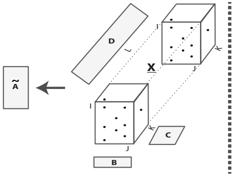
Model-Driven Sparse CP Decomposition for Higher-Order Tensors

Jiajia Li[1], Jee Choi[2], Ioakeim Perros[1], Jimeng Sun[1], Richard Vuduc[1]

[1] Computational Science & Engineering, Georgia Institute of Technology, GA, USA
[2] IBM Thomas J. Watson Research Center, Yorktown Heights, NY, USA

IPDPS'17, June 1st 2017

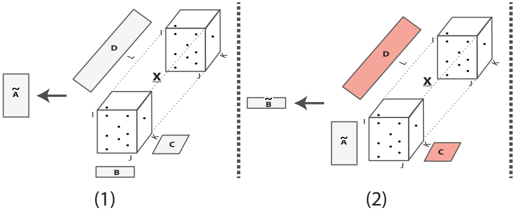
The problem



(1)

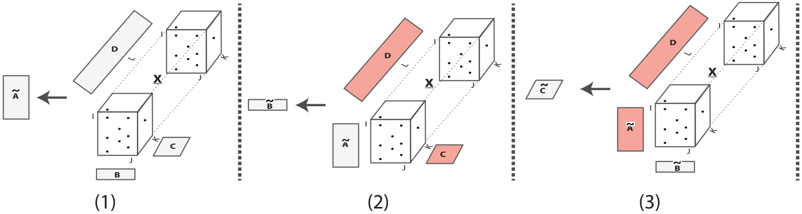
A 4th-Matriced Tensor Times Khatri-Rao Product (MTTKRP) sequence from a tensor decomposition.

The problem



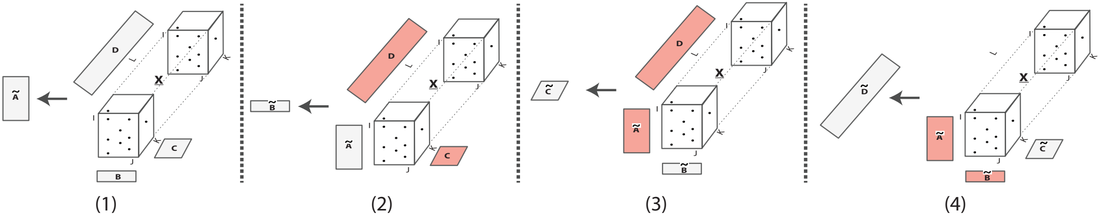
A 4th-MTTKRP sequence from a tensor decomposition.

The problem



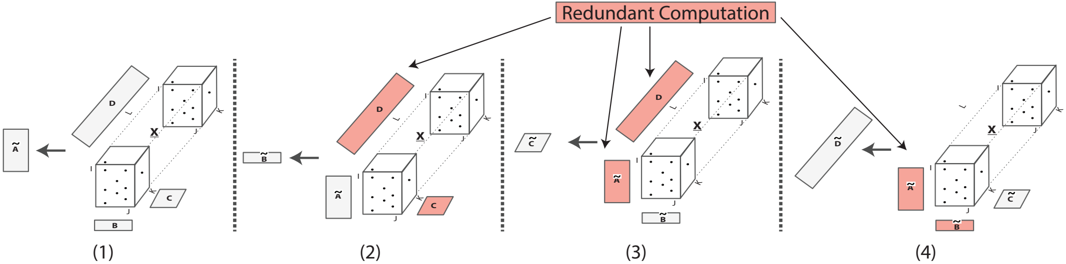
A 4th-MTTKRP sequence from a tensor decomposition.

The problem



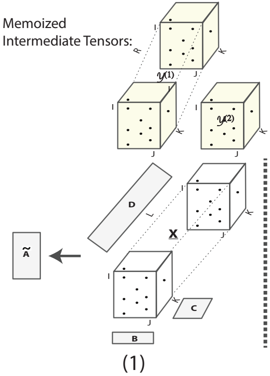
A 4th-MTTKRP sequence from a tensor decomposition.

The problem



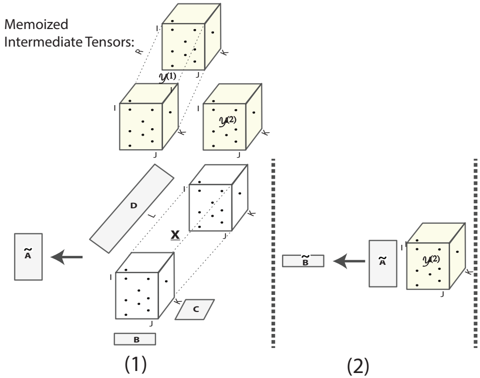
A 4th-MTTKRP sequence from a tensor decomposition.

Solution A



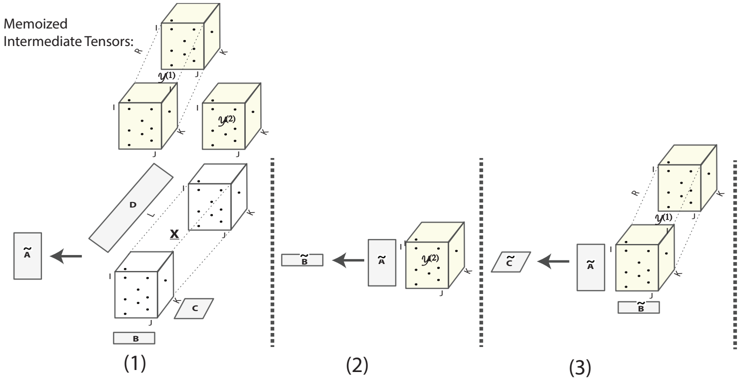
One scheme to save flops at the cost of increasing intermediate storage.

Solution A



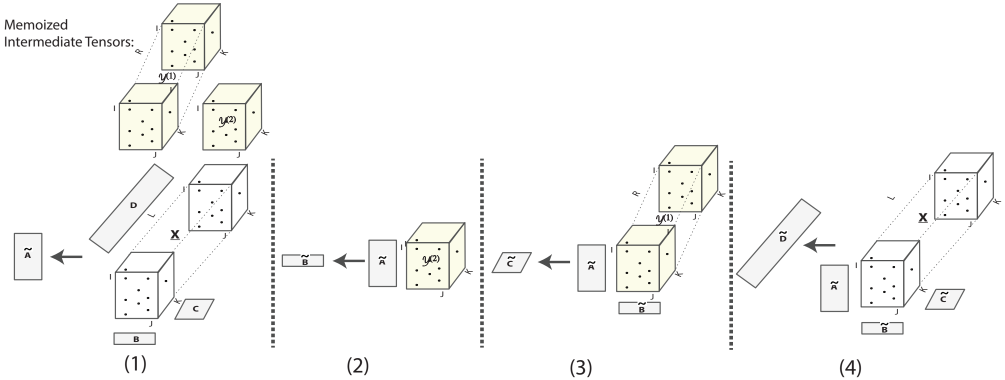
One scheme to save flops at the cost of increasing intermediate storage.

Solution A



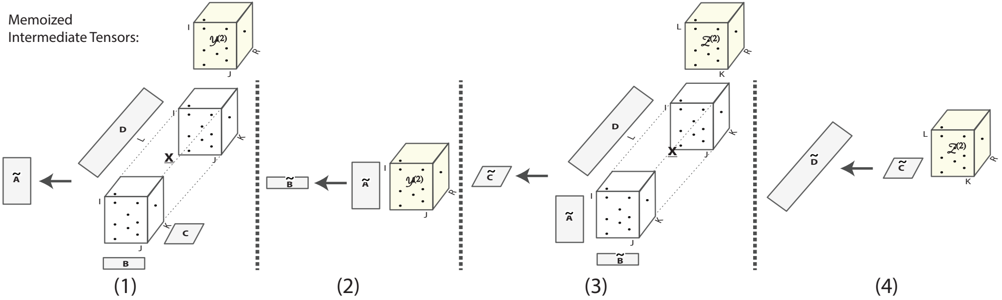
One scheme to save flops at the cost of increasing intermediate storage.

Solution A



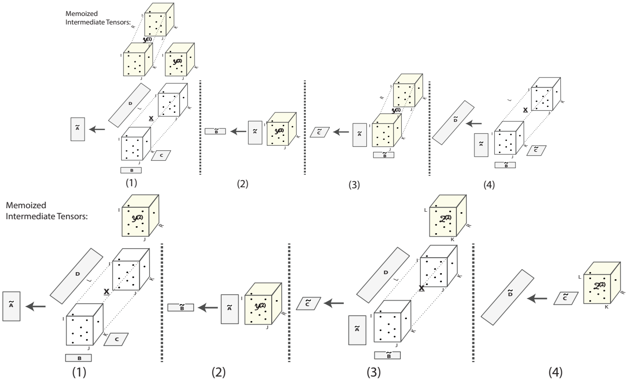
One scheme to save flops at the cost of increasing intermediate storage.

Solution B



Another scheme to save flops but minimize the amount of storage.

Overview



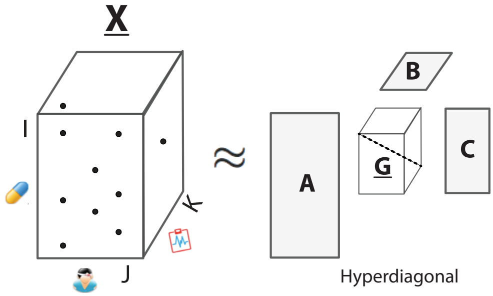
- Parameterize our algorithm
- A flexible tradeoff of storage for time
- Build a model-driven framework (AdaTM).

Outline

- Background
- Motivation
- Properties and Formats of Sparse Tensors
- Adaptive Tensor Memoization (AdaTM)
- Experiments
- Conclusion

Tensors

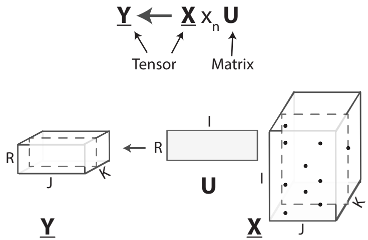
- Tensors, multi-way arrays, provide a natural way to represent multidimensional data.
  - Special cases: matrices (**U**) - 2D tensors, vectors (**x**) - 1D tensors.
  - Tensor mode (N): also called dimension or order.
- A sparse tensor, a tensor consisting mostly of zero entries, widely exist in real applications.
- Tensor analysis is usually factorizing a tensor into interpretable components.
  - E.g. CP decomposition, where MTTKRP is a critical computational kernel.



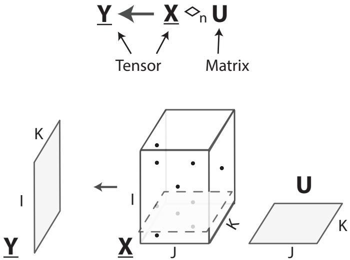
A 3D CP decomposition on a sparse tensor from healthcare data.

Basic Tensor Operations

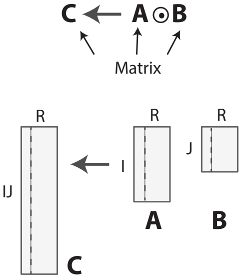
Tensor-Times-Matrix Multiply (TTM)



Quasi-TTM (q-TTM)

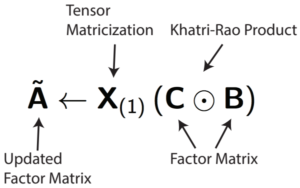


Khatri-Rao Product



Matriced Tensor Times Khatri-Rao Product (MTTKRP)

- Matriced Tensor Times Khatri-Rao Product (MTTKRP)

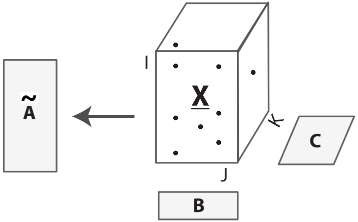


$\Leftrightarrow$

$\underline{\mathbf{Y}} \leftarrow \underline{\mathbf{X}} \times_n \mathbf{C}$ TTM

$\tilde{\mathbf{A}} \leftarrow \underline{\mathbf{Y}} \diamond_n \mathbf{B}$ q-TTM

$$\tilde{A}(i,r) = \sum_{j=1}^{J} B(j,r) \sum_{k=1}^{K} \underline{\mathbf{X}}(i,j,k) C(k,r).$$



Proposed by Smith et al. in SPLATT (IPDPS'15).

CP Decomposition

Input: An N^{th}-order sparse tensor $\underline{\mathbf{X}} \in R^{I \times \cdots \times I}$ and an integer rank R;
Output: Dense factors $\mathbf{A}^{(1)}, \ldots, \mathbf{A}^{(N)}$, $\mathbf{A}^{(i)} \in R^{I \times R}$ and weights λ;
1: Initialize $\mathbf{A}^{(1)}, \ldots, \mathbf{A}^{(N)}$;
2: **do**
3: **for** $n = 1, \ldots, N$ **do**
4: $\mathbf{V} \leftarrow \mathbf{A}^{(1)\dagger}\mathbf{A}^{(1)} * \cdots \mathbf{A}^{(n-1)\dagger}\mathbf{A}^{(n-1)} * \mathbf{A}^{(n+1)\dagger}\mathbf{A}^{(n+1)} * \cdots * \mathbf{A}^{(N)\dagger}\mathbf{A}^{(N)}$;
5: $\tilde{\mathbf{A}}^{(n)} \leftarrow \mathbf{X}_{(n)}(\mathbf{A}^{(N)} \odot \cdots \odot \mathbf{A}^{(n+1)} \odot \mathbf{A}^{(n-1)} \odot \cdots \odot \mathbf{A}^{(1)})$;
6: $\mathbf{A}^{(n)} \leftarrow \tilde{\mathbf{A}}^{(n)}\mathbf{V}^{\dagger}$;
7: Normalize columns of $A^{(n)}$ and store the norms as λ;
8: **end for**
9: **while** Fit ceases to improve or maximum iterations exhausted.
10: **Return:** $[\![\lambda, \mathbf{A}^{(1)}, \ldots, \mathbf{A}^{(N)}]\!]$;

MTTKRP is the performance bottleneck.

$T_{CP} \approx N(N^{\epsilon} mR + NIR^2) \approx NT_M$,
$IR \ll m$,
where $T_M = \mathcal{O}(N^{\epsilon} mR)$, $\epsilon \in (0, 1]$ is the time for a single MTTKRP.

Motivation

$$\tilde{\mathbf{A}} \leftarrow \mathbf{X}_{(1)} \left(\mathbf{D} \odot \mathbf{C} \odot \mathbf{B}\right) \Leftrightarrow \begin{cases} \underline{\mathbf{Y}}^{(1)} = \underline{\mathbf{X}} \times_4 \mathbf{D}; \\ \underline{\mathbf{Y}}^{(2)} = \underline{\mathbf{Y}}^{(1)} \diamond_3 \mathbf{C}; \\ \tilde{\mathbf{A}} = \underline{\mathbf{Y}}^{(2)} \diamond_2 \mathbf{B}; \end{cases}$$

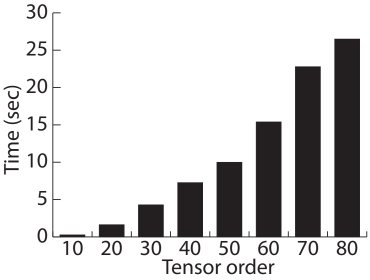
$$\tilde{\mathbf{B}} \leftarrow \mathbf{X}_{(2)} \left(\mathbf{D} \odot \mathbf{C} \odot \tilde{\mathbf{A}}\right) \Leftrightarrow \begin{cases} \underline{\mathbf{Y}}^{(1)} = \underline{\mathbf{X}} \times_4 \mathbf{D}; \\ \underline{\mathbf{Y}}^{(2)} = \underline{\mathbf{Y}}^{(1)} \diamond_3 \mathbf{C}; \\ \tilde{\mathbf{B}} = \underline{\mathbf{Y}}^{(2)} \diamond_1 \tilde{\mathbf{A}}; \end{cases}$$

An MTTKRP sequence has arithmetic redundancy.

Motivation

$$\tilde{\mathbf{A}} \leftarrow \mathbf{X}_{(1)} \left(\mathbf{D} \odot \mathbf{C} \odot \mathbf{B}\right) \Leftrightarrow \begin{cases} \underline{\mathbf{Y}}^{(1)} = \underline{\mathbf{X}} \times_4 \mathbf{D}; \\ \underline{\mathbf{Y}}^{(2)} = \underline{\mathbf{Y}}^{(1)} \diamond_3 \mathbf{C}; \\ \tilde{\mathbf{A}} = \underline{\mathbf{Y}}^{(2)} \diamond_2 \mathbf{B}; \end{cases}$$

$$\tilde{\mathbf{B}} \leftarrow \mathbf{X}_{(2)} \left(\mathbf{D} \odot \mathbf{C} \odot \tilde{\mathbf{A}}\right) \Leftrightarrow \begin{cases} \underline{\mathbf{Y}}^{(1)} = \underline{\mathbf{X}} \times_4 \mathbf{D}; \\ \underline{\mathbf{Y}}^{(2)} = \underline{\mathbf{Y}}^{(1)} \diamond_3 \mathbf{C}; \\ \tilde{\mathbf{B}} = \underline{\mathbf{Y}}^{(2)} \diamond_1 \tilde{\mathbf{A}}; \end{cases}$$



Synthetic, hypercubical, sparse tensors with $m = 100000$, $I = 1000$, $R = 16$.

An MTTKRP sequence has arithmetic redundancy.

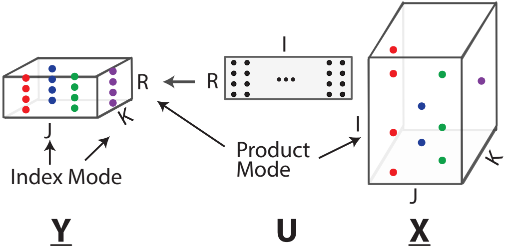
The time of an MTTKRP sequence grows with tensor order.

Special properties of Sparse TTM and q-TTM

Sparse TTM

Sparse TTM outputs a semi-sparse tensor:

- Its product mode becomes dense;
- Its index modes are unchanged.

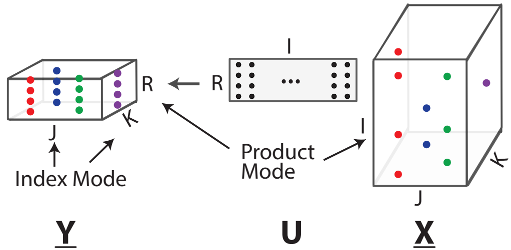


Special properties of Sparse TTM and q-TTM

Sparse TTM

Sparse TTM outputs a semi-sparse tensor:

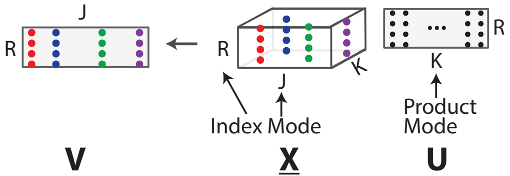
- Its product mode becomes dense;
- Its index modes are unchanged.



Sparse q-TTM

The q-TTM of a semi-sparse tensor and a dense matrix yields another semi-sparse tensor:

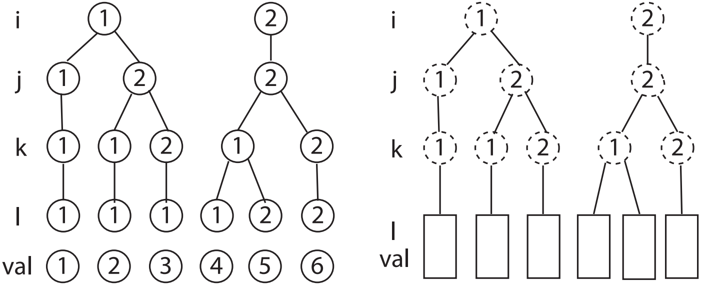
- Its index modes are unchanged;
- Its product mode disappears.



Tensor Formats

i	j	k	l	val
1	1	1	1	1
1	2	1	1	2
1	2	2	1	3
2	2	1	1	4
2	2	1	2	5
2	2	2	2	6

(a) COO



(b) CSF

(c) vCSF

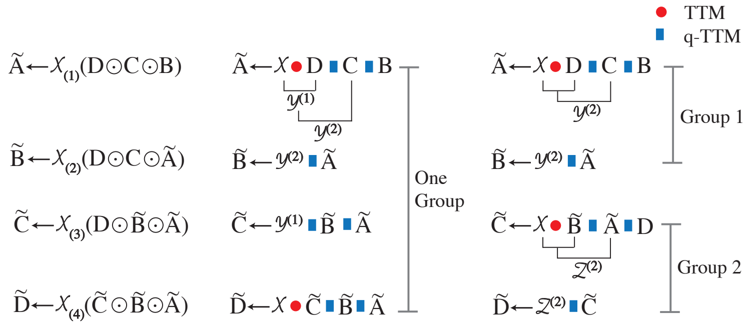
vCSF

- The dashed indices are not actually stored, but reuse the indices in CSF tree [Smith et al].
- vCSF is associated with CSF format.

Adaptive Tensor Memoization (AdaTM)

- Two example 4-D tensor memoization algorithms.
- MTTKRP sequence analysis.
- Adaptive Tensor Memoization (AdaTM)
  - The model-driven framework
  - Parameter selection
  - Predictive model
  - Parallelism

Two example 4-D tensor memoization algorithms



(a) Traditional 4th-order MTTKRP sequence

(b) Simple TM algorithm

(c) Optimal TM algorithm

Comparison:

Storage: $\underline{\mathbf{Y}}^{(1)} + \underline{\mathbf{Y}}^{(2)}$ vs $\underline{\mathbf{Y}}^{(2)} + \underline{\mathbf{Z}}^{(2)} +$ permuted $\underline{\mathbf{X}}$. – Depend on the input sparse tensor.

#Products: 9 vs 8.

Performance Analysis of an MTTKRP sequence

Problem: *Find the number of memoized* MTTKRPs n_p^* *that minimizes the total number of products (*TTM *and q-*TTM*)* n_O *in an* N^{th}*-order* MTTKRP *sequence, given infinite storage space.*

Suppose the input tensor $\underline{\mathbf{X}} \in \mathbb{R}^{I \times \cdots \times I}$ is hypercubical and dense,

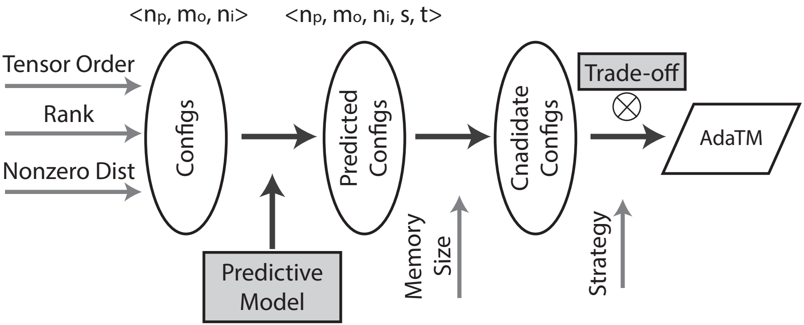
Lemma

$n_p^* = \sqrt{N/2}$ *minimizes the number of products* n_O *for an* N^{th}*-order* MTTKRP *sequence.*

$$\begin{cases} n_p = 1, & n_O = N(N+1)/2 = \mathcal{O}(N^2) \\ n_p = N/2, & n_O = N^2/2 = \mathcal{O}(N^2) \\ n_p = n_p^*, & n_O = n_O^* = \mathcal{O}(N^{1.5}) \end{cases}$$

which is asymptotically better for higher-order tensors.

The model-driven framework



Parameter selection

- $\mathbf{n_p}$: The number of producer modes, with one per memoized MTTKRP. Its range is $n_p \in \{1, \ldots, \sqrt{N/2}\}$.
- $\mathbf{m_o}$: The order of modes of each sparse tensor.
- $\mathbf{n_i}$: The number of intermediate semi-sparse tensors saved from each memoized MTTKRP. Its range is $\{1, \ldots, N/n_p - 1\}$.

For any choice of preceding parameters, we have a model that estimates the storage $s(n_p, m_o, n_i)$ in bytes and time $t(n_p, m_o, n_i)$ in flops

Predictive model

$$t = 2\sum_{i=1}^{n_p}\left(\sum_{l=2}^{N} m_l R + \sum_{l=1}^{\frac{N}{n_p}-1}\sum_{j=2}^{l+1} m_j\right) R \triangleq 2\tilde{N}mR; \quad s = \sum_{i=1}^{n_p}\left(m^i_{CSF} + 8\sum_{l=\frac{N}{n_p}-n_i+1}^{\frac{N}{n_p}} m_l R\right).$$

Algorithms		#Flops	Tensor Storage Space (Bytes)
Product	TTM	$2mR$	m_{CSF}
	q-TTM	$2mR$	$8m$
One MTTKRP group	Memoized MTTKRP	$2\sum_{l=2}^{N} m_l R$	$m_{CSF} + 8\sum_{l=\frac{N}{n_p}-n_i+1}^{\frac{N}{n_p}} m_l R$
	Partial MTTKRPS	$2\sum_{l=1}^{\frac{N}{n_p}-1}\sum_{j=2}^{l+1} m_j R$	-
MTTKRP sequence	AdaTM	$2\sum_{i=1}^{n_p}\left(\sum_{l=2}^{N} m_l R + \sum_{l=1}^{\frac{N}{n_p}-1}\sum_{j=2}^{l+1} m_j\right) R$	$\sum_{i=1}^{n_p}\left(m^i_{CSF} + 8\sum_{l=\frac{N}{n_p}-n_i+1}^{\frac{N}{n_p}} m_l R\right)$
	SPLATT	$2NmR$	m_{CSF}

Indices and values use "uint64_t" and "double" respectively. m_l is the number of fibers at the l^{th}-level of a CSF tree, $m_{CSF} = 16\sum_{l=1}^{N} m_l$.

Platforms and Datasets

Experimental Platforms Configuration

Parameters	Intel Core i7-4770K	Intel Xeon E7-4820
Microarchitecture	Haswell	Westmere
Frequency	3.5 GHz	2.0 GHz
#Physical cores	4	16
Memory size	32 GiB	512 GB
Memory bandwidth	25.6 GB/s	34.2 GB/s
Compiler	gcc 4.7.3	gcc 4.4.7

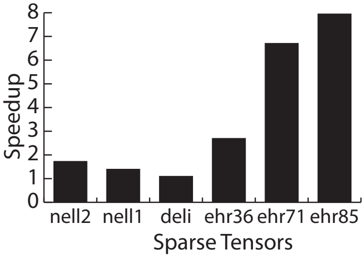
Sparse tensors

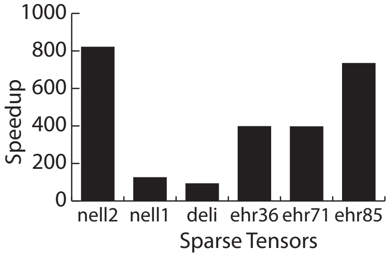
Dataset	Order	Max Mode size	NNZ	Density
nell2	3	30K	77M	1.3e-05
nell1	3	25M	144M	3.1e-13
deli	3	17M	140M	6.1e-12
ehr36	36	19	11K	4.7e-26
ehr71	71	21	221K	1.4e-55
ehr85	85	21	920K	7.9e-68

Tensor source:

- Never Ending Language Learning (NELL) project, "nell1, nell2 with noun-verb-noun".
- Data crawled from tagging systems, "deli with user-item-tag".
- Electronic Health Records (EHR) by considering a specific group of similar diseases as one mode and the co-occurrence counts of different diagnoses as values to build the higher-order tensors.

Performance





SPLATT [Smith et al.]

Multi-threaded, using CSF format.

Tensor Toolbox [Bader and Kolda]

Sequential, using COO format.

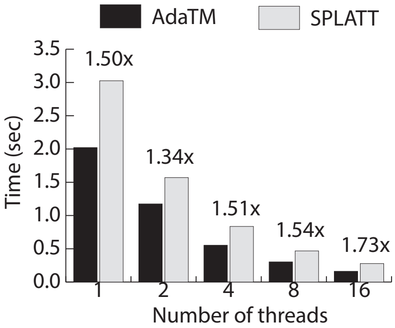
Storage

	Storage Space (MBytes)			Ratios	
Dataset	COO	CSF	CSF+vCSF	/CSF	/COO
nell2	2290	2540	2581	102%	113%
nell1	4280	6430	8510	132%	199%
deli	4180	5570	11090	199%	265%
ehr36	3.04	1.94	7.97	411%	262%
ehr71	121	62	205	333%	169%
ehr85	604	200	470	236%	78%

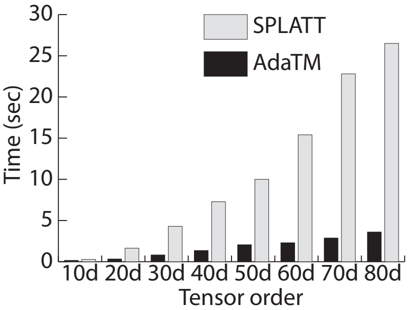
Storage range:

- /CSF: 1-4×;
- /COO: 0.8-2.7×

Scalability



Tensor *nell2*.

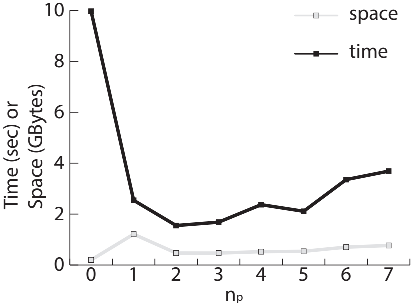


Synthetic sparse tensors.

Comparable multi-threading Scalability

Better scalability in dimensionality.

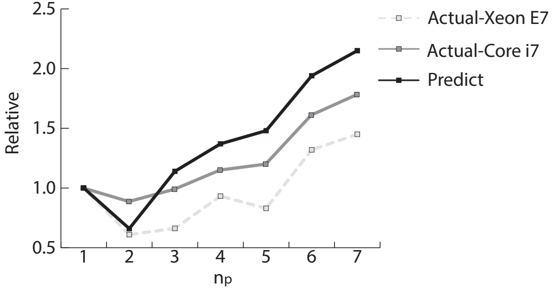
Model Analysis



Tensor *ehr85*.

$n_p^* = 2$

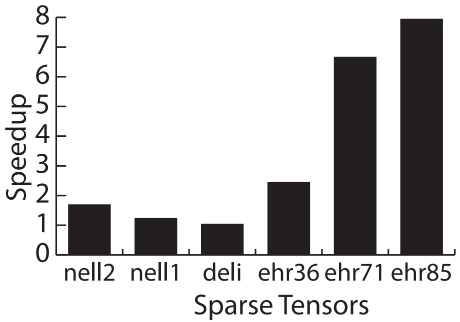
Space: 236% of SPLATT's;
Performance: 6.4× speedup.



Tensor *ehr85*.

Acceptable prediction.

CPD Application



The speedup of AdaTM over Splatt on CP-ALS.

Conclusion

Summary

- We consider the MTTKRP sequence as it arises in the context of CPD.
- We identify a memoization technique that permits a gradual tradeoff of storage for time.
- We parameterize our algorithm and build a model-driven and user-guided framework for it.

Future

- Apply our adaptive tensor memoization algorithm to other tensor decompositions;
- We also believe a closer inspection of not just the arithmetic but also communication properties of our method coupled with more architecture-specific tuning are ripe opportunities.

Source code: `https://github.com/hpcgarage/AdaTM`.

References

- B. W. Bader and T. G. Kolda. Efficient MATLAB computations with sparse and factored tensors, SIAM Journal on Scientific Computing 30(1):205-231, December 2007.
- S. Smith, N. Ravindran, N. Sidiropoulos, and G. Karypis, "Splatt: Efficient and parallel sparse tensor-matrix multiplication," IPDPS, 2015.
- O. Kaya and B. Ucar, "Scalable sparse tensor decompositions in distributed memory systems," SC'15. New York, NY, USA: ACM, 2015, pp. 77:1–77:11.
- O. Kaya and B. Uar, "High performance parallel algorithms for the tucker decomposition of sparse tensors," ICPP, Aug 2016, pp. 103–112.
- J. H. Choi and S. Vishwanathan, "Dfacto: Distributed factorization of tensors," NIPS, 2014, pp. 1296–1304.
- M. Baskaran, B. Meister, N. Vasilache, and R. Lethin, "Efficient and scalable computations with sparse tensors," HPEC, Sept 2012, pp. 1–6.
- ... and so on