Scalable and Accurate CP Decomposition using Khatri-Rao Volume Sampling

Anonymous authors
Paper under double-blind review

Abstract

CP decomposition is a popular tool for analyzing the large volumes of multi-modal data, generated in diverse applications ranging from analytical chemistry to modern computer vision. Despite being the workhorse algorithm for CP decomposition, the alternating least-squares (ALS) is computationally intensive. This is attributed to the high dimensionality of the underlying Khatri-Rao product regression problem in each iteration. This work proposes a fast sampling based approach, called Khatri-Rao volume sampling (KRVS) to efficiently solve Khatri-Rao product regression problem. The proposed KRVS based solution is an unbiased estimate, and achieves $(1 + \epsilon)$ relative error approximation of the optimal least-squares solution. Building on the results of KRVS, a fast and accurate sampling based algorithm called CP-KRVS is proposed for CP decomposition. Theoretical and numerical analysis reveals that the proposed KRVS approach achieves a comparable accuracy, at faster runtime than exact leverage score sampling. Further, KRVS approach has higher accuracy, at slightly slower runtime than implicit leverage score sampling . At higher sample sizes, the KRVS based solution approximately converges with the optimal least-squares and ALS based solutions for Khatri–Rao product regression and CP decomposition, respectively, while achieving a faster runtime.

1 Introduction

Modern-day applications like computational neuroscience (Miwakeichi et al., 2004; Cong et al., 2015; Hunyadi et al., 2017), image processing/computer vision (Derkach et al., 2019; Panagakis et al., 2021; Pham Minh et al., 2022), social network analysis (Acar et al., 2006; 2009; Dunlavy et al., 2011; Spiegel et al., 2011), quantum physics (Orús, 2014; 2019), etc. inherently generate massive multi-modal data. For instance, the time series data from social network/chatroom communications can be aptly represented using three (or more) modes *viz.,* users, keywords, and time windows. Tensors are a generalization of vectors/matrices to N dimensions, and offer a natural and compact means to represent such multi-modal data (Kolda & Bader, 2009). Unlike the standard pairwise interactions in matrices, tensor representation helps to capture multi-linear/multi-modal relations in the data. This helps to unravel the hidden structures in the data, which are otherwise difficult to capture in matrix form (Acar & Yener, 2008).

The multi-modal data stored in tensor format can be further analyzed to perform feature extraction, classification, clustering, source separation, etc. One approach is to flatten/matricize the tensor, followed by using well established two-way component analysis schemes like PCA, LDA, etc. However, a flattened view of a tensor fails to exhibit the multiple couplings/correlations among different modes (Cichocki, 2014). As a result, the two-way component analysis schemes prove to be less efficient for tensor data. Tensor decompositions generalize the idea of factor analysis to N dimensions, and offer a better alternative for multi-modal data analysis (Cichocki et al., 2015). The utility of tensor decompositions was demonstrated in the context of latent semantic indexing problem by Chew et al. (2007), where matrix factorization-based methods were able to cluster the documents only based on language, regardless of the topic. In the same work, tensor decomposition-based approach was able to cluster the documents with similar topics across the languages. Similar observations are made in the domains of computer vision (Wang & Ahuja, 2008), quantum physics (Orús, 2014), signal processing (Cichocki et al., 2015), etc.

The Canonical Polyadic (CP) decomposition is a popular tensor decomposition framework in the literature (Kolda & Bader, 2009), and was independently discovered across various disciplines (Refer (Kolda & Bader, 2009, Table 3.1)). The CP decomposition factorizes a mode N tensor $\mathcal{X} \in \mathbb{R}^{I \times I \times \ldots I}$ as a sum of R components $\mathcal{X} = \sum_{r=1}^{R} \mathbf{a}_r^{(1)} \circ \mathbf{a}_r^{(2)} \circ \ldots \mathbf{a}_r^{(N)}$. Here $'\circ'$ denotes outer product (Definition 4), and R denotes the CP rank of $\mathcal{X}$ (Definition 11). The CP decomposition of $\mathcal{X}$ is compactly denoted as $\mathcal{X} = [\![\mathbf{A}^{(1)}, \mathbf{A}^{(2)} \ldots, \mathbf{A}^{(N)}]\!]$, where $\mathbf{A}^{(n)} := [\mathbf{a}_1^{(n)} \ \ldots \ \mathbf{a}_R^{(n)}]$ for $n \in \{1, 2, \ldots N\}$. Computing the CP rank is hard (Håstad, 1990), and it is approximated in practice using various methods, e.g., CORCONDIA (Papalexakis & Faloutsos, 2015). One of the distinguishing features of CP decomposition is that it is unique up to scaling and permutation, unlike popular matrix factorizations like SVD, PCA, etc. (Cichocki et al., 2015). This property is useful in a variety of applications, including source separation Gong et al. (2018); Ruan et al. (2022), integrated sensing and communications (Li et al., 2023; Zeng et al., 2024), etc.

The alternating least-squares (ALS) has been a workhorse algorithm for computing CP decomposition (Kolda & Bader, 2009). Given a mode N tensor $\mathcal{X} \in \mathbb{R}^{I \times I \times \ldots I}$ and its CP rank R, the ALS optimizes one factor matrix $\hat{\mathbf{A}}^{(n)}$ at time by keeping the rest of the factor matrices $\hat{\mathbf{A}}^{(m)}, m \neq n$ fixed. The process is repeated iteratively till convergence. In a given iteration, the optimization problem for updating nth factor matrix $\mathbf{A}^{(n)}$ is given as $\hat{\mathbf{A}}^{(n)} = \arg\min_{\mathbf{A}^{(n)}} \|\hat{\mathbf{A}}_{\neq n}(\hat{\mathbf{A}}^{(n)})^{\mathrm{T}} - \mathbf{X}_{(n)}^{\mathrm{T}}\|_F$, where $\hat{\mathbf{A}}_{\neq n} = \odot_{m=N, m\neq n}^{1} \hat{\mathbf{A}}^{(m)}$ is the Khatri-Rao product of $N-1$ factor matrices, and $\mathbf{X}_{(n)}$ denotes mode n matricization (Definition 6) of $\mathcal{X}$. The least-squares solution for $\hat{\mathbf{A}}^{(n)}$ has a runtime complexity of $O(I^N R)$, and is computationally intensive even for moderate values of N.

The last decade has seen a rising interest in designing randomization-based scalable algorithms for high-dimensional regression problems, with applications to CP decomposition (Wang et al., 2015; Cheng et al., 2016; Battaglino et al., 2018; Gittens et al., 2020; Malik & Becker, 2020; Jin et al., 2021; Malik, 2022; Malik et al., 2022b; Woodruff & Zandieh, 2022; Bharadwaj et al., 2023). The ALS update for nth factor matrix i.e., $\hat{\mathbf{A}}^{(n)}$ is obtained via randomization framework as $\bar{\mathbf{A}}^{(n)} = \arg\min_{\mathbf{A}^{(n)}} \|\mathbf{S}\hat{\mathbf{A}}_{\neq n}(\mathbf{A}^{(n)})^{\mathrm{T}} - \mathbf{S}\mathbf{X}_{(n)}^{\mathrm{T}}\|_F$. Here $\mathbf{S} \in \mathbb{R}^{K \times I^{N-1}}$ denotes sketching/sampling matrix. Refer to related works (Section 3) for more details. One of the initial works on sketching based framework for CP decomposition was by Wang et al. (2015), where TensorSketch (Pham & Pagh, 2013; Pagh, 2013) was employed for approximating the CP-ALS algorithm. Later works, including (Battaglino et al., 2018; Malik & Becker, 2020; Jin et al., 2021), proposed a fast JLT-based scheme called Kronecker fast JLT (KFJLT) to sketch Kronecker structured vectors with applications to CP decomposition. The KFJLT-based method has a high storage requirement, since the FFT operation destroys sparsity Larsen & Kolda (2022). Moreover, it suffers from lower accuracy compared to the sampling-based method in scenarios with dense tensors (Malik & Becker, 2020).

Cheng et al. (2016) proposed a sampling-based approach to approximate the factor matrix update problem in CP-ALS. Their sampling distribution was based on leverage scores of the factor matrices and entries of the tensor, and incurred an initial runtime cost of $nnz(\mathcal{X})$. This runtime dependency on $\mathcal{X}$ was later improved by Larsen & Kolda (2022), where the sampling distribution was constructed based solely on leverage scores of the factor matrices. The algorithm proposed by Larsen & Kolda (2022) for a rank R CP decomposition of an N mode tensor $\mathcal{X}$ required a sample size of $O(R^{N-1}/\epsilon)$ to provide an $(1+\epsilon)$ relative error approximation of the optimal CP-ALS-based solution. This was an improvement over $O(R^{2(N-1)}/\epsilon)$ sketch size of KFJLT, and $O(R^{N-1}/\epsilon^2)$ sample size of Cheng et al. (2016) methods. Recent years have seen the development of binary tree-based approaches (Malik, 2022; Malik et al., 2022b; Woodruff & Zandieh, 2022; Bharadwaj et al., 2023) for constructing the exact/approximate leverage scores of the Khatri-Rao product matrix, using factor matrices. This approach eliminates the exponential dependence of sample size on the number of modes N for $(1+\epsilon)$ relative error approximation, where the algorithm proposed by Bharadwaj et al. (2023) requires a sample size as small as $O(R/\epsilon)$. However, reduced sample size requirement in Bharadwaj et al. (2023) comes with high runtime complexity, as the cost of sampling one row from a Khatri-Rao product of N factor matrices $\mathbf{A}^{(n)} \in \mathbb{R}^{I \times R}$, is $O(NR^3 + NR^2 \log I)$, as opposed to $O(R)$ cost in work of Larsen & Kolda (2022).

As the tensor datasets continue to grow massive in size, with billions of non-zero entries (Smith et al., 2017), the distributed memory implementation of existing tensor decomposition algorithms is being explored (Bharadwaj et al., 2024). One approach is to independently compute the sampled least-squares estimate locally in each cluster/compute node, followed by averaging of local estimates. The leverage score

sampling-based approaches (Cheng et al., 2016; Larsen & Kolda, 2022; Malik, 2022; Malik et al., 2022b; Woodruff & Zandieh, 2022; Bharadwaj et al., 2023) give a biased solution to the regression problem, and hence are not suitable for model averaging (Dereziński et al., 2022). In this context, our work proposes a scalable and accurate sampling algorithm for approximating the Khatri-Rao product regression, with applications to CP decomposition under the following contributions.

1. Khatri-Rao volume sampling (KRVS) - A novel sampling approach to efficiently solve the Khatri-Rao product regression. The proposed KRVS solution is an unbiased estimate of the optimal least-squares solution, unlike baseline sketching/sampling approaches. The unbiasedness of KRVS is crucial for model averaging (Dereziński et al., 2022), and could be exploited in its distributed memory implementation.

2. The KRVS solution $\bar{\mathbf{w}} = \arg\min_{\mathbf{w}} \|\mathbf{SXw} - \mathbf{Sy}\|_2$ achieves a $(1+\epsilon)$ relative error approximation of the optimal least-squares solution $\mathbf{w}^* = \arg\min_{\mathbf{w}} \|\mathbf{Xw} - \mathbf{y}\|_2$, where $\mathbf{X} = \odot_{n=1}^{N} \mathbf{A}^{(n)}$, $\mathbf{A}^{(n)} \in \mathbb{R}^{I \times R}$, and $\mathbf{S} \in \mathbb{R}^{K \times I^N}$. The required sample size $K = O(R^N/\epsilon)$ is same as implicit (Larsen & Kolda, 2022), and higher than $O(R/\epsilon)$ for exact (Bharadwaj et al., 2023) leverage score sampling. However, the corresponding sampling time $O(KR^2)$ of KRVS is slightly greater than $O(KR)$ for implicit, and smaller than $O(NR^3 + KNR^2 \log I)$ for exact leverage score sampling.

3. Building on KRVS based Khatri-Rao product regression, a fast ALS algorithm called CP-KRVS is proposed for CP-decomposition. In the CP-decomposition of a tensor $\mathcal{X} \in \mathbb{R}^{I \times I \times \dots I}$, the factor matrix update obtained via CP-KRVS in a given iteration is an $(1+\epsilon)$ relative error approximation of optimal CP-ALS based update. The required sample size K, and the corresponding sampling time of CP-KRVS, implict (Larsen & Kolda, 2022), and exact (Bharadwaj et al., 2023) leverage score sampling follow a similar trend as in the Khatri-Rao product regression (see Table 1).

4. Numerical experiments for Khatri-Rao product regression and CP-decomposition, on synthetic and real world data sets, highlight that KRVS achieves higher accuracy than implicit (Larsen & Kolda, 2022) leverage score sampling, at a comparable runtime. Further, the KRVS is less accurate than the exact leverage score (Bharadwaj et al., 2023) sampling method, but compensates it with faster runtime. At higher sample sizes, the proposed KRVS achieves a comparable accuracy to the optimal least-squares/ALS based solutions, with a significant speed-up in running time (upto 60× speed-up observed empirically in Khatri-Rao product regression experiments, Section 6.1).

The rest of the paper is organized as follows. Section 2 gives a background on notations, tensor algebra, and sketching/sampling-based approximate regression. Related works on randomization approaches for Khatri-Rao product regression and the CP decomposition are discussed in Section 3. The proposed KRVS based regression approximation, along with its concentration and complexity analysis, is discussed in Section 4. Using the proposed KRVS framework for approximate Khatri-Rao product regression, Section 5 elaborates on fast CP decomposition algorithm, followed by its complexity and concentration guarantees. Numerical results for the proposed and baseline approximate regression and CP decomposition methods is discussed in Section 6, followed by the conclusion in Section 7.

2 Background

Notations: The real field is denoted by $\mathbb{R}$. A scalar product of K elements $\{n_1, \dots, n_K\}$ is denoted as $\prod_{k=1}^{K} n_k$. Similar notation is followed for the Kronecker product, Khatri-Rao product, elementwise/Hadamard product, and outer product, denoted by $\otimes$, $\odot$, $\diamond$, and $\circ$, respectively. The vectors, matrices, and tensors are denoted by boldface lowercase, boldface uppercase, and boldface calligraphic letters, respectively. The row i, column j, and element (i,j) of a matrix $\mathbf{A}^{I \times J}$ are denoted by $\mathbf{A}_{(i,:)}$, $\mathbf{A}_{(:,j)}$ and A_{ij} respectively. The ith entry of a vector $\mathbf{x}$ is denoted by x_i. A set of discrete indices $\{1, \dots, K\}$ is denoted as $\langle K \rangle$. An identity tensor of suitable order is denoted as $\mathcal{I}$.

The transpose, determinant, inverse, and pseudo-inverse of a matrix $\mathbf{A}$ is denoted by $\mathbf{A}^{\mathrm{T}}$, $|\mathbf{A}|$, $\mathbf{A}^{-1}$, and $\mathbf{A}^{\dagger}$, respectively. The mode n product of a tensor $\mathcal{X}$ with matrix $\mathbf{A}$ is denoted as $\mathcal{X} \times_n \mathbf{A}$ (see Definition 5).

The CP decomposition of a tensor $\mathcal{X}$ characterized by factor matrices $\mathbf{A}^{(1)}, \ldots, \mathbf{A}^{(N)}$ is defined as $\mathcal{X} = \sum_{r=1}^{R} \mathbf{a}_r^{(1)} \circ \mathbf{a}_r^{(2)} \circ \ldots \mathbf{a}_r^{(N)}$, where $\mathbf{A}^{(n)} = [\mathbf{a}_1^{(n)} \ \mathbf{a}_2^{(n)} \ \ldots \mathbf{a}_R^{(n)}]$ for $n \in \langle N \rangle$. For brevity, the CP decomposition of $\mathcal{X}$ denoted as $[\![\mathcal{I}; \mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \ldots, \mathbf{A}^{(N)}]\!]$. The symbol " := " denotes "equals by definition". The following definitions and properties from numerical linear/tensor algebra are employed in the text.

Definition 1 (**Least-squares regression**). *The least-squares regression problem is to obtain an optimal weight vector $\mathbf{w}^* \in \mathbb{R}^{R \times 1}$, given the design matrix $\mathbf{X} \in \mathbb{R}^{I \times R}$, and the observation vector $\mathbf{y} \in \mathbb{R}^{I \times 1}$ as*

$$\mathbf{w}^* = \arg\min_{\mathbf{w}} \|\mathbf{X}\mathbf{w} - \mathbf{y}\|_2. \tag{1}$$

The least-squares regression problem in Equation (1) is denoted as $(\mathbf{X}, \mathbf{y})$ for brevity.

Definition 2 (**Kronecker Product** Kolda & Bader (2009)). *The Kronecker product of two matrices $\mathbf{A} \in \mathbb{R}^{I \times J}$ and $\mathbf{B} \in \mathbb{R}^{M \times N}$ results in a matrix of size $IM \times JN$ given as follows*

$$\mathbf{A} \otimes \mathbf{B} = \begin{bmatrix} A_{11}\mathbf{B} & \cdots & A_{1J}\mathbf{B} \\ \vdots & \ddots & \vdots \\ A_{I1}\mathbf{B} & \cdots & A_{IJ}\mathbf{B}. \end{bmatrix} \tag{2}$$

Definition 3 (**Khatri-Rao Product** Kolda & Bader (2009)). *The Khatri-Rao product of two matrices $\mathbf{A} \in \mathbb{R}^{I \times J}$ and $\mathbf{C} \in \mathbb{R}^{K \times J}$ with same number of columns results in a matrix of size $IK \times J$, and is obtained by taking column-wise Kronecker product as*

$$\mathbf{A} \odot \mathbf{C} = [\mathbf{A}_{(:,1)} \otimes \mathbf{C}_{(:,1)} \ \mathbf{A}_{(:,2)} \otimes \mathbf{C}_{(:,2)} \ \cdots \ \mathbf{A}_{(:,J)} \otimes \mathbf{C}_{(:,J)}]. \tag{3}$$

Definition 4 (**Outer Product** Kolda & Bader (2009)). *The outer product of three vectors $\mathbf{x} \in \mathbb{R}^{I \times 1}$, $\mathbf{y} \in \mathbb{R}^{J \times 1}$ and $\mathbf{z} \in \mathbb{R}^{K \times 1}$ results into a rank-1 tensor $\mathcal{W} \in \mathbb{R}^{I \times J \times K}$, $\mathcal{W} = \mathbf{x} \circ \mathbf{y} \circ \mathbf{z}$ of mode-3. Its element corresponding to index (i, j, k) is denoted by W_{ijk}, and given as $= x_i y_j z_k$.*

The next fact states a few useful properties of the Kronecker and Khatri-Rao products, further employed in the text.

Fact 1 (**Matrix Product Properties** Sidiropoulos et al. (2017)). *Consider the matrices $\mathbf{A} \in \mathbb{R}^{I \times I}$, $\mathbf{B} \in \mathbb{R}^{I \times I}$, $\mathbf{C} \in \mathbb{R}^{I \times I}$, $\mathbf{D} \in \mathbb{R}^{I \times I}$, $\mathbf{G} \in \mathbb{R}^{I \times I}$ and $\mathbf{H} \in \mathbb{R}^{J \times J}$. Then*

$$(\mathbf{A} \otimes \mathbf{B})(\mathbf{C} \otimes \mathbf{D}) = (\mathbf{AC}) \otimes (\mathbf{BD}) \tag{4}$$

$$(\mathbf{A} \otimes \mathbf{B})(\mathbf{C} \odot \mathbf{D}) = (\mathbf{AC}) \odot (\mathbf{BD}) \tag{5}$$

$$(\mathbf{A} \odot \mathbf{B})^T(\mathbf{C} \odot \mathbf{D}) = (\mathbf{A}^T\mathbf{C}) \diamond (\mathbf{B}^T\mathbf{D}) \tag{6}$$

$$|\mathbf{A} \otimes \mathbf{H}| = |\mathbf{A}|^J |\mathbf{H}|^I \tag{7}$$

Definition 5 (**Mode$-n$ Product** Kolda & Bader (2009)). *The mode$-n$ product of a tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \ldots I_N}$ and a matrix $\mathbf{P} \in \mathbb{R}^{J \times I_n}$ denoted as $\mathcal{X} \times_n \mathbf{P}$ results into a tensor of size $I_1 \times I_2 \times \ldots I_{n-1} \times J \times I_{n+1} \times \ldots I_N$ and is given as*

$$(\mathcal{X} \times_n \mathbf{P})_{i_1 i_2 \ldots i_{n-1} j i_{n+1} \ldots i_N} = \sum_{i=1}^{I_n} X_{i_1 i_2 \ldots i_N} P_{j i_n}. \tag{8}$$

Definition 6 (**Matricization** Kolda & Bader (2009)). *The matricization of a tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \ldots I_N}$ along mode n, $n \in \langle N \rangle$ is denoted by $\mathbf{X}_{(n)} \in \mathbb{R}^{I_n \times J}$, where $J = \prod_{m=1, m \neq n}^{N} I_m$, and results into a matrix with fibers along mode n as its columns. The $(i_1, i_2, \ldots, i_N)$th element of $\mathcal{X}$ maps to (i_n, j)th element of $\mathbf{X}_{(n)}$ where*

$$j = 1 + \sum_{k=1, k \neq n}^{N} (i_k - 1) J_k, \ \textit{for } J_k = \prod_{m=1, m \neq n}^{k-1} I_m.$$

Definition 7. *Given a discrete random variable $z \in \langle I \rangle$, denote $z \sim$ MULTINOMIAL$(\mathbf{p})$ if $\mathbf{p} := (p_1, p_2, \ldots p_I)$ is a discrete probability distribution, and $\Pr[z = i] = p_i$.*

Definition 8 (**Weighted Sampling** (Larsen & Kolda, 2022)). *A sampling matrix* $\mathbf{S} \in \mathbb{R}^{K \times I} \sim$ *RANDSAMPLE*$(K, \mathbf{p})$*, if* $K \leq I$*,* $\mathbf{p} = (p_1, p_2, \ldots p_I)$ *is a discrete probability distribution, and the entries on* $\mathbf{S}$ *are defined as follows. Let* $\zeta_k \sim$ *MULTINOMIAL*$(\mathbf{p})$ *for* $k = 1 \ldots K$*; then*

$$S_{ki} = \begin{cases} \frac{1}{\sqrt{Kp_i}} & \zeta_k = i; \\ 0 & \textit{Otherwise.} \end{cases} \tag{9}$$

Definition 9 (**Volume Sampling** (Derezinski & Warmuth, 2017)). *The volume sampling selects a small subset of rows from a tall matrix* $\mathbf{X} \in \mathbb{R}^{I \times R}$*, with a probability proportional to volume of the submatrix (the volume of a matrix* $\mathbf{X}$ *refers to* $|\mathbf{X}^T\mathbf{X}|$*). A size* K *volume sampling on* $\mathbf{X}$ *is denoted as VolumeSample*$(\mathbf{X}, K)$*, and gives sampled row indices as output.*

Definition 10 (**Leverage score sampling** (Woodruff et al., 2014)). *Let* $\mathbf{U} \in \mathbb{R}^{I \times R}$ *denote a matrix with top* R *left singular vectors of* $\mathbf{X} \in \mathbb{R}^{I \times R}$ *as columns, such that* $l_i := \|\mathbf{U}_{(i,:)}\|_2^2$ *denotes the* i*th leverage score of* $\mathbf{X}$*. Then, leverage score sampling selects a subset of rows from* $\mathbf{X}$*, via the probability distribution* $(q_1, q_2, \ldots q_I)$*, where* $q_i \geq l_i/R$*, for* $i \in \{1, 2, \ldots I\}$ *denotes the probability of sampling* i*th row from* $\mathbf{X}$*.*

Fact 2 (Kolda & Bader (2009)). *Let the CP decomposition of* $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \ldots I_N}$ *be expressed as*

$$\mathcal{Y} = \sum_{r=1}^{R} \mathbf{a}_r^{(1)} \circ \mathbf{a}_r^{(2)} \circ \ldots \mathbf{a}_r^{(N)} = [\![\mathcal{I}; \mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \ldots \mathbf{A}^{(N)}]\!], \tag{10}$$

where $\mathbf{A}^{(n)} \in \mathbb{R}^{I_n \times R}$*,* $n \in \langle N \rangle$ *denote the factor matrices. Then, matricization of* $\mathcal{X}$ *along mode* n *gives*

$$\mathbf{X}_{(n)} = \mathbf{A}^{(n)}(\mathbf{A}^{(N)} \odot \mathbf{A}^{(N-1)} \odot \ldots \odot \mathbf{A}^{(n+1)} \odot \mathbf{A}^{(n-1)} \odot \ldots \odot \mathbf{A}^{(1)})^T. \tag{11}$$

2.1 Review of CP decomposition

The CP decomposition factors a given tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \ldots I_N}$ into a sum of R rank 1 component tensors as shown below.

$$\mathcal{X} = \sum_{r=1}^{R} \mathbf{a}_r^{(1)} \circ \mathbf{a}_r^{(2)} \circ \ldots \mathbf{a}_r^{(N)}. \tag{12}$$

Here R is called the CP rank of $\mathcal{X}$, defined as follows.

Definition 11 (CP Rank (Kolda & Bader, 2009)). *The CP rank* R *of a tensor* $\mathcal{X}$ *is defined as the smallest number of rank* 1 *tensors that generate* $\mathcal{X}$ *as their sum.*

The vectors $\mathbf{a}_r^{(n)}$ are stacked as columns to construct the factor matrices $\mathbf{A}^{(n)} \in \mathbb{R}^{I_n \times R}$ as $\mathbf{A}^{(n)} = [\mathbf{a}_1^{(n)}\ \mathbf{a}_2^{n}\ \ldots \mathbf{a}_R^{n}]$, for $n \in \langle N \rangle$.

In practice, there is no finite algorithm to compute the exact CP decomposition of a given tensor $\mathcal{X}$, since obtaining the exact value of R is hard (Håstad, 1990). Many approaches try to fit CP decomposition on $\mathcal{X}$ for different values of R until they find a good fit. Once the value of R is fixed, the CP decomposition of $\mathcal{X}$ can be approximately solved as follows.

$$\min_{\hat{\mathcal{X}}} \|\mathcal{X} - \hat{\mathcal{X}}\|_F, \text{ where } \hat{\mathcal{X}} = \sum_{r=1}^{R} \hat{\mathbf{a}}_r^{(1)} \circ \hat{\mathbf{a}}_r^{(2)} \circ \ldots \hat{\mathbf{a}}_r^{(N)} = [\![\mathcal{I}; \hat{\mathbf{A}}^{(1)}, \hat{\mathbf{A}}^{(2)}, \ldots \hat{\mathbf{A}}^{(N)}]\!]. \tag{13}$$

The alternating least squares (ALS) approach, also known as CP-ALS, is considered the workhorse algorithm for solving Equation (13) and is shown in Algorithm 1. For a given value of R, the CP-ALS algorithm iteratively solves for each factor matrix $\hat{\mathbf{A}}^{(n)}$, while fixing the rest $N-1$ factor matrices $\hat{\mathbf{A}}^{(p)}$, $p \neq n$. The iterations are continued till convergence. The subproblem of updating nth factor matrix can be obtained by taking nth mode matricization in Equation (13) as follows

$$\hat{\mathbf{A}}^{(n)} = \arg\min_{\mathbf{A}^{(n)}} \|\mathbf{X}_{(n)} - \mathbf{A}^{(n)}(\odot_{p=N, p\neq n}^{1} \hat{\mathbf{A}}^{(p)})^{\mathrm{T}}\|_F. \tag{14}$$

Algorithm 1: CP-ALS

Data: $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \ldots I_N}$, CP rank R
Result: $\hat{\mathbf{A}}^{(n)} \in \mathbb{R}^{I_n \times R}$, for $n \in \langle N \rangle$
Initialize $\hat{\mathbf{A}}^{(n)}$, $n = \{2, 3, \ldots N\}$;
repeat
  for $i = 1 : N$ **do**
    $\hat{\mathbf{A}}_{\neq i} \leftarrow \odot_{p=N, p\neq n}^{1} \hat{\mathbf{A}}^{(p)}$;
    $\mathbf{X}_{(i)} \leftarrow$ Mode i Matricization of $\mathcal{X}$;
    $\hat{\mathbf{A}}^{(n)} \leftarrow (\hat{\mathbf{A}}_{\neq i})^{\dagger} \mathbf{X}_{(i)}^{\mathrm{T}}$;
  end
until *Until Convergence*;
return $\hat{\mathbf{A}}^{(n)} \in \mathbb{R}^{I_n \times R}$, for $n \in \langle N \rangle$

Let $\hat{\mathbf{A}}_{\neq n} := \odot_{p=N, p\neq n}^{1} \hat{\mathbf{A}}^{(p)}$. Taking the transpose of Equation (14) gives

$$\hat{\mathbf{A}}^{(n)} = \arg\min_{\mathbf{A}^{(n)}} \|\hat{\mathbf{A}}_{\neq n} (\hat{\mathbf{A}}^{(n)})^{\mathrm{T}} - \mathbf{X}_{(n)}^{\mathrm{T}}\|_F. \tag{15}$$

The factor matrix update problem of CP-ALS defined in Equation (15) is a high dimensional regression problem, where the design matrix $\hat{\mathbf{A}}_{\neq n} \in \mathbb{R}^{I_{\neq n} \times R}$, $I_{\neq n} := \prod_{p=1, p\neq n}^{N}$ is a Khatri-Rao product of several matrices. The exact solution to the problem in Equation in (15) is given as $\hat{\mathbf{A}}^{(n)} = (\hat{\mathbf{A}}_{\neq n})^{\dagger} \mathbf{X}_{(n)}^{\mathrm{T}}$ and has high computational cost. The next subsection briefly reviews the randomization approach to approximately solve Khatri-Rao product regression problem similar to (15), with less computational cost.

2.2 Fast Khatri-Rao Product Regression

The nth factor matrix update problem in Equation (15) is a Khatri-Rao product regression problem defined next. Consider a linear regression (refer Defn. 1) problem $(\mathbf{X}, \mathbf{y})$, where $\mathbf{X} = \odot_{m=1}^{N} \mathbf{A}^{(m)}$, $\mathbf{A}^{(m)} \in \mathbb{R}^{I_m \times R}$, $\forall m \in \langle N \rangle$, with optimal solution $\mathbf{w}^* := \arg\min_{\mathbf{w} \in \mathbb{R}^{R \times 1}} \|\mathbf{X}\mathbf{w} - \mathbf{y}\|_2^2 = \mathbf{X}^{\dagger}\mathbf{y}$. Without loss of generality, assume $I_1 = I_2 = \ldots = I_N = I$. The runtime complexity of computing $\mathbf{w}^*$ is $O(I^N R^2)$, which is computationally intensive even for moderate values of N.

Randomization approach has been extensively used over the years for speeding-up linear regression (Sarlos, 2006; Drineas et al., 2006b; Mahoney et al., 2011; Woodruff et al., 2014; Drineas & Mahoney, 2016). The regression probelm $(\mathbf{X}, \mathbf{y})$ is solved using randomization in two essential steps: Generate a sketch of $\mathbf{X}$, $\mathbf{y}$ as $\mathbf{X}_K$, $\mathbf{y}_K$ respectively, where $\mathbf{X}_K \in \mathbb{R}^{K \times R}$, $\mathbf{y}_K \in \mathbb{R}^{K \times 1}$, and $K \ll I^N$, followed by solving $(\mathbf{X}_K, \mathbf{y}_K)$. Denote $\hat{\mathbf{w}} := \arg\min_{\mathbf{w}} \|\mathbf{X}_K \mathbf{w} - \mathbf{y}_K\|_2^2$ as the randomization based solution to $(\mathbf{X}, \mathbf{y})$. Then, the following properties of $\hat{\mathbf{w}}$ are desirable

$$(a) : \|\mathbf{X}\hat{\mathbf{w}} - \mathbf{y}\|_2 \leq (1 + \epsilon)\|\mathbf{X}\mathbf{w}^* - \mathbf{y}\|_2 \qquad (b) : \mathbb{E}[\hat{\mathbf{w}}] = \mathbf{w}^*. \tag{16}$$

The property (a) is termed as $(1 + \epsilon)$ relative error approximation, and (b) denotes unbiased estimation. The two properties in Equation (16) together show that the obtained solution $\hat{\mathbf{w}}$ is a good approximation of the optimal solution $\mathbf{w}^*$. Baseline randomization approaches for Khatri-Rao product regression, and subsequently, CP decomposition are reviewed next.

3 Related Works

The factor matrix update problem in Equation (15) can be approximately solved via randomization as follows

$$\bar{\mathbf{A}}^{(n)} = \arg\min_{\mathbf{A}^{(n)}} \|\mathbf{S}\hat{\mathbf{A}}_{\neq n} (\mathbf{A}^{(n)})^{\mathrm{T}} - \mathbf{S}\mathbf{X}_{(n)}^{\mathrm{T}}\|_F. \tag{17}$$

Here $\mathbf{S} \in \mathbb{R}^{K \times I_{\neq n}}$ is a matrix that reduces the number of constraints in Equation (15) to $K \ll I_{\neq n}$. Sketching and sampling are two popular approaches for randomization. In sketching, the matrix $\mathbf{S}$ is constructed

using randomly sampled entries from a probability distribution like Gaussian (Dasgupta & Gupta, 2003), Rademacher (Achlioptas, 2003), sparse random projections (Li et al., 2006; Clarkson & Woodruff, 2017), etc. In sampling, $\mathbf{S}$ is very sparse, with only one non-zero entry per row. Multiplication by $\mathbf{S}$ performs selection and scaling of a subset of rows from $\mathbf{A}_{\neq n}$ and $\mathbf{X}_{(n)}^{\mathrm{T}}$. The selection can be uniformly at random (Drineas et al., 2006a) or based on some measure of importance, like leverage scores Frieze et al. (2004); Drineas et al. (2006b; 2011), volume measure Dereziński & Warmuth (2018); Derezinski et al. (2018) etc. However, the high dimensionality of Khatri-Rao product regression makes the naive application of sketching/sampling based approach intractable. Recent years have seen a lot of research focused of efficient sketching/sampling of structured matrices, and are discussed next.

Implicit sampling: The implicit sampling approach for solving the problem in Equation (17) involves sketching/sampling the factor matrices as $\mathbf{A}_K^{(m)} = \mathbf{S}^{(m)}\mathbf{A}^{(m)}$, $m \in \{1, 2, \ldots N\}, m \neq n$ in the first step. This is followed by equivalently constructing the sampled design matrix $\mathbf{S}\mathbf{A}_{\neq n}$ using $\mathbf{A}_K^{(m)}$. The next theorem from Cheng et al. (2016) upper bounds the leverage scores of a Khatri-Rao product matrix, using leverage scores of its constituent factors.

Theorem 1 (Theorem 3.2 of Cheng et al. (2016))**.** *Consider the matrices* $\mathbf{C} \in \mathbb{R}^{I_1 \times R}$*,* $\mathbf{B} \in \mathbb{R}^{I_2 \times R}$*, and* $\mathbf{Y} \in \mathbb{R}^{I \times R}$*, such that* $\mathbf{Y} = \mathbf{C} \odot \mathbf{B}$*,* $I = I_1 I_2$*. Let* l_i*,* l_j*, and* l_k *denote the leverage scores of* i*th,* j*th, and* k*th rows of* $\mathbf{C}$*,* $\mathbf{B}$*, and* $\mathbf{Y}$ *respectively, where* $k = (i-1)I_2 + j$*. Then,*

$$l_k \leq l_i l_k. \tag{18}$$

Cheng et al. (2016) proposed a sampling distribution for $\hat{\mathbf{A}}_{\neq n}$ and $\mathbf{X}_{(n)}^{\mathrm{T}}$ based on leverage scores of $\mathbf{A}^{(m)}$ and row norms of $\mathbf{X}_{(n)}^{\mathrm{T}}$. Their approach required a sample size $K = O(R^{N-1} \log I/\epsilon^2)$ to provide the $(1+\epsilon)$ relative error approximation (property (a) in Equation (16)). It also incurs a one-time cost of $nnz(\mathcal{X})$ for computing the row norms of $\mathbf{X}_{(n)}^{\mathrm{T}}$. A fast Johnson-Lindenstrauss transform (FJLT) based approach was employed by Battaglino et al. (2018); Jin et al. (2021), where each factor matrix $\mathbf{A}^{(m)}$ is sketched using FJLT, thereby only constructing sketched rows of $\hat{\mathbf{A}}_{\neq n}$. However, sketching RHS i.e., $\mathbf{X}_{(n)}^{\mathrm{T}}$ involves taking FFT, and destroys the inherent sparsity (if present) of $\mathcal{X}$. Also, the required sketch size for relative error approximation is $O(R^{2(N-1)}/\epsilon)$. Larsen & Kolda (2022) proposed a modified scheme compared to Cheng et al. (2016), by deterministically including high leverage rows in the sampled design matrix, thus improving the sample size requirement to $O(R^{N-1}/\epsilon)$. Other approaches using implicit sampling for tensor decompositions include Diao et al. (2019); Ma & Solomonik (2021); Fahrbach et al. (2022). Note that implicit sampling requires a sample size with exponential dependence on N. Recent works on randomization approaches tend to avoid this exponential dependence on N, and are discussed next.

Recursive/binary tree based methods: Woodruff & Zandieh (2022) proposed a sampling method for Khatri-Rao product of N matrices via approximate ridge leverage scores, using the ideas from tree-based recursive sketch (Ahle et al., 2020; Woodruff & Zandieh, 2020). Despite having a linear dependence on the number of columns R, the proposed sampling method is computationally intensive due to dependence on the order N and accuracy parameter ϵ being $O(N^8)$ and $O(\epsilon^{-4})$, respectively. Malik (2022) later improved the runtime dependence on N and R to $O(N^3)$ and $O(R^4)$, respectively using a similar approach as Woodruff & Zandieh (2022), with further applications to CP and tensor train decomposition. The sampling method in Malik (2022) was later improved and generalized for tensor decompositions into any arbitrary tensor networks in Malik et al. (2022a). Recently, Bharadwaj et al. (2023) proposed an efficient data structure for sampling rows of a Khatri-Rao product matrix according to exact leverage scores, offering a faster runtime complexity compared to Malik (2022). Note that the sampling distribution employed in Malik (2022); Malik et al. (2022a); Bharadwaj et al. (2023) is based on exact leverage scores of the Khatri-Rao product. This results in a sample size requirement for $(1+\epsilon)$ relative error approximation of nth factor matrix update problem in Equation (17) to be $O(R/(\epsilon\delta))$.

Baselines comparison with our work: This work presents a novel sampling framework for solving Khatri-Rao product regression, with applications to CP decomposition. The proposed solution is an unbiased estimator of the optimal least-squares solution, unlike the baseline implicit (Cheng et al., 2016; Jin et al., 2021; Larsen & Kolda, 2022)) and recursive/tree-based (Malik, 2022; Malik et al., 2022a; Bharadwaj et al.,

Approach	**Sample size (K)**	**Sampling time (K samples)**	**Guarantees**
Implicit leverage score sampling (Larsen & Kolda, 2022)	$O(R^{N-1}/\epsilon)$	$O(KR)$	$(1+\epsilon)$ Relative Error
Exact leverage score sampling (Bharadwaj et al., 2023)	$O(R/\epsilon)$	$O((N-1)R^3 + K(N-1)R^2 \log I)$	$(1+\epsilon)$ Relative Error
KRVS (**This work**)	$O(R^{N-1}/\epsilon)$	$O(KR^2)$	$(1+\epsilon)$ Relative Error, Unbiased Estimation

Table 1: Comparison of KRVS with baseline sampling algorithms for rank R CP decomposition of a mode N tensor $\mathcal{X} \in \mathbb{R}^{I \times I \times \dots I}$. The metrics compared are sampling time, required sample size for $(1+\epsilon)$ relative error approximation. The pre-processing cost of $O(NIR^2)$, to compute the factor matrices' leverage scores, and the cost for solving sampled least-squares is same for all considered sampling schemes, and is not included in asymptotic runtime comparison.

2023) sampling methods. This is a useful property for model averaging (Dereziński et al., 2022), and could be exploited in the distributed memory implementation of regression/CP decomposition. Further, the proposed solution requires a sample size of $O(R^{N-1}/(\epsilon\delta))$ for $(1+\epsilon)$ relative error approximation. This is the best possible sample size in implicit sampling(see Section C.3.3 of Malik (2022)). For a fixed sample size K, the sampling time of KRVS-based approach to solve factor matrix update problem via Equation (17) is $O(KR^2)$. This is less compared to the $O((N-1)R^3 + K(N-1)R^2 \log I)$ sampling time of exact leverage score sampling approach in Bharadwaj et al. (2023). In solving Khatri-Rao product regression/CP-decomposition, the proposed KRVS approach is shown (empirically) to have higher accuracy than implicit (Larsen & Kolda, 2022), and comparable/lower accuracy than the exact (Bharadwaj et al., 2023) leverage score sampling-based approaches. The Table 1 gives a summary of comparison between different CP decomposition algorithms for an N mode tensor $\mathcal{X} \in \mathbb{R}^{I \times I \times \dots I}$, at a decomposition rank R. Recall that $(1+\epsilon)$ relative error and unbiased estimation guarantees refer to the properties (a) and (b) of Equation (16), respectively.

4 Khatri-Rao Volume Sampling (KRVS)

This section discusses the proposed sampling scheme for Khatri-Rao product regression. The Section 4.1 reviews the volume rescaled sampling approach for approximate regression, followed by our contributions in Section 4.2. The Subsections 4.2.1 and 4.2.2 highlight the sampling algorithm and concentration/complexity analysis, respectively.

4.1 Review of volume rescaled sampling

Consider a linear regression problem $(\mathbf{X}, \mathbf{y})$, $\mathbf{X} \in \mathbb{R}^{I \times R}$, $\mathbf{y} \in \mathbb{R}^{I \times 1}$. The optimal solution $\mathbf{w}^*$ for $(\mathbf{X}, \mathbf{y})$ is obtained as $\mathbf{w}^* = \arg\min_{\mathbf{w}} \|\mathbf{X}\mathbf{w} - \mathbf{y}\|_2$, with a runtime complexity of $O(IR^2)$ i.e., the SVD complexity of $\mathbf{X}$. This is computationally intensive when $I \gg R$. An alternative approach for solving $(\mathbf{X}, \mathbf{y})$ is to select a subset of rows from $\mathbf{X}$ and corresponding entries from $\mathbf{y}$ to get $\mathbf{X}_K$ and $\mathbf{y}_K$ respectively, followed by solving $(\mathbf{X}_K, \mathbf{y}_K)$. Let $\mathbf{q} = (q_1, q_2, \dots q_I)$ denote a discrete probability distribution on row indices of $\mathbf{X}$, where the probability of sampling $\mathbf{X}_{(i,:)}$, i.e., row i of $\mathbf{X}$ is q_i. Then, the volume rescaled sampling (VRS) by Dereziński et al. (2022) selects a subset of K row indices $\boldsymbol{\omega} = (\omega_1, \omega_2, \dots \omega_K)$ with replacement from $\mathbf{X}$, to construct a sampling matrix $\bar{\mathbf{S}}_{\boldsymbol{\omega}} \in \mathbb{R}^{K \times I}$, with entries defined as follows

$$(\bar{\mathbf{S}}_{\boldsymbol{\omega}})_{ki} = \begin{cases} 1/\sqrt{q_{\omega_k}}, & i = \omega_k \\ 0, & \text{otherwise} \end{cases}, \text{ with probability } \frac{|(\bar{\mathbf{S}}_{\boldsymbol{\omega}}\mathbf{X})^{\mathrm{T}}\bar{\mathbf{S}}_{\boldsymbol{\omega}}\mathbf{X}| \prod_{k=1}^{K} q_{\omega_k}}{Q}. \tag{19}$$

Here, Q denotes the normalization factor for the joint sampling distribution. Let $\langle I \rangle^K$ denote all the size K row indices set, sampled with replacement from $\langle I \rangle$. Then,

$$Q = \sum_{\boldsymbol{\omega} \in \langle I \rangle^K} |\mathbf{X}^{\mathrm{T}}\bar{\mathbf{S}}_{\boldsymbol{\omega}}^{\mathrm{T}}\bar{\mathbf{S}}_{\boldsymbol{\omega}}\mathbf{X}| \prod_{k=1}^{K} q_{\omega_k} = (R)! \binom{K}{R} |\mathbf{X}^{\mathrm{T}}\mathbf{X}|. \tag{20}$$

The sampled design matrix $\mathbf{X}_K$ and the observation vector $\mathbf{y}_K$ are obtained as $\mathbf{X}_K = \bar{\mathbf{S}}_{\boldsymbol{\omega}}\mathbf{X}$ and $\mathbf{y}_K = \bar{\mathbf{S}}_{\boldsymbol{\omega}}\mathbf{y}$. Further, the sampled regression problem $(\mathbf{X}_K, \mathbf{y}_K)$ is solved as

$$\mathbf{w}_{\text{VRS}} := \arg\min_{\mathbf{w}} \|\mathbf{X}_K\mathbf{w} - \mathbf{y}_K\|_2 = \mathbf{X}_K^{\dagger}\mathbf{y}_K. \tag{21}$$

Computing $\mathbf{w}_{\text{VRS}}$ has a runtime of $O(IR^2 + (4R^2 + 2K)R^2)$ (Derezinski et al., 2018, Theorem 6). Further, the obtained solution $\mathbf{w}_{\text{VRS}}$ is an unbiased estimator of optimal solution $\mathbf{w}^*$.

4.2 Our Contribution

4.2.1 Sampling Algorithm for Khatri-Rao Product Regression

Let $\mathbf{X} \in \mathbb{R}^{I\times R}$ in regression problem $(\mathbf{X}, \mathbf{y})$, be defined as $\mathbf{X} = \odot_{n=1}^{N}\mathbf{A}^{(n)}$, where $\mathbf{A}^{(n)} \in \mathbb{R}^{I_n\times R}$ for $n \in \{1, 2, \dots N\}$. Note that the number of rows of $\mathbf{X}$ i.e., $I = \prod_{n=1}^{N} I_n$, grows very large i.e., $I \gg R$, even for modest values of N. The volume rescaled sampling-based solution to $(\mathbf{X}, \mathbf{y})$ is an unbiased estimator of the optimal solution, and is highly desirable. However, its implementation is difficult due to the high runtime complexity. Our proposed KRVS approach addresses this problem.

To obtain the size K regression approximation $(\mathbf{X}_K, \mathbf{y}_K)$ of the original problem $(\mathbf{X}, \mathbf{y})$, the KRVS method samples K row indices from each factor matrix $\mathbf{A}^{(n)}$ independently, and uses them to construct the equivalent sampled row indices of $\mathbf{X}$. Thus obtained row indices can be shown to follow the joint sampling distribution in Equation (19). Denote $\mathbf{q}^{(n)} := (q_1^{(n)}, q_2^{(n)}, \dots q_{I_n}^{(n)})$ as a discrete probability distribution on the row indices of $\mathbf{A}^{(n)}$, $\boldsymbol{\omega}^{(n)} := (\omega_1^{(n)}, \omega_2^{(n)}, \dots \omega_K^{(n)})$ as K row indices sampled independently, with replacement from $\langle I_n\rangle$, via $\mathbf{q}^{(n)}$, for $n \in \langle N\rangle$. The sampled row index set $\boldsymbol{\omega} = (\omega_1, \omega_2, \dots \omega_K)$ for $\mathbf{X}$ is constructed using $\boldsymbol{\omega}^{(n)}$ as

$$\omega_k = \omega_k^{(N)} + (\omega_k^{(N-1)} - 1)I_N + \sum_{i=1}^{N-2}(\omega_k^{(i)} - 1)\prod_{j=i+1}^{N} I_j. \tag{22}$$

Further, the marginal probability $\Pr[\omega_k \in \boldsymbol{\omega}]$ is given as

$$\Pr[\omega_k \in \boldsymbol{\omega}] = \prod_{n=1}^{N}\Pr[\omega_k^{(n)} \in \boldsymbol{\omega}^{(\boldsymbol{n})}] = \prod_{n=1}^{N} q^{(n)}_{\omega_k^{(n)}}. \tag{23}$$

The marginal probabilities of sampled rows indices ω_k in Equation (23) together form a discrete probability distribution over row indices of $\mathbf{X}$ i.e., $\langle I\rangle$, as shown next.

Proposition 2. *Let $\mathbf{q}^{(n)} = (q_1^{(n)}, q_2^{(n)}, \dots q_{I_n}^{(n)})$ denote a discrete probability distribution over the row indices of $\mathbf{A}^{(n)} \in \mathbb{R}^{I_n\times R}$, for $n \in \{1, 2, \dots N\}$ such that, the probability of sampling row j from $\mathbf{A}^{(n)}$ is given as $q_j^{(n)}$. Then $\mathbf{q} = \odot_{n=1}^{N}\mathbf{q}^{(n)}$ is a discrete probability distribution over the row indices of $\mathbf{X} = \odot_{n=1}^{N}\mathbf{A}^{(n)}$.*

Proof. The two sufficient conditions for $\mathbf{q} = (q_1, q_2, \dots q_I)$ to be a discrete probability distribution are q_i, $1 \le i \le I$, being positive and summing up to 1. Each element of $\mathbf{q}$ is constructed from the product of elements of probability distributions $\mathbf{q}^{(n)}$, and is positive. Next, the sum of the elements of $\mathbf{q}$ is resolved as

$$\sum_{i=1}^{I} q_i = \prod_{n=1}^{N}\sum_{j=1}^{I_n} q_j^{(n)} = \prod_{n=1}^{N}(1) = 1. \tag{24}$$

□

Let $\bar{\mathbf{T}}_{\boldsymbol{\omega}^{(n)}} \in \mathbb{R}^{K\times I_n}$ denote a matrix which samples and rescales K rows from $\mathbf{A}^{(n)}$, corresponding to the indices in $\boldsymbol{\omega}^{(n)}$ such that $(\bar{\mathbf{T}}_{\boldsymbol{\omega}^{(n)}})_{ki} = \begin{cases} 1/\sqrt{q_{\omega_k^{(n)}}}, & i = \omega_k^{(n)} \\ 0, & \text{Else} \end{cases}$. Using Equation (22) and Proposition 2, the

equivalent sampling matrix $\bar{\mathbf{T}}_{\boldsymbol{\omega}} \in \mathbb{R}^{K \times I}$, which samples and rescales K rows corresponding to $\boldsymbol{\omega}$ from $\mathbf{X}$ is expressed as

$$\bar{\mathbf{T}}_{\boldsymbol{\omega}} = \left(\odot_{n=1}^{N} \bar{\mathbf{T}}_{\boldsymbol{\omega}^{(n)}}^{\mathrm{T}}\right)^{\mathrm{T}}. \tag{25}$$

The regression problem $(\mathbf{X}, \mathbf{y})$ can be approximated via KRVS approach as shown in Algorithm 2. Given

Algorithm 2: Khatri-Rao Volume Sampling (KRVS)

Data: $\mathbf{A}^{(n)} \in \mathbb{R}^{I_n \times R}$, $n \in \langle N \rangle$, Sample size $K \geq R$
Result: Sampling matrix $\bar{\mathbf{T}}_{\boldsymbol{\omega}}$, Sub-sampled design matrix $\mathbf{X}_K$

1 $B \leftarrow \max(K, 4R^2)$;
2 $A \leftarrow |\diamond_{n=1}^{N} (\mathbf{A}^{(n)})^{\mathrm{T}} \mathbf{A}^{(n)}|$;
3 $\mathbf{q}^{(n)} \leftarrow$ Leverage scores of $\mathbf{A}^{(n)}$, for all $n \in \langle N \rangle$;
4 **repeat**
5 Sample B row indices i.i.d., with replacement from $\langle I_n \rangle$ via distribution $\mathbf{q}^{(n)}$, and store in $\boldsymbol{\omega}^{(n)}$, for all $n \in \langle N \rangle$;
6 $\mathbf{A}_B^{(n)} \leftarrow \bar{\mathbf{T}}_{\boldsymbol{\omega}^{(n)}} \mathbf{A}^{(n)}$ for all $n \in \langle N \rangle$;
7 $(\mathbf{X}_B) \leftarrow (\mathbf{A}_B^{(1)}) \diamond (\mathbf{A}_B^{(2)}) \diamond \ldots (\mathbf{A}_B^{(N)})$;
8 Sample *Accept* $\sim$ Bernoulli $\left(\frac{|(1/B)\mathbf{X}_B^{\mathrm{T}}\mathbf{X}_B|}{A}\right)$
9 **until** $Accept = True$;
10 **if** $B > K$ **then**
11 $\boldsymbol{\omega} \leftarrow$ VolumeSample$(\mathbf{X}_B, K)$; `// see Definition 9`
12 Construct $\bar{\mathbf{T}}_{\boldsymbol{\omega}}$ from $\boldsymbol{\omega}$ and q, similar to Equation (19);
13 Obtain $\mathbf{X}_K$ by sampling the rows corresponding to $\boldsymbol{\omega}$ from $\mathbf{X}_B$;
14 **else**
15 Construct $\bar{\mathbf{T}}_{\boldsymbol{\omega}}$ from $\bar{\mathbf{T}}_{\boldsymbol{\omega}^{(n)}}$, for all $n \in \{1, 2, \ldots N\}$ using Equation (25);
16 $\mathbf{X}_K \leftarrow \mathbf{X}_B$;
17 **end**
18 **return** $\mathbf{X}_K$, $\bar{\mathbf{T}}_{\boldsymbol{\omega}}$

factor matrices $\mathbf{A}^{(n)} \in \mathbb{R}^{I_n \times R}$, $n \in \langle N \rangle$ and sample size K as input, Algorithm 2 returns a sampled version $\mathbf{X}_K \in \mathbb{R}^{K \times R}$ of the design matrix $\mathbf{X} = \odot_{n=1}^{N} \mathbf{A}^{(n)}$, and the equivalent sampling matrix $\bar{\mathbf{T}}_{\boldsymbol{\omega}}$. The sampling is performed in two stages, where the first stage involves a size $B = \max(K, 4R^2)$ joint sampling according to distribution in Equation (19). This is followed by a size K volume sampling (Refer Definition 9) on the output of the first stage. The volume rescaled sampling approach by Derezinski et al. (2018) follows a similar two-stage sampling procedure. The next lemma proves that the sampled design matrix $\mathbf{X}_K$ in Algorithm 2 has the same joint sampling distribution as direct size K volume rescaled sampling on $\mathbf{X}$.

Lemma 3. *Let $\bar{\mathbf{S}}_{\boldsymbol{\omega}} \in \mathbb{R}^{K \times I}$ and $\bar{\mathbf{T}}_{\boldsymbol{\omega}} \in \mathbb{R}^{K \times I}$ denote the row sampling matrices on $\mathbf{X} = \odot_{n=1}^{N} \mathbf{A}^{(n)}$ obtained via volume rescaled sampling in Equation (19), and Algorithm 2 respectively. Here $\mathbf{X} \in \mathbb{R}^{I \times R}$, $\mathbf{A}^{(n)} \in \mathbb{R}^{I_n \times R}$, for $n \in \langle N \rangle$ and $I = \prod_{n=1}^{N} I_n$. Then $\bar{\mathbf{S}}_{\boldsymbol{\omega}}$ and $\bar{\mathbf{T}}_{\boldsymbol{\omega}}$ have the same joint sampling distribution.*

Proof. The Algorithm 2 follows a two-step sampling process, where the second stage consists of size K volume sampling on the output of the first stage. The volume rescaled sampling (Derezinski et al., 2018) also follows two stage sampling approach, with a similar second-stage sampling as Algorithm 2. Hence, it is sufficient to show that the sampling probability at the end of first stage in Algorithm 2 (line 8) is same as volume rescaled sampling.

The rows of sampled matrix $\mathbf{X}_B$ in the first stage sampling are constructed (Refer line 7 of Algorithm 2) by taking Hadamard product of the sampled rows from the factor matrices. Thus, the sampled design $\mathbf{X}_B$ is expressed in terms of sampled factor matrices as follows

$$\mathbf{X}_B = \diamond_{n=1}^{N} (\bar{\mathbf{T}}_{\boldsymbol{\omega}^{(n)}} \mathbf{A}^{(n)}) = \diamond_{n=1}^{N} (\bar{\mathbf{T}}_{\boldsymbol{\omega}^{(n)}}^{\mathrm{T}})^{\mathrm{T}} \mathbf{A}^{(n)} \overset{(a)}{=} (\odot_{n=1}^{N} \bar{\mathbf{T}}_{\boldsymbol{\omega}^{(n)}}^{\mathrm{T}})(\odot_{n=1}^{N} \mathbf{A}^{(n)}) \overset{(b)}{=} \bar{\mathbf{T}}_{\boldsymbol{\omega}} \mathbf{X}. \tag{26}$$

Here, equalities (a) and (b) follows from property of Khatri-Rao product in Equation (6), and the definition of $\bar{\mathbf{T}}_{\boldsymbol{\omega}}$ in Equation (25) respectively. Further, the denominator term A of the acceptance probability in line 8 is resolved using Equation (6) as

$$|\diamond_{n=1}^{N} (\mathbf{A}^{(n)})^{\mathrm{T}}\mathbf{A}^{(n)}| = |(\odot_{n=1}^{N}\mathbf{A}^{(n)})^{\mathrm{T}}(\odot_{n=1}^{N}\mathbf{A}^{(n)})| = |\mathbf{X}^{\mathrm{T}}\mathbf{X}| \tag{27}$$

Lastly, the fact that the sampling probability at the end of the first stage in volume rescaled sampling (Derezinski et al., 2018) is given by $\frac{|(1/B)\mathbf{X}^{\mathrm{T}}\mathbf{T}_{\boldsymbol{\omega}}^{\mathrm{T}}\mathbf{T}_{\boldsymbol{\omega}}\mathbf{X}|}{|\mathbf{X}^{\mathrm{T}}\mathbf{X}|}$ completes the proof. □

4.2.2 Concentration and Runtime Analysis : KRVS based Khatri-Rao Product Regression

This section evaluates the accuracy (Theorem 4, Theorem 6) and runtime complexity (Theorem 9) of the proposed KRVS-based solution to the Khatri-Rao product regression. The next theorem highlights the unbiased estimation property of the KRVS-based solution.

Theorem 4. *Consider a linear regression problem* $(\mathbf{X}, \mathbf{y})$*, where* $\mathbf{X} \in \mathbb{R}^{I\times R}$ *is given as* $\mathbf{X} = \odot_{n=1}^{N}\mathbf{A}^{(n)}$*, with optimal solution* $\mathbf{w}^* = \arg\min_{\mathbf{w}} \|\mathbf{X}\mathbf{w} - \mathbf{y}\|_2$*. Let* $\hat{\mathbf{w}} = \arg\min_{\mathbf{w}} \|\bar{\mathbf{T}}_{\boldsymbol{\omega}}\mathbf{X}\mathbf{w} - \bar{\mathbf{T}}_{\boldsymbol{\omega}}\mathbf{y}\|_2$ *denote the KRVS based solution to the regression problem* $(\mathbf{X}, \mathbf{y})$*, where* $\bar{\mathbf{T}}_{\boldsymbol{\omega}} \in \mathbb{R}^{K\times I}$ *is obtained via Algorithm 2. Then,* $\hat{\mathbf{w}}$ *is an unbiased estimator of* $\mathbf{w}^*$*, i.e.,*

$$\mathbb{E}[\hat{\mathbf{w}}] = \mathbf{w}^*. \tag{28}$$

Proof. The proof follows by extending the unbiased estimate property of volume rescaled sampling based solution $\mathbf{w}_{\mathrm{VRS}}$ to the KRVS-based solution $\hat{\mathbf{w}}$. Let $\mathbf{w}_{\mathrm{VRS}} = \arg\min_{\mathbf{w}} \|\bar{\mathbf{S}}_{\boldsymbol{\omega}}\mathbf{X}\mathbf{w} - \bar{\mathbf{S}}_{\boldsymbol{\omega}}\mathbf{y}\|_2$ denote the volume rescaled sampling based solution to the regression problem $(\mathbf{X}, \mathbf{y})$. Then, the following result is stated from (Derezinski et al., 2018, Theorem 3)

$$\mathbb{E}[\mathbf{w}_{\mathrm{RVS}}] = \mathbf{w}^*. \tag{29}$$

Further, the proof is completed using Lemma 3, which states that volume rescaled sampling matrix $\bar{\mathbf{S}}_{\boldsymbol{\omega}}$ and KRVS matrix $\bar{\mathbf{T}}_{\boldsymbol{\omega}}$ have the same joint sampling distribution. □

The next proposition gives a way of choosing the probability distribution $\mathbf{q}^{(n)}$ for the rows of $\mathbf{A}^{(n)}$, $n \in \langle N\rangle$.

Proposition 5. *Choose* $\mathbf{q}^{(n)} = \frac{1}{R}\left(l_1^{(n)}, l_2^{(n)}, \dots l_{I_n}^{(n)}\right)$ *as the sampling distribution for* n*th factor matrix* $\mathbf{A}^{(n)}$*, where* $l_i^{(n)}$ *denotes the leverage score of* i*th row of* $\mathbf{A}^{(n)}$*. The equivalent sampling distribution* $q = \odot_{n=1}^{N}\mathbf{q}^{(n)}$ *for the rows of* $\mathbf{X} = \odot_{n=1}^{N}\mathbf{A}^{(n)}$ *is given by* $q = \frac{1}{R^N}(\tau_1, \tau_2, \dots \tau_I)$*, where*

$$\tau_i = \prod_{n=1}^{N} l_{i_n}^{(n)}, \text{ such that, } i = i_N + (i_{N-1} - 1)I_N + \sum_{g=1}^{N-2}(i_g - 1)\prod_{h=g+1}^{N} I_h. \tag{30}$$

Remark 1 (Cheng et al. (2016))**.** *Denote* $l_i :=$ *leverage score of* i*th row of* $\mathbf{X}$*, then* $l_i \leq \tau_i$*,* $\forall i \in \langle I\rangle$*.*

Relative error approximation: The next theorem gives our main result i.e., $(1+\epsilon)$ relative error approximation property of the proposed KRVS based Khatri-Rao product regression.

Theorem 6 (Relative Error-KRVS)**.** *Consider a linear regression problem* $(\mathbf{X}, \mathbf{y})$ *with optimal solution* $\mathbf{w}^* = \arg\min_{\mathbf{w}} \|\mathbf{X}\mathbf{w} - \mathbf{y}\|_2$*, where* $\mathbf{X} \in \mathbb{R}^{I\times R}$*,* $\mathbf{X} = \odot_{n=1}^{N}\mathbf{A}^{(n)}$*,* $\mathbf{A}^{(n)} \in \mathbb{R}^{I_n\times R}$*, and* $\prod_{n=1}^{N} I_n = I$*. Let* $\bar{\mathbf{T}}_{\boldsymbol{\omega}} \in \mathbb{R}^{K\times I}$ *denote the KRVS matrix on* $\mathbf{X}$*, and* $\hat{\mathbf{w}} := \arg\min_{\mathbf{w}} \|\bar{\mathbf{T}}_{\boldsymbol{\omega}}\mathbf{X}\mathbf{w} - \bar{\mathbf{T}}_{\boldsymbol{\omega}}\mathbf{y}\|_2$*. For* $K \geq R^N \max\{C\log(R/\delta), 1/(\epsilon\delta)\}$*, where* $\epsilon, \delta \in (0,1)$*, the following inequality holds with probability at least* $1-\delta$

$$\|\mathbf{X}\hat{\mathbf{w}} - \mathbf{y}\|_2^2 \leq (1+\epsilon)\|\mathbf{X}\mathbf{w}^* - \mathbf{y}\|_2^2. \tag{31}$$

Denote ith row of $\mathbf{X}$ as $\mathbf{x}_i$. The key idea to prove Theorem 6 is to note that KRVS is essentially a weighted sampling matrix (see Definition 8), for an appropriately defined marginal distribution p_i i.e., probability of sampling row $\mathbf{x}_i$, $\forall i \in \{1, 2, \dots I\}$. Further, obtaining the value of p_i allows us to extend the relative error

approximation guarantee for weighted sampling (Larsen & Kolda, 2022, Theorem 6) to KRVS. Recall that the leverage score of $\mathbf{x}_i$ is given as $\mathbf{x}_i(\mathbf{X}^{\mathrm{T}}\mathbf{X})^{-1}\mathbf{x}_i^{\mathrm{T}}$ (Mahoney et al., 2011). Let $\mathbf{q} = (q_1, q_2, \dots q_I)$ denote a probability distribution on rows of $\mathbf{X}$. The next definition is adapted from (Dereziński et al., 2022, Defn. 2.5), and gives a generalization of leverage scores to arbitrary distribution $\mathbf{q}$.

Definition 12 (Generalized leverage scores)**.** *Denote $\mathbf{q} = (q_1, q_2, \dots q_I)$ as probability distribution on the rows of $\mathbf{X} \in \mathbb{R}^{I\times R}$, such that $\Pr[\mathbf{x}_i] = q_i$. Then, the leverage score $Lev_{\mathbf{q}}(\mathbf{x}_i)$ of ith row $\mathbf{x}_i$ of $\mathbf{X}$, with respect to the distribution $\mathbf{q}$ is given as*

$$\mathrm{Lev}_{\mathbf{q}}(\mathbf{x}_i) = \frac{\mathbf{x}_i \Theta_{\mathbf{q}}^{-1} \mathbf{x}_i^T}{R}, \text{ where } \Theta_q := \sum_{i=1}^{I} q_i \mathbf{x}_i^T \mathbf{x}_i. \tag{32}$$

The next lemma employs the notion of generalized leverage scores to give the marginal probability for KRVS.

Lemma 7 (Marginal probability for KRVS)**.** *Let $\mathbf{\Xi} = (\mathbf{x}_{\omega_1}, \mathbf{x}_{\omega_2}, \dots \mathbf{x}_{\omega_K})$ denote the set of K rows sampled from $\mathbf{X} \in \mathbb{R}^{I\times R}$ via KRVS in Algorithm 2, $\mathbf{q} = (q_1, q_2, \dots q_I)$ denote the equivalent distribution on rows of $\mathbf{X}$. Then, marginal probability of sampling ith row $\mathbf{x}_i$ is*

$$p_i := \Pr[\mathbf{x}_i \in \mathbf{\Xi}] = \left(\frac{R}{K}\right) \mathrm{Lev}_{\mathbf{q}}(\mathbf{x}_i) + \left(1 - \frac{R}{K}\right) q_i. \tag{33}$$

Proof. Denote $\mathbf{\Phi} := (\mathbf{x}_{\phi_1}, \mathbf{x}_{\phi_2}, \dots \mathbf{x}_{\phi_K})$ as a set of rows sampled with replacement from $\mathbf{X}$ via a direct size K volume rescaled sampling in Equation (19). Then, (Dereziński et al., 2022, Theorem 2.7) gives the marginal probability $\Pr[\mathbf{x}_i \in \mathbf{\Phi}] = (R/K)\mathrm{Lev}_{\mathbf{q}}(\mathbf{x}_i) + (1 - R/K)q_i$. Further, Lemma 3 states that sampled rows $\mathbf{\Xi}$ via KRVS have same probability distribution as $\mathbf{\Phi}$, and completes the proof. □

The next theorem states that the weighted sampling (See Definition 8) based solution to the linear regression problem achieves relative error approximation of the optimal solution, for an appropriate sample size K.

Theorem 8. *(Larsen & Kolda, 2022, Theorem 6) Consider a linear regression problem $(\mathbf{X}, \mathbf{y})$, with optimal solution $\mathbf{w}^*$, where $\mathbf{X} \in \mathbb{R}^{I\times R}$, $I \gg R$, and rank$(\mathbf{X}) = R$. Let $\mathbf{p} \in [0, 1]^I$ be a probability distribution and assume there exists a fixed $\beta \in (0, 1]$ such that*

$$\beta \le \min_{i\in\langle I\rangle} \frac{p_i R}{l_i(\mathbf{X})}, \forall i \in \langle I\rangle, \tag{34}$$

where $l_i(\mathbf{X})$ denotes leverage score of ith row of $\mathbf{X}$. For any $\epsilon, \delta \in (0, 1)$, let $K = (R/\beta)\max\{C\log(R/\delta), (1/\delta\epsilon)\}$, where $C = 144/(1 - \sqrt{2})^2$ and let $\mathbf{S} \sim$ RANDSAMPLE$(K, \mathbf{p})$. Then $\bar{\mathbf{w}} := \arg\min_{\mathbf{w}} \|\mathbf{SXw} - \mathbf{Sy}\|_2^2$ satisfies the following with probability at least $1 - \delta$

$$\|\mathbf{X}\bar{\mathbf{w}} - \mathbf{y}\|_2^2 \le (1 + \epsilon)\|\mathbf{Xw}^* - \mathbf{y}\|_2^2. \tag{35}$$

The Lemma 7 and Theorem 8 are together employed to construct the proof of Theorem 6 as follows

Proof of Theorem 6

Proof. Let $p_i \in [0, 1]$ denote the marginal probability of sampling $\mathbf{x}_i$, i.e., ith row of $\mathbf{X}$ via KRVS, given by Equation (33), l_i denote the leverage score of $\mathbf{x}_i$. Recall from Proposition 5 that $\mathbf{q} = \frac{1}{R^N}(\tau_1, \tau_2, \dots \tau_I)$, where τ_i is defined in Equation (30). Then,

$$\frac{p_i R}{l_i} \overset{(a)}{=} \frac{R}{l_i}\left(\frac{R}{K}\mathrm{Lev}_{\mathbf{q}}(\mathbf{x}_i) + \left(1 - \frac{R}{K}\right)\frac{\tau_i}{R^N}\right) \overset{(b)}{\ge} \frac{R}{l_i}\left(1 - \frac{R}{K}\right)\frac{\tau_i}{R^N} \overset{(c)}{\ge} \frac{R}{l_i}\left(1 - \frac{R}{K}\right)\frac{l_i}{R^N} = \frac{(1 - R/K)}{R^{N-1}}.$$

In the above expression, Equations (a), (b), and (c) follows from Lemma 7, Definition 12 (since generalized leverage scores are non-negative) and Remark 1, respectively. Further, denote $\beta \in (0, 1]$ as follows

$$\beta := \min_{i\in\langle I\rangle} \frac{p_i R}{l_i} \le \frac{1 - (R/K)}{R^{N-1}} \le \frac{1}{R^{N-1}}. \tag{36}$$

Hence, relative error guarantee in Equation (31) is proved by extending Theorem 8 to KRVS matrix $\bar{\mathbf{T}}_{\boldsymbol{\omega}}$, for $K \geq (R/\beta)\max\{C\log(R/\delta), 1/(\epsilon\delta)\} = R^N \max\{C\log(R/\delta), 1/(\epsilon\delta)\}$. □

The asymptotic runtime complexity of proposed KRVS based Khatri-Rao product regression is stated next.

Theorem 9 (Runtime complexity-KRVS)**.** *Consider a linear regression problem* $\mathbf{w}^* = \arg\min_{\mathbf{w}} \|\mathbf{X}\mathbf{w} - \mathbf{y}\|_2^2$, *where* $\mathbf{X} \in \mathbb{R}^{I\times R}$, $\mathbf{X} = \odot_{n=1}^N \mathbf{A}^{(n)}$, $\mathbf{A}^{(n)} \in \mathbb{R}^{I_n \times R}$, *and* $I = \prod_{n=1}^N$. *Let* $\hat{\mathbf{w}} = \|\mathbf{X}_K\mathbf{w} - \bar{\mathbf{T}}_{\boldsymbol{\omega}}\mathbf{y}\|_2^2$ *denote the KRVS based solution, where* $\mathbf{X}_K$ *and* $\bar{\mathbf{T}}_{\boldsymbol{\omega}} \in \mathbb{R}^{K\times I}$ *are obtained using Algorithm 2. Then, the asymptotic runtime complexity of computing* $\hat{\mathbf{w}}$ *is* $O((\sum_{n=1}^N I_n + 4R^2 + 2K)R^2\log(1/\delta))$ *with probability at least* $1-\delta$.

Proof. The steps $1-3$ and $10-18$ have a deterministic runtime of $O(\sum_{n=1}^N I_n R^2 + BR^2)$ due to the cost of computing leverage scores $\mathbf{q}^{(n)}$ of factor matrices $\mathbf{A}^{(n)}$ and volume sampling of $\mathbf{X}_B$. Refer the loop in lines $4-9$ as the sampling loop. The overall runtime complexity of the sampling loop is upper-bounded as follows. Let p denote the number of iterations of the sampling loop. Then, the expected value of p is

$$\mathbb{E}[p] = \sum_{\boldsymbol{\omega}\in\langle I\rangle^B} \frac{(1/B)|\mathbf{X}_B^{\mathrm{T}}\mathbf{X}_B|}{|\mathbf{X}^{\mathrm{T}}\mathbf{X}|}\prod_{b=1}^B q_{\omega_b} = \frac{1}{B^R}\sum_{\boldsymbol{\omega}\in\langle I\rangle^B} \frac{|\mathbf{X}_B^{\mathrm{T}}\mathbf{X}_B|}{|\mathbf{X}^{\mathrm{T}}\mathbf{X}|}\prod_{b=1}^B q_{\omega_b} \tag{37}$$

Since $\bar{\mathbf{T}}_{\boldsymbol{\omega}}$ and volume rescaled sampling matrix $\bar{\mathbf{S}}_{\boldsymbol{\omega}}$ have same joint sampling distributions (Lemma 3), the Equation (37) is further resolved using the normalization constant (20) as

$$\begin{aligned}\mathbb{E}[p] = \frac{B(B-1)\dots(B-R+1)}{B^R} &= (1-1/B)\,(1-2/B)\dots(1-R/B) \geq (1-R/B)^R \\ &\overset{(a)}{\geq} \left(1-R^2/B\right) \overset{(b)}{\geq} 3/4,\end{aligned}$$

where inequalities (a) and (b) follow from Bernoulli's inequality, $(1+x)^n \geq 1+nx$, and the fact that $B \geq 4R^2$). Thus, the expected value of acceptance probability is lower bounded by $(3/4)$. Further application of Markov's inequality yields that the number of iterations p of the sampling loop is upper bounded by $p \leq \log(1/\delta)/\log(4/3)$. The operations inside the sampling loop are constructing $\mathbf{X}_B \in \mathbb{R}^{B\times R}$ and computing the determinant $|\mathbf{X}_B^{\mathrm{T}}\mathbf{X}_B|$, with runtime complexities $O(BRN)$ and $O(BR^2)$, respectively. Hence, the overall runtime complexity of the sampling loop is $O(BR^2\log(1/\delta))$. The overall runtime complexity of Algorithm 2 is obtained by substituting $B = \max(K, 4R^2)$ as $O((\sum_{n=1}^N I_n + 4R^2 + K)R^2\log(1/\delta))$. Adding the cost $O(KR^2)$ of solving sampled regression completes the proof. □

Remark 2. *For* $K = O(R^2)$, *sampling* K *rows from* $\mathbf{X} = \odot_{n=1}^N \mathbf{A}^{(n)}$, $\mathbf{A}^{(n)} \in \mathbb{R}^{I\times R}$ *via KRVS has asymptotic runtime complexity of* $O((NI+K)R^2)$.

5 KRVS based Fast CP Decomposition

This section build on the results from KRVS based Khatri-Rao product regression to give an ALS algorithm for CP decomposition. The proposed algorithm, along with its concentration and runtime complexity analysis are provided in the Sections 5.1, 5.2.

5.1 Algorithm for CP decomposition

Consider a mode N tensor $\mathcal{X} \in \mathbb{R}^{I\times I\times\dots I}$. The rank R CP decomposition of $\mathcal{X}$ gives N factor matrices $\mathbf{A}^{(n)} \in \mathbb{R}^{I\times R}$ such that

$$\mathcal{X} = \mathcal{I} \times_1 \mathbf{A}^{(1)} \times_2 \mathbf{A}^{(2)} \dots \mathbf{A}^{(N)}. \tag{38}$$

Here $\mathcal{I} \in \mathbb{R}^{R\times R\times\dots R}$ is an identity tensor of order R. The Equation (38) is compactly denoted as $\mathcal{X} = [\![\mathcal{I}; \mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \dots \mathbf{A}^{(N)}]\!]$. The CP-ALS algorithm (Kolda & Bader, 2009, Fig 3.3) iteratively computes the estimates $\hat{\mathbf{A}}^{(n)}$ of factor matrices $\mathbf{A}^{(n)}$, $n \in \langle N\rangle$, such that $\hat{\mathcal{X}} = [\![\mathcal{I}; \hat{\mathbf{A}}^{(1)}, \hat{\mathbf{A}}^{(2)}, \dots \hat{\mathbf{A}}^{(N)}]\!]$ is close to $\mathcal{X}$, i.e., $\|\mathcal{X} - \hat{\mathcal{X}}\|_F$ is small. The convergence criteria in CP-ALS (Kolda & Bader, 2009, Fig 3.3) can be set

on the basis of residual error $r = \|\mathcal{X} - \hat{\mathcal{X}}\|_F$. The asymptotic runtime complexity of each iteration of CP-ALS is $O(N(R+I)I^{N-1}R)$, driven mainly by high dimensionality of Khatri-Rao product. Denote $\hat{\mathbf{A}}_{\neq n} := \odot_{m=N,m\neq n}^{1} \hat{\mathbf{A}}^{(m)}$, such that $\hat{\mathbf{A}}_{\neq n} \in \mathbb{R}^{I^{N-1}\times R}$. Then, the update equation of nth factor matrix $\hat{\mathbf{A}}^{(n)}$ in CP-ALS is given as

$$\hat{\mathbf{A}}^{(n)} = \arg\min_{\mathbf{A}^{(n)}} \|\hat{\mathbf{A}}_{\neq n}\mathbf{A}^{(n)\mathrm{T}} - \mathbf{X}_{(n)}^{\mathrm{T}}\|_F^2 = \left(\hat{\mathbf{A}}_{\neq n}\right)^{\dagger} \mathbf{X}_{(n)}^{\mathrm{T}}. \tag{39}$$

The problem in Equation (39) is a Khatri-Rao product regression, and can be efficiently solved using KRVS. Let $\mathbf{Z}_{\neq n} \in \mathbb{R}^{K\times R}$ denote the sampled design matrix, obtained via KRVS on $\hat{\mathbf{A}}_{\neq n}$ using Algorithm 2. The corresponding sampled data is denoted by $\bar{\mathbf{X}}_{(n)}^{\mathrm{T}} = \bar{\mathbf{T}}_n \mathbf{X}_{(n)}^{\mathrm{T}}$, where $\bar{\mathbf{T}}_n \in \mathbb{R}^{K\times I^{N-1}}$ is KRVS matrix. Then, the Khatri-Rao product regression problem in Equation (39) is efficiently solved as

$$\bar{\mathbf{A}}^{(n)} = \arg\min_{\mathbf{A}^{(n)}} \|\mathbf{Z}_{\neq n}\mathbf{A}^{(n)\mathrm{T}} - \bar{\mathbf{X}}_{(n)}^{\mathrm{T}}\|_F^2 = (\mathbf{Z}_{\neq n})^{\dagger} \bar{\mathbf{X}}_{(n)}^{\mathrm{T}}. \tag{40}$$

Algorithm 3, known henceforth as CP-KRVS, summarizes the KRVS-based CP decomposition of $\mathcal{X}$. In line 4, the notation $\{\bar{\mathbf{A}}_{(\neq i)}\}$ refers to set of matrices $\bar{\mathbf{A}}^{(m)}$, where $m \in \{1, 2, \ldots N\}$ and $m \neq i$. The tensor $\mathcal{X}$, CP rank R, and sample size K are given as input to CP-KRVS, and its CP decomposition $[\![\mathcal{I}; \bar{\mathbf{A}}^{(1)}, \bar{\mathbf{A}}^{(2)}, \ldots \bar{\mathbf{A}}^{(N)}]\!]$ is obtained as output. The next subsection highlights the accuracy and runtime complexity of CP-KRVS.

Algorithm 3: KRVS based CP decomposition (CP-KRVS)

Data: $\mathcal{X} \in \mathbb{R}^{I\times I\times\ldots I}$, Sample size $K \geq R$
Result: $\bar{\mathbf{A}}^{(n)} \in \mathbb{R}^{I\times R}$, for $n \in \langle N\rangle$
1 Initialize $\bar{\mathbf{A}}^{(n)}$, $n = \{2, 3, \ldots N\}$;
2 **repeat**
3 **for** $i = 1 : N$ **do**
4 $[\mathbf{Z}_{\neq i}, \bar{\mathbf{T}}_i] \leftarrow \mathrm{KRVS}(\{\bar{\mathbf{A}}_{(\neq i)}\}, K)$; `// see Algorithm 2`
5 $\bar{\mathbf{X}}_{(i)}^{\mathrm{T}} \leftarrow \bar{\mathbf{T}}_i \mathbf{X}_{(i)}^{\mathrm{T}}$;
6 $\bar{\mathbf{A}}^{(i)} \leftarrow (\mathbf{Z}_{\neq i})^{\dagger}\bar{\mathbf{X}}_{(i)}^{\mathrm{T}}$
7 **end**
8 **until** *Until Convergence*;
9 **return** $\bar{\mathbf{A}}^{(n)} \in \mathbb{R}^{I\times R}$, for $n \in \langle N\rangle$

5.2 Accuracy and Runtime Complexity of CP-KRVS

This section constitutes of two key results in Theorem 10 and Theorem 11, that describe accuracy and runtime complexity of CP-KRVS respectively. The Theorem 10 stated next shows that the factor matrix update in proposed CP-KRVS achieves a $(1+\epsilon)$ relative error approximation of the ALS based optimal factor matrix update.

Theorem 10 (Relative Error Approximation : CP-KRVS)**.** *Consider the rank R CP decomposition of a mode N tensor $\mathcal{X} \in \mathbb{R}^{I\times I\times\ldots I}$, where $I \gg R$. Let $\hat{\mathbf{A}}^{(n)} = \arg\min_{\mathbf{A}^{(n)}} \|\hat{\mathbf{A}}_{\neq n}\mathbf{A}^{(n)T} - \mathbf{X}_{(n)}^{T}\|_F^2$ denote the optimal nth factor matrix update using ALS, where $\mathbf{X}_{(n)}$ denotes mode n matricization of $\mathcal{X}$ and $\mathbf{A}_{\neq n} := \odot_{m=N,m\neq n}^{1}\mathbf{A}^{(m)}$. Denote $\bar{\mathbf{A}}^{(n)} := \arg\min_{\mathbf{A}^{(n)}} \|\hat{\mathbf{Z}}_{\neq n}\mathbf{A}^{(n)T} - \bar{\mathbf{X}}_{(n)}^{T}\|_F^2$, where $\hat{\mathbf{Z}}_{\neq n} \in \mathbb{R}^{K\times R}$ is sampled design matrix from KRVS, $\bar{\mathbf{T}}_n \in \mathbb{R}^{K\times I^{N-1}}$ is KRVS matrix, and $\bar{\mathbf{X}}_{(n)}^{T} = \bar{\mathbf{T}}_n\mathbf{X}_{(n)}^{T}$. For $K \geq R^{(N-1)}\max\{C\log(R/\delta), 1/(\epsilon\delta)\}$, $\epsilon, \delta \in (0,1)$, the following holds with probability at least $1-\delta$*

$$\|\hat{\mathbf{A}}_{\neq n}\bar{\mathbf{A}}^{(n)T} - \bar{\mathbf{X}}_{(n)}^{T}\|_F^2 \leq (1+\epsilon)\|\hat{\mathbf{A}}_{\neq n}\hat{\mathbf{A}}^{(n)T} - \bar{\mathbf{X}}_{(n)}^{T}\|_F^2. \tag{41}$$

Proof. The nth factor matrix update $\bar{\mathbf{A}}^{(n)}$ is obtained as a KRVS based solution to the regression problem $\arg\min_{\mathbf{A}^{(n)}} \|\hat{\mathbf{A}}_{\neq n}\mathbf{A}^{(n)\mathrm{T}} - \mathbf{X}_{(n)}^{\mathrm{T}}\|_F^2$, where $\mathbf{A}_{\neq n} := \odot_{m=N,m\neq n}^{1}\mathbf{A}^{(m)}$ is Khatri-Rao product of $(N-1)$ factor matrices. The proof completes by extending Theorem 6 to the solution $\bar{\mathbf{A}}^{(n)}$, for appropriate sample size $K \geq R^{(N-1)}\max\{C\log(R/\delta), 1/(\epsilon\delta)\}$. □

The per-iteration runtime complexity of CP-KRVS algorithm is stated next.

Theorem 11 (Runtime Complexity : CP-KRVS)**.** *The rank R CP decomposition of a mode N tensor $\mathcal{X} \in \mathbb{R}^{I \times I \times \ldots I}$, where $I \gg R$, using CP-KRVS in Algorithm 3 at a sample size K, has a per-iteration runtime complexity of $O(N(NIR + 4R^3 + KI)R)$.*

Proof. Consider the subproblem of updating ith factor matrix in a given iteration of Algorithm 3. The dominant steps and their runtime complexities are given as

1. (**line** 3) Sampling the design matrix $\bar{\mathbf{A}}_{\neq i} := \odot_{m=N, m\neq i}^{1} \bar{\mathbf{A}}^{(i)}$ via KRVS (Algorithm 2) to obtain $\mathbf{Z}_{\neq i}$ and $\bar{\mathbf{T}}_i$ with runtime $O((N-1)IR^2 + 4R^4 + KR^2)$. Refer Theorem 9 for more details.
2. (**line** 5) Computing $(\mathbf{Z}_{\neq i})^{\dagger}$ with a runtime $O(KR^2)$, i.e., SVD complexity of $\mathbf{Z}_{\neq i} \in \mathbb{R}^{K \times R}$, $K > R$.
3. (**line** 5) Computing the update $\bar{\mathbf{A}}^{(i)} = (\mathbf{Z}_{\neq i})^{\dagger} \bar{\mathbf{X}}_{(i)}^{\mathrm{T}}$ with runtime $O(KRI)$, due to matrix multiplication complexity of $\mathbf{Z}_{\neq i}$ and $\bar{\mathbf{X}}_{(i)}^{\mathrm{T}} \in \mathbb{R}^{K \times I}$.

Hence, the overall runtime of ith factor matrix update in a given iteration resolves to $O(NIR^2 + 4R^4 + KRI)$. Combining the runtime for N factor matrices completes the proof. □

6 Numerical Results

6.1 Khatri-Rao Product Regression

This section compares the performance of proposed KRVS approach for Khatri-Rao product regression (Section 6.1) and CP decomposition (Section 6.2, 6.3), with the leverage score sampling by Cheng et al. (2016); Larsen & Kolda (2022); Bharadwaj et al. (2023), and optimal least-squares based solution.

Simulation Setup: Let $(\mathbf{X}, \mathbf{y})$ denote a Khatri-rao product regression, where $\mathbf{X} \in \mathbb{R}^{I \times R}$, $\mathbf{X} = \mathbf{A}^{(1)} \odot \mathbf{A}^{(2)}$, $\mathbf{A}^{(n)} \in \mathbb{R}^{I_n \times R}$ for $n = \{1, 2\}$. Further, the observation $\mathbf{y} \in \mathbb{R}^{I \times 1}$, $\mathbf{y} = \mathbf{y}^{(1)} \odot \mathbf{y}^{(2)}$. The factors $\mathbf{A}^{(1)}$, $\mathbf{y}^{(1)}$ and $\mathbf{A}^{(2)}$, $\mathbf{y}^{(2)}$ are sampled from *abalone*, *cpusmall*, and *mg* regression datasets of *libsvm* library (Chang & Lin, 2011). The following values are considered for simulation $I_1 = I_2 = 500$, $R = 6$. The experiments were run using Python environment, on a system with an Intel Core $i7 - 14700$ processor and 48GB RAM.

Dataset	Instances	Features
abalone	4177	36
cpusmall	8192	12
mg	1385	21

Table 2: The regression datasets *abalone*, *housing*, and *mg* from the *libsvm* library by Chang & Lin (2011).

Baselines and Evaluation Metric: Let $\bar{\mathbf{w}} \in \mathbb{R}^{R \times 1}$ denote a solution to $(\mathbf{X}, \mathbf{y})$. Then, the regression error corresponding to $\bar{\mathbf{w}}$ is defined as $\frac{1}{I}\|\mathbf{X}\bar{\mathbf{w}} - \mathbf{y}\|_2^2$. The regression error is considered as a metric to evaluate the performance of the proposed KRVS-based solution $\hat{\mathbf{w}}$ against the following baselines.

- [Implicit leverage score sampling (ImpLSS)] $\bar{\mathbf{w}} := \arg\min_{\mathbf{w} \in \mathbb{R}^{R \times 1}} \|\mathbf{SXw} - \mathbf{Sy}\|_2^2$, where $\mathbf{S} \in \mathbb{R}^{K \times I}$, $K \ll I$ is constructed using implicit leverage score sampling (Larsen & Kolda, 2022, Lemma 9).
- [Exact leverage score sampling (ExactLSS)] $\tilde{\mathbf{w}} := \arg\min_{\mathbf{w} \in \mathbb{R}^{R \times 1}} \|\mathbf{PXw} - \mathbf{Py}\|_2^2$, where sampling matrix $\mathbf{P} \in \mathbb{R}^{K \times I}$, $K \ll I$ is constructed using exact leverage scores [a] of $\mathbf{X}$ (Bharadwaj et al., 2023, Theorem 1.1).
- [Least-Squares (LS)] $\mathbf{w}^* = \arg\min \|\mathbf{Xw} - \mathbf{y}\|_2^2$, least-squares based optimal solution.

The runtime of KRVS, ImpLSS, ExactLSS, and LS based solutions is also considered for comparison.

Insights: Fig. 1 shows the regression error *v/s* sample size variation for LS, KRVS, ImpLSS, and ExactLSS solutions to $(\mathbf{X}, \mathbf{y})$, across the *abalone* ((1a), (1b)), *cpusmall* ((1c),(1d)), and *mg* ((1e),(1f)) datasets. The sample sizes considered are $K = \{15, 20, 25, 30, 35, 40, 45, 50, 100, 200, 400, 800\}$. At smaller sample sizes,

[a] Adapted from: github.com/vbharadwaj-bk/fast-tensor-leverage/tree/main/reference-implementation

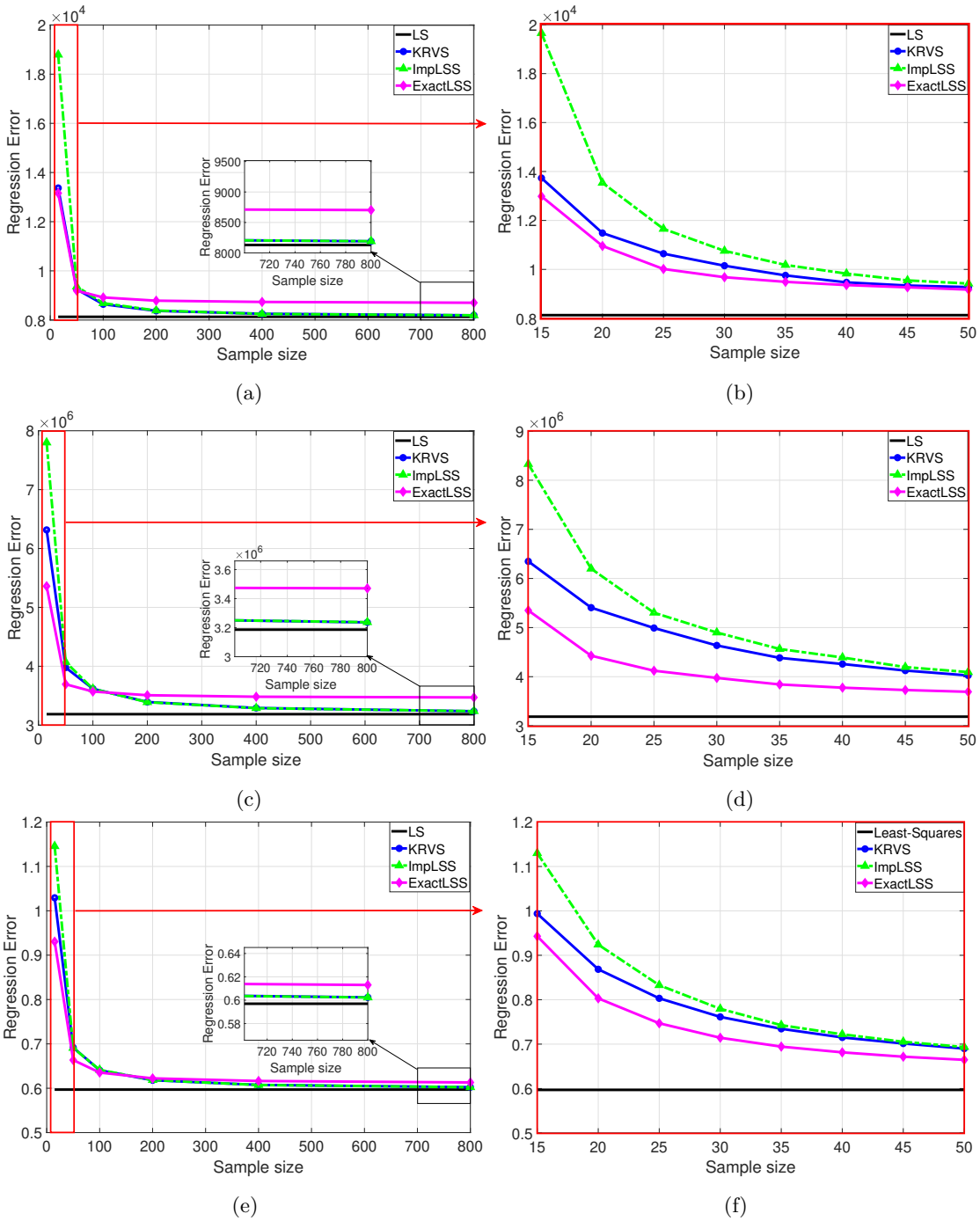


Figure 1: Comparison of regression error for the proposed KRVS (Algorithm 2), with implicit leverage score sampling (ImpLSS) by Cheng et al. (2016); Larsen & Kolda (2022), exact leverage score sampling (ExactLSS) by Bharadwaj et al. (2023), and optimal least-squares (LS) solutions to the regression problem $(\mathbf{X}, \mathbf{y})$. Here $\mathbf{X} = \mathbf{C} \odot \mathbf{B}$, $\mathbf{y} = \mathbf{y}_C \odot \mathbf{y}_B$, such that $\mathbf{C} \in \mathbb{R}^{500\times 6}$, $\mathbf{B} \in \mathbb{R}^{500\times 6}$, $\mathbf{y}_C \in \mathbb{R}^{500\times 1}$, and $\mathbf{y}_B \in \mathbb{R}^{500\times 1}$ are chosen from *abalone* ((1a), (1b)), *cpusmall* ((1c), (1d)), and *mg* ((1e), (1f)) datasets of *libsvm* Chang & Lin (2011) library.

	abalone		***cpusmall***		***mg***	
Solution	$(K = 200)$	$(K = 800)$	$(K = 200)$	$(K = 800)$	$(K = 200)$	$(K = 800)$
ImpLSS	0.00022	0.00058	0.00023	0.00075	0.00022	0.00057
KRVS (**This work**)	0.00040	0.00079	0.00041	0.00093	0.00039	0.00077
ExactLSS	0.01929	0.07320	0.01962	0.07370	0.01933	0.07356
LS	0.02438	0.02438	0.02674	0.02674	0.02584	0.02584

Table 3: Runtime (in sec.) comparison of sampling based solutions for the Khatri-Rao product regression $(\mathbf{X}, \mathbf{y})$, where $\mathbf{X} \in \mathbb{R}^{I \times R}$, $\mathbf{X} = \mathbf{A}^{(1)} \odot \mathbf{A}^{(2)}$, $\mathbf{y} = \mathbf{y}^{(1)} \odot \mathbf{y}^{(2)}$. The factors $\mathbf{A}^{(1)} \in \mathbb{R}^{500 \times 6}$, $\mathbf{y} \in \mathbb{R}^{500 \times 1}$ and $\mathbf{A}^{(2)} \in \mathbb{R}^{500 \times 6}$, $\mathbf{y} \in \mathbb{R}^{500 \times 1}$ are chosen from the datasets *cpusmall*, *abalone*, and *mg* of *libsvm* (Chang & Lin, 2011). The sampling matrix $\mathbf{S} \in \mathbb{R}^{K \times I}$ is constructed using KRVS (Algorithm 2), ImpLSS by Cheng et al. (2016); Larsen & Kolda (2022), and ExactLSS by Bharadwaj et al. (2023).

the ExactLSS solution is observed to have least regression error, closely followed by the proposed KRVS solution. This behavior is attributed to smaller sample size requirement of ExactLSS based approach, for an $(1 + \epsilon)$ relative error approximation (see Table 1). The ImpLSS solution is seen to be suffering from low accuracy at smaller sample sizes. Further, an increase in the value of sample size K for KRVS, ImpLSS, and ExactLSS based solutions results in smaller regression error at the cost of runtime complexity (see Table 1). This highlights the runtime *v/s* accuracy trade-off in sampling-based solutions to the Khatri-Rao product regression. Lastly, the regression error of ExactLSS, ImpLSS, and KRVS solutions converge to that of optimal LS based solution at higher sample sizes. This highlights the accuracy of sampling based approximation to Khatri-Rao product regression.

Table 3 summarizes the runtime of ImpLSS, KRVS, ExactLSS, and LS based solutions to the Khatri-Rao product regression $(\mathbf{X}, \mathbf{y})$. The runtime is evaluated for sample size $K \in \{200, 800\}$ across the *abalone*, *cpusmall*, and *mg* datasets. The proposed KRVS solution achieves a significantly faster runtime than LS ($60\times$ or more at $K = 200$) and ExactLSS ($47\times$ or more at $K = 200$) solutions. The empirical speed-up observed in KRVS concurs with the asymptotic runtime (see Table 1), and is attributed to its lower sampling time compared to ExactLSS. The ImpLSS exhibits slightly faster runtime ($1.7\times$ at $K = 200$) than KRVS, but it has low accuracy at smaller sample sizes (see Figures (1b), (1d), (1f)). Lastly, the increased sample size K in KRVS, ImpLSS, and ExactLSS results in a higher runtime, reaffirming their runtime *v/s* accuracy trade-off.

In summary, the KRVS achieves a comparable accuracy to ExactLSS, and LS solutions at a significantly faster runtime. Further, the KRVS exhibits higher accuracy than ImpLSS solution at small sample sizes, with a modest extra runtime cost. Thereby, the proposed KRVS achieves the best runtime *v/s* accuracy trade-off at smaller sample sizes, and is preferable among the baseline sampling approaches to solve Khatri-Rao product regression.

6.2 CP Decomposition

Simulation Setup: Consider the CP decomposition of a tensor $\mathcal{Y} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$ at rank R, where the $\mathcal{Y}$ is generated as follows

$$\mathcal{Y} = \mathcal{Y}_1 + \mathcal{V} = \mathcal{I} \times_1 \mathbf{A} \times_2 \mathbf{B} \times_3 \mathbf{C} + \mathcal{V} \tag{42}$$

$$\mathcal{Y} = \mathcal{Y}_2 + \mathcal{V} = \text{SparsifyData}(\mathcal{Y}_1, p) + \mathcal{V} \tag{43}$$

The entries of $\mathbf{A} \in \mathbb{R}^{I_1 \times R}$, $\mathbf{B} \in \mathbb{R}^{I_2 \times R}$, and $\mathbf{C} \in \mathbb{R}^{I_3 \times R}$ are generated i.i.d. from the standard normal distribution i.e., $\mathcal{N}(0, 1)$. The function SparsifyData takes a dense tensor $\mathcal{Y}_1$ and probability measure p as inputs, and returns a sparse version of $\mathcal{Y}_1$ retaining the non-zero values with probability p. The tensors $\mathcal{Y}_1$ and $\mathcal{Y}_2$ denote instances of a random dense and sparse tensor, and have been previously used (Ma & Solomonik, 2021) for benchmarking tensor decomposition algorithms. Further, $\mathcal{Y}$ is assumed to be corrupted by noise $\mathcal{V}$, where the entries of $\mathcal{V}$ are generated from a normal distribution with mean zero, and variance $\sigma = (\eta / \max\{I_1, I_2, I_3\})$. The following values are considered for experiments: $I_1 = I_2 = I_3 = 100$, $R = 10$,

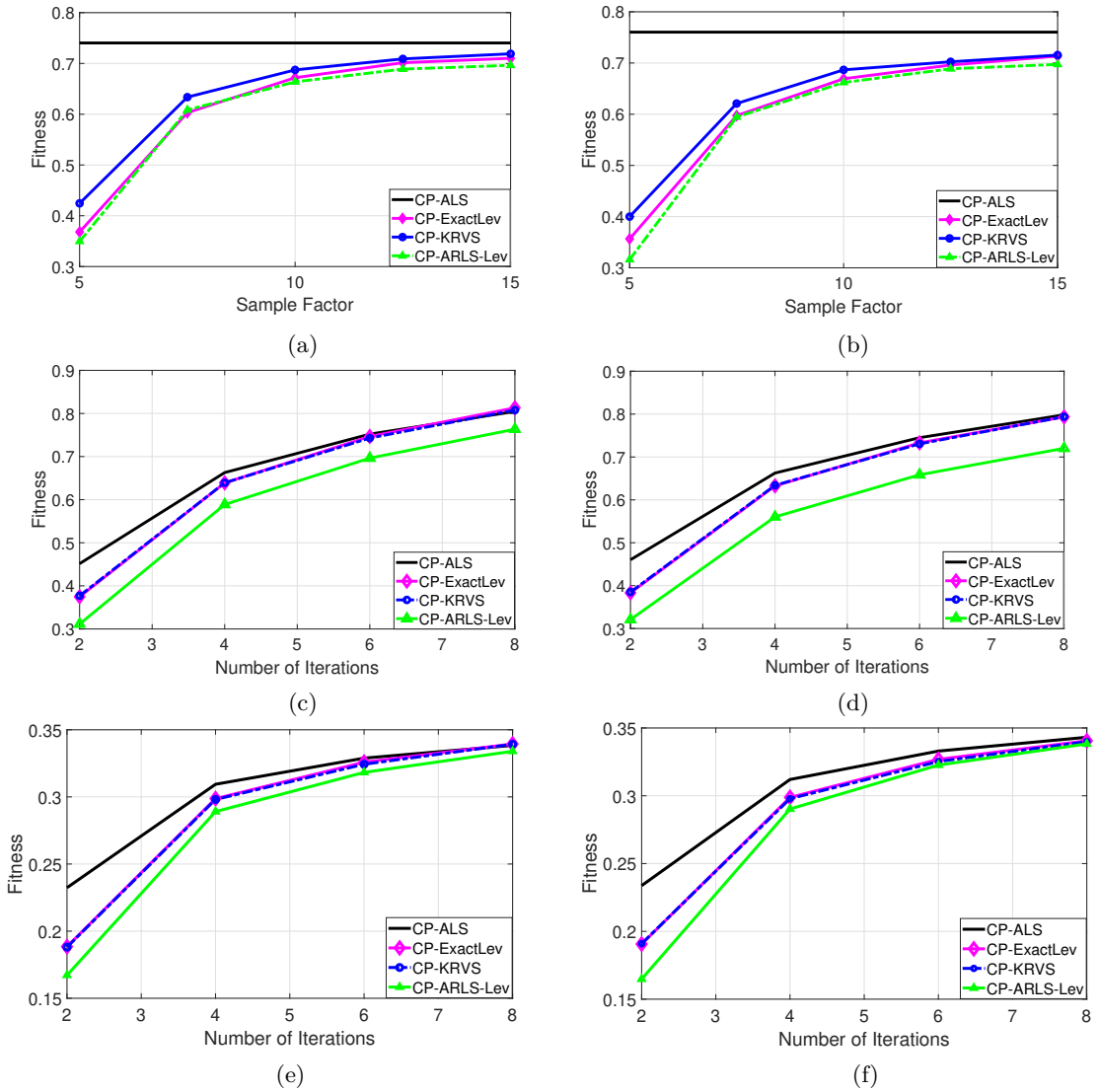


Figure 2: CP decomposition of tensor $\mathcal{Y} \in \mathbb{R}^{100\times100\times100}$ at rank $R = 10$ using proposed KRVS (Algorithm 3), optimal CP-ALS algorithm, and the baseline leverage score based algorithms CP-ARLS-Lev (Larsen & Kolda, 2022), CP-ExactLev (Bharadwaj et al., 2023). Fig. 2a and Fig. 2b plots Fitness *v/s* sample factor κ for dense tensor $\mathcal{Y}$ in Equation (42), at the noise variance $\sigma \in \{1, 10\}$ respectively. The number of ALS iterations is 6. Fig. (2c, 2d) / (2e, 2f) plots Fitness *v/s* number of ALS iterations for dense/sparse tensor $\mathcal{Y}$ in Equation (42)/(43) at the noise variance $\sigma \in \{1, 10\}$ respectively. The sample factor $\kappa = 200$.

$\eta \in \{1, 10\}$, $p = 0.6$. The experiments were run using MATLAB 2024a on a system with an Intel Core $i7 - 14700$ processor and 48GB RAM.

Evaluation metric and Baselines: Denote $\bar{\mathcal{Y}}_i$, $i \in \{1, 2\}$ as the reconstructed tensor from the considered CP decomposition algorithms. Then, fitness f is defined as $f := 1 - ((\|\mathcal{Y}_i - \bar{\mathcal{Y}}_i\|_F^2)/(\|\mathcal{Y}_i\|_F^2))$. A higher value of f indicates a good quality of CP decomposition. The proposed CP-KRVS (Algorithm 3) for CP decomposition is compared with the following baseline schemes.

- [CP-ALS] The ALS algorithm (optimal) for CP decomposition (Kolda & Bader, 2009).
- [CP-ARLS-Lev] An alternate randomized least-squares [b] (ARLS) for CP decomposition by Larsen & Kolda (2022) based on implicit leverage score sampling .
- [CP-ExactLev] Sampling based ALS algorithm for CP decomposition using exact leverage scores, same accuracy as work of Bharadwaj et al. (2023).

The fitness of considered CP decomposition algorithms are plotted against (i) sample factor κ, and (ii) number of ALS iterations, such that the sample size $K = \kappa R$.

Insights: In fitness *v/s* sample factor plots of Fig. 2a and Fig. 2b, the proposed KRVS is observed to have higher accuracy as compared to baseline leverage score sampling schemes *viz.,* ExactLev by Bharadwaj et al. (2023) and CP-ARLS-Lev by Larsen & Kolda (2022). This highlights the fact that, among the CP decomposition algorithms operating at small sample size K, the proposed KRVS based approach outperforms the baseline leverage score sampling schemes in terms of accuracy. Further, the runtime *v/s* accuracy trade-off is observed through improved fitness of proposed and baseline sampling based CP decomposition algorithms with increase in sample size (i.e., increase in κ value), at the cost of runtime.

Fig. 2c and Fig. 2d plots the fitness *v/s* number of ALS iterations for dense tensor i.e., $\mathcal{Y}$ in Equation (42). The fitness of the proposed KRVS method is observed to be overlapping with the ExactLev (Bharadwaj et al., 2023) and better than the CP-ARLS-Lev Larsen & Kolda (2022). Further, the fitness of all the considered CP decomposition algorithms improves with an increase in the number of ALS iterations. A similar trend in performance is also observed for the sparse tensor $\mathcal{Y}$ in Equation (43) through Fig. 2e and Fig. 2f. Lastly, a decrease in fitness is observed in Fig. 2d and Fig. 2f, as compared to Fig. 2c and Fig. 2e, respectively. This is due to increase in the noise variance.

The next section shows the application of baseline CP decomposition algorithms on image data.

6.3 CP Decomposition Experiments on Image Data

Image/Video is a naturally occurring multi-modal data, with the spatial dimensions, color, illuminations, viewpoints/orientations, etc., representing its different modes (Wang & Ahuja, 2008; Derkach et al., 2019). An RGB (red, green, blue) colored image of resolution $H \times W$ can be represented using a rank R CP decomposition in $(H + W + 3)R$ parameters. For $R \ll \max\{H, W\}$, this is an order of magnitude lesser than $3HW$ parameters required in naive representation. Thus, the CP decomposition provides a compact way to accurately represent the image data, and has been employed in key image processing applications like copy detection (Liang et al., 2024), motion capture (Mohaoui & Dmytryshyn, 2024), etc. This section presents the CP decomposition experiments on COIL-100 dataset (Nene et al., 1996), which is a popular image dataset for benchmarking tensor decomposition algorithms (Zhou et al., 2014; Battaglino et al., 2018).

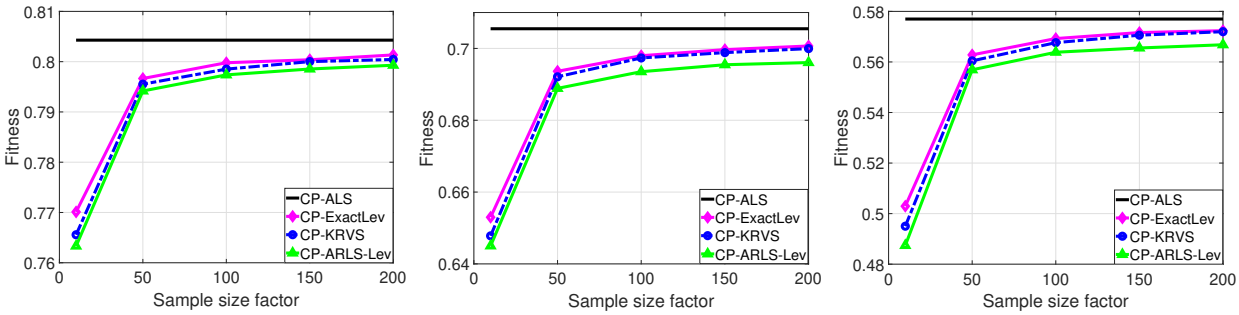
Dataset description: The COIL-100 dataset contains images of 100 objects, in 72 different orientations. Each image is colored, and of size 128×128. Assuming the standard RGB (red, green, and blue) splitting, each object from the dataset can be represented as a mode 4 tensor $\mathcal{Y} \in \mathbb{R}^{128 \times 128 \times 72 \times 3}$. For the first experiment, a given orientation of an object is represented as a sub-tensor $\mathcal{Z}_i \in \mathbb{R}^{128 \times 128 \times 3}$, where $\mathcal{Z}_i := \mathcal{Y}_{(:,:,i,:)}$. For the second experiment, the image data of $\mathcal{Y}$ is split into color channels $\mathcal{X}_j \in \mathbb{R}^{128 \times 128 \times 72}$, where $\mathcal{X}_j = \mathcal{Y}_{(:,:,:,j)}$. In both experiments, the rank $R = 20$ CP decomposition of the tensors *viz.,* $\mathcal{Z}_i$, $\mathcal{Y}_j$ is computed.

[b] Adapted from Brett W. Bader, Tamara G. Kolda and others, Tensor Toolbox for MATLAB, Version 3.8, December 7, 2025.

Evaluation metric and Baselines: The evaluation metric (fitness) and the baselines (CP-ALS, CP-ExactLev, CP-ARLS-Lev) remain same as in Section 6.2. The fitness (f) of different CP decomposition algorithms, for each color channel, is plotted as a function of sample size factor κ, where sample size $K = \kappa R$.

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

Figure 3: (3a) An object $\mathcal{Y}$ from COIL-100 dataset in orientation 1; Reconstructed image post rank R CP decomposition via (3b) CP-ALS, $R = 20$; (3c) CP-KRVS, $R = 20$, $\kappa = 10$; (3d) CP-KRVS, $R = 30$, $\kappa = 10$.



(a) (b) (c)

Figure 4: CP decomposition of image tensor $\mathcal{Y} \in \mathbb{R}^{128\times128\times72\times3}$, split across different color channels, *viz.,* red ($\mathcal{X}_1 = \mathcal{Y}_{(:,:,:,1)}$), green ($\mathcal{X}_2 = \mathcal{Y}_{:,:,:,2}$), and blue ($\mathcal{X}_3 = \mathcal{Y}_{(:,:,:,3)}$), in Figures (4a), (4b), and (4c), respectively. The rank R CP decomposition is computed using proposed CP-KRVS (Algorithm 3), optimal CP-ALS algorithm, and the baseline leverage score sampling algorithms CP-ARLS-Lev Larsen & Kolda (2022), CP-ExactLev Bharadwaj et al. (2023). The number of ALS iterations is 6, and CP rank $R = 20$.

Insights: Figure 3 shows a set of four images of an object $\mathcal{Y}$ from COIL-100 dataset, *viz.,* (3a) original image; (3b) reconstructed image after rank $R = 20$ CP decomposition via CP-ALS; (3c), (3d) reconstructed images at rank $R = \{20, 30\}$ CP decomposition via CP-KRVS, respectively. The reconstructed images (Figures (3b), (3c), and (3d)) after CP decomposition approximate the features of the original image accurately, including borders, color grading, etc. Further, increasing the value CP decomposition rank from $R = 20$ in Figure (3c), to $R = 30$ in Figure (3d), helps to capture additional details from the original image. Hence, increasing the rank of CP decomposition improves the approximation quality with a trade-off in complexity.

Figure 4 compares the proposed CP-KRVS algorithm with baseline CP-ALS, CP-ExactLev, and CP-ARLS-Lev based CP decomposition algorithms. The CP-ExactLev achieves highest fitness among the considered sampling based CP decomposition algorithms, closely followed by CP-KRVS. The CP-ARLS-Lev is observed to have lower accuracy than the CP-ExactLev and CP-KRVS. This observation concurs with the experiments on synthetic dataset (Section 6.2). Next, the increase in sample size leads to improved fitness in all considered sampling based CP decomposition algorithms at the cost of runtime complexity (Table 1), highlighting their accuracy *v/s* runtime complexity trade-off. Lastly, the fitness of CP-ExactLev, CP-KRVS, and CP-ARLS-Lev is seen to approach the CP-ALS at higher sample sizes.

7 Conclusion

This work proposed a sampling based approach, called as KRVS, to solve Khatri-Rao product regression. Theoretical analysis shows that the proposed KRVS based solution is an unbiased estimator, and achieves $(1+\epsilon)$ relative error approximation of the optimal least-squares based solution. Numerical results on standard regression datasets shows that KRVS solution achieves higher accuracy than the baseline implicit leverage score sampling solution at a comparable runtime. Moreover, the KRVS solution has a faster runtime and lower accuracy than exact leverage score sampling solution. At higher sample size, the KRVS solution achieves a close approximation of the optimal LS solution with significantly faster runtime. The proposed KRVS based framework is further employed to design an ALS algorithm (CP-KRVS) for CP decomposition. The theoretical analysis reveals that the CP-KRVS achieves a $(1 + \epsilon)$ relative error approximation of the optimal ALS based factor matrix update. Numerical analysis on synthetic and real world datasets highlight the superior accuracy of CP-KRVS over baseline implicit leverage score sampling, and comparable/slightly lower accuracy than exact leverage score sampling approach for CP decomposition. At higher sample size, the proposed KRVS achieves a close approximation of optimal ALS algorithm. The KRVS framework could be further extended to other popular tensor decompositions like tensor train (Oseledets, 2011), tensor ring (Malik & Becker, 2021) etc. Moreover, the unbiased estimation property of KRVS could be further exploited via its distributed memory implementation. These directions can be pursued as part of future work.

References

Evrim Acar and Bülent Yener. Unsupervised multiway data analysis: A literature survey. *IEEE transactions on knowledge and data engineering*, 21(1):6–20, 2008.

Evrim Acar, Seyit A Camtepe, and Bülent Yener. Collective sampling and analysis of high order tensors for chatroom communications. In *International Conference on Intelligence and Security Informatics*, pp. 213–224. Springer, 2006.

Evrim Acar, Daniel M. Dunlavy, and Tamara G. Kolda. Link prediction on evolving data using matrix and tensor factorizations. In *2009 IEEE International Conference on Data Mining Workshops*, pp. 262–269, 2009. doi: 10.1109/ICDMW.2009.54.

Dimitris Achlioptas. Database-friendly random projections: Johnson-Lindenstrauss with binary coins. *Journal of computer and System Sciences*, 66(4):671–687, 2003.

Thomas D Ahle, Michael Kapralov, Jakob BT Knudsen, Rasmus Pagh, Ameya Velingker, David P Woodruff, and Amir Zandieh. Oblivious sketching of high-degree polynomial kernels. In *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 141–160. SIAM, 2020.

Casey Battaglino, Grey Ballard, and Tamara G Kolda. A practical randomized CP tensor decomposition. *SIAM Journal on Matrix Analysis and Applications*, 39(2):876–901, 2018.

Vivek Bharadwaj, Osman Asif Malik, Riley Murray, Laura Grigori, Aydin Buluc, and James Demmel. Fast exact leverage score sampling from Khatri-Rao products with applications to tensor decomposition. *Advances in Neural Information Processing Systems*, 36:47874–47901, 2023.

Vivek Bharadwaj, Osman Asif Malik, Riley Murray, Aydın Buluç, and James Demmel. Distributed-memory randomized algorithms for sparse tensor CP decomposition. In *Proceedings of the 36th ACM Symposium on Parallelism in Algorithms and Architectures*, pp. 155–168, 2024.

Chih-Chung Chang and Chih-Jen Lin. LIBSVM: a library for support vector machines. *ACM transactions on intelligent systems and technology (TIST)*, 2(3):1–27, 2011.

Dehua Cheng, Richard Peng, Yan Liu, and Ioakeim Perros. SPALS: Fast alternating least squares via implicit leverage scores sampling. *Advances in neural information processing systems*, 29, 2016.

Peter A Chew, Brett W Bader, Tamara G Kolda, and Ahmed Abdelali. Cross-language information retrieval using PARAFAC2. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 143–152, 2007.

Andrzej Cichocki. Era of big data processing: A new approach via tensor networks and tensor decompositions. *arXiv preprint arXiv:1403.2048*, 2014.

Andrzej Cichocki, Danilo Mandic, Lieven De Lathauwer, Guoxu Zhou, Qibin Zhao, Cesar Caiafa, and Huy Anh Phan. Tensor decompositions for signal processing applications: From two-way to multiway component analysis. *IEEE signal processing magazine*, 32(2):145–163, 2015.

Kenneth L Clarkson and David P Woodruff. Low-rank approximation and regression in input sparsity time. *Journal of the ACM (JACM)*, 63(6):1–45, 2017.

Fengyu Cong, Qiu-Hua Lin, Li-Dan Kuang, Xiao-Feng Gong, Piia Astikainen, and Tapani Ristaniemi. Tensor decomposition of EEG signals: a brief review. *Journal of neuroscience methods*, 248:59–69, 2015.

Sanjoy Dasgupta and Anupam Gupta. An elementary proof of a theorem of Johnson and Lindenstrauss. *Random Structures & Algorithms*, 22(1):60–65, 2003.

Michał Dereziński and Manfred K Warmuth. Reverse iterative volume sampling for linear regression. *Journal of Machine Learning Research*, 19(23):1–39, 2018.

Michal Derezinski and Manfred KK Warmuth. Unbiased estimates for linear regression via volume sampling. *Advances in Neural Information Processing Systems*, 30, 2017.

Michal Derezinski, Manfred KK Warmuth, and Daniel J Hsu. Leveraged volume sampling for linear regression. *Advances in Neural Information Processing Systems*, 31, 2018.

Michał Dereziński, Manfred K Warmuth, and Daniel Hsu. Unbiased estimators for random design regression. *Journal of Machine Learning Research*, 23(167):1–46, 2022.

Dmytro Derkach, Adria Ruiz, and Federico M Sukno. Tensor decomposition and non-linear manifold modeling for 3D head pose estimation. *International Journal of Computer Vision*, 127(10):1565–1585, 2019.

Huaian Diao, Rajesh Jayaram, Zhao Song, Wen Sun, and David Woodruff. Optimal sketching for kronecker product regression and low rank approximation. *Advances in neural information processing systems*, 32, 2019.

Petros Drineas and Michael W. Mahoney. RandNLA: randomized numerical linear algebra. *Commun. ACM*, 59(6):80–90, May 2016. ISSN 0001-0782. doi: 10.1145/2842602. URL `https://doi.org/10.1145/2842602`.

Petros Drineas, Ravi Kannan, and Michael W Mahoney. Fast Monte Carlo algorithms for matrices I: Approximating matrix multiplication. *SIAM Journal on Computing*, 36(1):132–157, 2006a.

Petros Drineas, Michael W Mahoney, and Shan Muthukrishnan. Sampling algorithms for ℓ_2 regression and applications. In *Proceedings of the seventeenth annual ACM-SIAM symposium on Discrete algorithm*, pp. 1127–1136, 2006b.

Petros Drineas, Michael W Mahoney, Shan Muthukrishnan, and Tamás Sarlós. Faster least squares approximation. *Numerische mathematik*, 117(2):219–249, 2011.

Daniel M. Dunlavy, Tamara G. Kolda, and Evrim Acar. Temporal link prediction using matrix and tensor factorizations. *ACM Trans. Knowl. Discov. Data*, 5(2), February 2011. ISSN 1556-4681. doi: 10.1145/1921632.1921636. URL `https://doi.org/10.1145/1921632.1921636`.

Matthew Fahrbach, Gang Fu, and Mehrdad Ghadiri. Subquadratic kronecker regression with applications to tensor decomposition. *Advances in Neural Information Processing Systems*, 35:28776–28789, 2022.

Alan Frieze, Ravi Kannan, and Santosh Vempala. Fast Monte-Carlo algorithms for finding low-rank approximations. *Journal of the ACM (JACM)*, 51(6):1025–1041, 2004.

Alex Gittens, Kareem Aggour, and Bülent Yener. Adaptive sketching for fast and convergent canonical polyadic decomposition. In *International Conference on Machine Learning*, pp. 3566–3575. PMLR, 2020.

Xiao-Feng Gong, Qiu-Hua Lin, Feng-Yu Cong, and Lieven De Lathauwer. Double coupled canonical polyadic decomposition for joint blind source separation. *IEEE Transactions on Signal Processing*, 66(13):3475–3490, 2018.

Borbála Hunyadi, Patrick Dupont, Wim Van Paesschen, and Sabine Van Huffel. Tensor decompositions and data fusion in epileptic electroencephalography and functional magnetic resonance imaging data. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 7(1):e1197, 2017.

Johan Håstad. Tensor rank is NP-complete. *Journal of Algorithms*, 11(4):644–654, 1990. ISSN 0196-6774. doi: https://doi.org/10.1016/0196-6774(90)90014-6. URL `https://www.sciencedirect.com/science/article/pii/0196677490900146`.

Ruhui Jin, Tamara G Kolda, and Rachel Ward. Faster Johnson–Lindenstrauss transforms via kronecker products. *Information and Inference: A Journal of the IMA*, 10(4):1533–1562, 2021.

Tamara G Kolda and Brett W Bader. Tensor decompositions and applications. *SIAM review*, 51(3):455–500, 2009.

Brett W Larsen and Tamara G Kolda. Practical leverage-based sampling for low-rank tensor decomposition. *SIAM Journal on Matrix Analysis and Applications*, 43(3):1488–1517, 2022.

Ping Li, Trevor J Hastie, and Kenneth W Church. Very sparse random projections. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 287–296, 2006.

Xue Li, Bo Chang, and Zhi Chen. Tensor decomposition based THz channel estimation in OTFS for integrated sensing and communications. In *GLOBECOM 2023 - 2023 IEEE Global Communications Conference*, pp. 3996–4001, 2023. doi: 10.1109/GLOBECOM54140.2023.10437761.

Xiaoping Liang, Wanting Liu, Xianquan Zhang, and Zhenjun Tang. Robust image hashing via CP decomposition and DCT for copy detection. *ACM Transactions on Multimedia Computing, Communications and Applications*, 20(7):1–22, 2024.

Linjian Ma and Edgar Solomonik. Fast and accurate randomized algorithms for low-rank tensor decompositions. *Advances in neural information processing systems*, 34:24299–24312, 2021.

Michael W Mahoney et al. Randomized algorithms for matrices and data. *Foundations and Trends® in Machine Learning*, 3(2):123–224, 2011.

Osman Asif Malik. More efficient sampling for tensor decomposition with worst-case guarantees. In *International conference on machine learning*, pp. 14887–14917. PMLR, 2022.

Osman Asif Malik and Stephen Becker. Guarantees for the kronecker fast Johnson–Lindenstrauss transform using a coherence and sampling argument. *Linear Algebra and its Applications*, 602:120–137, 2020.

Osman Asif Malik and Stephen Becker. A sampling-based method for tensor ring decomposition. In *International conference on machine learning*, pp. 7400–7411. PMLR, 2021.

Osman Asif Malik, Vivek Bharadwaj, and Riley Murray. Sampling-based decomposition algorithms for arbitrary tensor networks. *arXiv preprint arXiv:2210.03828*, 2022a.

Osman Asif Malik, Yiming Xu, Nuojin Cheng, Stephen Becker, Alireza Doostan, and Akil Narayan. Fast algorithms for monotone lower subsets of kronecker least squares problems. *arXiv preprint arXiv:2209.05662*, 2022b.

Fumikazu Miwakeichi, Eduardo Martınez-Montes, Pedro A Valdés-Sosa, Nobuaki Nishiyama, Hiroaki Mizuhara, and Yoko Yamaguchi. Decomposing EEG data into space–time–frequency components using parallel factor analysis. *NeuroImage*, 22(3):1035–1045, 2004.

Souad Mohaoui and Andrii Dmytryshyn. CP decomposition-based algorithms for completion problem of motion capture data. *Pattern Analysis and Applications*, 27(4):133, 2024.

Sameer A Nene, Shree K Nayar, Hiroshi Murase, et al. Columbia object image library (coil-100). Technical report, Technical report CUCS-006-96, 1996.

Román Orús. A practical introduction to tensor networks: Matrix product states and projected entangled pair states. *Annals of physics*, 349:117–158, 2014.

Román Orús. Tensor networks for complex quantum systems. *Nature Reviews Physics*, 1(9):538–550, 2019.

Ivan V Oseledets. Tensor-train decomposition. *SIAM Journal on Scientific Computing*, 33(5):2295–2317, 2011.

Rasmus Pagh. Compressed matrix multiplication. *ACM Transactions on Computation Theory (TOCT)*, 5 (3):1–17, 2013.

Yannis Panagakis, Jean Kossaifi, Grigorios G. Chrysos, James Oldfield, Mihalis A. Nicolaou, Anima Anandkumar, and Stefanos Zafeiriou. Tensor methods in computer vision and deep learning. *Proceedings of the IEEE*, 109(5):863–890, 2021. doi: 10.1109/JPROC.2021.3074329.

Evangelos E. Papalexakis and Christos Faloutsos. Fast efficient and scalable core consistency diagnostic for the parafac decomposition for big sparse tensors. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5441–5445, 2015. doi: 10.1109/ICASSP.2015.7179011.

Ninh Pham and Rasmus Pagh. Fast and scalable polynomial kernels via explicit feature maps. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 239–247, 2013.

Hoang Pham Minh, Nguyen Nguyen Xuan, and Son Tran Thai. TT-ViT: Vision transformer compression using tensor-train decomposition. In *International Conference on Computational Collective Intelligence*, pp. 755–767. Springer, 2022.

Haoxin Ruan, Tong Lei, Kai Chen, and Jing Lu. An explicit connection between independent vector analysis and tensor decomposition in blind source separation. *IEEE Signal Processing Letters*, 29:1277–1281, 2022.

Tamas Sarlos. Improved approximation algorithms for large matrices via random projections. In *2006 47th annual IEEE symposium on foundations of computer science (FOCS'06)*, pp. 143–152. IEEE, 2006.

Nicholas D Sidiropoulos, Lieven De Lathauwer, Xiao Fu, Kejun Huang, Evangelos E Papalexakis, and Christos Faloutsos. Tensor decomposition for signal processing and machine learning. *IEEE Trans. Signal Process.*, 65(13):3551–3582, 2017.

Shaden Smith, Jee W. Choi, Jiajia Li, Richard Vuduc, Jongsoo Park, Xing Liu, and George Karypis. FROSTT: The formidable repository of open sparse tensors and tools, 2017. URL `http://frostt.io/`.

Stephan Spiegel, Jan Clausen, Sahin Albayrak, and Jérôme Kunegis. Link prediction on evolving data using tensor factorization. In *Pacific-Asia conference on knowledge discovery and data mining*, pp. 100–110. Springer, 2011.

Hongcheng Wang and Narendra Ahuja. A tensor approximation approach to dimensionality reduction. *International Journal of Computer Vision*, 76:217–229, 2008.

Yining Wang, Hsiao-Yu Tung, Alexander J Smola, and Anima Anandkumar. Fast and guaranteed tensor decomposition via sketching. *Advances in neural information processing systems*, 28, 2015.

David Woodruff and Amir Zandieh. Near input sparsity time kernel embeddings via adaptive sampling. In *International Conference on Machine Learning*, pp. 10324–10333. PMLR, 2020.

David Woodruff and Amir Zandieh. Leverage score sampling for tensor product matrices in input sparsity time. In *International Conference on Machine Learning*, pp. 23933–23964. PMLR, 2022.

David P Woodruff et al. Sketching as a tool for numerical linear algebra. *Foundations and Trends® in Theoretical Computer Science*, 10(1–2):1–157, 2014.

Xianlong Zeng, Peilan Wang, and Jun Fang. Millimeter-wave sensing and mapping: A low-rank CP decomposition approach. *IEEE Wireless Communications Letters*, 13(6):1675–1679, 2024. doi: 10.1109/LWC.2024.3386164.

Guoxu Zhou, Andrzej Cichocki, and Shengli Xie. Decomposition of big tensors with low multilinear rank. *arXiv preprint arXiv:1412.1885*, 2014.