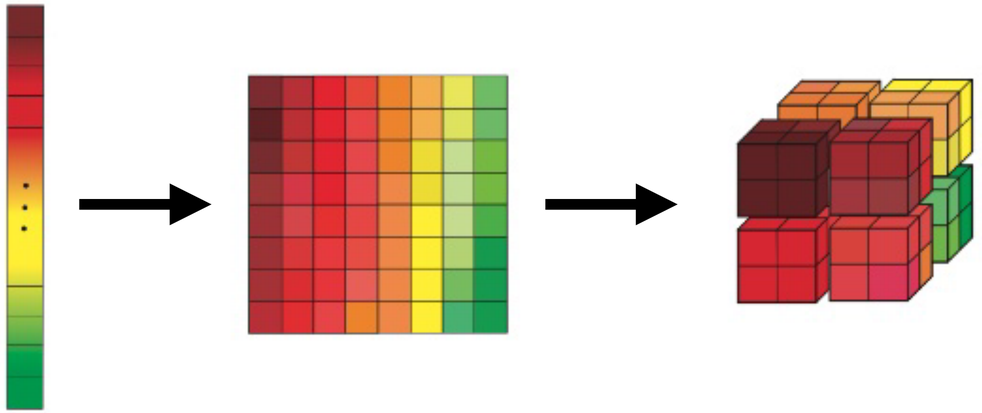


HiCOO: Hierarchical Storage of Sparse Tensors

Jiajia Li [1,2], Jimeng Sun [1], Richard Vuduc [1]

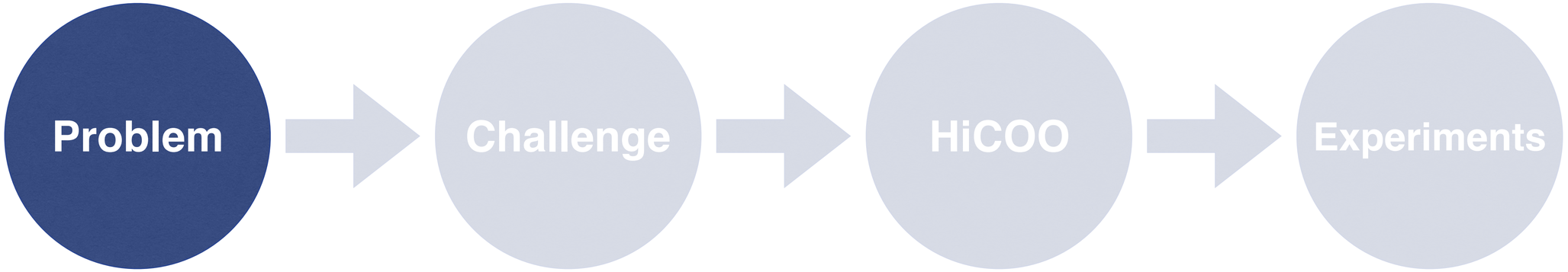
[1] Georgia Institute of Technology

[2] Pacific Northwest National Laboratory

November 13, 2018 @ SC18

Figure sources: “A brief survey of tensors” by Berton Earnshaw and NVIDIA Tensor Cores

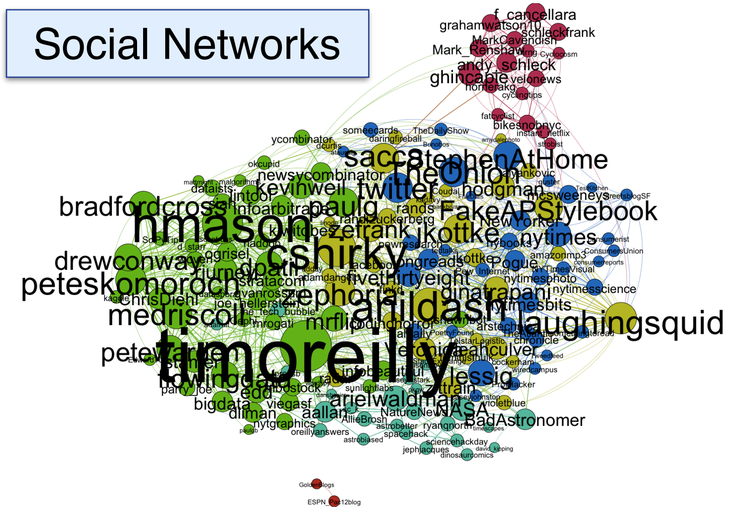
Outline



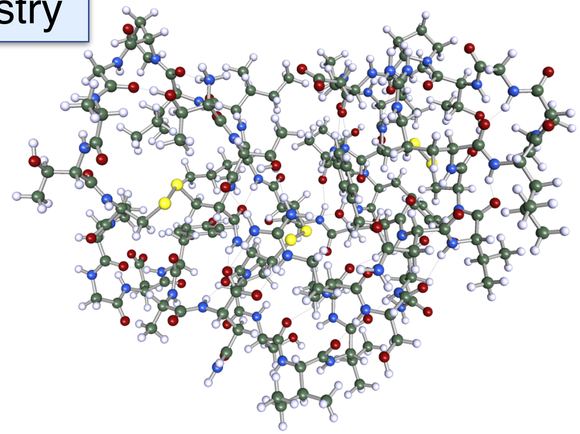
- Tensors
- Applications
- Tensor decomposition

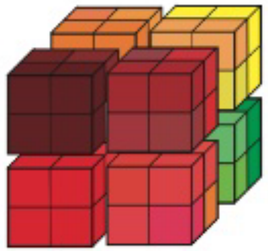
Tensors

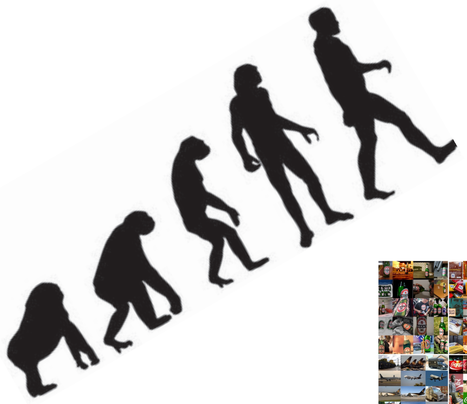
Social Networks

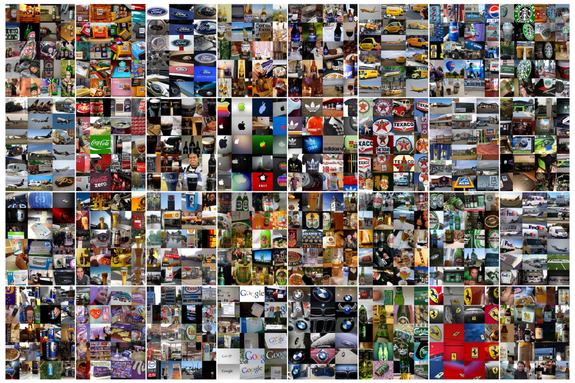


Quantum Chemistry



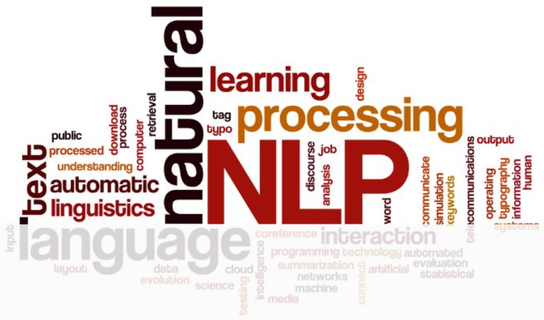


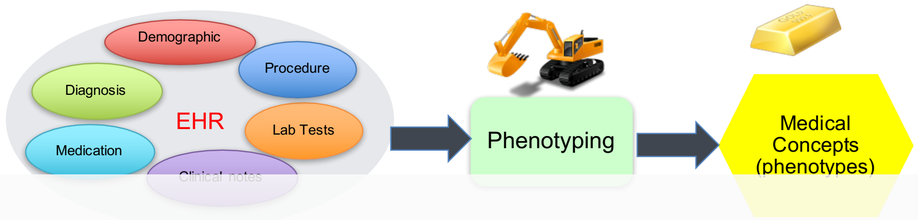




Deep Learning

Applications of Tensor Methods



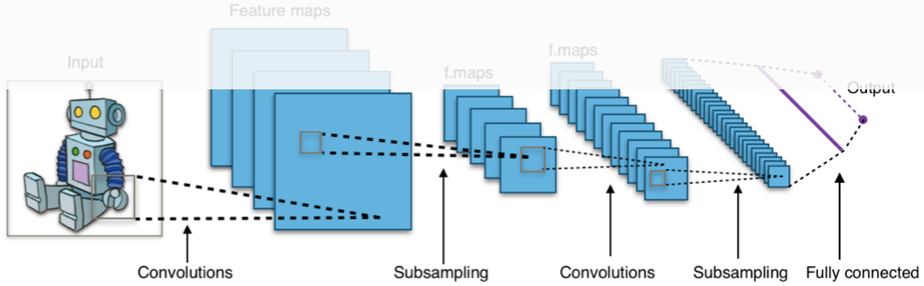


- Data analytics and compression

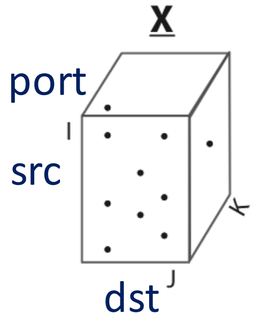
  Natural language processing, healthcare analytics, social network analytics, brain signal processing, and deep learning

- Scientific computing

  Quantum chemistry, computational physics, quantum mechanics



Tensor Decomposition for Anomaly Detection



Network data

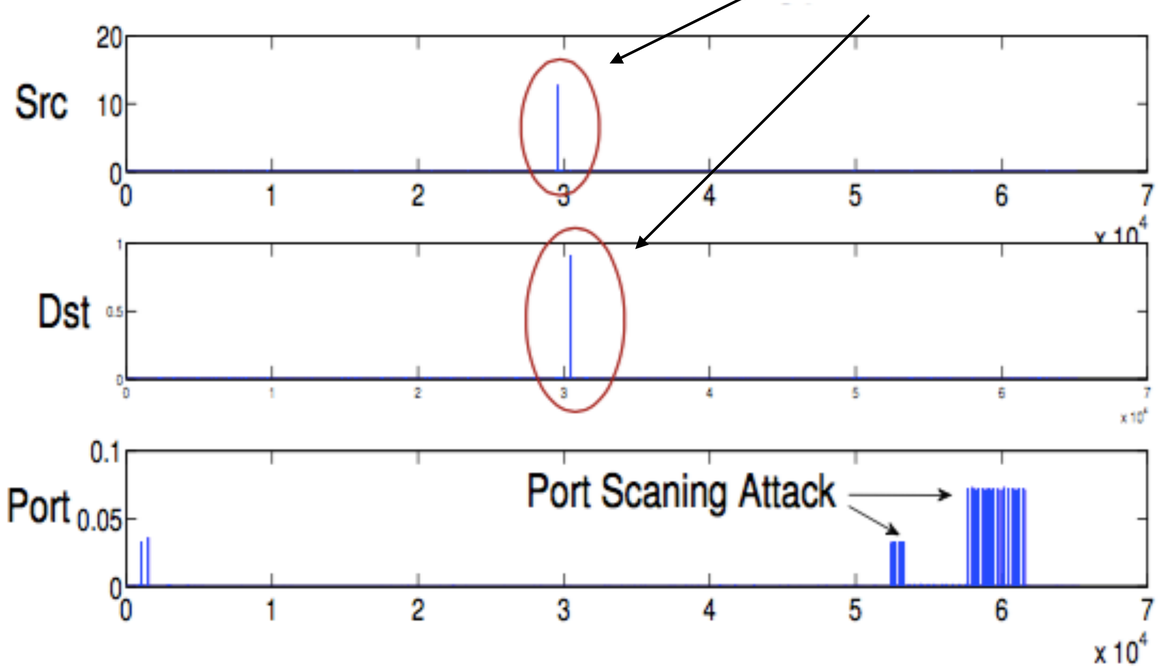
Tensor decomposition

A

B

C

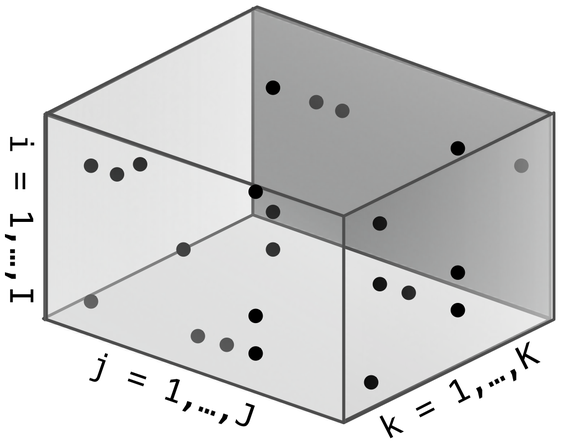
Anomalies



Source: Papalexakis et al. ECML-PKDD 2012

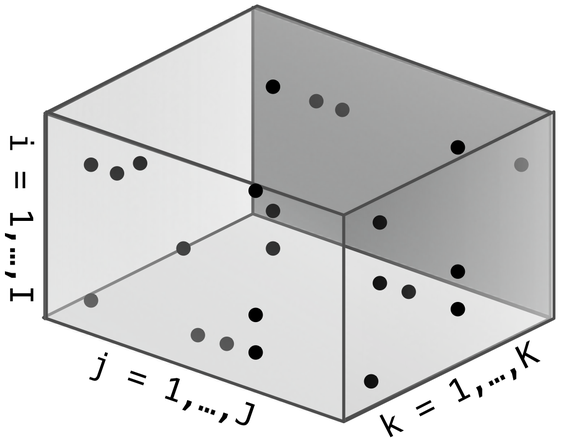
Tensors & Decompositions

- Tensors, multi-way arrays, provide a natural way to represent multi-relational data.
  - Special cases: matrices, vectors
  - Tensor mode or order: tensor dimension.
  - Data tensors in applications are usually SPARSE, meaning consisting mostly of zero entries.



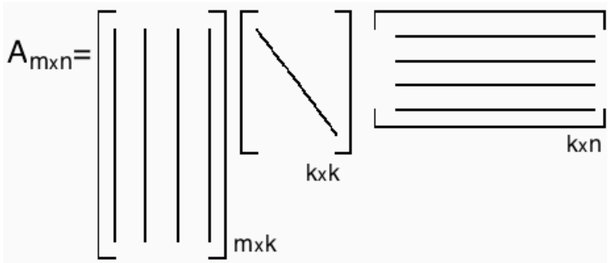
Tensors & Decompositions

- Tensors, multi-way arrays, provide a natural way to represent multi-relational data.
  - Special cases: matrices, vectors
  - Tensor mode or order: tensor dimension.
  - Data tensors in applications are usually SPARSE, meaning consisting mostly of zero entries.

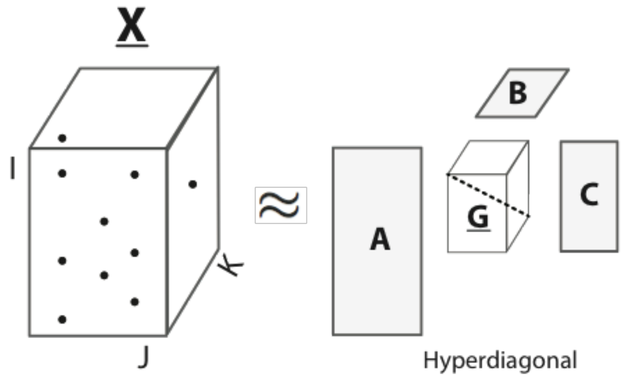


- Tensor decompositions: the natural generalization of matrix decompositions in data mining to tensors.

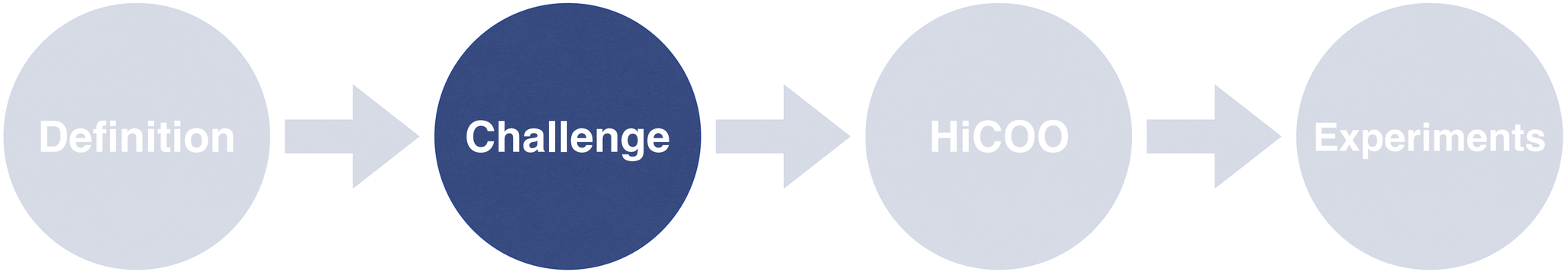
Singular Value Decomposition (SVD)



CANDECOMP/PARAFAC Decomposition (CPD)



Outline



Mode Orientation

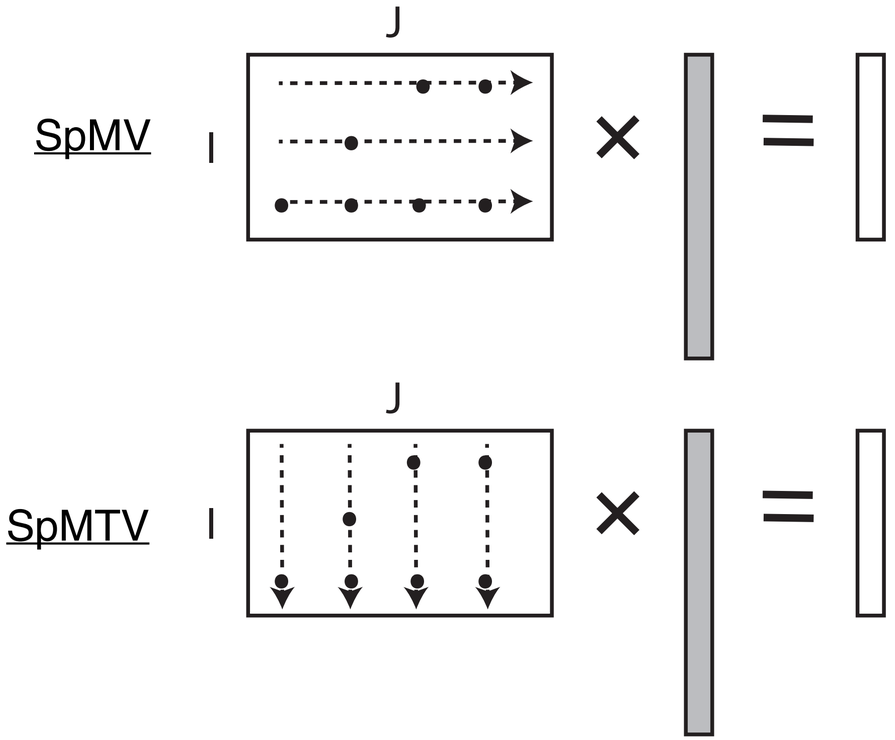
- Each mode has similar program behavior.

  Better not to favor any mode over the other.

  Matrix case: Do both matrix-vector multiplication and matrix-transpose-vector multiplication.

Mode-Specific

Row-oriented (CSR)



The concepts “mode-specific” and “mode-generic” orientations are inherited from [Baskaran et al. 2012].

[Baskaran et al. 2012] M. Baskaran et al., “Efficient and scalable computations with sparse tensors,” HPEC2012

Efficient

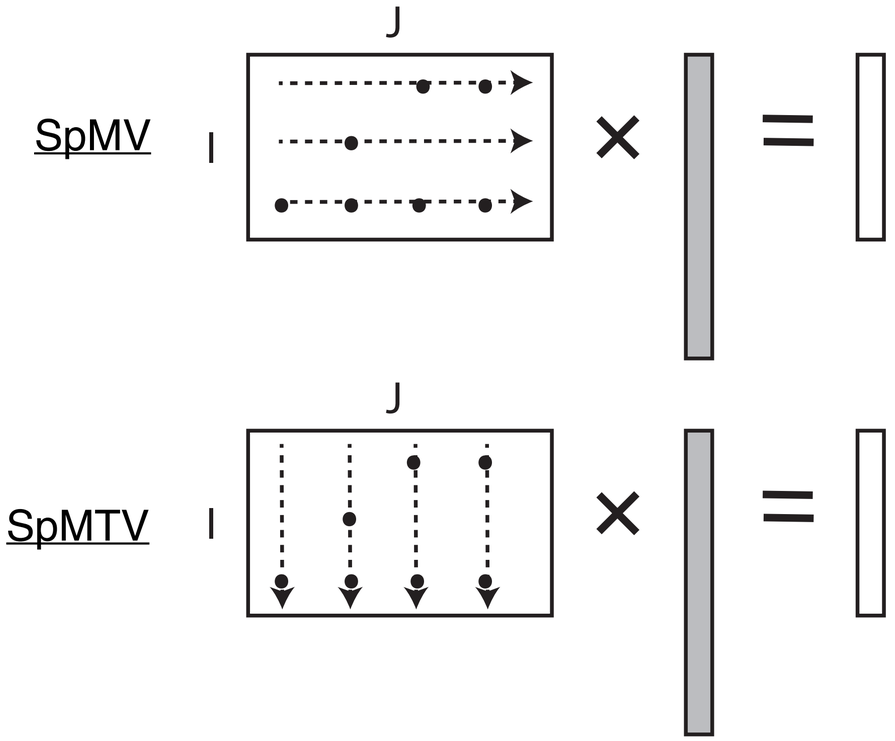
In-efficient

Mode Orientation

- Each mode has similar program behavior.

  Better not to favor any mode over the other.

  Matrix case: Do both matrix-vector multiplication and matrix-transpose-vector multiplication.

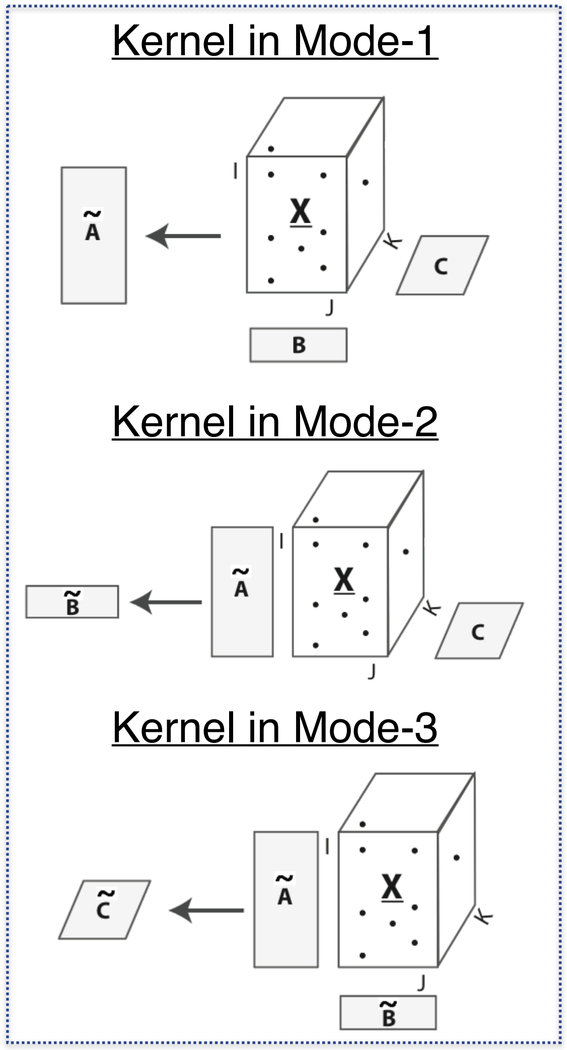


Mode-Specific	Mode-Generic
Row-oriented (CSR)	Coordinate (COO)
✓	✗
✗	✗

✓ Efficient ✗ In-efficient

Mode Orientation

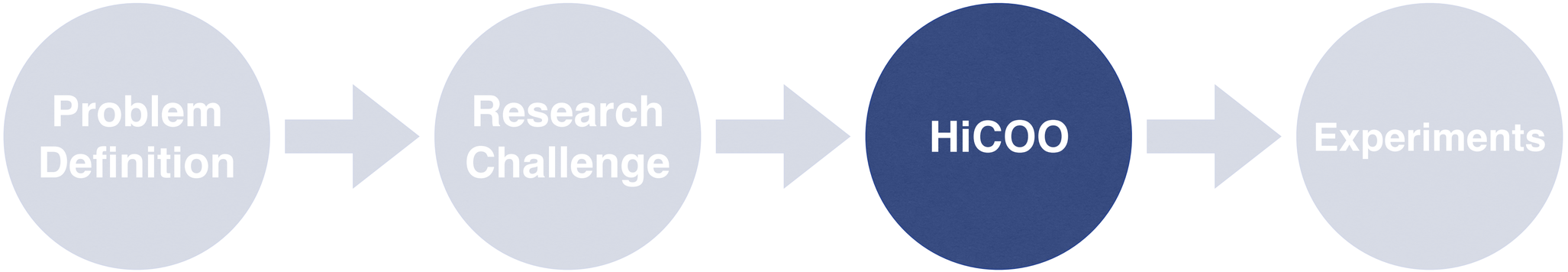
Tensor decomposition



	Mode-Specific	Mode-Generic
Mode-1 oriented (CSF/FCOO)	✓	✗
Mode-2	✗	✗
Mode-3	✗	✗

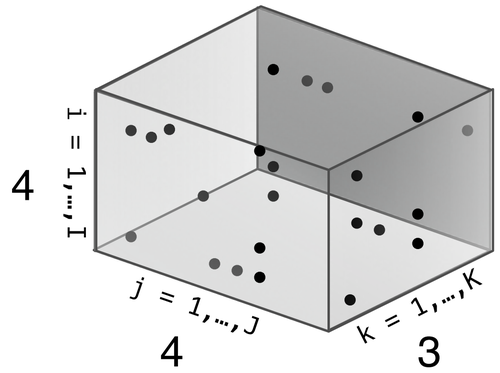
✓ Efficient ✗ In-efficient

Outline


Problem Definition
Research Challenge
HiCOO
Experiments

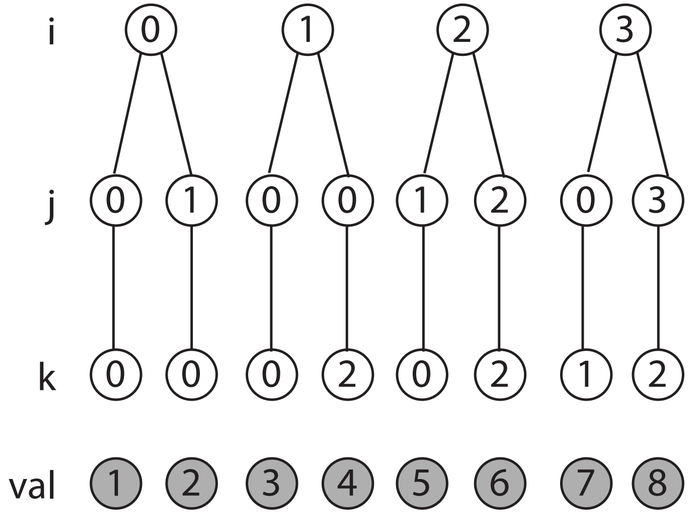
Current Sparse Tensor Formats

- COO: coordinate formats [Bader et al., 2006]
- CSF: Compressed Sparse Fibers, extension of CSR. [Smith et al. 2015]
- F-COO: Flagged COO format [Liu et al., 2017]



i	j	k	val
0	0	0	1
0	1	0	2
1	0	0	3
1	0	2	4
2	1	0	5
2	2	2	6
3	0	1	7
3	3	2	8

(a) COO



(b) CSF

	bf	j	k	val
sf[0]=1	1	0	0	1
	0	1	0	2
	1	0	0	3
	0	0	2	4
sf[1]=1	1	1	0	5
	0	2	2	6
	1	0	1	7
	0	3	2	8

(c) F-COO

Mode-Generic

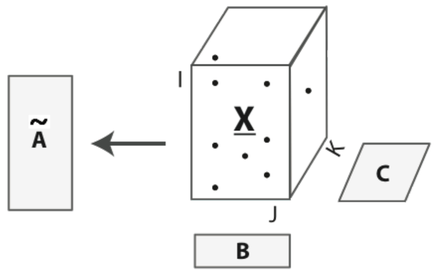
Mode-Specific
prefer different representations for different modes.

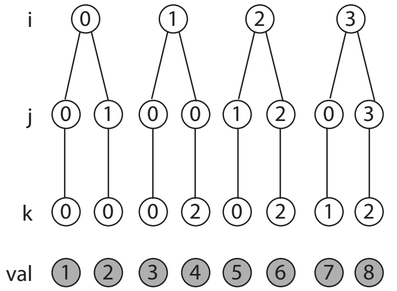
Mode-Specific Tensor Formats

- Three CSF/F-COO representations are required/preferred for three MTTKRPs.

Tensor Decomposition

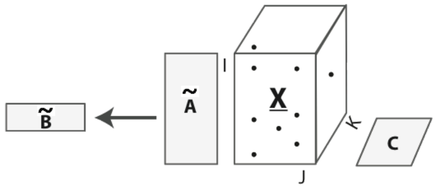
Kernel in Mode-1

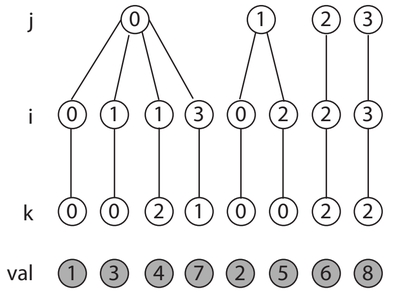




CSF-1

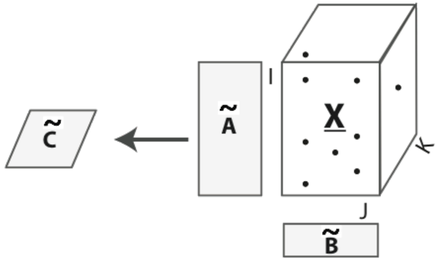
Kernel in Mode-2

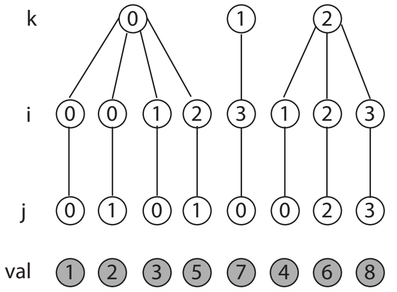




CSF-2

Kernel in Mode-3





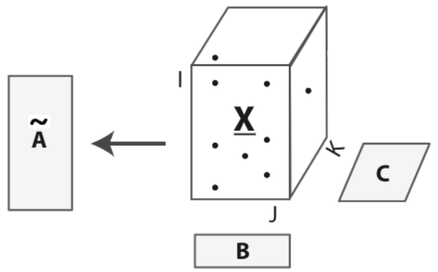
CSF-3

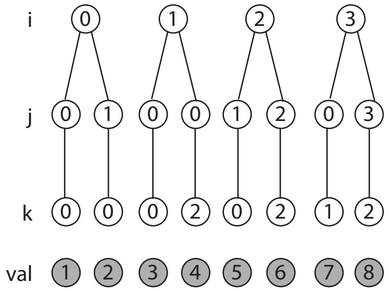
Mode-Specific Tensor Formats

- Three CSF/F-COO representations are required/preferred for three MTTKRPs.

Tensor Decomposition

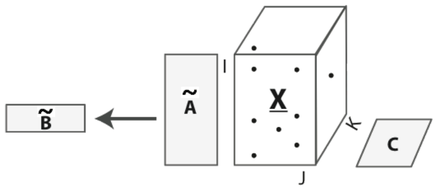
Kernel in Mode-1





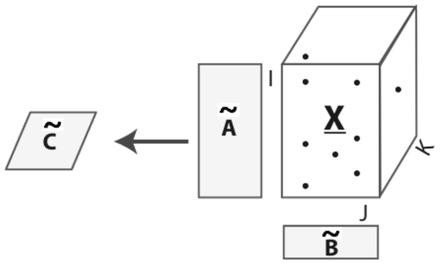
CSF-1

Kernel in Mode-2

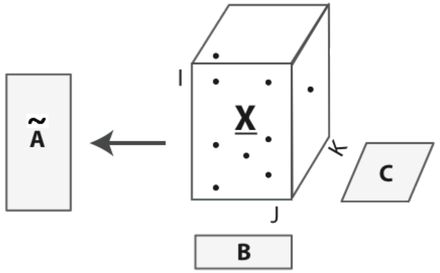
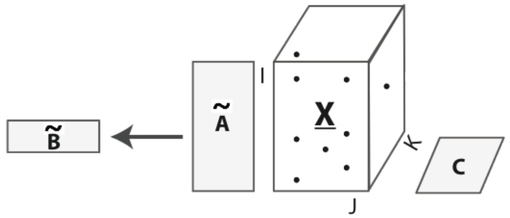
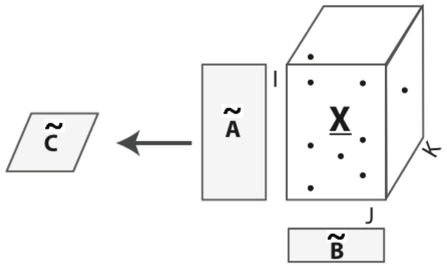


Performance payoff

Kernel in Mode-3



Mode Orientation

Tensor decomposition	Mode-Specific	Mode-Generic	
	Mode-1 oriented (CSF/FCOO)	Coordinate (COO)	**HiCOO**
Kernel in Mode-1 	✓	✗	✓
Kernel in Mode-2 	✗	✗	✓
Kernel in Mode-3 	✗	✗	✓

✓ Efficient ✗ In-efficient

HiCOO Format

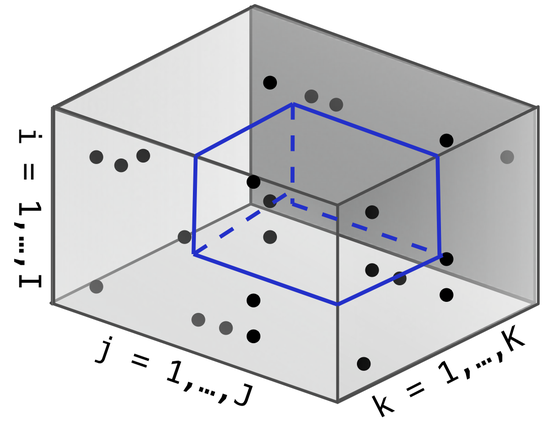
- Store a sparse tensor in units of small sparse blocks
  - Shorten the bit-length of element indices
  - Compress the number of block indices

i	j	k	val
0	0	0	1
0	1	0	2
1	0	0	3
1	0	2	4
2	1	0	5
2	2	2	6
3	0	1	7
3	3	2	8

COO

	bptr	bi	bj	bk	ei	ej	ek	val
B1	0	0	0	0	0	0	0	1
					0	1	0	2
					1	0	0	3
B2	3	0	0	1	1	0	0	4
B3	4	1	0	0	0	1	0	5
					1	0	1	7
B4	6	1	1	1	0	0	0	6
					1	1	0	8

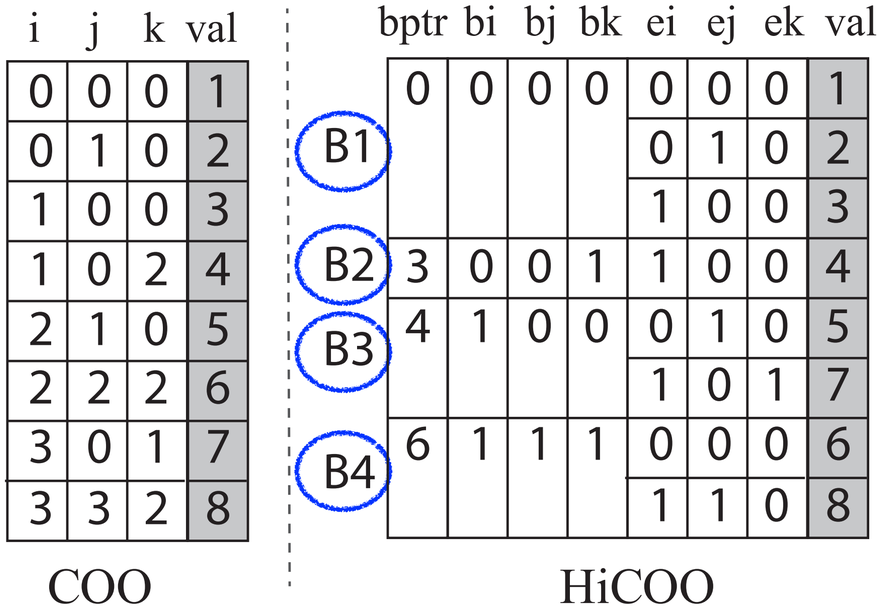
HiCOO



Block size: 2*2*2

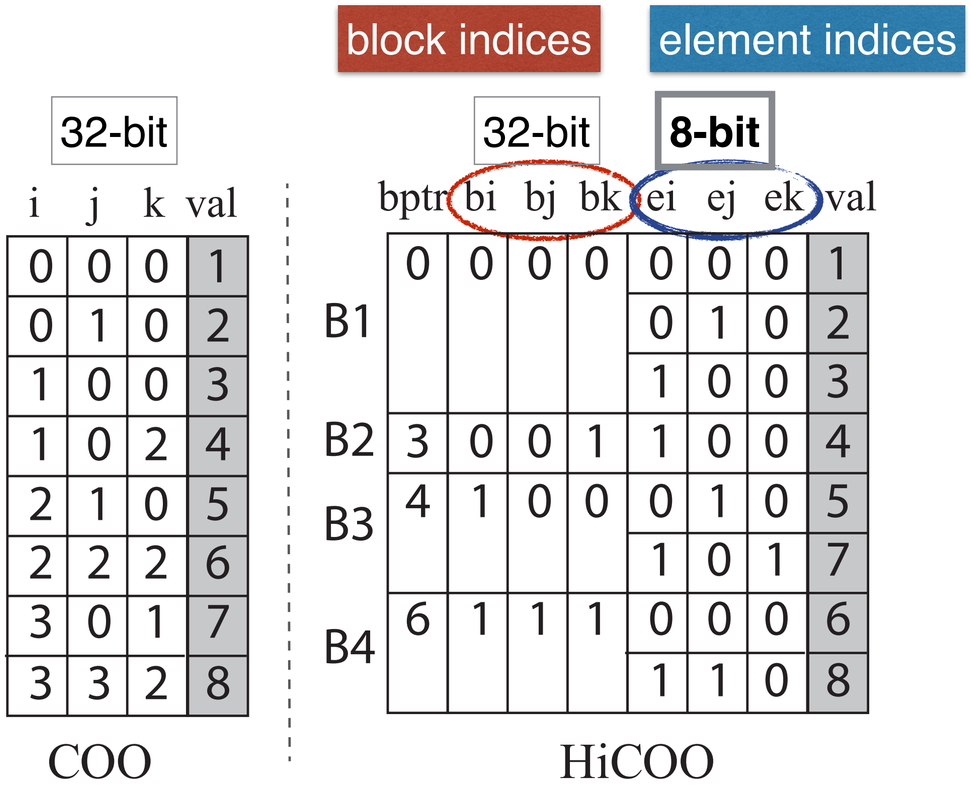
HiCOO Format

- Store a sparse tensor in units of small sparse blocks
  - Shorten the bit-length of element indices
  - Compress the number of block indices



HiCOO Format

- Store a sparse tensor in units of small sparse blocks
  - Shorten the bit-length of element indices
  - Compress the number of block indices

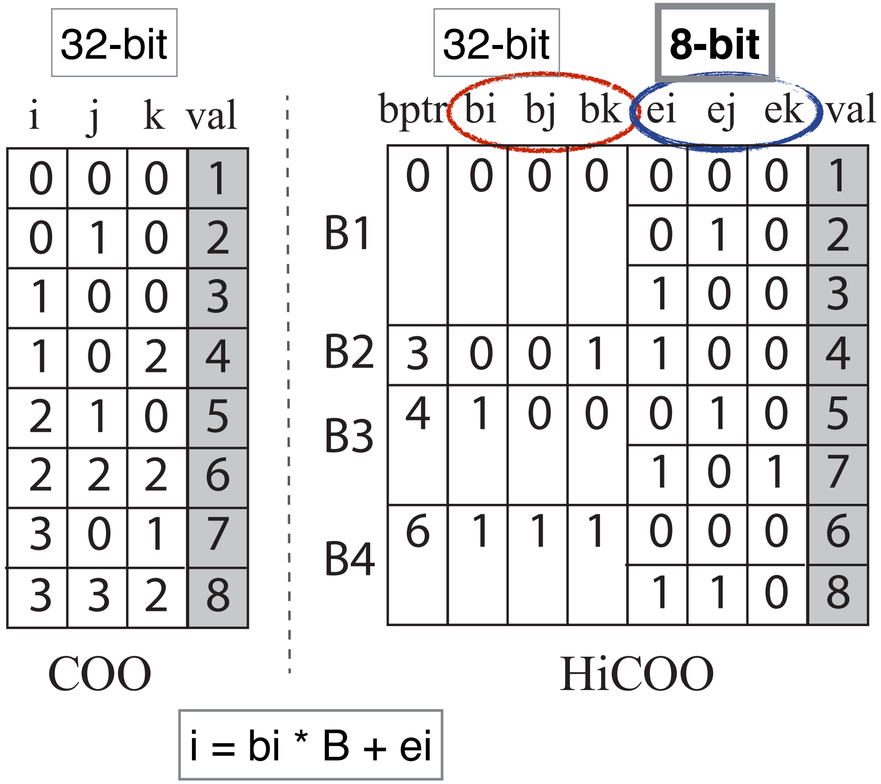


i	j	k	val
0	0	0	1
0	1	0	2
1	0	0	3
1	0	2	4
2	1	0	5
2	2	2	6
3	0	1	7
3	3	2	8

	bptr	bi	bj	bk	ei	ej	ek	val
B1	0	0	0	0	0	0	0	1
					0	1	0	2
					1	0	0	3
B2	3	0	0	1	1	0	0	4
B3	4	1	0	0	0	1	0	5
					1	0	1	7
B4	6	1	1	1	0	0	0	6
					1	1	0	8

HiCOO Format

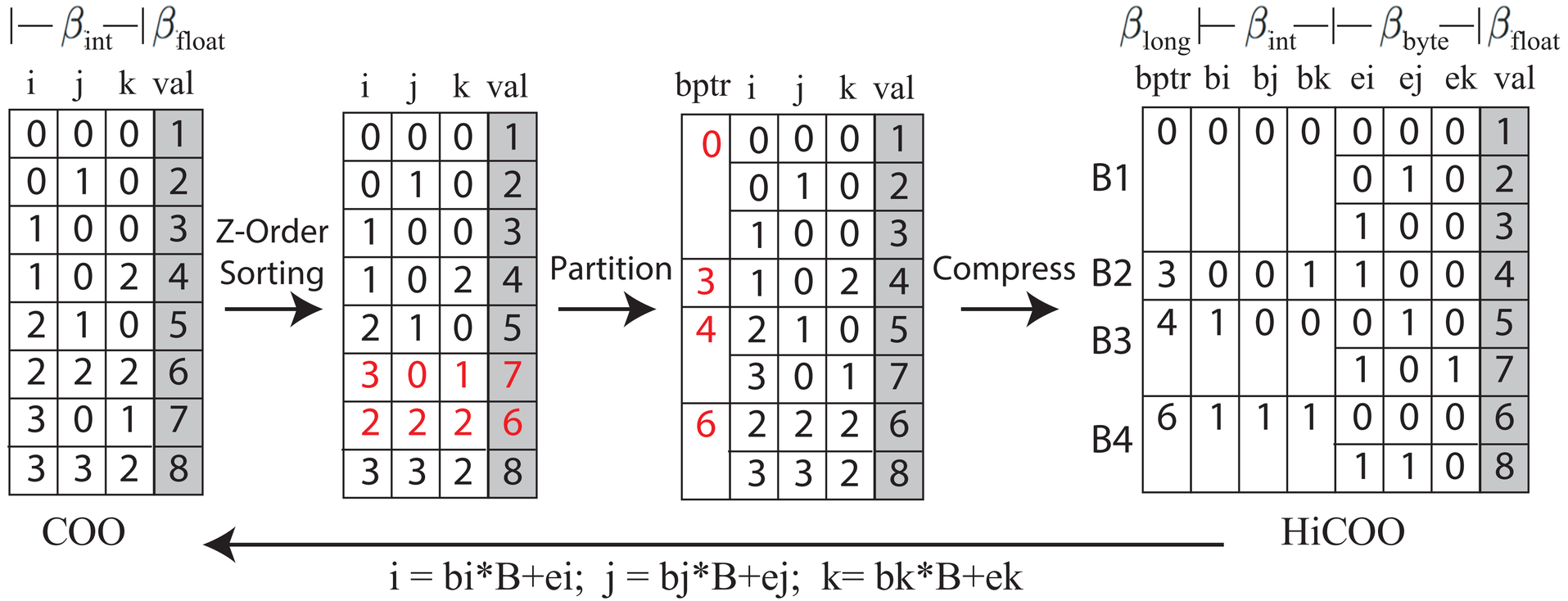
- Store a sparse tensor in units of small sparse blocks
  - Shorten the bit-length of element indices
  - Compress the number of block indices
  - For arbitrary-order sparse tensors.



For the tensor: Reduce its storage and memory footprints

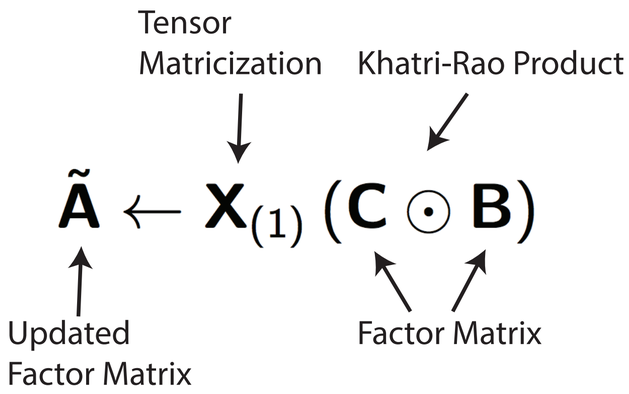
For matrices: Better data locality

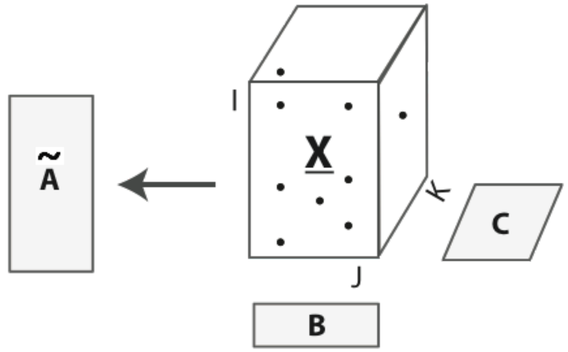
Format Conversion



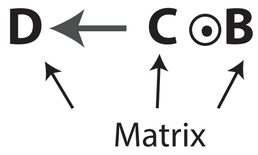
MTTKRP Operation

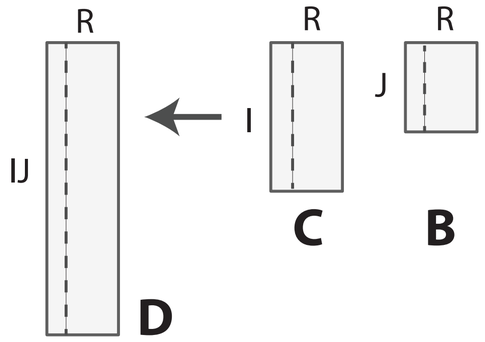
- Matriced Tensor Times Khatri-Rao Product (MTTKRP)





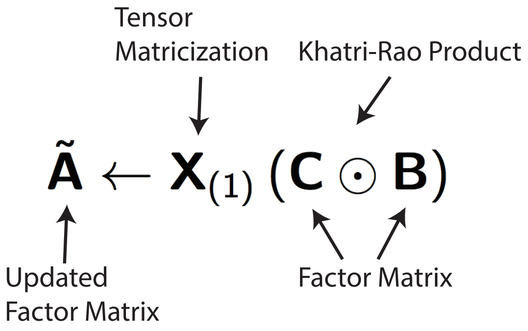
- Khatri-Rao Product

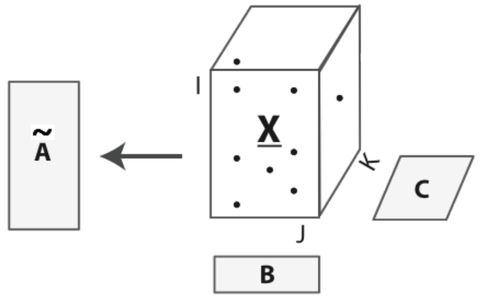




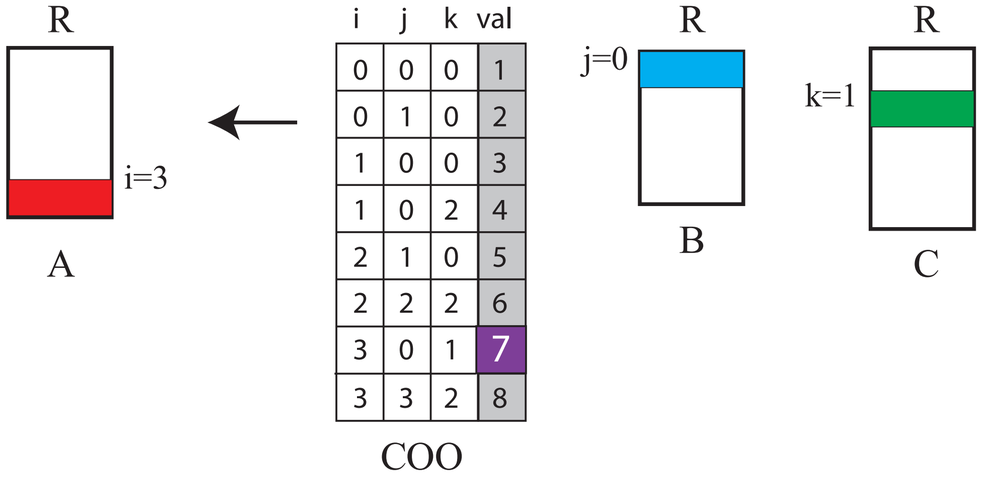
MTTKRP is the performance bottleneck of CP decomposition.

COO-MTTKRP

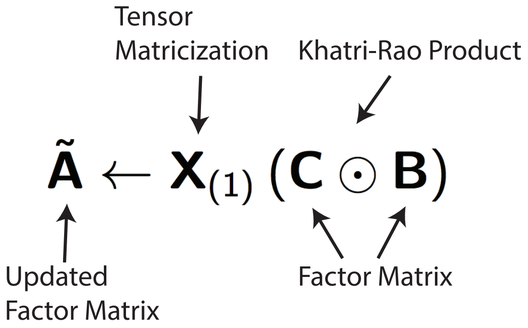


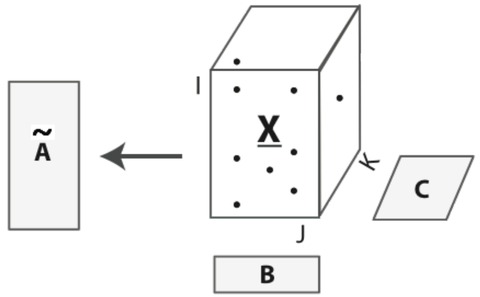


COO-MTTKRP algorithm in mode-1

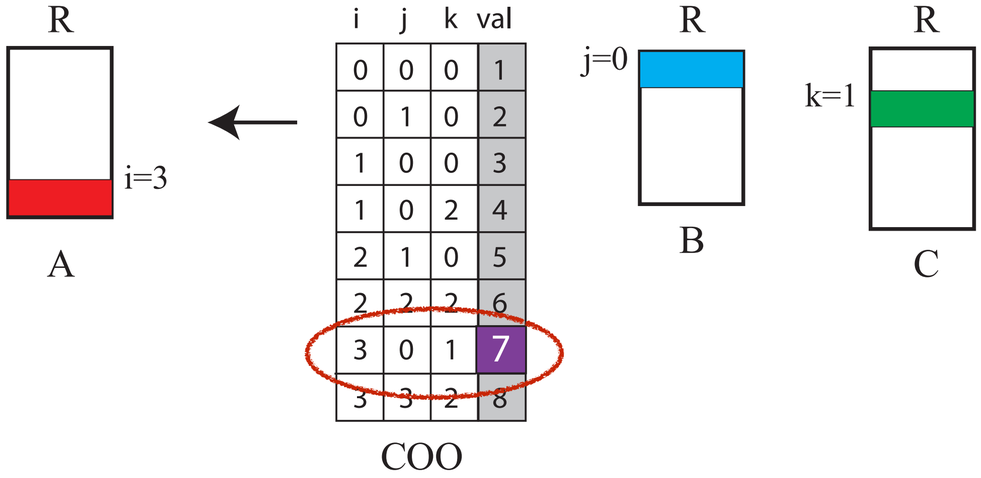


COO-MTTKRP

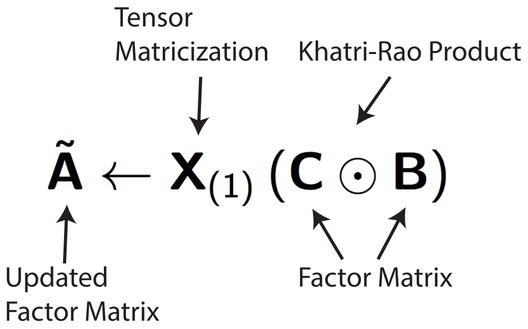


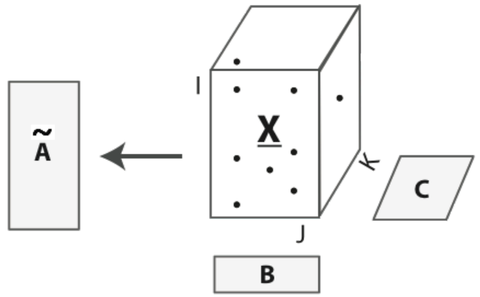


COO-MTTKRP algorithm in mode-1

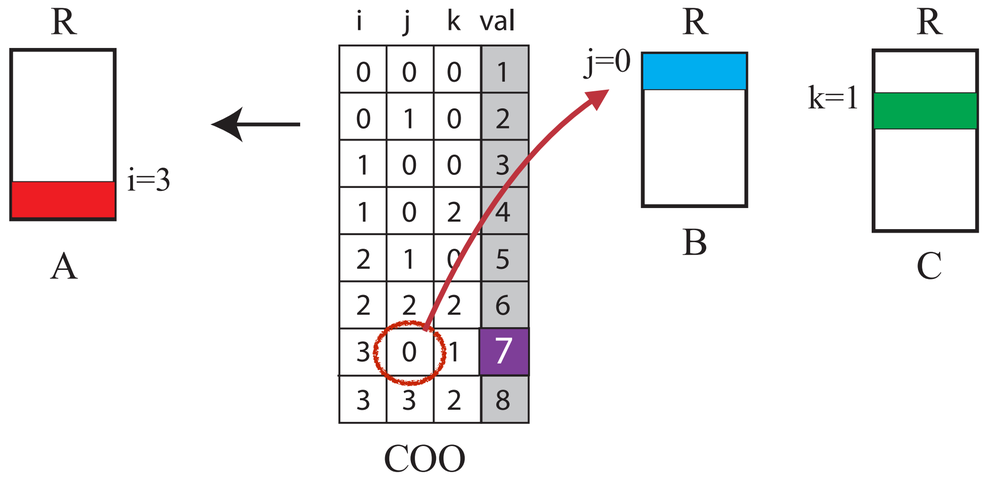


COO-MTTKRP

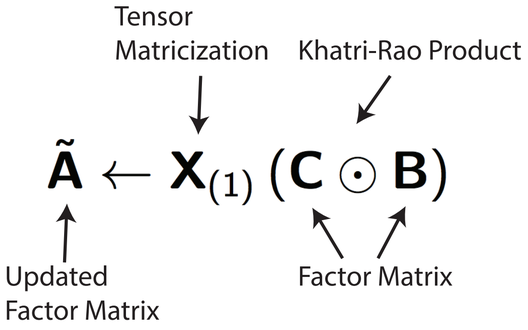


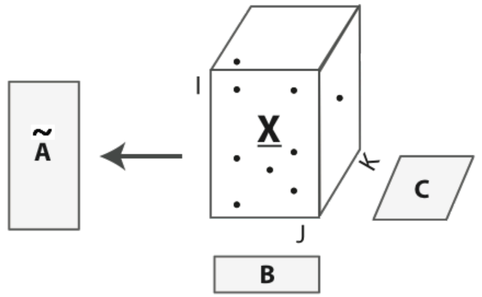


COO-MTTKRP algorithm in mode-1

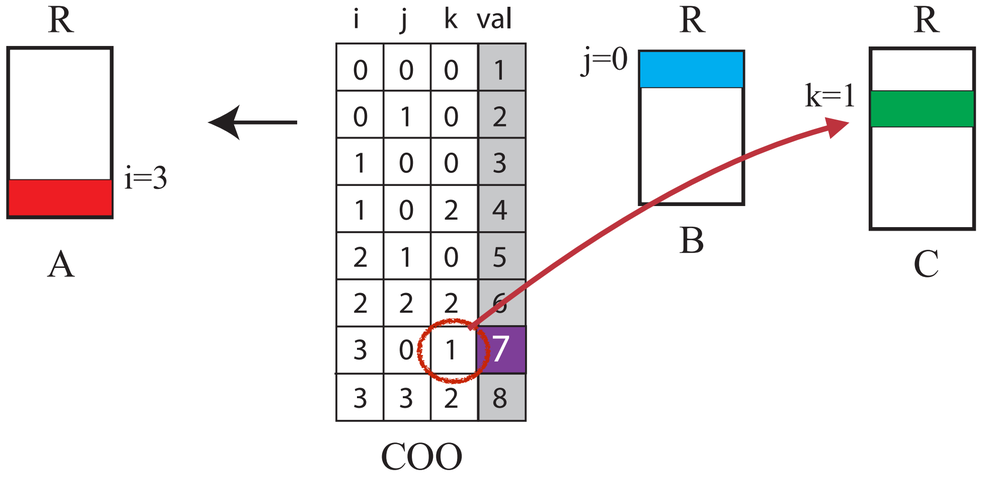


COO-MTTKRP

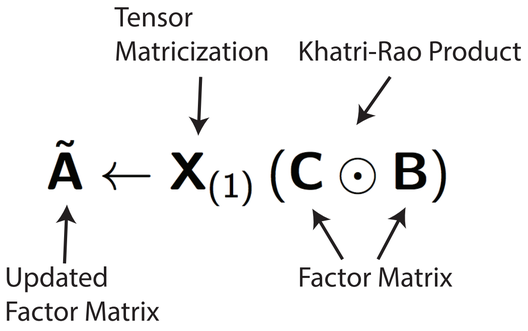


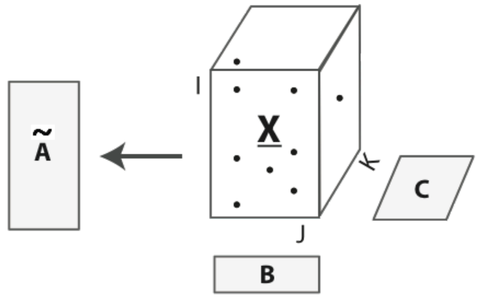


COO-MTTKRP algorithm in mode-1

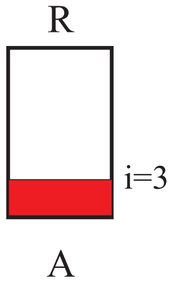


COO-MTTKRP



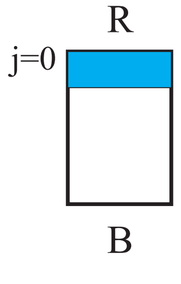


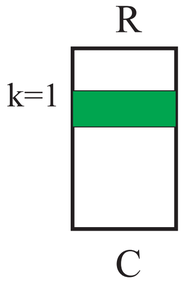
COO-MTTKRP algorithm in mode-1



i	j	k	val
0	0	0	1
0	1	0	2
1	0	0	3
1	0	2	4
2	1	0	5
2	2	2	6
3	0	1	7
3	3	2	8

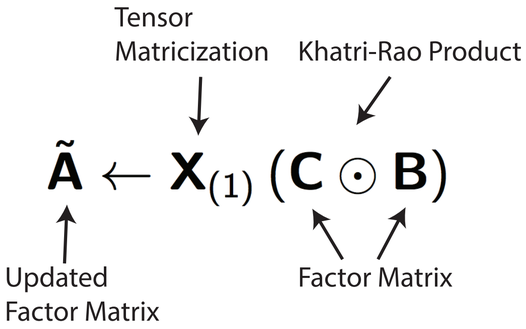
COO

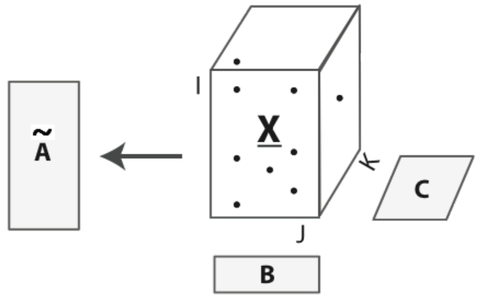




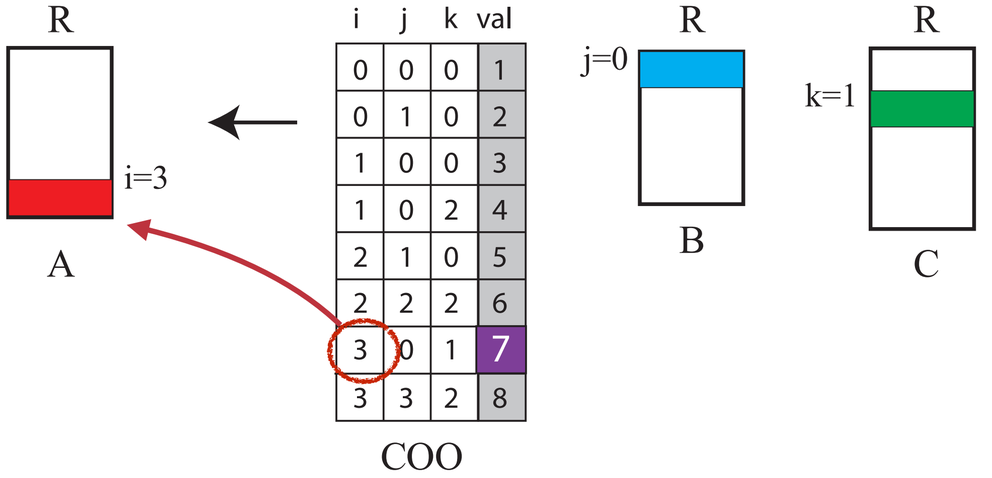
Entry-wise

COO-MTTKRP

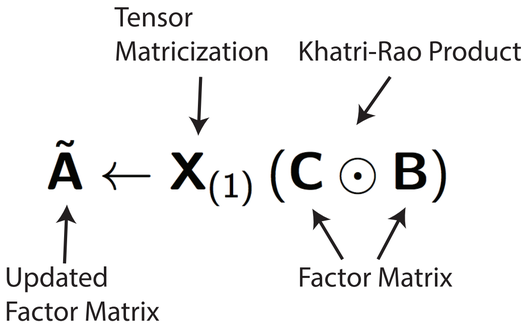


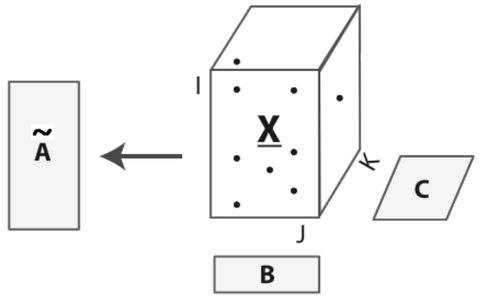


COO-MTTKRP algorithm in mode-1

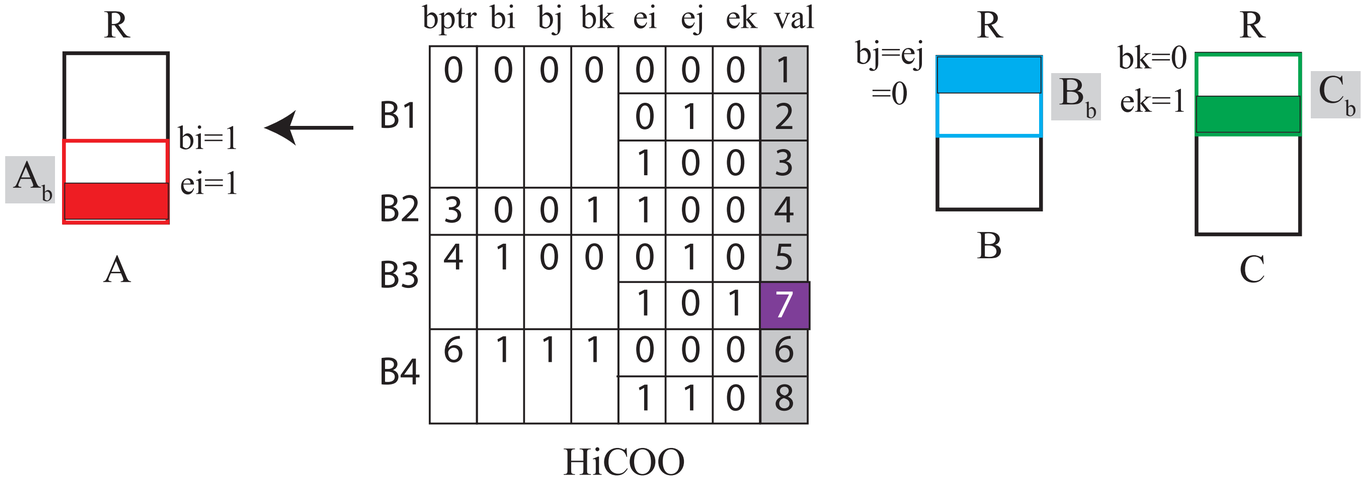


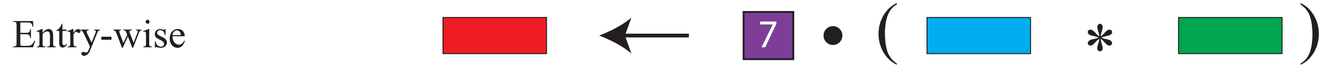
HiCOO-MTTKRP



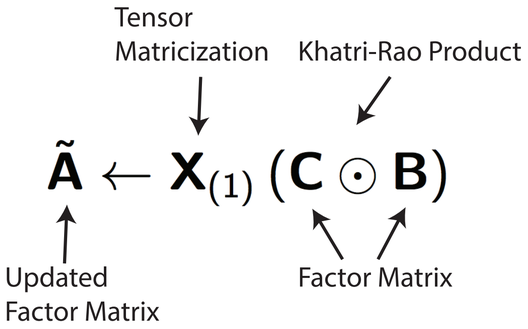


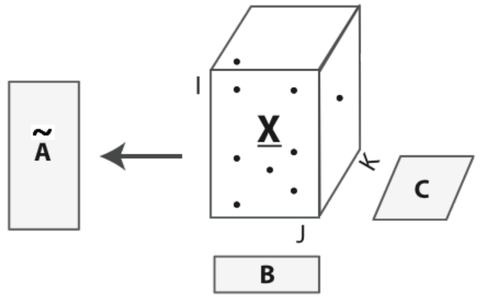
HiCOO-MTTKRP algorithm in mode-1



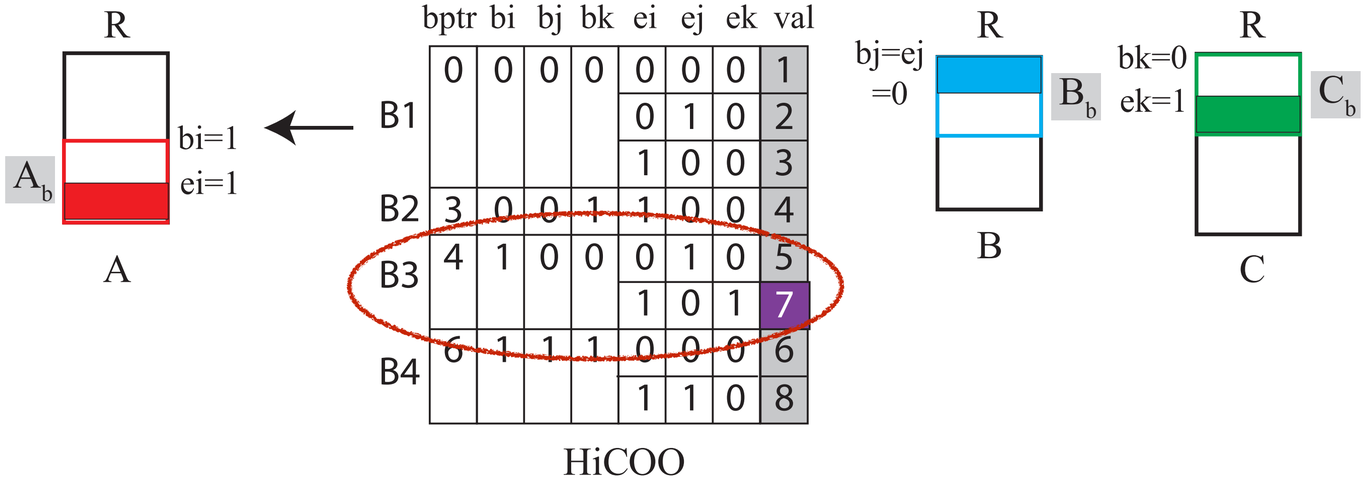


HiCOO-MTTKRP

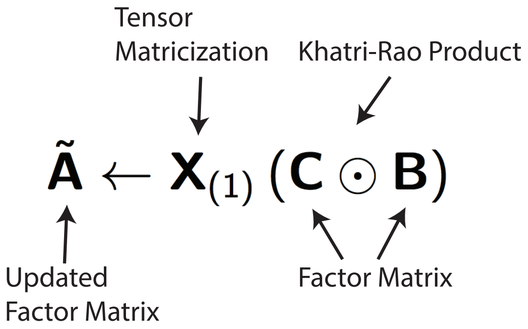


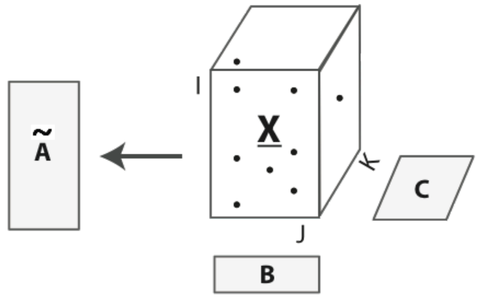


HiCOO-MTTKRP algorithm in mode-1

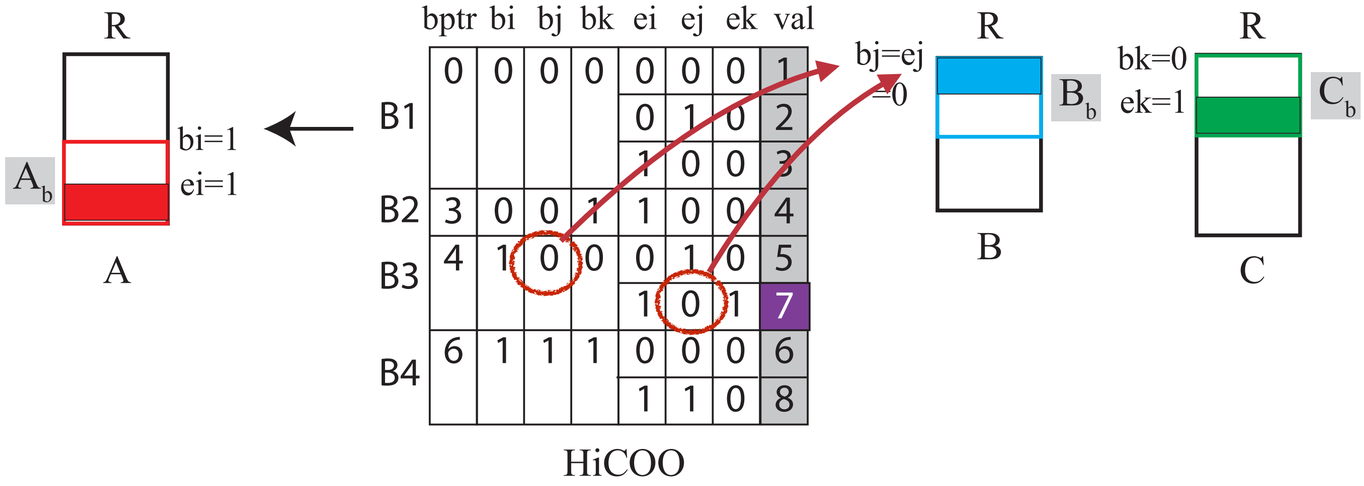


HiCOO-MTTKRP

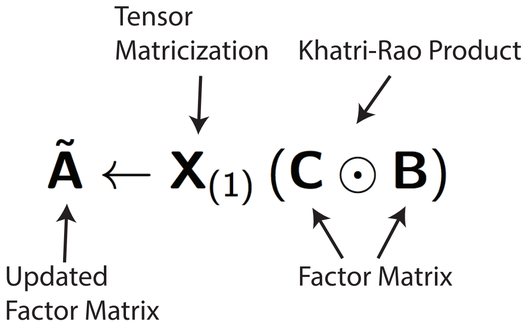


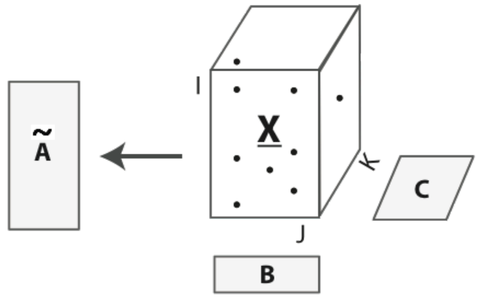


HiCOO-MTTKRP algorithm in mode-1

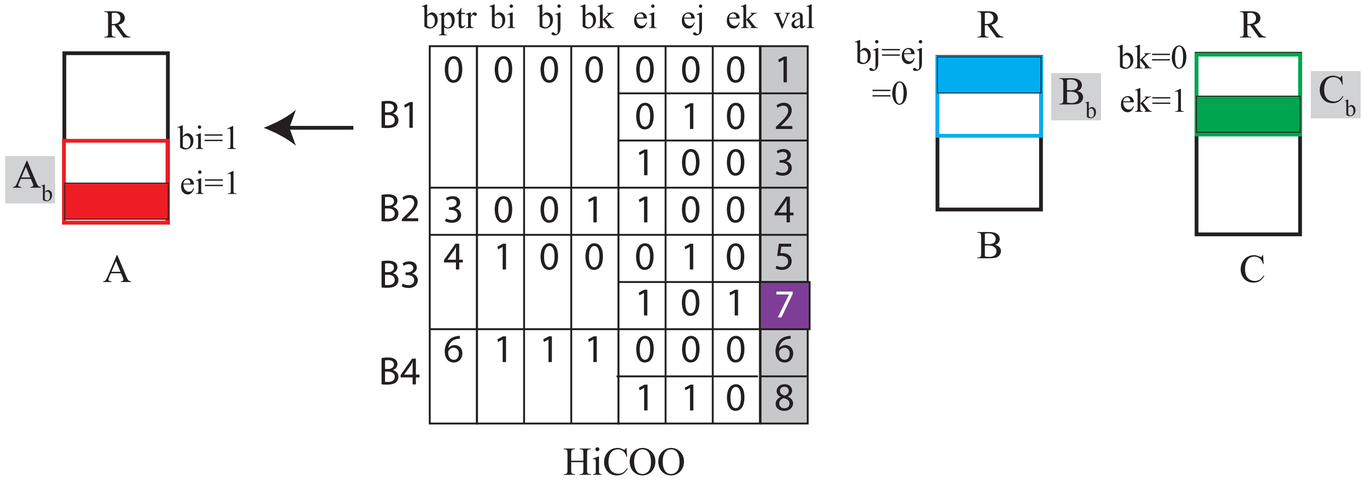


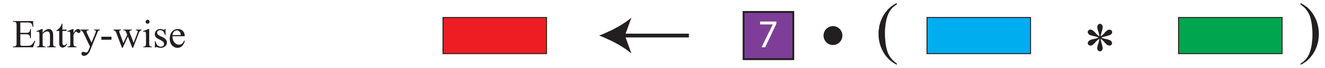
HiCOO-MTTKRP





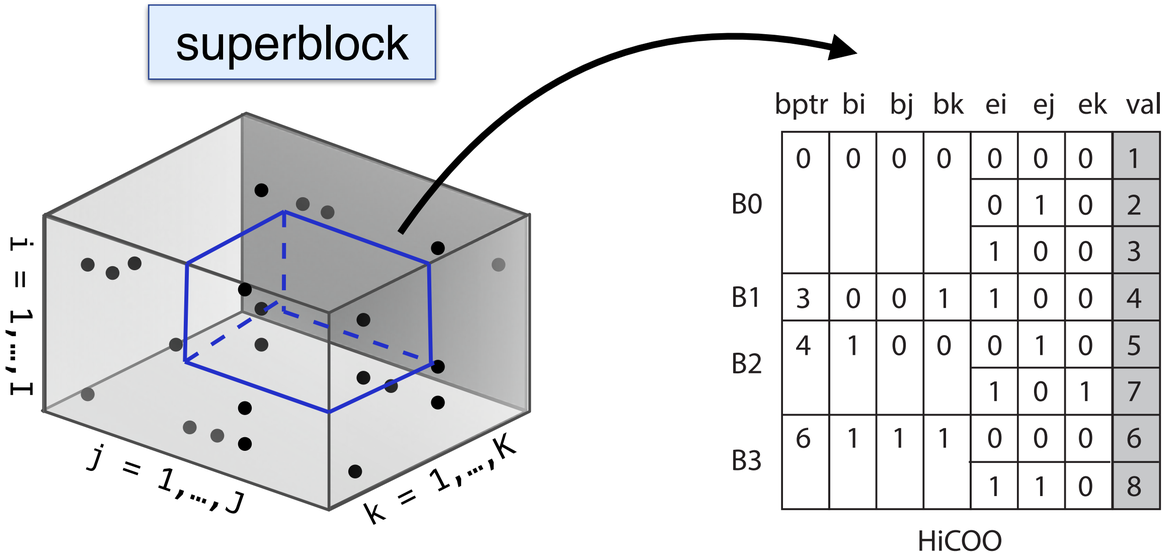
HiCOO-MTTKRP algorithm in mode-1





Two-level Blocking for Efficient Thread Parallelism

- Use two-level blocking strategy
  - Large superblocks (logical) + small blocks (physical)

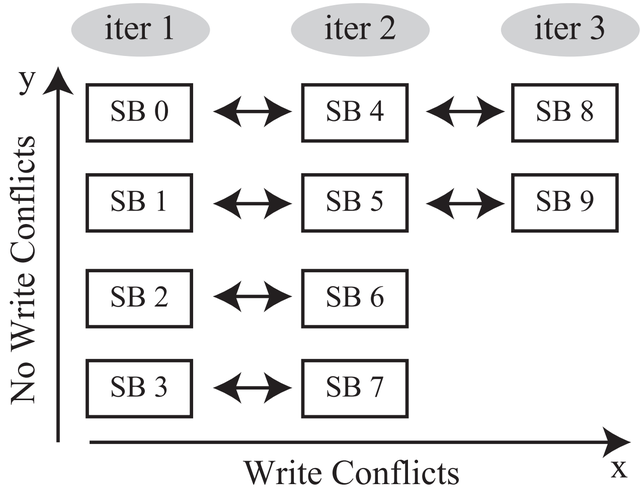


	bptr	bi	bj	bk	ei	ej	ek	val
B0	0	0	0	0	0	0	0	1
					0	1	0	2
					1	0	0	3
B1	3	0	0	1	1	0	0	4
B2	4	1	0	0	0	1	0	5
					1	0	1	7
B3	6	1	1	1	0	0	0	6
					1	1	0	8

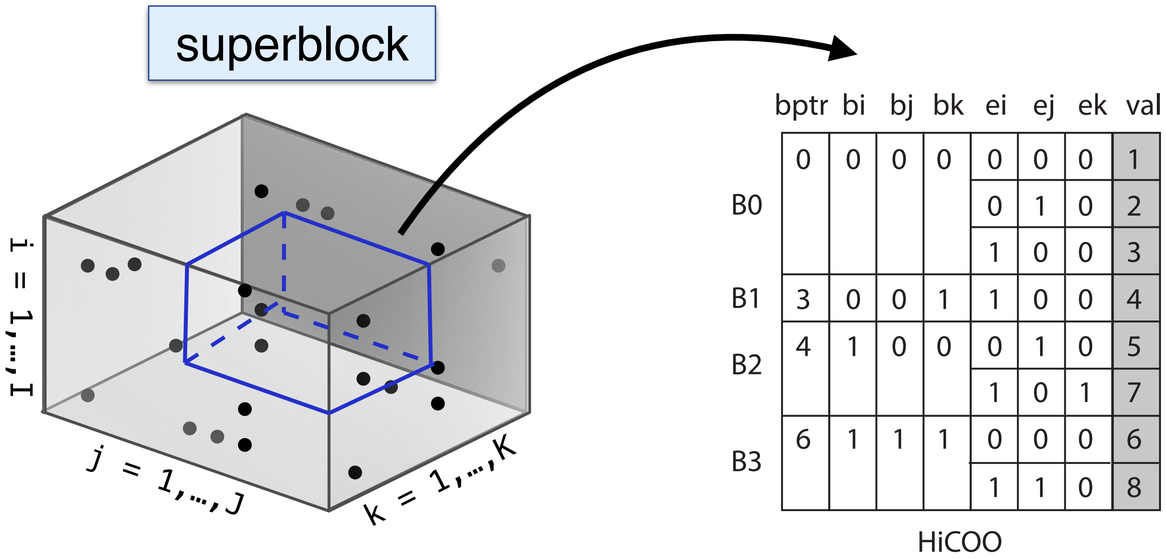
HiCOO

Two-level Blocking for Efficient Thread Parallelism

- Use two-level blocking strategy
  - Large superblocks (logical) + small blocks (physical)
  - To avoid using locks, we schedule superblocks according to scheduler with two parallel strategies (direct + privatization).
  - Increase only a bit extra storage.



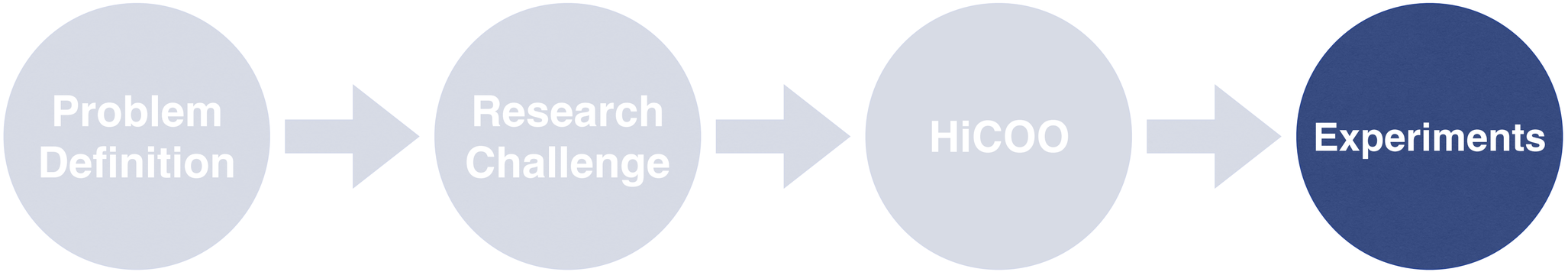
Superblock scheduler



	bptr	bi	bj	bk	ei	ej	ek	val
B0	0	0	0	0	0	0	0	1
					0	1	0	2
					1	0	0	3
B1	3	0	0	1	1	0	0	4
B2	4	1	0	0	0	1	0	5
					1	0	1	7
B3	6	1	1	1	0	0	0	6
					1	1	0	8

HiCOO

Outline


Problem Definition
Research Challenge
HiCOO
Experiments

Platform and Dataset

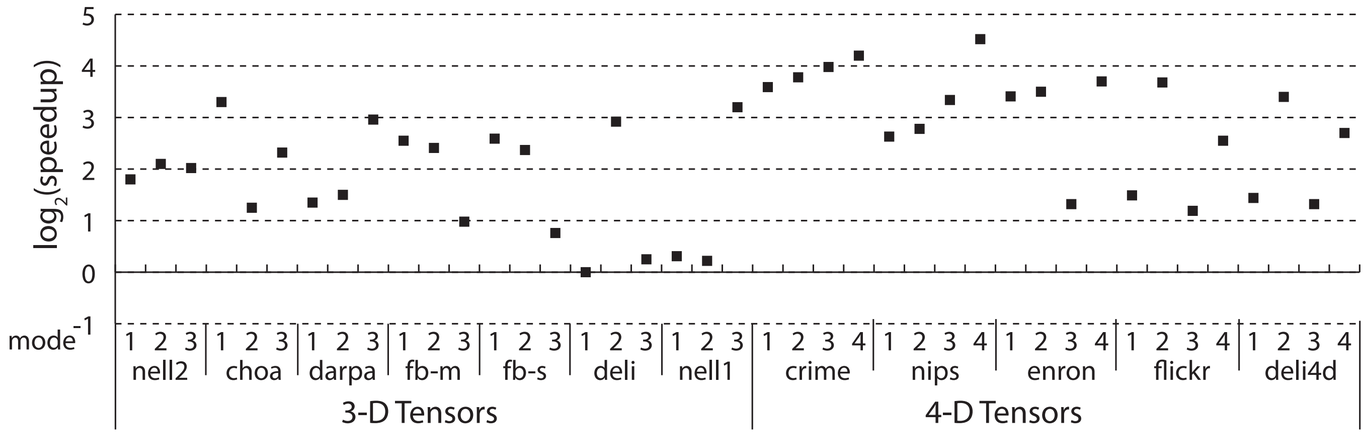
- **Platform**: Intel Xeon CPU E7-4850 v3 platform consisting 56 physical cores with icc 18.0.2 and parallelized by OpenMP.
- **Dataset**: FROSTT [Smith et al. 2017], HaTen2 [Jeon et al. 2015], and healthcare data.

DESCRIPTION OF SPARSE TENSORS.

Tensors	Order	Dimensions	#Nonzeros	Density
nell2	3	$12K \times 9K \times 29K$	77M	2.4×10^{-5}
choa	3	$712K \times 10K \times 767$	27M	5.0×10^{-6}
darpa	3	$22K \times 22K \times 24M$	28M	2.4×10^{-9}
fb-m	3	$23M \times 23M \times 166$	100M	1.1×10^{-9}
fb-s	3	$39M \times 39M \times 532$	140M	1.7×10^{-10}
deli	3	$533K \times 17M \times 2.5M$	140M	6.1×10^{-12}
nell1	3	$3M \times 2M \times 25M$	144M	9.1×10^{-13}
crime	4	$6K \times 24 \times 77 \times 32$	5M	1.5×10^{-2}
nips	4	$2K \times 3K \times 14K \times 17$	3M	1.8×10^{-6}
enron	4	$6K \times 6K \times 244K \times 1K$	54M	5.5×10^{-9}
flickr	4	$320K \times 28M \times 2M \times 731$	113M	1.1×10^{-14}
deli4d	4	$533K \times 17M \times 2M \times 1K$	140M	4.3×10^{-15}

Multicore MTTKRP

- ParTI! library: Speedups of HiCOO over COO

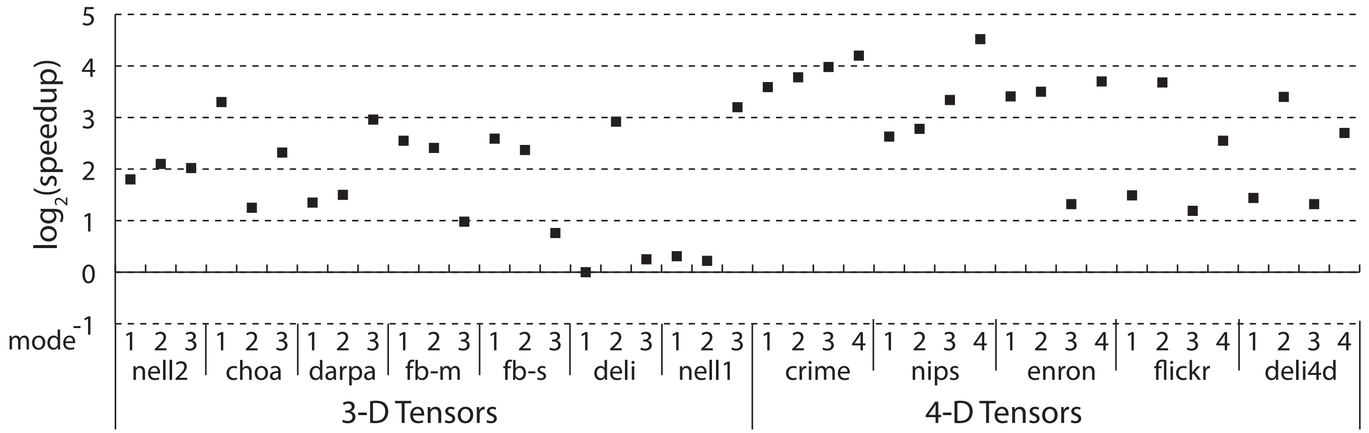


6.8x

Average speedup

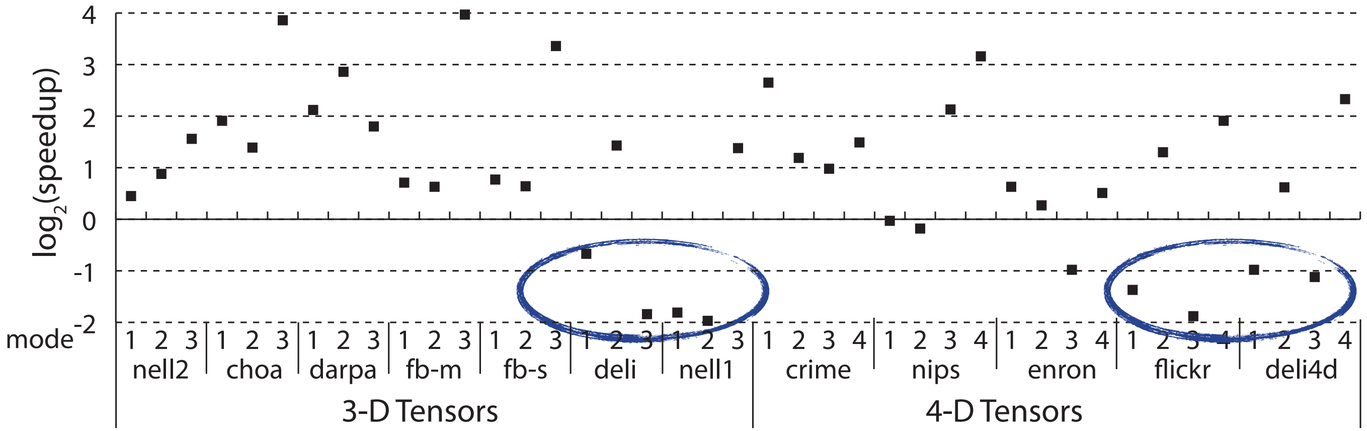
Multicore MTTKRP

- ParTI! library: Speedups of HiCOO over COO



6.8x

- SPLATT library: Speedups of HiCOO over CSF

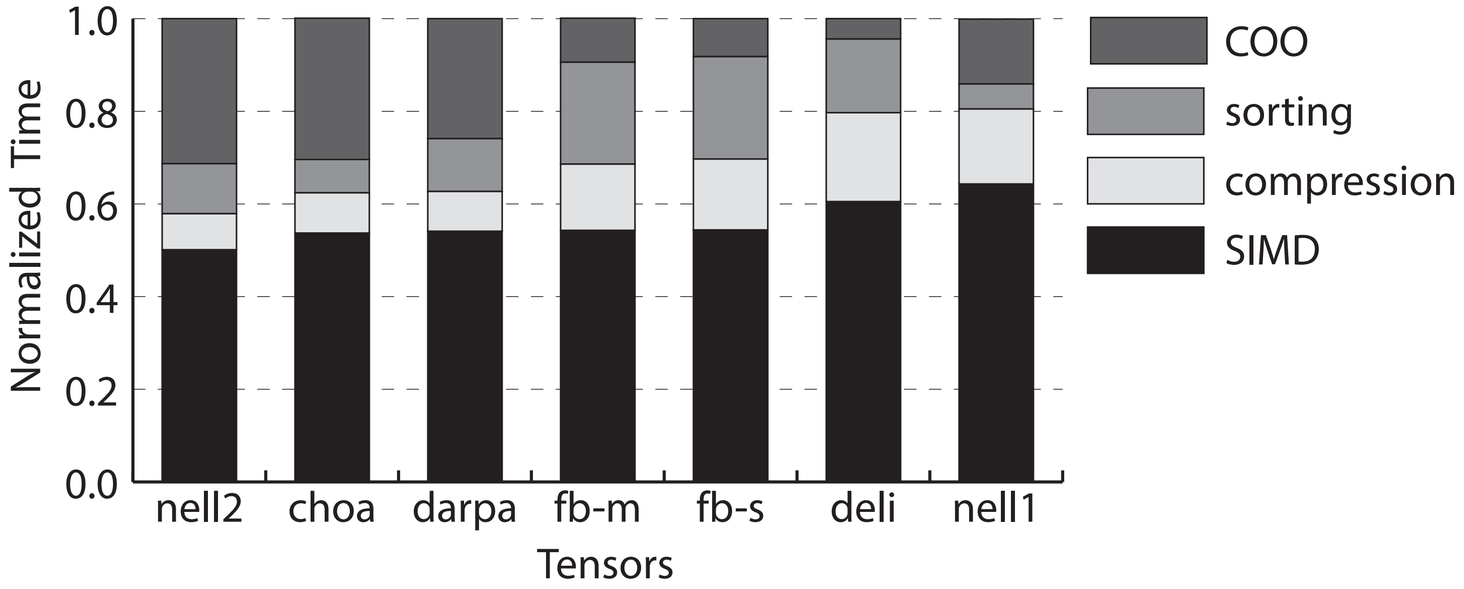


Average speedup

3.1x

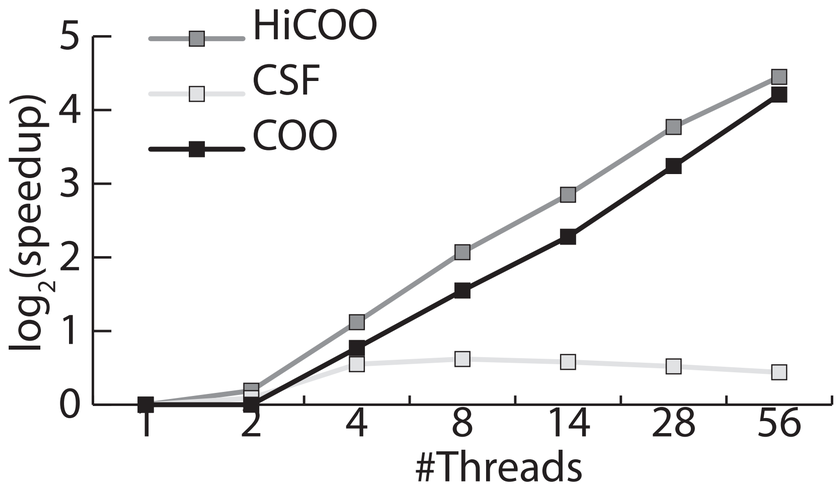
HiCOO Optimization Breakdown

- Z-order sorting: +18%
- Index compression: +20%
- SIMD: +22%

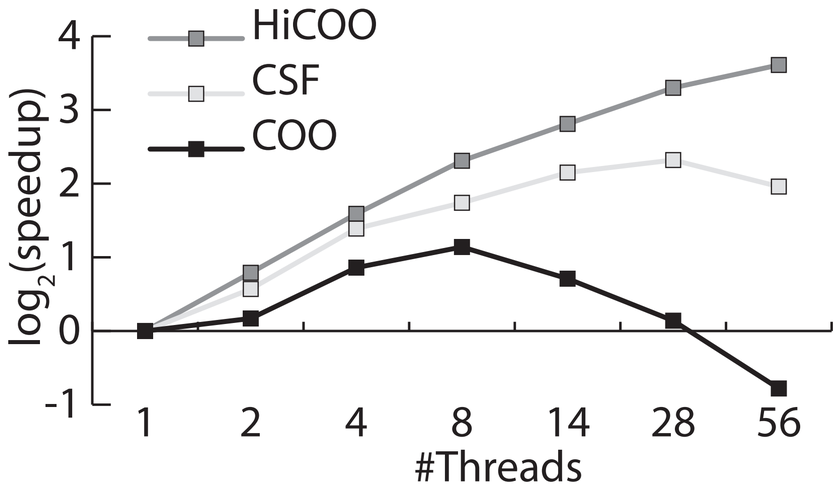


Thread Scalability

- Thread scalability of parallel COO, CSF, and HiCOO MTTKRPs on two representative cases.
- HiCOO achieves the best scalability.



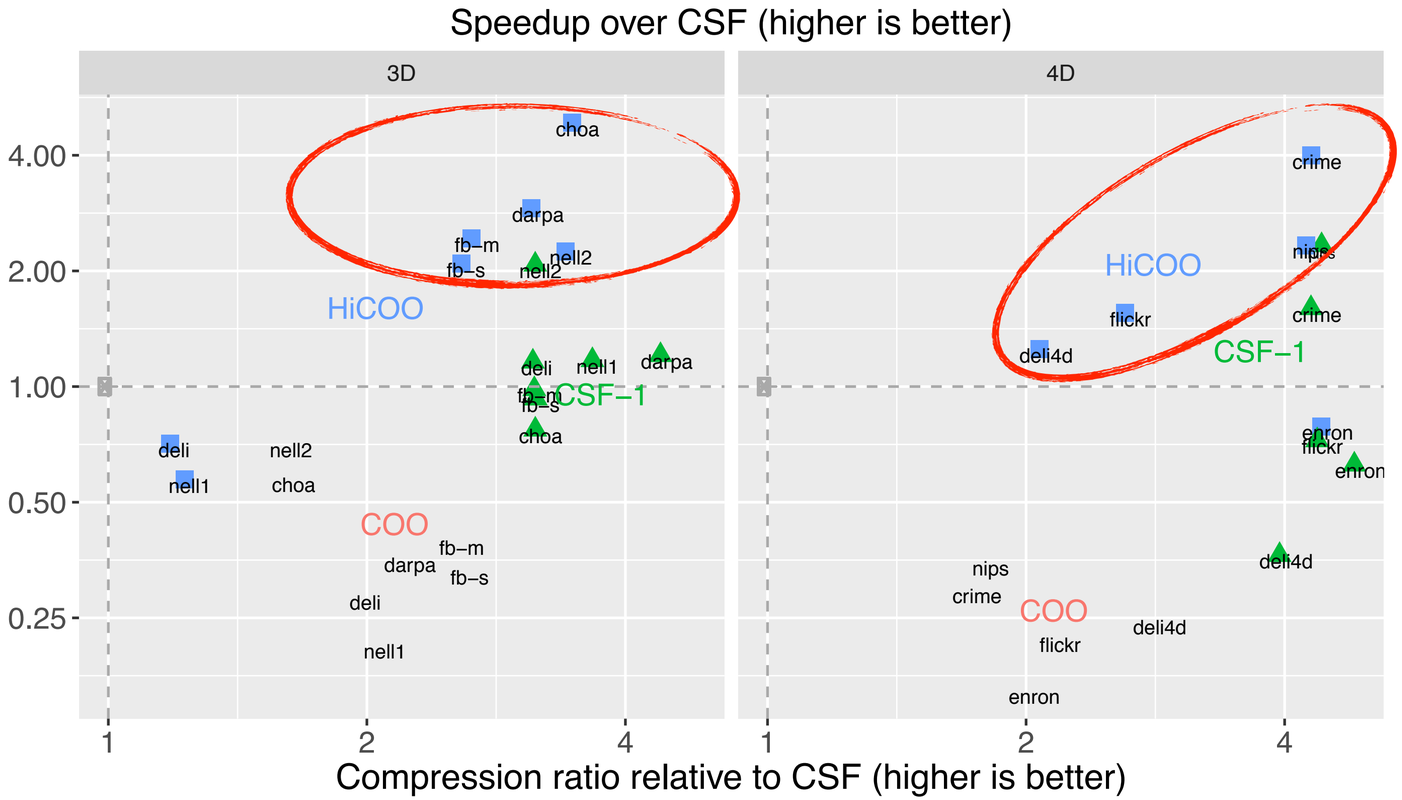
tensor fb-s in mode 3
(shortest mode)



tensor choa in mode 1
(longest mode)

Multicore CP-ALS

- HiCOO outperforms COO by 6.2× and CSF up to 2.1× on average.

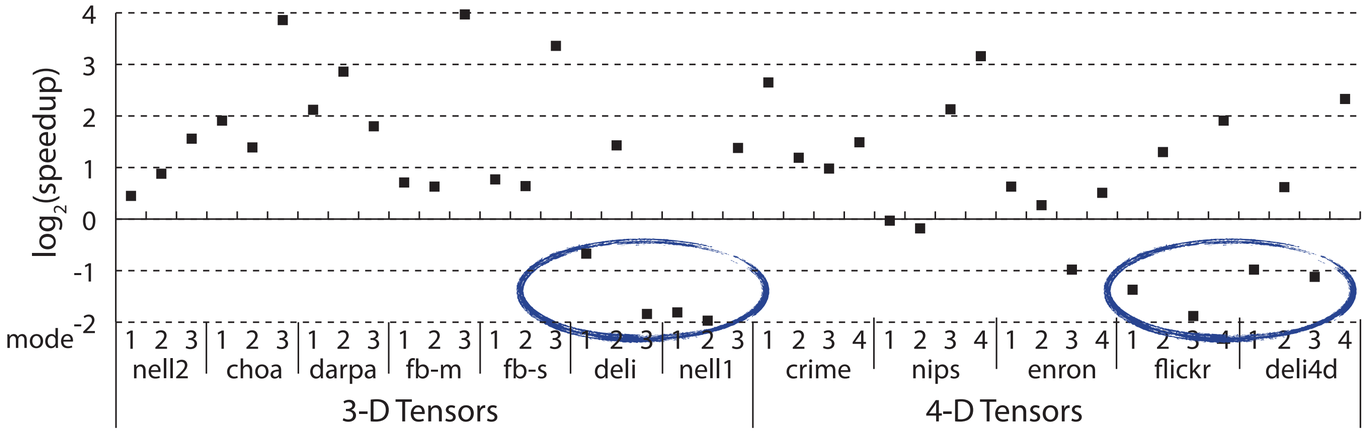


HiCOO Parameters

Parameters	Meaning	Effect	Preferable values
α_b	Block ratio	Tensor storage	small $\alpha_b < \frac{\beta_{\text{int}} - \beta_{byte}}{\beta_{\text{int}} + \beta_{long} / N}$
$\overline{c_b}$	Average slice size per tensor block	Memory traffic	large
L	Superblock size	Parallel granularity	depends
B	Block size	Data locality	$B \leq \frac{S_{cache}}{NR\beta_{float}}$

HiCOO Parameters cont.

- Reorder a sparse tensor to obtain denser blocks.
- Better for cache utilization and HiCOO algorithm performance.

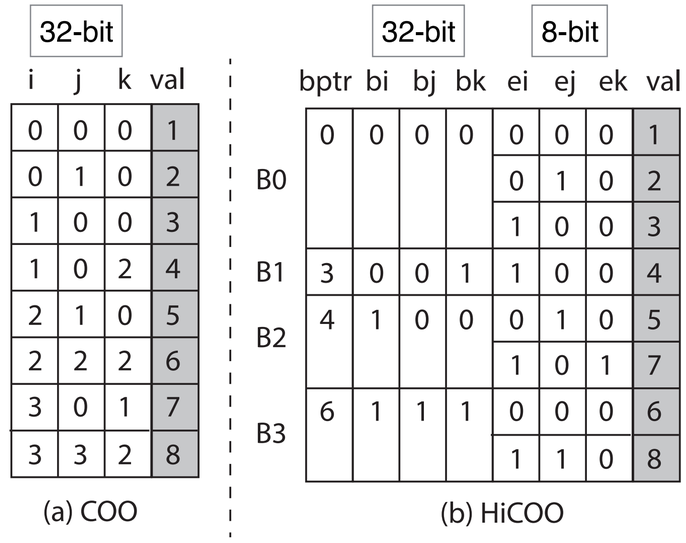


Speedups of HiCOO over CSF

Tensors	No reordering		
	α_b	$\overline{c_b}$	L
vast	0.003	2.210	8
nell2	0.020	0.302	10
choa	0.023	0.070	10
darpa	0.217	0.016	15
fb-m	0.416	0.011	18
fb-s	0.456	0.010	18
flickr	0.358	0.014	13
deli	0.988	0.008	16
nell1	0.998	0.008	18
crime	0.000	666.892	4
uber	0.000	0.998	4
nips	0.016	0.416	7
enron	0.037	0.031	8
flickr4d	0.358	0.014	15
deli4d	0.797	0.009	16

HiCOO: Hierarchical Storage of Sparse Tensors

- Neutral mode orientation format for arbitrary-order sparse tensors.
- Code: https://github.com/hpcgarage/ParTI
- Future steps:
  - Extend to sparse TTM and Tucker decomposition.
  - Optimize HiCOO-MTTKRP on GPUs.
  - Accelerate tensor reordering and format construction process.



A haiku for HiCOO

— By Richard W. Vuduc

Flexible format
Of hierarchical sparse blocks
Small, and often fast

Acknowledgement