\mathbf{C} \odot \mathbf{B})$
$\mathbf{A} \leftarrow \hat{\mathbf{A}}(\mathbf{B}^T\mathbf{B} * \mathbf{C}^T\mathbf{C})^\dagger$
$\hat{\mathbf{B}} \leftarrow \mathbf{X}_{(2)}(\mathbf{A} \odot \mathbf{B})$
$\mathbf{B} \leftarrow \hat{\mathbf{B}}(\mathbf{A}^T\mathbf{A} * \mathbf{C}^T\mathbf{C})^\dagger$
$\hat{\mathbf{C}} \leftarrow \mathbf{X}_{(1)}(\mathbf{B} \odot \mathbf{C})$
$\mathbf{C} \leftarrow \hat{\mathbf{C}}(\mathbf{A}^T\mathbf{A} * \mathbf{B}^T\mathbf{B})^\dagger$
until no improvement or max iterations achieved

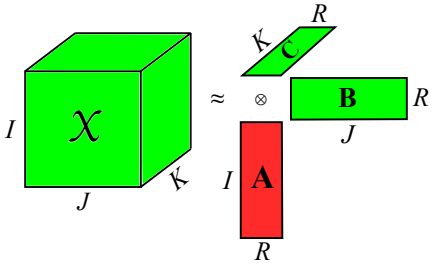
Computing CP Decomposition



Algorithm CP-ALS for 3rd order tensors

Input: $\mathcal{X}$: tensor
R: The rank of approximation
Output: CP decomposition $[\![\lambda; \mathbf{A}, \mathbf{B}, \mathbf{C}]\!]$
repeat
$\hat{\mathbf{A}} \leftarrow \mathbf{X}_{(1)}(\mathbf{C} \odot \mathbf{B})$
$\mathbf{A} \leftarrow \hat{\mathbf{A}}(\mathbf{B}^T\mathbf{B} * \mathbf{C}^T\mathbf{C})^\dagger$
$\hat{\mathbf{B}} \leftarrow \mathbf{X}_{(2)}(\mathbf{A} \odot \mathbf{B})$
$\mathbf{B} \leftarrow \hat{\mathbf{B}}(\mathbf{A}^T\mathbf{A} * \mathbf{C}^T\mathbf{C})^\dagger$
$\hat{\mathbf{C}} \leftarrow \mathbf{X}_{(1)}(\mathbf{B} \odot \mathbf{C})$
$\mathbf{C} \leftarrow \hat{\mathbf{C}}(\mathbf{A}^T\mathbf{A} * \mathbf{B}^T\mathbf{B})^\dagger$
until no improvement or max iterations achieved

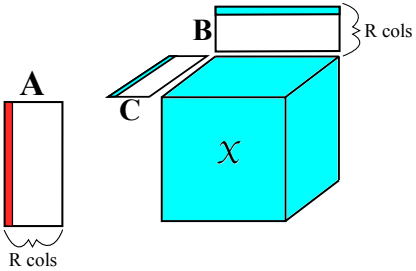
Computing CP Decomposition



Algorithm CP-ALS for 3rd order tensors

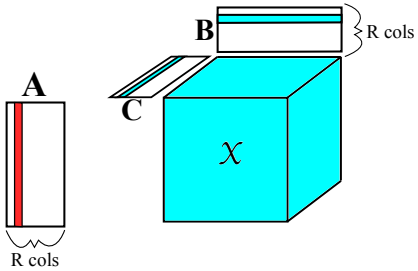
Input: $\mathcal{X}$: tensor
R: The rank of approximation
Output: CP decomposition $[\![\lambda; \mathbf{A}, \mathbf{B}, \mathbf{C}]\!]$
repeat
$\hat{\mathbf{A}} \leftarrow \mathbf{X}_{(1)}(\mathbf{C} \odot \mathbf{B})$
$\mathbf{A} \leftarrow \hat{\mathbf{A}}(\mathbf{B}^T\mathbf{B} * \mathbf{C}^T\mathbf{C})^\dagger$
$\hat{\mathbf{B}} \leftarrow \mathbf{X}_{(2)}(\mathbf{A} \odot \mathbf{B})$
$\mathbf{B} \leftarrow \hat{\mathbf{B}}(\mathbf{A}^T\mathbf{A} * \mathbf{C}^T\mathbf{C})^\dagger$
$\hat{\mathbf{C}} \leftarrow \mathbf{X}_{(1)}(\mathbf{B} \odot \mathbf{C})$
$\mathbf{C} \leftarrow \hat{\mathbf{C}}(\mathbf{A}^T\mathbf{A} * \mathbf{B}^T\mathbf{B})^\dagger$
until no improvement or max iterations achieved

Tensor-Times-Vector Multiplication (TTV)



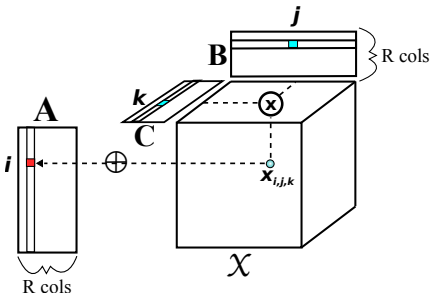
- $\mathbf{A}(:, r) \leftarrow \mathcal{X} \times_2 \mathbf{B}(:, r) \times_3 \mathbf{C}(:, r)$ for $r = 1, \ldots, R$.
- Updating $\mathbf{A}$ takes $R(N-1)$ TTVs.

Tensor-Times-Vector Multiplication (TTV)



- $\mathbf{A}(:, r) \leftarrow \mathcal{X} \times_2 \mathbf{B}(:, r) \times_3 \mathbf{C}(:, r)$ for $r = 1, \ldots, R$.
- Updating $\mathbf{A}$ takes $R(N-1)$ TTVs.

Tensor-Times-Vector Multiplication (TTV)



- Reminiscent of SpMV
- Each nonzero of $\mathcal{X}$ incurs multiplication of $N-1$ elements and 1 addition

Sparse Tensor Storage - Coordinate Format

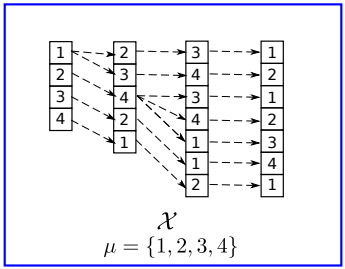
1	2	3	1
1	3	4	2
2	4	3	1
2	4	4	2
2	4	1	3
3	2	1	4
4	1	2	1

$\mathcal{X}$

$\mu = \{1, 2, 3, 4\}$

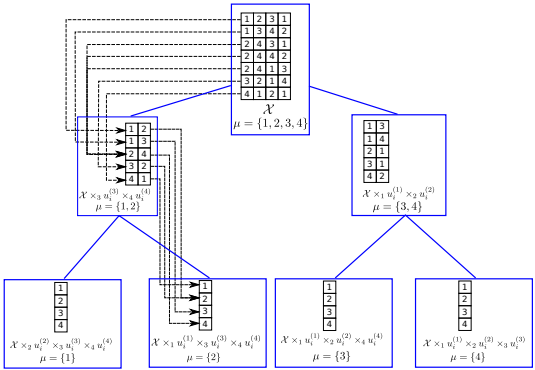
- N index arrays, N nnz($\mathcal{X}$) memory
- $R(N-1)$ TTVs per dimension, **RN(N - 1)** TTVs per iteration.
- Cost per TTV: nnz($\mathcal{X}$)

Sparse Tensor Storage - Compressed Sparse Fiber (CSF, Smith and Karypis)



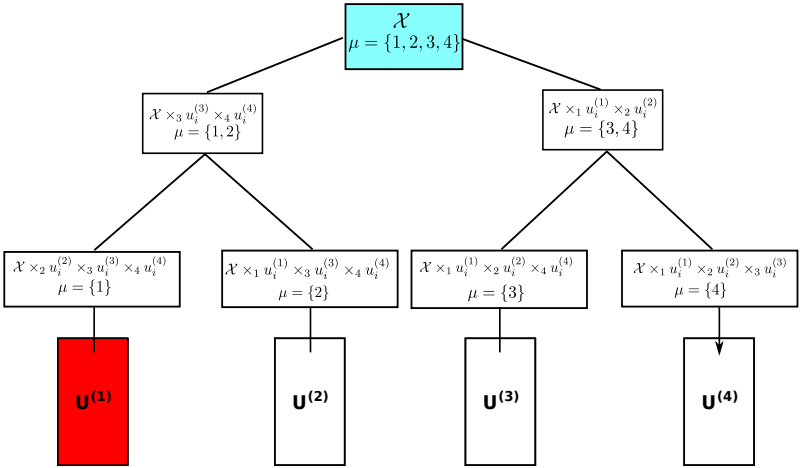
- Extension of compressed sparse row/column (CSR/CSC).
- N index and pointer arrays.
- $R(N-1)$ TTVs per dimension, **RN(N - 1)** TTVs per iteration.
- Cost per TTV: **at most** $\text{nnz}(\mathcal{X})$.
  - Index compression possible after multiplying in a dimension.

Sparse Tensor Storage - Dimension Tree (DT)

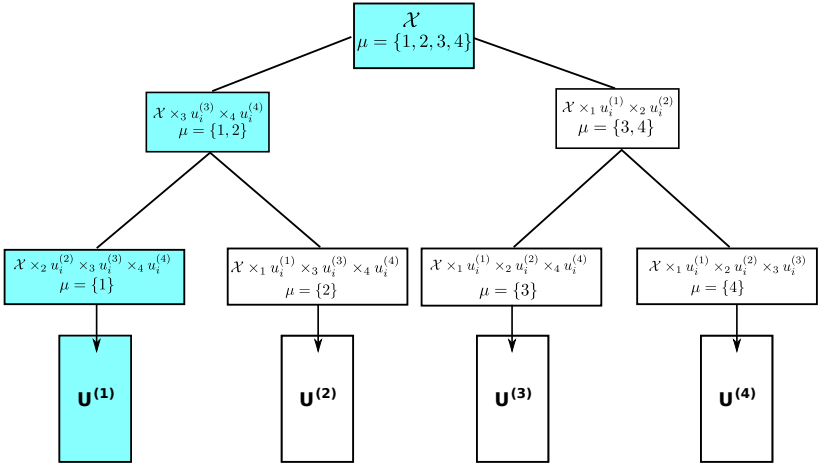


- Hierarchical storage scheme.
- Index arrays at each level **partitions** the dimension set.
- $N \log N$ index arrays.
- Index compression still possible.

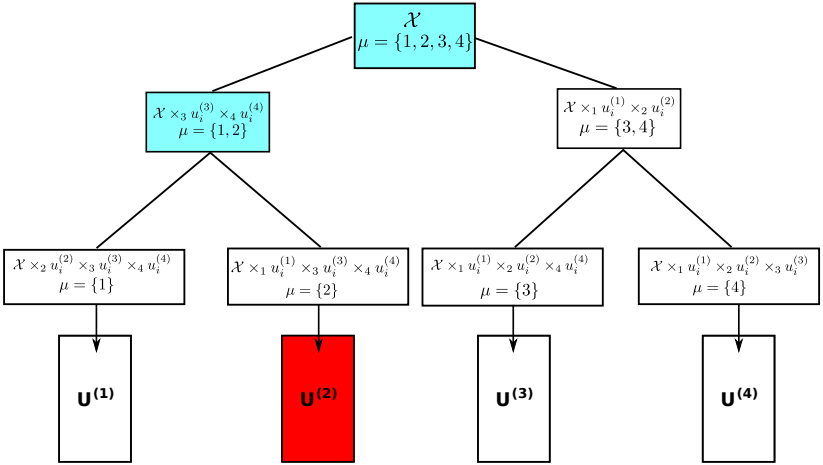
TTV Using Dimension Tree Storage



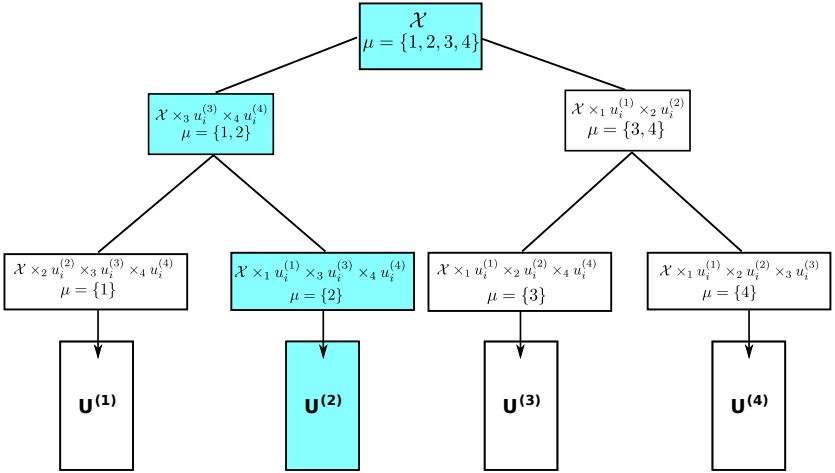
TTV Using Dimension Tree Storage



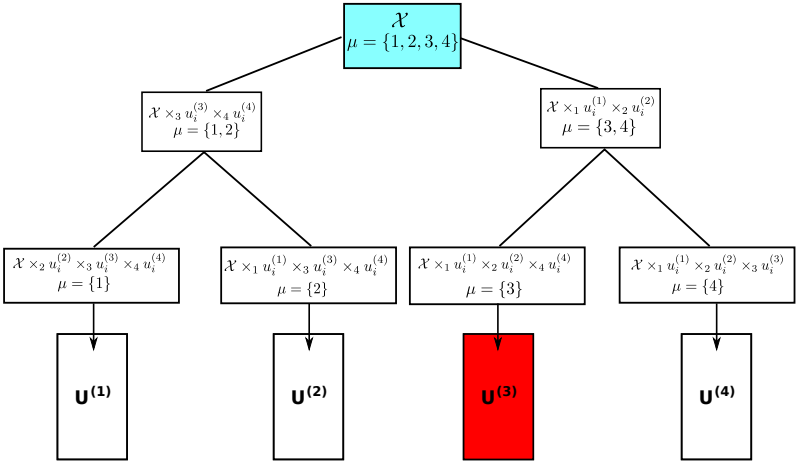
TTV Using Dimension Tree Storage



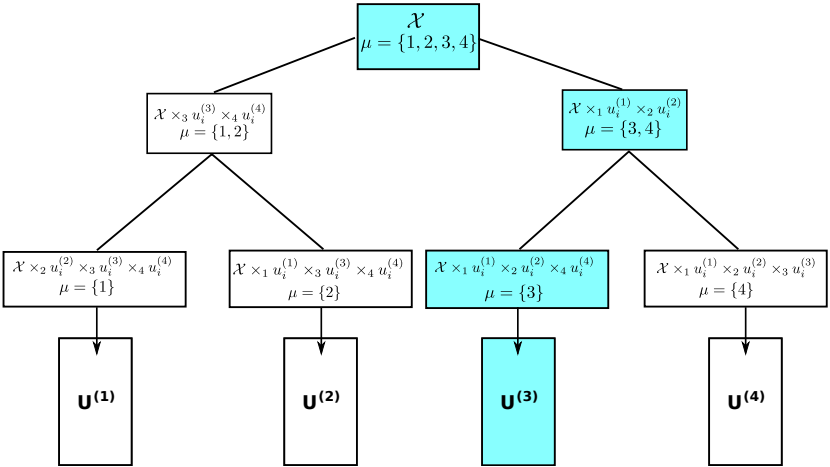
TTV Using Dimension Tree Storage



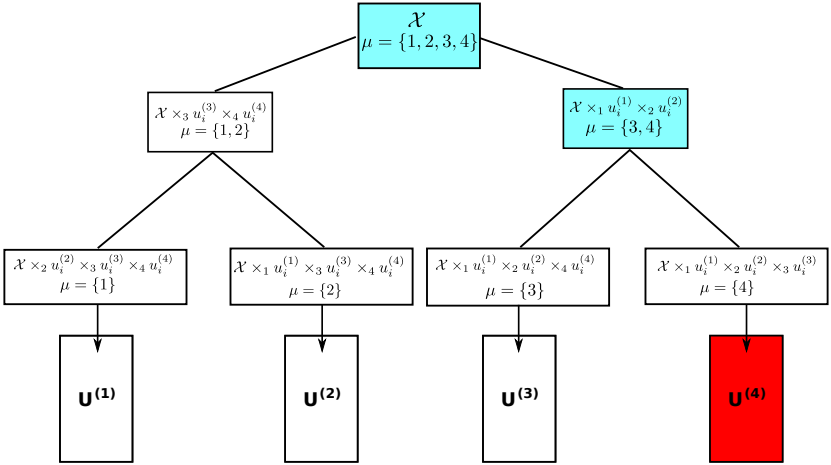
TTV Using Dimension Tree Storage



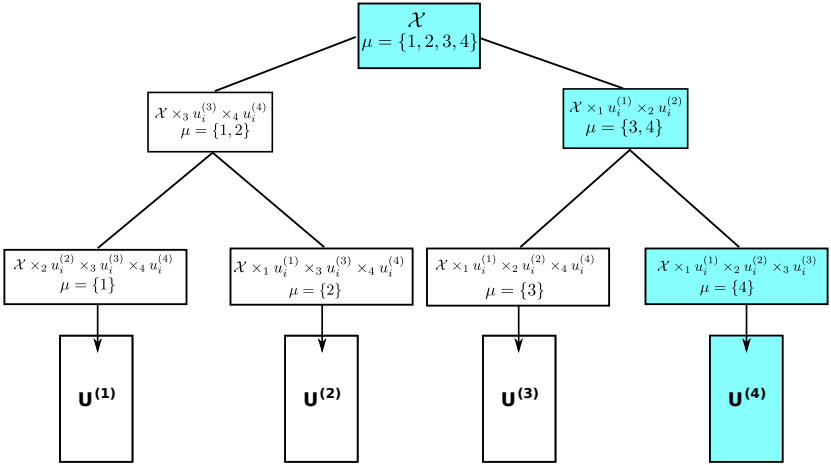
TTV Using Dimension Tree Storage



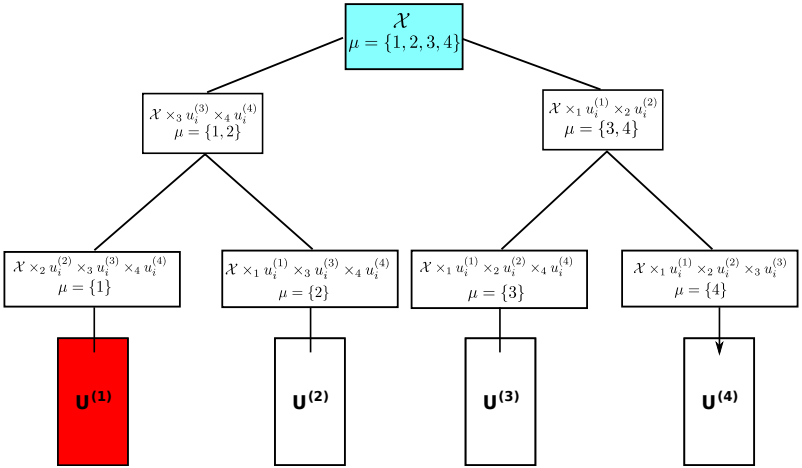
TTV Using Dimension Tree Storage



TTV Using Dimension Tree Storage



TTV Using Dimension Tree Storage



Sparse CP Decomposition using Dimension Trees

Theorem

Let $\mathcal{X}$ be an N-dimensional tensor. The total number of TTVs at each iteration of CP-ALS is at most $RN\lceil \log N \rceil$ using dimension tree storage.

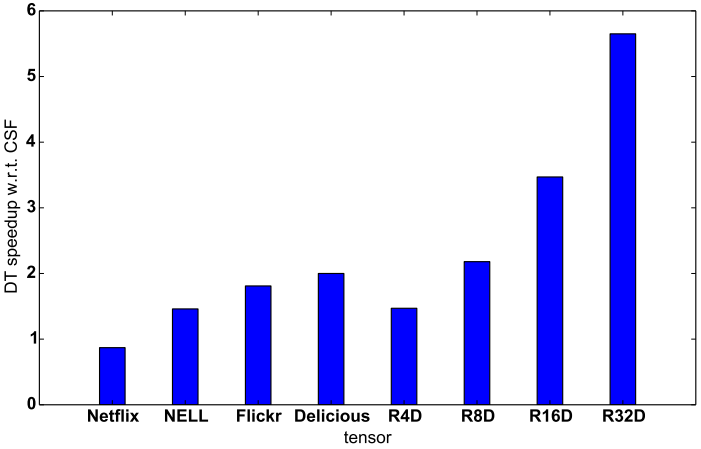
- In the traditional formats, each iteration takes $RN(N-1)$ TTVs.

Theorem

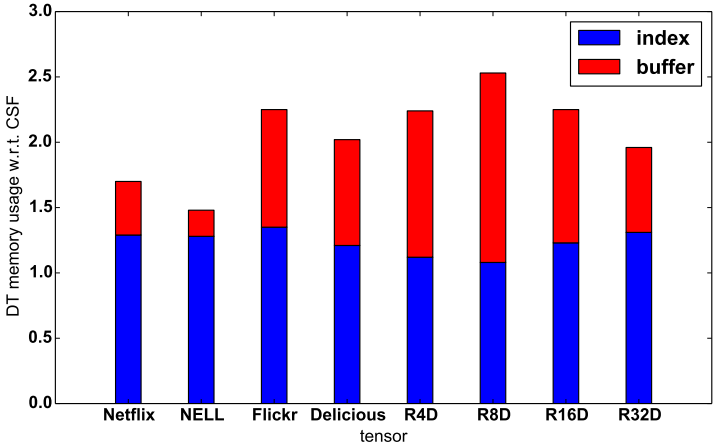
For an N-dimensionaltensor $\mathcal{X}$, the total number of intermediate TTV results is at most $R\lceil \log N \rceil$ at any instant of CP-ALS using dimension tree storage.

- At any instant, only the nodes on a path from a leaf to the root are active.

Experiments - Runtime



Experiments - Memory Usage



Conclusion

- We introduce a new sparse tensor data structure and associated computational scheme.
- DT format provides up to **5.65x faster TTVs** using higher dimensional tensors, at the expense of **1.5x-2.5x more memory usage**.
- Possibility to execute other tensor operations such as tensor-times-matrix multiply (TTM).

Contact

Oguz Kaya
INRIA and LIP, ENS Lyon, France
oguz.kaya@outlook.com
www.oguzkaya.com

Joint work with (advisors): Bora Uçar and Yves Robert