HaTen2: Billion-scale Tensor Decompositions - Supplementary Document

Abstract—**In this supplementary document, we give additional preliminaries, analyses, and discoveries all of which supplement the main paper.**

I. Preliminaries: Tensor

Tensor. Tensor is a multi-dimensional array. An N-dimensional tensor is denoted by $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$.

Fibers and Slices. A fiber is defined by fixing all indices but one index. In 3-way tensor, It is denoted by $\mathbf{x}_{:jk}, \mathbf{x}_{i:k}$, and $\mathbf{x}_{ij:}$. A slice is defined by fixing all indices but two indices. In 3-way tensor, it is denoted by $\mathbf{X}_{i::}, \mathbf{X}_{:j:}$ and $\mathbf{X}_{::k}$.

Matricization of tensor. The mode-n matricization of a tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ is denoted by $\mathbf{X}_{(n)} \in \mathbb{R}^{I_n \times (\prod_{k \neq n} I_k)}$ and arranges the mode-n fibers to be the columns of the resulting matrix.

n**-mode matrix product.** The n-mode matrix product of a tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ with a matrix $\mathbf{U} \in \mathbb{R}^{J \times I_n}$ is denoted by $\mathcal{X} \times_n \mathbf{U}$ and is of size $I_1 \times \cdots I_{n-1} \times J \times I_{n+1} \cdots \times I_N$. It is defined as

$$(\mathcal{X} \times_n \mathbf{U})_{i_1 \dots i_{n-1} j i_{n+1} \dots i_N} = \sum_{i_n=1}^{I_n} x_{i_1 i_2 \dots i_N} u_{j i_n}.$$

n**-mode vector product.** The n-mode vector product of a tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ with a vector $\mathbf{v} \in \mathbb{R}^{I_n}$ is denoted by $\mathcal{X} \bar{\times}_n \mathbf{v}$ and is of size $I_1 \times \cdots I_{n-1} \times I_{n+1} \cdots \times I_N$. It is defined as

$$(\mathcal{X} \bar{\times}_n \mathbf{v})_{i_1 \dots i_{n-1} i_{n+1} \dots i_N} = \sum_{i_n=1}^{I_n} x_{i_1 i_2 \dots i_N} v_{i_n}.$$

Kronecker product. The Kronecker product of matrices $\mathbf{A} \in \mathbb{R}^{I \times J}$ and $\mathbf{B} \in \mathbb{R}^{K \times L}$ is denoted by $\mathbf{A} \otimes \mathbf{B}$. The result is a matrix of size $(IK) \times (JL)$ and defined by

$$\mathbf{A} \otimes \mathbf{B} = \begin{bmatrix} \mathbf{a}_{11}\mathbf{B} & \mathbf{a}_{12}\mathbf{B} & \cdots & \mathbf{a}_{1J}\mathbf{B} \\ \mathbf{a}_{21}\mathbf{B} & \mathbf{a}_{22}\mathbf{B} & \cdots & \mathbf{a}_{2J}\mathbf{B} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{a}_{I1}\mathbf{B} & \mathbf{a}_{I2}\mathbf{B} & \cdots & \mathbf{a}_{IJ}\mathbf{B} \end{bmatrix}$$

$$= \begin{bmatrix} \mathbf{a}_1 \otimes \mathbf{b}_1 & \mathbf{a}_1 \otimes \mathbf{b}_2 & \mathbf{a}_1 \otimes \mathbf{b}_3 & \cdots & \mathbf{a}_J \otimes \mathbf{b}_{L-1} & \mathbf{a}_J \otimes \mathbf{b}_L \end{bmatrix}$$

Khatri-Rao product. The Khatri-Rao product (or column-wise Kronecker product) $(\mathbf{A} \odot \mathbf{B})$, where $\mathbf{A}, \mathbf{B}$ have the same number of columns, say R, is defined as:

$$\mathbf{A} \odot \mathbf{B} = \begin{bmatrix} \mathbf{A}(:,1) \otimes \mathbf{B}(:,1) \cdots \mathbf{A}(:,R) \otimes \mathbf{B}(:,R) \end{bmatrix}$$

If $\mathbf{A}$ is of size $I \times R$ and $\mathbf{B}$ is of size $J \times R$ then $(\mathbf{A} \odot \mathbf{B})$ is of size $IJ \times R$.

Hadamard product. The Hadamard product $\mathbf{A} * \mathbf{B}$ is the elementwise matrix product, where $\mathbf{A}$ and $\mathbf{B}$ have the same size $(I \times J)$, and is defined as:

$$\mathbf{A} * \mathbf{B} = \begin{bmatrix} \mathbf{a}_{11}\mathbf{b}_{11} & \mathbf{a}_{12}\mathbf{b}_{12} & \cdots & \mathbf{a}_{1J}\mathbf{b}_{1J} \\ \mathbf{a}_{21}\mathbf{b}_{21} & \mathbf{a}_{22}\mathbf{b}_{22} & \cdots & \mathbf{a}_{2J}\mathbf{b}_{2J} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{a}_{I1}\mathbf{b}_{I1} & \mathbf{a}_{I2}\mathbf{b}_{I2} & \cdots & \mathbf{a}_{IJ}\mathbf{b}_{IJ} \end{bmatrix}$$

II. Analysis

We provide additional analysis results on HaTen2.

A. Cost Comparison

We compare the costs of the steps of all HaTen2 methods in terms of the intermediate data size, the number of MapReduce jobs, and the number of the floating point operations in Tables I and II. Note that HaTen2-Tucker-DRI which contains all the proposed ideas, has the minimum intermediate data size. In PARAFAC, the intermediate data size of HaTen2-PARAFAC-DNN seems smaller than that of HaTen2-PARAFAC-DRI. However, HaTen2-PARAFAC-DNN has lower scalability than HaTen2-PARAFAC-DRI because of the skewness of $\mathcal{T}_r$, which may lead to out of memory in the corresponding machine. Since $\mathcal{T}_r$, the result of multiplication of a sparse tensor and a fully-dense matrix, is dense, there is high probability of skewness when computing $\mathcal{T}_r \bar{*}_3 \mathbf{c}_r^T$ in HaTen2-PARAFAC-DNN. In contrast, HaTen2-PARAFAC-DRI scales well by exploiting the sparsity of real-world tensors with the idea in Section III-B3. For both decompositions, HaTen2-DRI has the minimum number of jobs.

B. Equivalence of Operations

We give proofs to the equivalences of operations in Tucker and PARAFAC. In the following, $bin(\mathcal{X})$ is a function which converts the non-zero elements of a tensor $\mathcal{X}$ to 1.

Lemma 1 (CrossMerge): Given $\mathcal{X} \in \mathbb{R}^{I \times J \times K}$, $\mathbf{B} \in \mathbb{R}^{J \times Q}$, and $\mathbf{C} \in \mathbb{R}^{K \times R}$,

$$\mathcal{X} \times_2 \mathbf{B}^T \times_3 \mathbf{C}^T \Leftrightarrow CrossMerge(\mathcal{T}', \mathcal{T}'')_{(1)}$$

where $\mathcal{T}' \in \mathbb{R}^{I \times J \times K \times Q}$ is a tensor whose qth subtensor $\mathcal{T}'_{:::q}$ is given by $\mathcal{X} \bar{*}_2 \mathbf{b}_q^T$, and $\mathcal{T}'' \in \mathbb{R}^{I \times J \times K \times R}$ is a tensor whose rth subtensor $\mathcal{T}'_{:::r}$ is given by $bin(\mathcal{X}) \bar{*}_3 \mathbf{c}_r^T$.

Proof:

TABLE I: Summary of costs in the steps of all methods for computing $\mathcal{X} \times_2 \mathbf{B} \times_3 \mathbf{C}$ in Tucker decomposition. $\mathcal{T}$ denotes $\mathcal{X} \times_2 \mathbf{B}$.

Method	Step	Intermediate Data	Jobs
HATEN2-Tucker-Naive	$\mathcal{X}\bar{\times}_2\mathbf{b}_q^T$	$nnz(\mathcal{X}) + IJK$	Q
	$\mathcal{T}\bar{\times}_3\mathbf{c}_r^T$	$nnz(\mathcal{T}) + IQK$	R
HATEN2-Tucker-DNN	$\mathcal{X}\bar{*}_2\mathbf{b}_q^T$	$nnz(\mathcal{X}) + J$	Q
	$Collapse$	$nnz(\mathcal{X})Q$	1
	$\mathcal{T}\bar{*}_3\mathbf{c}_r^T$	$nnz(\mathcal{T}) + K$	R
	$Collapse$	$nnz(\mathcal{T})R$	1
HATEN2-Tucker-DRN	$\mathcal{X}\bar{*}_2\mathbf{b}_q^T$	$nnz(\mathcal{X}) + J$	Q
	$bin(\mathcal{X})\bar{*}_3\mathbf{c}_r^T$	$nnz(\mathcal{X}) + K$	R
	$CrossMerge$	$nnz(\mathcal{X})Q + nnz(\mathcal{X})R$	1
HATEN2-Tucker-DRI	$IMHP$	$2nnz(\mathcal{X}) + JQ + KR$	1
	$CrossMerge$	$nnz(\mathcal{X})Q + nnz(\mathcal{X})R$	1

TABLE II: Summary of costs in the steps of all methods for computing $\mathbf{X}_{(1)}(\mathbf{C} \odot \mathbf{B})$ in PARAFAC decomposition. $\mathcal{T}_r$ denotes $\mathcal{X}\bar{\times}_2\mathbf{b}_r^T$.

Method	Step	Intermediate Data	Jobs
HATEN2-PARAFAC-Naive	$\mathcal{X}\bar{\times}_2\mathbf{b}_r^T$	$nnz(\mathcal{X}) + IJK$	R
	$\mathcal{T}_r\bar{\times}_3\mathbf{c}_r^T$	$nnz(\mathcal{T}_r) + IK$	R
HATEN2-PARAFAC-DNN	$\mathcal{X}\bar{*}_2\mathbf{b}_r^T$	$nnz(\mathcal{X}) + J$	R
	$Collapse$	$nnz(\mathcal{X})$	R
	$\mathcal{T}\bar{*}_3\mathbf{c}_r^T$	$nnz(\mathcal{T}_r) + K$	R
	$Collapse$	$nnz(\mathcal{T}_r)$	R
HATEN2-PARAFAC-DRN	$\mathcal{X}\bar{*}_2\mathbf{b}_r^T$	$nnz(\mathcal{X}) + J$	R
	$bin(\mathcal{X})\bar{*}_3\mathbf{c}_r^T$	$nnz(\mathcal{X}) + K$	R
	$PairwiseMerge$	$2nnz(\mathcal{X})R$	1
HATEN2-PARAFAC-DRI	$IMHP$	$2nnz(\mathcal{X}) + JR + KR$	1
	$PairwiseMerge$	$2nnz(\mathcal{X})R$	1

The (i,q,k)-th element $\mathcal{M}_{iqk}$ of $\mathcal{M} = \mathcal{X} \times_2 \mathbf{B}^T$ is given by

$$\mathcal{M}_{iqk} = \sum_{j=1}^{J} \mathcal{X}(i,j,k)\mathbf{B}(j,q).$$

Then the (i,q,r)-th element of $(\mathcal{X} \times_2 \mathbf{B}^T) \times_3 \mathbf{C}^T$ is

$$\sum_{k=1}^{K} \mathcal{M}(i,q,k)\mathbf{C}(k,r)$$

$$= \sum_{k=1}^{K}(\sum_{j=1}^{J} \mathcal{X}(i,j,k)\mathbf{B}(j,q))\mathbf{C}(k,r)$$

$$= \sum_{(j,k)=(1,1)}^{(J,K)} \mathcal{X}(i,j,k)\mathbf{B}(j,q)\mathbf{C}(k,r) \quad (1)$$

The (i,j,k,q)-th element of subtensor $\mathcal{T}'_{:::q}$ is given by

$$\mathcal{X}(i,j,k)\mathbf{b}_q^T(j),$$

and (i,j,k,r)-th element of subtensor $\mathcal{T}''_{:::r}$ is given by

$$(bin(\mathcal{X})(i,j,k))\mathbf{c}_r^T(k).$$

Therefore, the (i,j,k,q)-th element of $\mathcal{T}'$ is

$$\mathcal{T}'_{ijkq} = \mathcal{X}(i,j,k)\mathbf{B}(j,q),$$

and the (i,j,k,r)-th element of $\mathcal{T}''$ is

$$\mathcal{T}''_{ijkr} = (bin(\mathcal{X})(i,j,k))\mathbf{C}(k,r).$$

The (i,q,r)-th element of $CrossMerge(\mathcal{T}', \mathcal{T}'')_{(1)}$ is

$$\sum_{(j,k)=(1,1)}^{(J,K)} \mathcal{T}'(i,j,k,q)\mathcal{T}''(i,j,k,r).$$

$$= \sum_{(j,k)=(1,1)}^{(J,K)} \mathcal{X}(i,j,k)\mathbf{B}(j,q)(bin(\mathcal{X})(i,j,k))\mathbf{C}(k,r).$$

Since $\mathcal{X}(i,j,k) \times (bin(\mathcal{X})(i,j,k)) = \mathcal{X}(i,j,k)$,

$$= \sum_{(j,k)=(1,1)}^{(J,K)} \mathcal{X}(i,j,k)\mathbf{B}(j,q)\mathbf{C}(k,r) \quad (2)$$

The equation (1) for (i,q,r)-th element of $\mathcal{X} \times_2 \mathbf{B}^T \times_3 \mathbf{C}^T$ is exactly the same as the equation (2) for (i,q,r)-th element of $CrossMerge(\mathcal{T}', \mathcal{T}'')_{(1)}$.

■

Lemma 2 (PairwiseMerge): Given $\mathcal{X} \in \mathbb{R}^{I \times J \times K}$, $\mathbf{B} \in \mathbb{R}^{J \times R}$, and $\mathbf{C} \in \mathbb{R}^{K \times R}$,

$$\mathbf{X}_{(1)}(\mathbf{C} \odot \mathbf{B}) \Leftrightarrow PairwiseMerge(\mathcal{F}', \mathcal{T}'')_{(1)}$$

where $\mathcal{F}' \in \mathbb{R}^{I \times J \times K \times R}$ is a tensor whose r-th subtensor $\mathcal{F}'_{:::r}$ is given by $\mathcal{X}\bar{*}_2\mathbf{b}_r^T$, and $\mathcal{T}'' \in \mathbb{R}^{I \times J \times K \times R}$ is a tensor whose r-th subtensor $\mathcal{T}'_{:::r}$ is given by $bin(\mathcal{X})\bar{*}_3\mathbf{c}_r^T$.

Proof:
The (i,r)-th element of $\mathbf{M} = \mathbf{X_{(1)}}(\mathbf{C} \odot \mathbf{B})$ is defined by

$$\mathbf{M}_{ir} = \sum_{(j,k)=(1,1)}^{(J,K)} \mathcal{X}(i,j,k)\mathbf{B}(j,r)\mathbf{C}(k,r) \quad (3)$$

The (i,j,k)-th element of subtensor $\mathcal{F'}_{::r}$ is given by

$$\mathcal{X}(i,j,k)\mathbf{b}_r^T(j),$$

and the (i,j,k)-th element of subtensor $\mathcal{T'}_{::r}$ is given by

$$(bin(\mathcal{X})(i,j,k))\mathbf{c}_r^T(k).$$

Therefore, the (i,j,k,r)-th element of $\mathcal{F'}$ is

$$\mathcal{F'}_{ijkr} = \mathcal{X}(i,j,k)\mathbf{B}(j,r),$$

and the (i,j,k,r)-th element of $\mathcal{T''}$ is

$$\mathcal{T''}_{ijkr} = (bin(\mathcal{X})(i,j,k))\mathbf{C}(k,r).$$

The (i,r)-th element of $PairwiseMerge(\mathcal{F'},\mathcal{T''})_{(1)}$ is

$$\sum_{(j,k)=(1,1)}^{(J,K)} \mathcal{F'}(i,j,k,r)\mathcal{T''}(i,j,k,r).$$

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

Since $\mathcal{X}(i,j,k) \times (bin(\mathcal{X})(i,j,k)) = \mathcal{X}(i,j,k)$,

$$= \sum_{(j,k)=(1,1)}^{(J,K)} \mathcal{X}(i,j,k)\mathbf{B}(j,r)\mathbf{C}(k,r) \cdots (2). \quad (4)$$

The equation (3) for (i,r)-th element of $\mathbf{X_{(1)}}(\mathbf{C} \odot \mathbf{B})$ is exactly the same as the equation (4) for (i,r)-th element of $PairwiseMerge(\mathcal{F'},\mathcal{T''})_{(1)}$.

■

III. DISCOVERY

Concept discovery on NELL data. NELL is a knowledge base dataset containing ('Noun Phrase 1', 'Noun Phrase 2', 'Context') triples from the 'Read the Web' project [1]. We filter the NELL data by removing entries whose values are below a threshold; the result is a tensor named NELL-2 whose size is $14545 \times 14545 \times 28818$ with 76 millions of nonzeros. We discover latent concept groups of NELL-2 by applying HATEN2-PARAFAC with rank 20, and HATEN2-Tucker with core tensor size $20 \times 20 \times 20$. Table III shows the concept discovery results from HATEN2-PARAFAC. We discovered several concepts: e.g., "Health Care System", "File Transfer", "Internet Service", and "Shopping". In PARAFAC decomposition, because the core tensor is diagonal, each 'Noun Phrase 1' group is combined only with a 'Noun Phrase 2' group and a 'Context' group. On the other hand, Tucker decomposition provides more diverse concepts compared with PARAFAC decomposition: e.g., a 'Noun Phrase 2' group may be combined with several 'Noun Phrase 1' groups and 'Context' groups. Table IV shows the groups in factors from Tucker decomposition: e.g., "Health", "Credit", "Network", "Algorithm", "Project", and "Information" in the 'Noun Phrase 1' mode. Table V shows the discovered concepts each of which combines the groups from the 'Noun Phrase 1', the 'Noun Phrase 2', and the 'Context' factors. The first concept represents "Health Care System" which contains the 'Noun Phrase 1' group S1 ("Health"), the 'Noun Phrase 2' group O2 ("Service"), and the 'Context' group C1 ("Care"). Note that a group of a factor appears in several concept groups in Tucker decomposition. For example, the 'Noun Phrase 2' group O2 appears in the first, the second, and the third concepts; the 'Context' group C6 appears in both the second and the third concepts.

TABLE III: Concept discovery result using HATEN2-PARAFAC on NELL-2 dataset.

Concepts	Noun Phrase1	Noun Phrase2	Context
Concept1: "Health Care System"	health child skin	providers systems organizations	'np1' 'care' 'np2' 'np1' 'insurance' 'np2' 'np1' 'and safety' 'np2'
Concept2: "File Transfer"	file hypertext FTP	protocol stack technology	'np1' 'stream' 'np2' 'np1' 'transfer' 'np2' 'np2' 'cable' 'np1'
Concept3: "Internet Service"	internet phone application	providers web sites roots	'np1' 'service' 'np2' 'np1' 'access' 'np2' 'np1' 'hosting' 'np2'
Concept4: "Shopping"	discount shop grocery	store service products	'np1' 'food' 'np2' 'np1' 'and nutrition' 'np2' 'np1' 'supplement' 'np2'

TABLE V: Concept discovery result using HATEN2-Tucker on NELL-2 dataset.

Concepts	Noun Phrase1	Noun Phrase2	Context
Concept1: (S1, O2, C1) "Health Care System"	health child skin	providers system professionals	'np1' 'care' 'np2' 'np1' 'insurance' 'np2' 'np1' 'service' 'np2'
Concept2: (S3, O2, C6) "Internet Service"	internet application email	providers system professionals	'np1' 'service' 'np2' 'np1' 'access' 'np2' 'np1' 'hosting' 'np2'
Concept3: (S6, O2, C6) "Information Access"	information details news	providers system professionals	'np2' 'service' 'np1' 'np2' 'access' 'np1' 'np2' 'hosting' 'np1'
Concept4: (S4, O3, C3) "Web Search Algorithm"	optimization rankings marketing	search website performance	'np2' 'engine' 'np1' 'np2' 'returned' 'np1' 'np2' 'results' 'np1'
Concept5: (S5, O4, C5) "Research Project Funding"	agency grants proposal	research training study	'np2' 'projects' 'np1' 'np2' 'funding' 'np1' 'np1' 'sponsoring' 'np2'

REFERENCES

[1] A. Carlson, J. Betteridge, B. Kisiel, B. Settles, E. R. H. Jr., and T. M. Mitchell, "Toward an architecture for never-ending language learning," in *AAAI*, 2010.

TABLE IV: Discovered factors from HATEN2-Tucker on NELL-2 dataset.

	NP S1: Health	**NP S2: Credit**	**NP S3: Network**	**NP S4: Algorithm**	**NP S5: Project**	**NP S6: Information**
Noun Phrase1	health child skin eye patient	credit charge bank ID account	internet phone email contact network	optimization rankings listings algorithms indexing	agency proposal management activities manager	information details news material pictures
	NP O1: Region	**NP O2: Service**	**NP O3: Web search**	**NP O4: Research**	**NP O5: Loan**	**NP O6: Network**
Noun Phrase2	world state planet region globe	providers system service insurance organization	search website page industry performance	research experience work training study	loan rates mortgage lender refinancing	roots speeds proxies ports routers
	Context C1: Care	**Context C2: Credit**	**Context C3: Function**	**Context C4: Transfer**	**Context C5: Support**	**Context C6: Service**
Context	'np1' 'care' 'np2' 'np1' 'insurance' 'np2' 'np1' 'service' 'np2' 'np1' 'safety' 'np2' 'np1' 'and fitness' 'np2'	'np1' 'card' 'np2' 'np1' 'report' 'np2' 'np2' 'management' 'np1' 'np1' 'account' 'np2' 'np1' 'debt' 'np2'	'np2' 'engine' 'np1' 'np2' 'returned' 'np1' 'np2' 'results' 'np1' 'np2' 'returns' 'np1' 'np2' 'machine' 'np1'	'np1' 'stream' 'np2' 'np1' 'transfer' 'np2' 'np1' 'communication' 'np2' 'np1' 'protocol' 'np2' 'np2' 'cable' 'np1'	'np2' 'project' 'np1' 'np2' 'and development' 'np1' 'np2 'funding' 'np1' 'np1' 'sponsoring' 'np2' 'np1' 'supporting' 'np2'	'np1' service' 'np2' 'np1 'access np2 'np1' 'hosting' 'np2' 'np1' 'broadband 'np2 'np1' 'infrastructure' 'np2'