A Additional Background

For higher-order tensors, we have similar notation definitions. The ith entry of a vector is denoted by a_i, element (i,j) of a matrix $\mathbf{A}$ is denoted by A_{ij}, and the element $(i_1,\ldots,i_k)$ of a k-dimensional tensor $\boldsymbol{\mathcal{T}} \in \mathbb{R}^{I_1\times\ldots\times I_k}$ is denoted by $\mathcal{T}_{i_1,\ldots,i_k}$. For notation simplicity, we assume that $(i_1,\ldots,i_k)$ also represents an index between 1 and $I_1 I_2 \cdots I_k$, which is location of the element with subscript $(i_1,\ldots,i_k)$.

Special Matrix Products

Our manipulation of tensors as matrices revolves around several matrix products. For a pair of matrices $\mathbf{A} \in \mathbb{R}^{I\times K}$ and $\mathbf{B} \in \mathbb{R}^{J\times L}$,

- the *Kronecker product* of matrices is denoted by $\mathbf{A}\otimes\mathbf{B} \in \mathbb{R}^{(IJ)\times(KL)}$, where the element $((i,j),(k,l))$ is $\mathbf{A}_{ik}\mathbf{B}_{jl}$;
- the *Khatri-Rao product* of matrices, $\mathbf{A}\odot\mathbf{B} \in \mathbb{R}^{(IJ)\times K}$ when $K = L$, has element $((i,j),k)$ as $\mathbf{A}_{ik}\mathbf{B}_{jk}$; and
- the *Hadamard product*, when $I = J$ and $K = L$, is the elementwise matrix product. It is denoted by $\mathbf{A} * \mathbf{B} \in \mathbb{R}^{I\times K}$, where the element (i,k) is $\mathbf{A}_{ik}\mathbf{B}_{ik}$.

More details on these products can be found in [21].

Tensor Matricization Here we consider only the case of mode-n matricization. For $n = 1,2,\ldots,k$, the mode-n matricization of a tensor $\boldsymbol{\mathcal{T}} \in \mathbb{R}^{I_1\times\ldots\times I_k}$ is denoted by $\mathbf{T}_{(n)} \in \mathbb{R}^{I_n\times\prod_{s\neq n} I_s}$, where the element $(i_n,(i_1,\ldots,i_{n-1},i_{n+1},\ldots,i_k))$ is $\mathcal{T}_{i_1,\ldots,i_k}$.

B Useful Identities

Lemma B.1.

$$(\mathbf{AU})\odot(\mathbf{BV}) = (\mathbf{A}\otimes\mathbf{B})(\mathbf{U}\odot\mathbf{V}). \tag{6}$$

Proof. Assume $\mathbf{A} \in \mathbb{R}^{I\times N}$, $\mathbf{B} \in \mathbb{R}^{J\times M}$, $\mathbf{U} \in \mathbb{R}^{N\times R}$, and $\mathbf{V} \in \mathbb{R}^{M\times R}$.

Then we have $(\mathbf{AU})\odot(\mathbf{BV}) \in \mathbb{R}^{IJ\times R}$ and the (ij,r)-th element is

$$\begin{aligned}
[(\mathbf{AU})\odot(\mathbf{BV})]_{ij,r} &= (\mathbf{AU})_{ir}(\mathbf{BV})_{jr} && (7)\\
&= (\sum_{p=1}^{N}\mathbf{A}_{ip}\mathbf{U}_{pr})(\sum_{q=1}^{M}\mathbf{B}_{jq}\mathbf{V}_{qr}) && (8)\\
&= \sum_{p=1}^{N}\sum_{q=1}^{M}\mathbf{A}_{ip}\mathbf{B}_{jq}\mathbf{U}_{pr}\mathbf{V}_{qr} && (9)\\
&= \sum_{p=1}^{N}\sum_{q=1}^{M}(\mathbf{A}\otimes\mathbf{B})_{ij,pq}(\mathbf{U}\odot\mathbf{V})_{pq,r} && (10)\\
&= [(\mathbf{A}\otimes\mathbf{B})(\mathbf{U}\odot\mathbf{V})]_{(ij,r)} && (11)
\end{aligned}$$

□

C Proof for Theorem 3.3

Theorem 3.3. *For matrices $\mathbf{A}^{(k)} \in \mathbb{R}^{I_k\times R}$ where $I_k > R$ for $k = 1,\ldots,K$, let $\tau_i^{(k)}$ be the statistical leverage score of the i-th row of $\mathbf{A}^{(k)}$. Then, for the $\prod_k I_k$-by-R matrix $\mathbf{A}^{(1)}\odot\mathbf{A}^{(2)}\odot\cdots\odot\mathbf{A}^{(K)}$ with statistical leverage score $\tau_{i_1,\ldots,i_K}$ for the row corresponding to $\tau_{i_1,\ldots,i_K}$, we have*

$$\tau^{1:K}_{i_1,\ldots,i_K} \leq \prod_{k=1}^{K}\tau^{(k)}_{i_k},$$

where $\tau^{1:K}_{i_1,\ldots,i_K}$ *denotes the statistical leverage score of the row of* $\mathbf{A}^{(1)} \odot \mathbf{A}^{(2)} \odot \cdots \odot \mathbf{A}^{(K)}$ *corresponding to the* i_k*-th row of* $\mathbf{A}^{(k)}$ *for* $k = 1, \ldots, K$.

Proof. Assume that the claim holds for $K-1$, that is, we have

$$\tau^{2:K}_{i_2,\ldots,i_K} \leq \prod_{k=2}^{K} \tau^{(k)}_{i_k}.$$

Let $\mathbf{A} = \mathbf{A}^{(1)}$ and let $\mathbf{B} = \mathbf{A}^{(2)} \odot \mathbf{A}^{(3)} \odot \cdots \odot \mathbf{A}^{(K)}$, then by Theorem 3.2, we have

$$\tau^{1:K}_{i_1,\ldots,i_K} = \tau^{\mathbf{A}\odot\mathbf{B}}_{i_1,(i_2,\ldots,i_K)} \leq \tau^{(1)}_{i_1} \tau^{2:K}_{i_2,\ldots,i_K} \leq \prod_{k=1}^{K} \tau^{(k)}_{i_k}.$$

Therefore, it also holds for K. By Theorem 3.2, it holds for $K = 2$. Then by induction, it holds for any $K \in \mathbb{Z}_+$. □

D Proof for Theorem 4.2

Theorem 4.2. *For matrix* $\mathbf{A} \in \mathbb{R}^{I\times M}$ *and matrix* $\mathbf{B} \in \mathbb{R}^{J\times N}$*, where* $I > M$ *and* $J > N$*, let* $\tau^{\mathbf{A}}_i$ *and* $\tau^{\mathbf{B}}_j$ *be the statistical leverage score of the* i*-th and* j*-th row of* $\mathbf{A}$ *and* $\mathbf{B}$*, respectively. Then, for matrix* $\mathbf{A} \otimes \mathbf{B} \in \mathbb{R}^{IJ\times MN}$ *with statistical leverage score* $\tau^{\mathbf{A}\otimes\mathbf{B}}_{i,j}$ *for the* $(iJ + j)$*-th row, we have* $\tau^{\mathbf{A}\otimes\mathbf{B}}_{i,j} = \tau^{\mathbf{A}}_i \tau^{\mathbf{B}}_j$.

Proof. Let the singular value decomposition of $\mathbf{A}$ and $\mathbf{B}$ be $\mathbf{A} = \mathbf{U}^a \mathbf{\Lambda}^a \mathbf{V}^{a\top}$ and $\mathbf{B} = \mathbf{U}^b \mathbf{\Lambda}^b \mathbf{V}^{b\top}$, where $\mathbf{U}^a \in \mathbb{R}^{I\times R}$, $\mathbf{U}^b \in \mathbb{R}^{J\times R}$, and $\mathbf{\Lambda}^a, \mathbf{\Lambda}^b, \mathbf{V}^a, \mathbf{V}^b \in \mathbb{R}^{R\times R}$.

By the definition of Khatri-Rao product from Section 2.6 of [21], we have that

$$\mathbf{A} \otimes \mathbf{B} = \left(\mathbf{U}^a \otimes \mathbf{U}^b\right) \left(\mathbf{\Lambda}^a \otimes \mathbf{\Lambda}^a\right) \left(\mathbf{U}^a \otimes \mathbf{U}^b\right)^\top \tag{12}$$

which is the legit SVD of $\mathbf{A} \otimes \mathbf{B}$. Therefore the leverage score of $\mathbf{A} \otimes \mathbf{B}$ can be computed from $\mathbf{U}_a \otimes \mathbf{U}_b$, that

$$\mathbf{H} = \left[\mathbf{U}_a \otimes \mathbf{U}_b\right] \left[\mathbf{U}_a \otimes \mathbf{U}_b\right]^\top \tag{13}$$

and for the index $k = iI + j$, we have

$$\begin{aligned}
\tau^{\mathbf{A}\odot\mathbf{B}}_{i,j} = \mathbf{H}_{k,k} =& \mathbf{e}_k^\top \mathbf{H} \mathbf{e}_k && (14)\\
=& \left\| \left[\mathbf{U}^a \otimes \mathbf{U}^b\right]^\top \mathbf{e}_k \right\|_2^2 && (15)\\
=& \sum_{p=1}^{R} \sum_{q=1}^{R} \left(\mathbf{U}^a_{i,p}\right)^2 \left(\mathbf{U}^b_{j,q}\right)^2 && (16)\\
=& \left(\sum_{p=1}^{R} \left(\mathbf{U}^a_{i,p}\right)^2\right) \left(\sum_{q=1}^{R} \left(\mathbf{U}^b_{j,q}\right)^2\right) && (17)\\
=& \tau^{\mathbf{A}}_i \tau^{\mathbf{B}}_j. && (18)
\end{aligned}$$

□

E Lemmas for Least Square Regression Approximation

For the least square regression problem

$$\min_{\mathbf{x}} \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2,$$

where design matrix $\mathbf{A} \in \mathbb{R}^{n \times p}$ with $n > p$ (assuming full column rank), coefficients $\mathbf{x} \in \mathbb{R}^{p \times 1}$ and response vector $\mathbf{b} \in \mathbb{R}^{n \times 1}$.

The optimal solution is

$$\mathbf{x}_{\text{opt}} = \left(\mathbf{A}^\top \mathbf{A}\right)^{-1} \mathbf{A}^\top \mathbf{b},$$

The percentage explained by $\mathbf{x}$ is $\rho(\mathbf{x})$:

$$\rho(\mathbf{x}) = 1 - \frac{\|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2}{\|\mathbf{b}\|_2^2}$$

The optimal result is

$$\rho(\mathbf{x}_{\text{opt}}) = \frac{\mathbf{b}^\top \mathbf{A} \left(\mathbf{A}^\top \mathbf{A}\right)^{-1} \mathbf{A}^\top \mathbf{b}}{\mathbf{b}^\top \mathbf{b}}$$

Lemma E.1. *Given* $\mathbf{S} \in \mathbb{R}^{m \times n}$*, such that*

$$\left\|\mathbf{U}^\top \mathbf{b} - \mathbf{U}^\top \mathbf{S}^\top \mathbf{S}\mathbf{b}\right\|_2 \leq \frac{\epsilon}{3} \|\mathbf{b}\|_2,$$

the solution $\mathbf{x}_{\mathbf{B}} = \left(\mathbf{A}^\top \mathbf{A}\right)^{-1} \mathbf{A}^\top \mathbf{S}^\top \mathbf{S}\mathbf{b}$*, satisfies that*

$$\rho(\mathbf{x}_{\mathbf{B}}) \geq \rho(\mathbf{x}_{opt}) - \epsilon.$$

Proof. Let the SVD of matrix $\mathbf{A}$ is of the form $\mathbf{A} = \mathbf{U}\mathbf{\Lambda}\mathbf{V}^\top$. We have that

$$\mathbf{x}_{\text{opt}} = \mathbf{V}\mathbf{\Lambda}^{-1}\mathbf{U}^\top \mathbf{b}$$

and

$$\mathbf{x}_{\mathbf{B}} = \mathbf{V}\mathbf{\Lambda}^{-1}\mathbf{U}^\top \mathbf{S}^\top \mathbf{S}\mathbf{b}$$

We have that

$$\rho(\mathbf{x}_{\text{opt}}) = 1 - \frac{\left\|\left(\mathbf{I} - \mathbf{U}\mathbf{U}^\top\right)\mathbf{b}\right\|_2^2}{\|\mathbf{b}\|_2^2}$$

Similarity, we have that

$$\begin{aligned}
\rho(\mathbf{x}_{\mathbf{B}}) =& 1 - \frac{\left\|\left(\mathbf{I} - \mathbf{U}\mathbf{U}^\top \mathbf{S}^\top \mathbf{S}\right)\mathbf{b}\right\|_2^2}{\|\mathbf{b}\|_2^2} \\
=& 1 - \frac{\left\|\left(\mathbf{I} - \mathbf{U}\mathbf{U}^\top + \mathbf{U}\mathbf{U}^\top - \mathbf{U}\mathbf{U}^\top \mathbf{S}^\top \mathbf{S}\right)\mathbf{b}\right\|_2^2}{\|\mathbf{b}\|_2^2} \\
\geq& \rho(\mathbf{x}_{\text{opt}}) \\
& - \left(2\frac{\left\|\left(\mathbf{I} - \mathbf{U}\mathbf{U}^\top\right)\mathbf{b}\right\|_2}{\|\mathbf{b}\|_2} + \frac{\left\|\left(\mathbf{U}\mathbf{U}^\top - \mathbf{U}\mathbf{U}^\top \mathbf{S}^\top \mathbf{S}\right)\mathbf{b}\right\|_2}{\|\mathbf{b}\|_2}\right) \\
& \frac{\left\|\left(\mathbf{U}\mathbf{U}^\top - \mathbf{U}\mathbf{U}^\top \mathbf{S}^\top \mathbf{S}\right)\mathbf{b}\right\|_2}{\|\mathbf{b}\|_2}
\end{aligned}$$

Note that

$$\left\|\mathbf{U}^\top \mathbf{b} - \mathbf{U}^\top \mathbf{S}^\top \mathbf{S}\mathbf{b}\right\|_2 \leq \frac{\epsilon}{3} \|\mathbf{b}\|_2,$$

and

$$2\frac{\left\|\left(\mathbf{I} - \mathbf{U}\mathbf{U}^\top\right)\mathbf{b}\right\|_2}{\|\mathbf{b}\|_2} + \frac{\left\|\left(\mathbf{U}\mathbf{U}^\top - \mathbf{U}\mathbf{U}^\top \mathbf{S}^\top \mathbf{S}\right)\mathbf{b}\right\|_2}{\|\mathbf{b}\|_2} \leq 3,$$

which concludes that

$$\rho(\mathbf{x}_{\mathbf{B}}) \geq \rho(\mathbf{x}_{\text{opt}}) - \epsilon.$$

□

Lemma E.2 (Matrix Multiplication Approximation [15]). *Given matrix* $\mathbf{A} \in \mathbb{R}^{m\times n}$ *and* $\mathbf{B} \in \mathbb{R}^{n\times p}$*, and sampling probability* $p_i, i = 1, 2, \ldots, n$ *satisfies that*

$$p_i \geq \beta \frac{\left\|\mathbf{A}^{(i)}\right\|_2 \left\|\mathbf{B}_{(i)}\right\|_2}{\|\mathbf{A}\|\|\mathbf{B}\|},$$

where $\mathbf{A}^{(i)}$ *and* $\mathbf{B}_{(i)}$ *is the* i*-th column and row of* $\mathbf{A}$ *and* $\mathbf{B}$*, respectively. Then with probability at least* $1 - \delta$*, the random matrix multiplication with* c *samples, we have that*

$$\|\mathbf{AB} - \mathbf{CD}\| \leq \frac{1 + \sqrt{(8/\delta)\log 1/\delta}}{\beta c}\|\mathbf{A}\|\|\mathbf{B}\|.$$

Proof. See Appendix A.3 in [15]. □

Corollary E.3. *Given matrix* $\mathbf{A} \in \mathbb{R}^{m\times n}$ *and* $\mathbf{B} \in \mathbb{R}^{n\times p}$*, and sampling probability* $p_i, i = 1, 2, \ldots, n$ *satisfies that*

$$p_i \geq \beta_1 \frac{\left\|\mathbf{A}^{(i)}\right\|_2^2}{\|\mathbf{A}\|^2}, \text{and}$$
$$p_i \geq \beta_2 \frac{\left\|\mathbf{B}^{(i)}\right\|_2^2}{\|\mathbf{B}\|^2}.$$

where $\mathbf{A}^{(i)}$ *and* $\mathbf{B}_{(i)}$ *is the* i*-th column and row of* $\mathbf{A}$ *and* $\mathbf{B}$*, respectively. Then with probability at least* $1 - \delta$*, the random matrix multiplication with* c *samples, we have that*

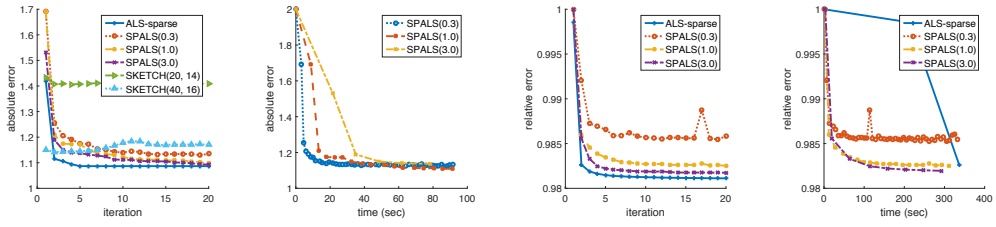
$$\|\mathbf{AB} - \mathbf{CD}\| \leq \frac{1 + \sqrt{(8/\delta)\log 1/\delta}}{\sqrt{\beta_1\beta_2}c}\|\mathbf{A}\|\|\mathbf{B}\|.$$

Proof. Note that

$$p_i \geq \frac{1}{2}\left(\beta_1 \frac{\left\|\mathbf{A}^{(i)}\right\|_2^2}{\|\mathbf{A}\|^2} + \beta_2 \frac{\left\|\mathbf{B}^{(i)}\right\|_2^2}{\|\mathbf{B}\|^2}\right) \geq \sqrt{\beta_1\beta_2}\frac{\left\|\mathbf{A}^{(i)}\right\|_2 \left\|\mathbf{B}_{(i)}\right\|_2}{\|\mathbf{A}\|\|\mathbf{B}\|}.$$

□

F Full Experimental Results



(a) (b) (c) (d)

Figure 1: (a) Absolute errors of various tensor factorization methods plotted against iteration count on a random high rank tensor with $nsr = 1$. (b) Error of SPALS with various rates plotted against running time in seconds on a random high rank tensor with $nsr = 1$. (c) Errors over iterations of deterministic and sampled variants of our sparse implementation on the sparse Amazon tensor. (d) Errors over time (seconds) of deterministic and sampled variants of our sparse implementation on the sparse Amazon tensor, ran with 16 threads.

We implemented and evaluated our algorithms in a single machine setting. Experiments are tested on a single machine with two Intel Xeon E5-2630 v3 CPU and 256GB memory. All methods are

implemented in C++ with OpenMP parallelization. As the algorithms are randomized, all numbers of randomized routines that we report are averages from 5 trials.

Due to the large size of the tensors involved, several implementation details were incorporated to facilitate testing. The dense/sparse tensors were preprocessed into bit format, and in the sparse case an index table was precomputed and stored along with the input data. In the sparse case, parallelism was also incorporated to facilitate testing, since the comparisons are between variants of our ALS implementations. As the implementation does not optimize cache performances, the costs of the ALS iterations on $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$ are uneven, and differ by factors of up to 4 in some cases. The parallel speedup with 16 threads are only about 8. We believe these are less crucial than studying the convergence properties of SPALS, and will discuss ways of address them in future works in Section 7.

F.1 Dense Synthetic Tensors

We start by comparing our method against the sketching based algorithm from [37], which was tested in the single thread setting. As a result, our experiments are ran in the single thread setting. The synthetic data we tested are third-order tensors with dimension $n = 1000$, as described in [37]. We first generated a rank-1000 tensor with harmonically decreasing weights on each rank-1 component. And then after normalization, random Gaussian noise with signal-to-noise $nsr = 0.1, 1, 10$ was added to the rank-1000 tensor. As with previous experimental evaluations [37], we focus on the low rank case, setting rank to $r = 10$. The performances of various routines are given in Table 1. We vary the sampling rate of our algorithm, i.e., SPALS(α) will sample $\alpha r^2 \log^2 n$ rows at each iteration.

	$nsr = 0.1$		$nsr = 1$		$nsr = 10$	
	error	time	error	time	error	time
ALS-dense	0.27	64.8	1.08	66.2	10.08	67.6
sketch(20, 14)	0.45	6.50	1.37	4.70	11.11	4.90
sketch(40, 16)	0.30	16.0	1.13	12.7	10.27	12.4
ALS-sparse	0.24	501	1.09	512	10.15	498
SPALS(0.3)	0.20	1.76	1.14	1.93	10.40	1.92
SPALS(1)	0.18	5.79	1.10	5.64	10.21	5.94
SPALS(3.0)	0.21	15.9	1.09	16.1	10.15	16.16

Table 1: Running times per iterations in seconds and errors of various alternating least squares implementations

On these instances, a call to SPALS with rate α samples was about $4.77\alpha \times 10^3$ rows, and as the tensor is dense, $4.77\alpha \times 10^6$ entries. The correspondence between running times and rates demonstrate the sublinear runtimes of SPALS with low sampling rates.

The runs that are most directly comparable to the experiments from [37] are the ones with noise-to-signal ratio 0.1. However, in our experiments, all routines converged in ≤ 3 iterations on these data sets Therefore we studied convergence behaviors on instances with high noise-to-signal ratios. In particular, the error over the first 20 iterations of SPALS with various rates are given in Figure 1 (a).

These results show that if one can tolerate a modest increase in error, a much faster convergence can be obtained in the first few iterations of SPALS. Also, a lower sampling rate does result in a higher final error. On the other hand, the gains of having a lower sampling rate is more clear when we plot the errors vs. time instead of iterations in Figure 1 (b). Note that the total running time given in this chart is comparable to the cost of a single step made by the dense ALS routine. Also, we were unable to include sketched methods in this comparison because outputting intermediate errors requires de-convolving the sketch, resulting in significant running time overheads.

In these evaluations, SPALS significantly outperforms existing routines under similar error requirements, and also has a smooth degrading towards the deterministic methods. This is despite the fact that SPALS is optimized for sparse tensors, and stores additional bookkeeping information in its intermediate data structures.

F.2 Sparse Data Tensor

Our original motivation for SPALS was to handle large sparse data tensors. We ran our algorithm on two sparse data tensors: a Foursquare dataset containing users' check-in records and a tensor generated from Amazon review data [24]. The sizes of these two tensors, and convergences of SPALS with various parameters are in Table 2.

	Check-In		Amazon	
dims	$44312 * 355 * 24$		$2.44m * 6.64m * 92.6k$	
nnz	689403		2025872645	
	error	time	error	time
ALS-sparse	0.70	0.269	0.981	142
SPALS(0.3)	0.76	0.088	0.987	6.97
SPALS(1)	0.71	0.269	0.983	15.7
SPALS(3.0)	0.70	0.694	0.982	38.9

Table 2: Sizes of the two large sparse tensors and the convergent relative error and running times per iteration of SPALS under various parameters.

The gains of sampling are less apparent on Check-in due to its smaller size and unbalanced dimensions. As a result, we will focus our comparisons on the Amazon data tensor, which also has a much higher noise to signal ratio than our other experiments. Running deterministic ALS with rank 10 on it leads to a relative error of 98.1%. The bounds from Theorem 4.1 with $\epsilon = 0.1$, gives an overhead factor of around 100. Nonetheless, SPALS still converges rapidly towards a good approximation: the errors over iterations and times of the deterministic variant as well as sampling with rates 0.3, 1.0, and 3.0 are shown in Figure 1 (c).

These comparisons are restricted to variants of SPALS because other methods are either for dense tensors, or have significant memory overheads. On the other hand, this did allow us to run the experiments in parallel with 16 threads. Although these parallelizations are for the purpose of speeding up the trials, we encountered several rather surprising issues related to randomizations. Standard random number generators leads to correlations between the threads, which affected both the memory access pattern and the rates of convergence. In particular, with the RAND() function, SPALS converges to a relative error of 98.8% on the Amazon tensor. Most of these issues were resolved by switching to a high quality parallel random number generator obtained from sitmo.com.