

Core-elements Subsampling for Alternating Least Squares

Dunyao Xue

Institute of Statistics and Big Data,
Renmin University of China, Beijing, China

Mengyu Li*

Department of Statistics and Data Science, Tsinghua University, Beijing, China

Jingyi Zhang

School of Science, Department of Mathematics
Beijing University of Posts and Telecommunications, Beijing, China

Cheng Meng

Center for Applied Statistics, Institute of Statistics and Big Data,
Renmin University of China, Beijing, China.

Abstract

In this paper, we propose a novel element-wise subset selection method for the alternating least squares (ALS) algorithm, focusing on low-rank matrix factorization involving matrices with missing values, as commonly encountered in recommender systems. While ALS is widely used for providing personalized recommendations based on user-item interaction data, its high computational cost, stemming from repeated regression operations, poses significant challenges for large-scale datasets. To enhance the efficiency of ALS, we propose a core-elements subsampling method that selects a representative subset of data and leverages sparse matrix operations to approximate ALS estimations efficiently. We establish theoretical guarantees for the approximation and convergence of the proposed approach, showing that it achieves similar accuracy with significantly reduced computational time compared to full-data ALS. Extensive simulations and real-world applications demonstrate the effectiveness of our method in various scenarios, emphasizing its potential in large-scale recommendation systems.

Keywords: Recommender systems, Matrix factorization, Element-wise subsampling, Missing values

*Corresponding author, mengyuli@tsinghua.edu.cn

1 Introduction

Recommender systems are designed to provide personalized suggestions to users by modeling their preferences over items (Lü et al., 2012; Bi et al., 2017; LeBlanc et al., 2024). These systems utilize explicit feedback, such as user ratings, and implicit feedback, including user activities such as purchasing, viewing, or searching for items.

Matrix factorization (MF) techniques have become fundamental in developing effective recommender systems (Lee and Seung, 2000; Sun and Luo, 2016). Among these, alternating least squares (ALS) (Zhou et al., 2008; Jain et al., 2013; Takács and Tikk, 2012) is a robust MF algorithm capable of handling both explicit and implicit feedback data. Unlike the classical singular value decomposition (SVD) (Klema and Laub, 1980; Abdi, 2007; Zhao, 2024), ALS efficiently manages missing values in the user-item matrix, thereby improving its applicability in a broader range of real-world scenarios.

Despite its effectiveness, low-rank matrix factorization using alternating least squares can be computationally intensive due to the numerous regression operations required, particularly when dealing with large initial matrices. For instance, given an initial matrix $\mathbf{R} \in \mathbb{R}^{n_u \times n_m}$, the time complexity of ALS algorithms is $O(n_f^2(\text{nnz}(\mathbf{R}) + n_f n_u + n_f n_m) n_t)$ (Zhou et al., 2008), where n_f denotes the rank of the decomposed low-rank matrix, n_t represents the number of iterations, and $\text{nnz}(\cdot)$ denotes the number of non-zero elements of a matrix.

An effective strategy to improve computational efficiency is to estimate the model on a subset of observations, commonly referred to as the coresets approach, subsampling, or subset selection. By strategically selecting representative observations using methods such as leverage scores (Ma and Sun, 2015; Ma et al., 2014; Han et al., 2025; Zhong et al., 2023; Shimizu et al., 2023), influence functions (Ting and Brochu, 2018), predictive inference-based subsampling (Wu et al., 2024), various optimality criteria (Wang et al., 2018; Yu et al., 2023; Wang et al., 2019), and other probabilistic or deterministic techniques, coresets substantially

reduce computational burdens while retaining most of the information in large datasets. These approaches have proven effective across a range of statistical learning tasks, including least squares regression (Ma et al., 2014; Ma and Sun, 2015; Ting and Brochu, 2018; Meng et al., 2017; Drineas et al., 2006; Wang et al., 2019, 2021; Ma et al., 2022), generalized linear models (Wang et al., 2018; Ai et al., 2021; Yu et al., 2022, 2024), nonparametric regression (Ma et al., 2015; Meng et al., 2020, 2022; Zhang et al., 2024), quantile regression (Wang and Ma, 2021; Ai et al., 2021), model-free methods (Meng et al., 2021; Yi and Zhou, 2023), machine learning (Wang et al., 2018; Han et al., 2025), time series analysis (Xie et al., 2019, 2023), and optimal transport (Li et al., 2023; Hu et al., 2024; Li et al., 2023). Beyond the computational benefits, coresets play a pivotal role in measurement-constrained problems (Zhang et al., 2023; Meng et al., 2021) and privacy-preserving contexts (Wang et al., 2019; Balle et al., 2020). For a comprehensive overview, we refer readers to Li and Meng (2020) and Yu et al. (2024).

There have been several efforts to accelerate the alternating least squares algorithm through subsampling techniques. Pan et al. (2008) introduced a sampling ALS ensemble method that repeatedly samples elements from the rating matrix according to a predefined probability distribution, applies ALS to each sampled matrix, and then aggregates the results through weighted averaging. Furthermore, Cheng et al. (2016) used a row sampling strategy based on leverage scores (Han et al., 2025; Ma et al., 2014) to improve the efficiency of ALS for tensor data. However, these sampling approaches exhibit some limitations. Element-wise sampling directly from the rating matrix can result in substantial information loss, and leverage score-based sampling incurs high computational costs due to the need to compute leverage scores. Although approximations of sampling probabilities can alleviate some of this computational burden, they often compromise accuracy. Therefore, we aim to develop a novel sampling method that significantly accelerates the traditional ALS algorithm while preserving high accuracy.

Beyond sampling techniques, various methods have been proposed to accelerate ALS. For example, Zhou et al. (2008) used parallel computing to reduce the computational costs associated with repeated regression operations. Pilászy et al. (2010) applied the Sherman–Morrison formula (SMF) to accelerate computations. Additionally, Hastie et al. (2015) explored the relationship between matrix completion and singular value decomposition, proposing a new iteration formulation for ALS. However, these approaches are beyond the scope of this paper, as sampling algorithms do not modify the core structure of the ALS algorithm and can be integrated with other acceleration techniques. Consequently, our focus is on developing advanced sampling algorithms within the fundamental ALS framework, rather than introducing entirely new algorithmic formulations.

In recommender systems, we observe that both non-negative matrix factorization and matrix factorization with regularization terms tend to produce low-rank matrices with a large number of small values, resulting in numerical sparsity, as illustrated in Fig. 1. Sampling rows from such numerically sparse matrices, \mathbf{U} or \mathbf{M} , can be significantly inefficient because most sampled elements are not informative. In contrast, the core-elements approach (Li et al., 2024), an element-wise subset selection method specifically designed for sparse matrices in linear and nonparametric regressions, effectively addresses this limitation by preserving the most informative components in the data. Our extensive experimental results indicate that ALS often exhibits similar numerical sparsity characteristics as shown in Fig. 1. Therefore, extending the core-elements method to the ALS framework can be an effective solution to address this challenge.

Building on the foundations laid by these previous studies, in this paper we propose an efficient and scalable approach for approximating alternating least squares estimation in matrix factorization.

Major contributions. We summarize our contributions as follows.

First, we propose a novel core-element sampling framework Core-ALS for alternating least

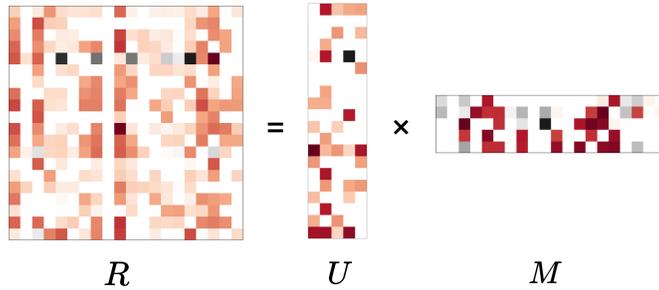


Figure 1: Illustration of matrix factorization. Values less than 5% of the maximum value are displayed as white, showing that both \mathbf{U} and \mathbf{M} are numerically sparse.

squares that selects important entries from the entire matrix and couples this with sparse matrix operations at each regression step. This formulation efficiently approximates penalized least squares estimation and fully uses the information in the rating matrix \mathbf{R} . *Second, we establish theoretical guarantees for the iterative setting.* We derive per-iteration $(1+\epsilon)$ approximation bounds for the sketched regressions and show convergence of the Core-ALS method under mild conditions. Time complexity reductions specific to alternating updates and sparse matrix multiplications are also provided. *Third, we further improve the computational efficiency of Li et al. (2024a) and Li et al. (2024b).* We integrate partial quicksort (Martinez, 2004) into our *Core Sparse Matrix Multiplication* so that thresholding-and-sampling is performed on the fly, which avoids full sorting and yields substantial computational speedups. Moreover, we develop a fast variant of Core-ALS that substantially lowers the computational cost. *Fourth, we show superior accuracy and efficiency from experiments.* Extensive simulations confirm the method’s effectiveness and efficiency for both model fitting and prediction, with particularly strong results on classical recommendation metrics including NDCG@k and Hit@k.

The remainder of this paper is organized as follows. In Section 2, we introduce the alternating least squares algorithm and core-elements for penalized regression splines. Section 3 develops the core-elements algorithm for ALS, and Section 4 discusses its theoretical

properties. We then assess the performance of the proposed estimator through extensive experiments on synthetic and real-world data in Sections 5 and 6, respectively. Additional details omitted from the main text and technical proofs can be found in the Supplementary Material. All R code used to reproduce the results in this paper is available at <https://github.com/sapphirexdy>.

2 Background

Here we summarize the notation used throughout the paper. Specifically, matrices are represented by uppercase boldface italic letters, such as \mathbf{X} , while vectors are denoted by lowercase boldface italics letters such as \mathbf{x} . Scalars are represented using regular, non-bold typefaces. For any vector \mathbf{x} , we denote its ℓ_p norm by $\|\mathbf{x}\|_p$, and its Euclidean norm (i.e., the ℓ_2 norm) is abbreviated as $\|\mathbf{x}\|$. When referring to matrices, the spectral norm is represented by $\|\mathbf{X}\|_2$, and the Frobenius norm is denoted as $\|\mathbf{X}\|_F$.

2.1 Alternating Least Squares Algorithm

Consider a sparse¹ rating matrix $\mathbf{R} = (r_{ij}) \in \mathbb{R}^{n_u \times n_m}$ representing n_u users and n_m items, and a chosen implicit vector dimension n_f . The objective is to find two factor matrices $\mathbf{U} \in \mathbb{R}^{n_u \times n_f}$ and $\mathbf{M} \in \mathbb{R}^{n_m \times n_f}$ such that the relationship $\mathbf{R} = \mathbf{U}\mathbf{M}^\top$ holds in the absence of missing values.

More specifically, let $\mathbf{U} = (\mathbf{u}_i)$, where $\mathbf{u}_i \in \mathbb{R}^{1 \times n_f}$ ($i = 1, \dots, n_u$) denotes the i th row of \mathbf{U} , and let $\mathbf{M} = (\mathbf{m}_j)$, where $\mathbf{m}_j \in \mathbb{R}^{1 \times n_f}$ ($j = 1, \dots, n_m$) is the j th row of \mathbf{M} . If the non-missing values of \mathbf{R} are fully predictable and n_f is sufficiently large, we could expect

¹Here, the term “sparse” does not refer to a matrix with a large number of zero elements, but rather to one with a large proportion of missing values.

that

$$r_{ij} = \mathbf{u}_i \mathbf{m}_j^\top \quad \forall (i, j) \in [n_u] \times [n_m].$$

In practice, to estimate \mathbf{U} and \mathbf{M} , the alternating least squares algorithm minimizes the empirical loss:

$$\mathcal{L}^{emp}(\mathbf{R}, \mathbf{U}, \mathbf{M}) = \frac{1}{n} \sum_{(i,j) \in I} (r_{ij} - \mathbf{u}_i \mathbf{m}_j^\top)^2,$$

where I is the index set corresponding to the non-missing values of \mathbf{R} and $n = |I|$.

To prevent overfitting, a common approach is to append a Tikhonov regularization term (Tikhonov, 1977) to the empirical risk:

$$\mathcal{L}_\lambda^{reg}(\mathbf{R}, \mathbf{U}, \mathbf{M}) = \mathcal{L}^{emp}(\mathbf{R}, \mathbf{U}, \mathbf{M}) + \lambda (\|\mathbf{U}\mathbf{\Gamma}_U\|^2 + \|\mathbf{M}\mathbf{\Gamma}_M\|^2), \quad (1)$$

for certain suitably selected Tikhonov matrices $\mathbf{\Gamma}_U$ and $\mathbf{\Gamma}_M$.

Among the various types of regularization terms, the weighted λ -regularization works well empirically, preventing overfitting even as the number of features, n_f , or the number of ALS iterations increases (Zhou et al., 2008). Therefore, we focus on the regularized formulation (1) of ALS. In the rest of this paper, ALS refers to the following objective function:

$$f(\mathbf{U}, \mathbf{M}) = \sum_{(i,j) \in I} (r_{ij} - \mathbf{u}_i \mathbf{m}_j^\top)^2 + \lambda \left(\sum_i n_{u_i} \|\mathbf{u}_i\|^2 + \sum_j n_{m_j} \|\mathbf{m}_j\|^2 \right), \quad (2)$$

where n_{u_i} and n_{m_j} denote the number of ratings of user i and item j , respectively. This corresponds to taking $\mathbf{\Gamma}_U = \text{diag}(\sqrt{n_{u_i}})$ and $\mathbf{\Gamma}_M = \text{diag}(\sqrt{n_{m_j}})$ in (1).

Let $f(\mathbf{U}, \mathbf{M})$ take the partial derivatives for \mathbf{U} and \mathbf{M} , respectively, the specific iteration format of alternating least squares algorithm can be obtained as follows:

Step 1 (Initialization). Initialize matrix \mathbf{M} randomly and let $\widehat{\mathbf{M}} = \mathbf{M}$.

Step 2 (Update \mathbf{U}). Fix $\widehat{\mathbf{M}}$ and solve for \mathbf{U} by (3):

$$\widehat{\mathbf{u}}_i = \left(\widehat{\mathbf{M}}_{I_i^U}^\top \widehat{\mathbf{M}}_{I_i^U} + \lambda n_{u_i} \mathbf{E} \right)^{-1} \widehat{\mathbf{M}}_{I_i^U}^\top \mathbf{R}^\top(i, I_i^U), \quad \forall i = 1, \dots, n_u, \quad (3)$$

where I_i^U denotes the set of indices of non-missing values in the i th row of \mathbf{R} , $\widehat{\mathbf{M}}_{I_i^U}$ denotes the sub-matrix of $\widehat{\mathbf{M}}$ where rows $r \in I_i^U$ are selected, and $\mathbf{R}(i, I_i^U)$ is the row vector where columns $j \in I_i^U$ of the i th row of \mathbf{R} are selected. \mathbf{E} is the $n_f \times n_f$ identity matrix.

Step 3 (Update \mathbf{M}). Fix $\widehat{\mathbf{U}}$ and solve for \mathbf{M} by (4):

$$\widehat{\mathbf{m}}_j = \left(\widehat{\mathbf{U}}_{I_j^M}^\top \widehat{\mathbf{U}}_{I_j^M} + \lambda n_{m_j} \mathbf{E} \right)^{-1} \widehat{\mathbf{U}}_{I_j^M}^\top \mathbf{R}(I_j^M, j), \quad \forall j = 1, \dots, n_m. \quad (4)$$

Similarly, I_j^M denotes the set of indices of non-missing values in the j th column of \mathbf{R} , $\widehat{\mathbf{U}}_{I_j^M}$ denotes the sub-matrix of $\widehat{\mathbf{U}}$ where rows $r \in I_j^M$ are selected, and $\mathbf{R}(I_j^M, j)$ is the column vector where rows $i \in I_j^M$ of the j th column of \mathbf{R} are selected.

Step 4 (Iteration). Repeat Steps 2 and 3 until the stopping criterion is satisfied.

The sequence of achieved errors (2) is proved monotone non-increasing and bounded below, hence this sequence converges (Lee and Stöger, 2023).

2.2 Core-Elements For Penalized Regression Splines

In large-scale data analysis tasks, various subsampling techniques such as uniform subsampling (Zhang et al., 2023), leverage-based subsampling (Ma and Sun, 2015; Zhong et al., 2023; Shimizu et al., 2023), and optimal information-based subsampling (Wang et al., 2019; Wu et al., 2024; Yu et al., 2023) have been extensively studied and widely employed. These methods effectively reduce computational costs and have shown strong performance in practice.

Beyond these row-wise sampling approaches, Li et al. (2024a) proposed the core-elements method for approximating the ordinary least squares (OLS) estimation in linear models. To approximate the OLS estimation $\widehat{\boldsymbol{\beta}} = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{y}$, where $\mathbf{y} \in \mathbb{R}^n$ is the response vector and $\mathbf{X} \in \mathbb{R}^{n \times p}$ is the covariate matrix, the core-elements method constructs a sparse sketch $\mathbf{X}^* \in \mathbb{R}^{n \times p}$ of \mathbf{X} . Specifically, given a sampling budget $s \in \mathbb{Z}_+$, let $\mathbf{P} \in \mathbb{R}^{n \times p}$ be a binary matrix containing s ones and $(np - s)$ zeros. The sparse sketch \mathbf{X}^* is then formed by

$\mathbf{X}^* = \mathbf{P} \odot \mathbf{X}$, where \odot denotes the element-wise product. Based on the sparse sketch, the core-element estimator $\tilde{\boldsymbol{\beta}} = (\mathbf{X}^{*\top} \mathbf{X})^{-1} \mathbf{X}^{*\top} \mathbf{y}$ was proposed, motivated by the unbiasedness of the estimation. This approach was later extended to nonparametric additive models by [Li et al. \(2024b\)](#), which approximated the penalized least squares (PLS) estimation of regression splines, taking the form

$$\tilde{\boldsymbol{\beta}} = (\mathbf{X}^{*\top} \mathbf{X} + \mathbf{S}_\lambda)^{-1} \mathbf{X}^{*\top} \mathbf{y}, \quad (5)$$

where \mathbf{S}_λ is a penalty term related to the smoothing parameter λ .

Subsequently, an upper bound on the estimator’s variance is derived and approximately minimized, yielding the principle of core-element selection. Concretely, the sparse sketch \mathbf{X}^* retains at most s non-zero entries by keeping the $\lfloor s/p \rfloor$ largest absolute values in each column of \mathbf{X} and zeroing out the rest. The resulting core-elements estimator provides theoretical approximation guarantees relative to the full-sample PLS and outperforms established subsampling methods in empirical studies.

3 Methods

In [Section 2.2](#), we introduced the core-elements approach designed for penalized regression splines. Notably, the iterative form of the alternating least squares (ALS) algorithm closely resembles that of penalized regression splines, suggesting the potential application of the core-elements method to ALS. Nonetheless, this extension is not straightforward mainly for three reasons.

Firstly, the formulation of penalized regression splines differs from that of ALS, necessitating the development of a new estimator tailored to the ALS framework. Additionally, ALS involves multiple regression computations, and mitigating computational complexity requires continuous sampling operations, which invariably incur significant time costs. This highlights the need for more efficient algorithms to optimize the sampling process. Moreover, the core-

elements method is inherently designed for sparse matrices, whereas the matrices involved in ALS are typically dense. This disparity underscores the need to evaluate the applicability and effectiveness of the core-elements approach within the context of dense matrices. Thus, directly applying the core-element algorithm to each regression step may inadvertently reduce the overall efficiency of the ALS algorithm, thereby necessitating careful algorithmic design.

In this section, we present our main algorithm. We first develop the core-elements estimation for ALS and introduce the principle of selecting core-elements motivated by approximately minimizing the upper bound for the mean squared Frobenius norm error of $\tilde{\mathbf{U}}^{(t)}$ and $\tilde{\mathbf{M}}^{(t)}$ in each iteration.

Core-elements estimation. Inspired by the formulation (5), we propose an approximation for the ALS estimation (3) and (4) based on sparse sketches $\tilde{\mathbf{M}}_{I_i^U}^{(t)*}$ and $\tilde{\mathbf{U}}_{I_j^M}^{(t)*}$ in each iteration, taking the form:

$$\tilde{\mathbf{u}}_i^{(t+1)} = (\tilde{\mathbf{M}}_{I_i^U}^{(t)*\top} \tilde{\mathbf{M}}_{I_i^U}^{(t)} + \lambda n_{u_i} \mathbf{E})^{-1} \tilde{\mathbf{M}}_{I_i^U}^{(t)*\top} \mathbf{R}^\top(i, I_i^U), \quad \text{for } i = 1, \dots, n_u, \quad (6)$$

$$\tilde{\mathbf{m}}_j^{(t+1)} = (\tilde{\mathbf{U}}_{I_j^M}^{(t+1)*\top} \tilde{\mathbf{U}}_{I_j^M}^{(t+1)} + \lambda n_{m_j} \mathbf{E})^{-1} \tilde{\mathbf{U}}_{I_j^M}^{(t+1)*\top} \mathbf{R}(I_j^M, j), \quad \text{for } j = 1, \dots, n_m, \quad (7)$$

where (t) denotes the t th iteration and $\tilde{\mathbf{M}}_{I_i^U}^{(0)} = \mathbf{M}_{I_i^U}$. We assume that $\tilde{\mathbf{M}}_{I_i^U}^{(t)*\top} \tilde{\mathbf{M}}_{I_i^U}^{(t)} + \lambda n_{u_i} \mathbf{E}$ and $\tilde{\mathbf{U}}_{I_j^M}^{(t)*\top} \tilde{\mathbf{U}}_{I_j^M}^{(t)} + \lambda n_{m_j} \mathbf{E}$ are of full rank. Based on the formulations in (6) and (7), our goal is to iteratively find the sketches $\tilde{\mathbf{M}}_{I_i^U}^{(t)*}$ and $\tilde{\mathbf{U}}_{I_j^M}^{(t)*}$ that approximately minimize the expectation of the mean squared Frobenius norm error (MSFE) of $\tilde{\mathbf{U}}^{(t)} = (\tilde{\mathbf{u}}_1^{(t)}, \dots, \tilde{\mathbf{u}}_{n_u}^{(t)})^\top$ and $\tilde{\mathbf{M}}^{(t)} = (\tilde{\mathbf{m}}_1^{(t)}, \dots, \tilde{\mathbf{m}}_{n_m}^{(t)})^\top$, respectively, as defined by:

$$\text{MFSE}(\tilde{\mathbf{U}}^{(t)}) = \mathbb{E}(\|\tilde{\mathbf{U}}^{(t)} - \mathbf{U}^{(t)}\|_F^2) = \sum_{i=1}^{n_u} \mathbb{E}(\|\tilde{\mathbf{u}}_i^{(t)} - \mathbf{u}_i^{(t)}\|^2), \quad (8)$$

$$\text{MFSE}(\tilde{\mathbf{M}}^{(t)}) = \mathbb{E}(\|\tilde{\mathbf{M}}^{(t)} - \mathbf{M}^{(t)}\|_F^2) = \sum_{j=1}^{n_m} \mathbb{E}(\|\tilde{\mathbf{m}}_j^{(t)} - \mathbf{m}_j^{(t)}\|^2), \quad (9)$$

where $\mathbf{U}^{(t)} = (\mathbf{u}_1^{(t)}, \dots, \mathbf{u}_{n_u}^{(t)})^\top$ and $\mathbf{M}^{(t)} = (\mathbf{m}_1^{(t)}, \dots, \mathbf{m}_{n_m}^{(t)})^\top$ represent the ground truth values of \mathbf{U} and \mathbf{M} at the t th iteration, respectively.

Considering that it is challenging to directly minimize (8) and (9), we provide upper bounds for $\text{MFSE}(\tilde{\mathbf{U}}^{(t)})$ and $\text{MFSE}(\tilde{\mathbf{M}}^{(t)})$ in Proposition 1 and aim to minimize these upper bounds instead.

Proposition 1. *Let (t) denote the t th iteration and $\mathbf{L}_{U_i}^{(t)} = \tilde{\mathbf{M}}_{I_i^U}^{(t)} - \tilde{\mathbf{M}}_{I_i^U}^{(t)*}$. Taylor expansions of $\mathbb{E}(\|\tilde{\mathbf{u}}_i^{(t+1)} - \mathbf{u}_i^{(t+1)}\|^2)$ at $\mathbf{L}_{U_i}^{(t)}$ near the origin provide an upper bound for $\text{MFSE}(\tilde{\mathbf{U}}^{(t)})$.*

$$\text{MFSE}(\tilde{\mathbf{U}}^{(t)}) \leq \mathbf{V}_u^{(t)} + \mathbf{B}_u^{(t)},$$

where

$$\mathbf{V}_u^{(t)} = \sum_{i=1}^{n_u} \sigma^2 \left\{ [1 + \mathcal{O}(\gamma_u^{(t)})] \left(\|\tilde{\mathbf{M}}_{I_i^U}^{(t)} \mathbf{D}_{U_i}^{(t)}\|_F^2 + \|\mathbf{D}_{U_i}^{(t)}\|_2^2 \|\mathbf{L}_{U_i}^{(t)}\|_F^2 \right) + \mathcal{O}(\gamma_u^{(t)}) \text{Tr}(\mathbf{D}_{U_i}^{(t)}) \right\},$$

and

$$\mathbf{B}_u^{(t)} = \sum_{i=1}^{n_u} [1 + \mathcal{O}(\gamma_u^{(t)})] \|\lambda n_{u_i} \mathbf{D}_{U_i}^{(t)} \mathbf{u}_i^{(t+1)}\|^2.$$

Here, the terms $\mathbf{V}_u^{(t)}$ and $\mathbf{B}_u^{(t)}$ represent the upper bounds of the variance and the squared bias, respectively. The matrix $\mathbf{D}_{U_i}^{(t)}$ is defined as

$$\mathbf{D}_{U_i}^{(t)} = \left(\tilde{\mathbf{M}}_{I_i^U}^{(t)\top} \tilde{\mathbf{M}}_{I_i^U}^{(t)} + \lambda n_{u_i} \mathbf{E} \right)^{-1},$$

and the spectral radius is given by

$$\gamma_u^{(t)} = \|\mathbf{D}_{U_i}^{(t)} \mathbf{L}_{U_i}^{(t)\top} \tilde{\mathbf{M}}_{I_i^U}^{(t)}\|_2,$$

which is assumed to satisfy $\gamma_u^{(t)} < 1$ to ensure the convergence of the matrix series.

Similar to Proposition 1, we also yield an upper bound for the $\text{MFSE}(\tilde{\mathbf{M}}^{(t)})$ and the specific form of this upper bound can be found in the Appendix. Given the analogous forms of $\text{MFSE}(\tilde{\mathbf{U}}^{(t)})$ and $\text{MFSE}(\tilde{\mathbf{M}}^{(t)})$, we focus our analysis solely on $\text{MFSE}(\tilde{\mathbf{U}}^{(t)})$. According to Proposition 1, the upper bound of the $\text{MFSE}(\tilde{\mathbf{U}}^{(t)})$ decreases as both $\|\mathbf{L}_{U_i}^{(t)}\|_F$ and the

spectral radius $\gamma_u^{(t)}$ diminish. Specifically, the spectral radius $\gamma_u^{(t)}$ can be further bounded as follows: for $i = 1, \dots, n_u$,

$$\gamma_u^{(t)} \leq \|\mathbf{D}_{U_i}^{(t)}\|_2 \|\widetilde{\mathbf{M}}_{I_i^U}^{(t)}\|_2 \|\mathbf{L}_{U_i}^{(t)}\|_2 \leq \|\mathbf{D}_{U_i}^{(t)}\|_2 \|\widetilde{\mathbf{M}}_{I_i^U}^{(t)}\|_2 \left(n_f \max_{j \in \{1, \dots, n_f\}} \mathbf{L}_{U_i}^{(t)(j)\top} \mathbf{L}_{U_i}^{(t)(j)} \right)^{1/2},$$

where $\mathbf{L}_{U_i}^{(t)(j)}$ denotes the j th column of $\mathbf{L}_{U_i}^{(t)}$. This inequality indicates that a smaller maximum column norm of $\mathbf{L}_{U_i}^{(t)}$ leads to a smaller $\gamma_u^{(t)}$. Consequently, to minimize the upper bound of MFSE ($\widetilde{\mathbf{U}}^{(t)}$), it is essential to maintain both $\|\mathbf{L}_{U_i}^{(t)}\|_F$ and the column norms of $\mathbf{L}_{U_i}^{(t)}$ at minimal levels.

This requirement motivates a core-elements selection criterion similar to those proposed in Li et al. (2024a) and Li et al. (2024b). Specifically, for each i , given a subsampling rate r , the sketch $\widetilde{\mathbf{M}}_{I_i^U}^{(t)*}$ is constructed by retaining $\lfloor |I_i^U| \times r \rfloor$ elements with the largest absolute values in each column of $\widetilde{\mathbf{M}}_{I_i^U}^{(t)}$ and setting the remaining elements to zero. Intuitively, this approach ensures that $\mathbf{L}_{U_i}^{(t)}$ has approximately minimal column norms for each column. As a result, both $\|\mathbf{L}_{U_i}^{(t)}\|_F$ and $\|\mathbf{L}_{U_i}^{(t)}\|_2$ are approximately minimized, thereby achieving a relatively small upper bound for the MFSE ($\widetilde{\mathbf{U}}^{(t)}$) as stated in Proposition 1.

Similarly, to minimize the upper bound of MFSE ($\widetilde{\mathbf{M}}^{(t)}$), we adopt an analogous core-elements selection criterion: for each j , given a subsampling rate r , the sketch $\widetilde{\mathbf{U}}_{I_j^M}^{(t)*}$ is constructed by retaining $\lfloor |I_j^M| \times r \rfloor$ elements with the largest absolute values in each column of $\widetilde{\mathbf{U}}_{I_j^M}^{(t)}$ and setting the remaining elements to zero. This strategy also ensures that $\mathbf{L}_{M_j}^{(t)}$ maintains approximately minimal column norms for each column, thereby minimizing both $\|\mathbf{L}_{M_j}^{(t)}\|_F$ and $\|\mathbf{L}_{M_j}^{(t)}\|_2$. Consequently, the upper bound of the MFSE ($\widetilde{\mathbf{M}}^{(t)}$) is kept relatively small.

We have also implemented a version using a fixed subsample size to ensure applicability across diverse experimental settings, and further compared row-wise and block-wise variants with the original column-wise scheme to demonstrate the advantages of column-wise core-elements sampling; see our Appendix for details.

According to the core-elements selection criterion abovementioned, Algorithm 1 provides the specific form of core-elements subsampling algorithm, which will be used in the acceleration of ALS.

Algorithm 1: Core-elements subsampling algorithm (CES)

- 1: **Input:** $\mathbf{X} = (x_{ij}) \in \mathbb{R}^{n \times p}$, subsampling rate $r \in (0, 1)$
 - 2: **Initialize** $\mathbf{S} = (0) \in \mathbb{R}^{n \times p}$
 - 3: **For** $j = 1, \dots, p$ **do**
 - 4: **Let** $\mathcal{J} = \{i_1, \dots, i_s\}$ be an index set, $s = r \times n$, s.t. $\{|x_{i_q j}|\}_{q=1}^s$ are the s largest ones among $\{|x_{ij}|\}_{i=1}^n$
 - 5: **Let** $s_{i_q j} = 1, q = 1, \dots, s$
 - 6: **End for**
 - 7: **Let** $\mathbf{X}^* = \mathbf{S} \odot \mathbf{X}$ where \odot represents the element-wise product
 - 8: **Return** \mathbf{X}^*
-

Combining the abovementioned procedures, Algorithm 2 summarizes the Core-ALS method, which uses the core-elements subsampling method to construct the sparse sketch for every regression. Schematic of the Core-ALS method is shown in Fig. 2.

Algorithm 2: Core-ALS

- 1: **Input:** rating matrix $\mathbf{R} \in \mathbb{R}^{n_u \times n_m}$, implicit vector dimension n_f , subsampling rate r
 - 2: **Initialize** Matrices \mathbf{M} with ranks of n_f and let $\widetilde{\mathbf{M}}^{(0)} = \mathbf{M}$
 - 3: **Repeat**
 - 4: **For** $i = 1, \dots, n_u$:
 - 5: Construct the sparse sketch: $\widetilde{\mathbf{M}}_{I_i^U}^{(t)*} = \text{CES}(\widetilde{\mathbf{M}}_{I_i^U}^{(t)}, r)$
 - 6: Update $\widetilde{\mathbf{u}}_i^{(t+1)}$ with (6)
 - 7: **For** $j = 1, \dots, n_m$:
 - 8: Construct the sparse sketch: $\widetilde{\mathbf{U}}_{I_j^M}^{(t+1)*} = \text{CES}(\widetilde{\mathbf{U}}_{I_j^M}^{(t+1)}, r)$
 - 9: Update $\widetilde{\mathbf{m}}_j^{(t+1)}$ with (7)
 - 10: **Until** Convergence
 - 11: **Return** $\widetilde{\mathbf{U}}$ and $\widetilde{\mathbf{M}}$
-

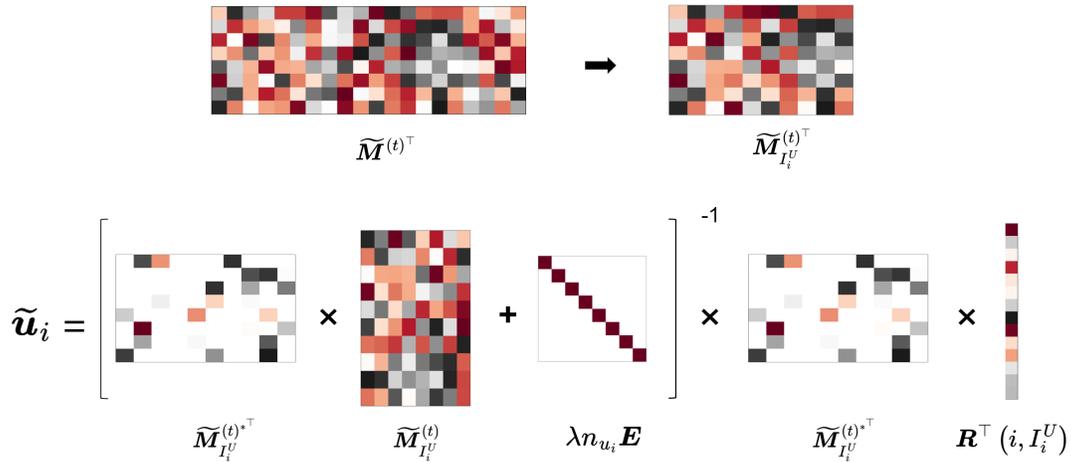


Figure 2: Schematic of the flow of the Core-ALS.

In Algorithm 2, selecting elements of the $\widetilde{\mathbf{U}}^{(t)}$ or $\widetilde{\mathbf{M}}^{(t)}$ slices before each regression step can lead to some time loss during execution. To mitigate this issue, we optimized the sampling algorithm by employing Partial Quicksort (Martinez, 2004) to sort the elements based on their magnitudes (see Appendix for details). This optimization ensures that our sampling process remains nearly lossless in the case of large-scale matrices. We reported sorting

costs in the Appendix to show the efficiency of our sorting procedure. Furthermore, our method provides a distinct advantage over alternative techniques that require computing sampling probabilities, such as leverage score-based sampling, which often incur substantial computational costs.

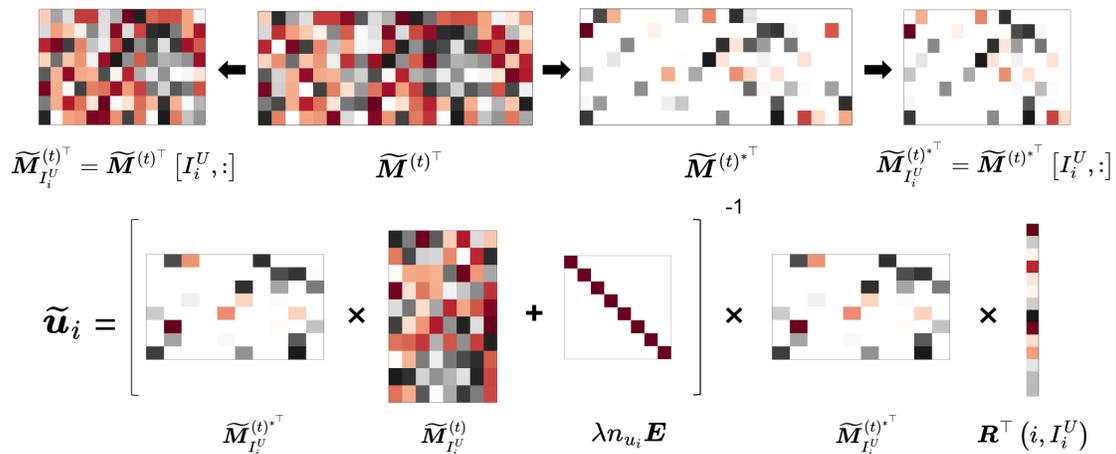


Figure 3: Fast variant of the Core-ALS.

In addition, we introduce a fast variant of the algorithm, which performs sampling on $\tilde{\mathbf{U}}^{(t)}$ and $\tilde{\mathbf{M}}^{(t)}$ before each iteration rather than repeatedly sampling on $\tilde{\mathbf{U}}_{I_j^M}^{(t)}$ and $\tilde{\mathbf{M}}_{I_i^U}^{(t)}$. While this approach does not guarantee exact adherence to the specified sampling ratio, it ensures that the time complexity does not exceed that of the original algorithm. Our experimental results suggest that this variant significantly lowers the sampling overhead while maintaining relatively high accuracy. Schematic of the fast variant of the Core-ALS is shown in Fig. 3 and a detailed description of this method is provided in the Appendix.

Moreover, the core-elements estimation proposed here is specifically designed for the explicit-feedback ALS. To extend core-elements estimation to the implicit-feedback setting, we combine it with the implicit-feedback variant of ALS in the Appendix and provide the corresponding experiments.

4 Theoretical Results

In this section, we first demonstrate that the core-elements estimation achieves the $(1 + \epsilon)$ -approximation w.r.t. the full sample estimation (3) and (4). Then, we show the convergence of the algorithm. Finally, the time complexity of the two proposed algorithms is given. Technical proofs are provided in the Appendix.

4.1 Approximation Guarantee

Theorems 1 and 2 provides non-asymptotic relative error bounds for the proposed core-elements estimation $\tilde{\mathbf{u}}_i$ and $\tilde{\mathbf{m}}_j$ in each iteration, respectively.

Theorem 1. Let $\tilde{\mathbf{M}}_{I_i^U}^{(t)*}$ be the sparse sketch of $\tilde{\mathbf{M}}_{I_i^U}^{(t)}$, and recall $\tilde{\mathbf{u}}_i^{(t+1)}$ defined in (6). Suppose

$$\|\tilde{\mathbf{M}}_{I_i^U}^{(t)} - \tilde{\mathbf{M}}_{I_i^U}^{(t)*}\|_2 \leq \epsilon'_m \|\tilde{\mathbf{M}}_{I_i^U}^{(t)}\|_2,$$

where

$$0 < \epsilon'_m \leq \frac{1}{c_m^{(t)}} \left[1 + \frac{c_m^{(t)} + 1}{(\sqrt{1 + \epsilon_m} - 1) \text{RSSE}(\hat{\mathbf{u}}_i^{(t+1)})} \right]^{-1}.$$

Under these conditions, we have

$$\|\mathbf{R}^\top(i, I_i^U) - \tilde{\mathbf{M}}_{I_i^U}^{(t)} \tilde{\mathbf{u}}_i^{(t+1)}\|^2 \leq (1 + \epsilon_m) \|\mathbf{R}^\top(i, I_i^U) - \tilde{\mathbf{M}}_{I_i^U}^{(t)} \hat{\mathbf{u}}_i^{(t+1)}\|^2. \quad (10)$$

In (10), $c_m^{(t)} = \|\tilde{\mathbf{M}}_{I_i^U}^{(t)}\|_2^2 \left\| \left(\tilde{\mathbf{M}}_{I_i^U}^{(t)\top} \tilde{\mathbf{M}}_{I_i^U}^{(t)} + \lambda n_{u_i} \mathbf{E} \right)^{-1} \right\|_2$ and $\text{RSSE}(\hat{\mathbf{u}}_i^{(t+1)}) = \|\mathbf{R}^\top(i, I_i^U) - \tilde{\mathbf{M}}_{I_i^U}^{(t)} \hat{\mathbf{u}}_i^{(t+1)}\| / \|\mathbf{R}^\top(i, I_i^U)\|$ is the relative sum of squares error (RSSE) of the full sample estimation $\hat{\mathbf{u}}_i^{(t+1)}$.

Theorem 2. Let $\tilde{\mathbf{U}}_{I_j^M}^*$ be the sparse sketch of $\tilde{\mathbf{U}}_{I_j^M}$, and $\tilde{\mathbf{m}}_j$ be defined in (7). Under conditions similar to those in Theorem 1, we also have

$$\|\mathbf{R}^\top(j, I_j^M) - \tilde{\mathbf{U}}_{I_j^M}^{(t+1)} \tilde{\mathbf{m}}_j^{(t+1)}\|^2 \leq (1 + \epsilon_u) \|\mathbf{R}^\top(j, I_j^M) - \tilde{\mathbf{U}}_{I_j^M}^{(t+1)} \hat{\mathbf{m}}_j^{(t+1)}\|^2. \quad (11)$$

The complete version of this theorem can be found in the Appendix.

Theorems 1 and 2 indicate that to achieve the $(1+\epsilon)$ -approximation, Algorithm 2 requires sketches $\widetilde{\mathbf{M}}_{I_i^U}^{(t)*}$ and $\widetilde{\mathbf{U}}_{I_j^M}^{(t+1)*}$ such that the ratios $\|\widetilde{\mathbf{M}}_{I_i^U}^{(t)} - \widetilde{\mathbf{M}}_{I_i^U}^{(t)*}\|_2 / \|\widetilde{\mathbf{M}}_{I_i^U}^{(t)}\|_2$ and $\|\widetilde{\mathbf{U}}_{I_j^M}^{(t+1)} - \widetilde{\mathbf{U}}_{I_j^M}^{(t+1)*}\|_2 / \|\widetilde{\mathbf{U}}_{I_j^M}^{(t+1)}\|_2$ are $O(\epsilon^{1/2})$, respectively.

4.2 Convergence Guarantee

Theorem 3. We use the superscript (t) to represent the t th step of the iteration. $\widetilde{\mathbf{M}}_{I_i^U}^{(t)*}$ is the sparse sketch of $\widetilde{\mathbf{M}}_{I_i^U}^{(t)}$ and $\widetilde{\mathbf{u}}_i^{(t+1)}$ is defined by (6). $\widetilde{\mathbf{U}}_{I_j^M}^{(t+1)*}$ is the sparse sketch of $\widetilde{\mathbf{U}}_{I_j^M}^{(t+1)}$ and $\widetilde{\mathbf{m}}_j^{(t+1)}$ is defined by (7). $\epsilon_m, \epsilon_u, \epsilon'_m, \epsilon'_u$ are defined in Theorems 1 and 2, respectively.

Suppose the full sample estimator's convergence rate is C . When C satisfies $C \leq \min\{1/(1+\epsilon_u), 1/(1+\epsilon_m)\}$, ϵ'_m and ϵ'_u always satisfy conditions in Theorems 1 and 2, i.e., are $O(\epsilon^{1/2})$.

We have

$$\sum_{j=1}^{n_m} \|\mathbf{R}^\top(j, I_j^M) - \widetilde{\mathbf{U}}_{I_j^M}^{(t+1)} \widetilde{\mathbf{m}}_j^{(t+1)}\|^2 \leq \sum_{i=1}^{n_u} \|\mathbf{R}^\top(i, I_i^U) - \widetilde{\mathbf{M}}_{I_i^U}^{(t)} \widetilde{\mathbf{u}}_i^{(t+1)}\|^2, \quad (12)$$

$$\sum_{i=1}^{n_u} \|\mathbf{R}^\top(i, I_i^U) - \widetilde{\mathbf{M}}_{I_i^U}^{(t)} \widetilde{\mathbf{u}}_i^{(t+1)}\|^2 \leq \sum_{j=1}^{n_m} \|\mathbf{R}^\top(j, I_j^M) - \widetilde{\mathbf{U}}_{I_j^M}^{(t)} \widetilde{\mathbf{m}}_j^{(t)}\|^2. \quad (13)$$

Combining the equation (12) and equation (13), we have

$$\sum_{j=1}^{n_m} \|\mathbf{R}^\top(j, I_j^M) - \widetilde{\mathbf{U}}_{I_j^M}^{(t+1)} \widetilde{\mathbf{m}}_j^{(t+1)}\|^2 \leq \sum_{j=1}^{n_m} \|\mathbf{R}^\top(j, I_j^M) - \widetilde{\mathbf{U}}_{I_j^M}^{(t)} \widetilde{\mathbf{m}}_j^{(t)}\|^2,$$

i.e.,

$$\mathcal{L}_\lambda^{\text{reg}}(\mathbf{R}, \mathbf{U}, \mathbf{M})^{(t+1)} \leq \mathcal{L}_\lambda^{\text{reg}}(\mathbf{R}, \mathbf{U}, \mathbf{M})^{(t)}.$$

Under these conditions, the error function is monotonically decreasing, so the algorithm converges.

Theorem 3 indicates that as long as the full sample estimator's convergence rate is small enough and the sampled matrix closely approximates the original matrix, Core-ALS achieves convergence.

4.3 Time Complexity Analysis

Theorem 4 provides specific time complexity for the proposed core-elements ALS method.

Theorem 4. *Suppose r is the subsampling rate, $nnz(R)$ is the number of non-missing values in the rating matrix \mathbf{R} . For the Core-ALS method, each step of updating \mathbf{U} takes*

$$O(n_f (nnz(R) \times (rn_f) + n_f^2 n_u)).$$

while each step of updating \mathbf{M} takes

$$O(n_f (nnz(R) \times (rn_f) + n_f^2 n_m)).$$

If the Core-ALS method takes a total of n_t rounds to stop, it runs in time

$$O(n_f (nnz(R) \times (rn_f) + n_f^2 n_m + n_f^2 n_u) n_t). \tag{14}$$

Theorem 4 shows that Core-ALS substantially reduces the computational cost of ALS. When the sampling probability is sufficiently small, the overall time complexity of the algorithm can reach $O(n_f^3 (n_m + n_u) n_t)$.

5 Simulation Studies

In this section, we evaluate the performance of the Core-ALS method using synthetic data. We use CORE to refer to the estimator in Algorithm 2. For comparison, we consider several state-of-the-art subsampling methods including uniform subsampling (UNIF), basic leverage subsampling (BLEV) (Ma and Sun, 2015; Cheng et al., 2016). We have also compared with two production-grade speed-up baselines, SparkALS (Winlaw et al., 2015) and SGD-ALS, in the Appendix. All experiments were implemented using the R programming language on a server with 256 GB RAM and 64 cores Intel[®] Xeon[®] Gold 5218 CPU.

5.1 Estimation Accuracy under Different Parameter Settings

To construct a low-rank rating matrix \mathbf{R} , we first generate two low-rank factor matrices \mathbf{U} and \mathbf{M} and then form an initial low-rank matrix $\mathbf{R}^o = \mathbf{U}\mathbf{M}^\top$ before sparsification. The factor matrices \mathbf{U} and \mathbf{M} are generated from one of the following widely used multivariate distributions:

D1. multivariate normal distribution, $N(\mathbf{0}, \mathbf{\Sigma})$;

D2. multivariate log-normal distribution, $LN(\mathbf{0}, \mathbf{\Sigma})$;

D3. multivariate t-distribution with 4 degrees of freedom, $t_4(\mathbf{0}, \mathbf{\Sigma})$, where $\mathbf{\Sigma} = (\sigma_{ij}) \in \mathbb{R}^{p \times p}$ is a covariance matrix with $\sigma_{ij} = 0.6^{|i-j|}$ for $i, j = 1, \dots, p$.

After constructing the initial low-rank matrix \mathbf{R}^o , we add Gaussian noise $\varepsilon_{ij} \sim N(0, 1)$ to each entry to introduce stochastic perturbations. Subsequently, we randomly remove a proportion of the entries to achieve a prescribed sparsity ratio α . Specifically, we randomly select $(1 - \alpha) \times 100\%$ of the entries and set them to NaN. We consider $\alpha \in \{0.4, 0.5, 0.6, 0.7\}$, which correspond to rating matrices denoted by **R1** through **R4**, respectively. Note that **R1** represents a relatively dense matrix, while **R4** is relatively sparse.

We fix the size of the rating matrix as $(n_u, n_m) = (3600, 3600)$ and choose the implicit vector dimension $n_f = 60$. For the three subsampling methods, i.e., UNIF, BLEV, and CORE, we select a subsampling rate $r \in \{0.1, 0.15, 0.2, 0.25\}$ for each regression problem. Under these chosen subsampling rates, all three methods maintain theoretically comparable time complexity.

Due to the sparsity of the rating matrix, we take the position of the non-missing value as the empirical set and the position of the missing value as the prediction set. We calculate the empirical relative mean squared error (ReMSE) for each of the estimators based on one

hundred replications, i.e.,

$$\text{ReMSE}(\tilde{\mathbf{R}}) = \frac{1}{100} \sum_{n=1}^{100} \frac{\sqrt{\sum_{(i,j) \in I} (r_{ij} - \tilde{r}_{ij})^2}}{\sqrt{\sum_{(i,j) \in I} r_{ij}^2}},$$

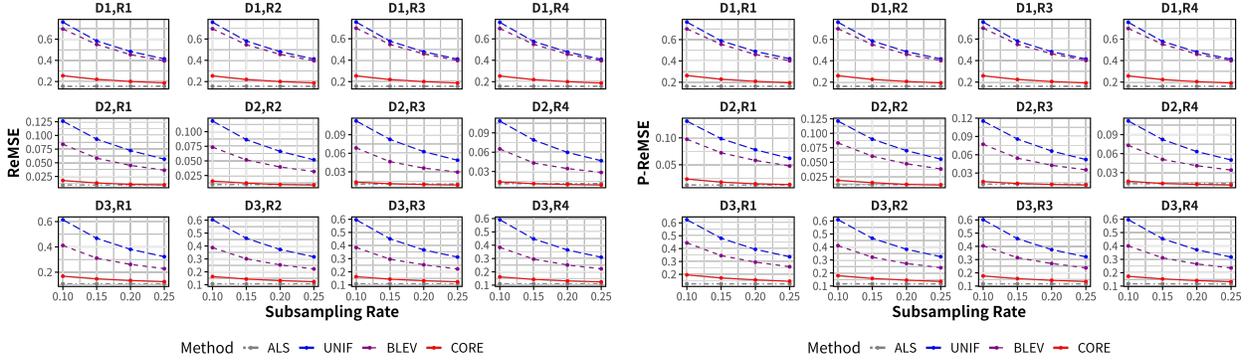
where I represents the set of non-missing value locations, \tilde{r} represents the elements of the approximate matrix $\tilde{\mathbf{R}}$ obtained by the ALS estimator and calculate the prediction relative MSE (P-ReMSE) for each of the estimators based on one hundred replications, i.e.,

$$\text{P-ReMSE}(\tilde{\mathbf{R}}) = \frac{1}{100} \sum_{n=1}^{100} \frac{\sqrt{\sum_{(i,j) \in I^c} (r_{ij} - \tilde{r}_{ij})^2}}{\sqrt{\sum_{(i,j) \in I^c} r_{ij}^2}},$$

where I^c represents the set of missing value locations.

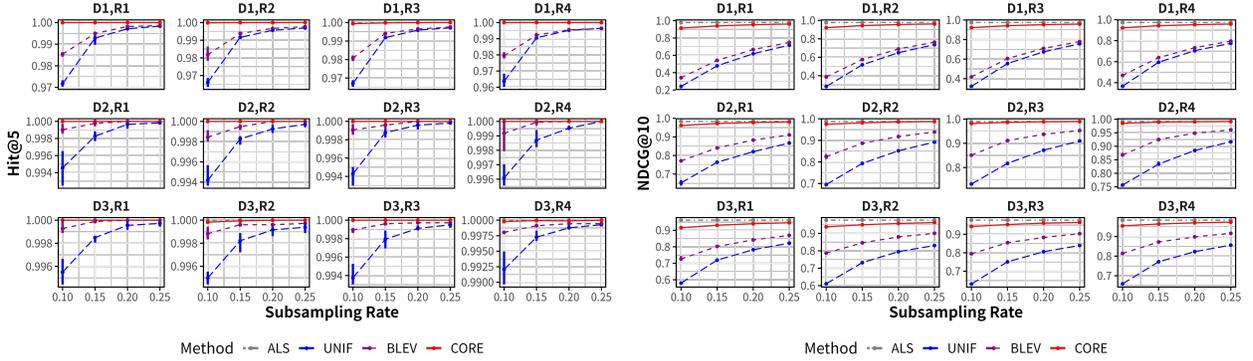
We also evaluate two widely-used metrics in recommender systems: Hit@5 and NDCG@10 (score-based version). Hit@5 measures whether the ground-truth item appears in the top-5 recommendations. NDCG@10 considers both the relevance and rank of recommended items. Our setting is explicit-feedback ALS for matrix reconstruction without a temporal dimension; recommendations are produced by ranking predicted ratings for unrated items. In contrast, classical leave-one-out or temporal split strategies were designed primarily for implicit-feedback scenarios or time-ordered recommendation scenarios. Randomly selecting a rating from the test set as the leave-one-out target would not reliably represent an item that should be recommended in this setting. Therefore, we consider the top 95% of scores in the test set as the items that are actually recommended. To demonstrate robustness, we also included complementary experiments under the leave-one-out setting in the Appendix. The results of four metrics versus different subsample sizes are shown in Fig. 4.

In Fig. 4, we observe that both MSE and PMSE w.r.t. all estimators decrease as r increases. We also observe that CORE consistently outperforms all other methods by a large margin on both Hit and NDCG metrics. This observation demonstrates that the proposed estimator achieves superior accuracy compared to existing methods by effectively leveraging the information embedded in the rating matrix. In particular, the core-elements approach



(a) ReMSE for different D and R.

(b) P-ReMSE for different D and R.



(c) Hit@5 for different D and R.

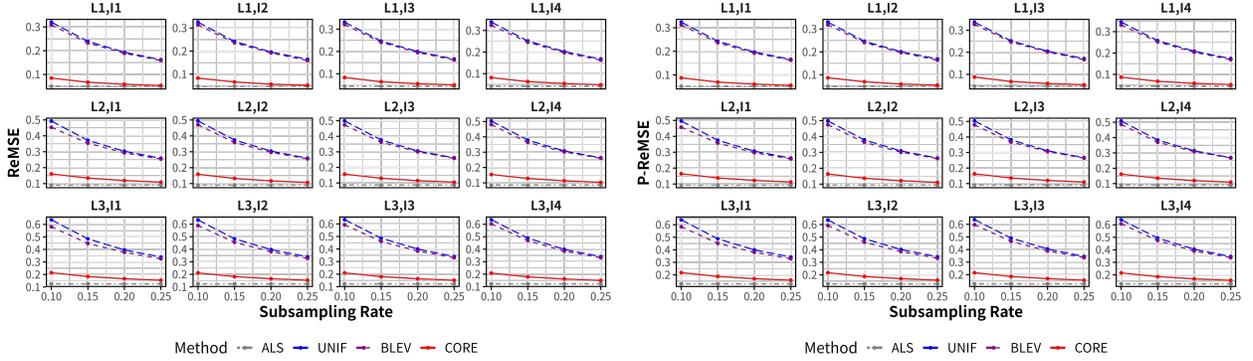
(d) NDCG@10 for different D and R.

Figure 4: Performance of four methods under different distributions and densities.

ensures that, at each iteration, the estimator remains approximately unbiased and maintains an approximately minimized estimation variance, thereby conferring a clear advantage over competing techniques.

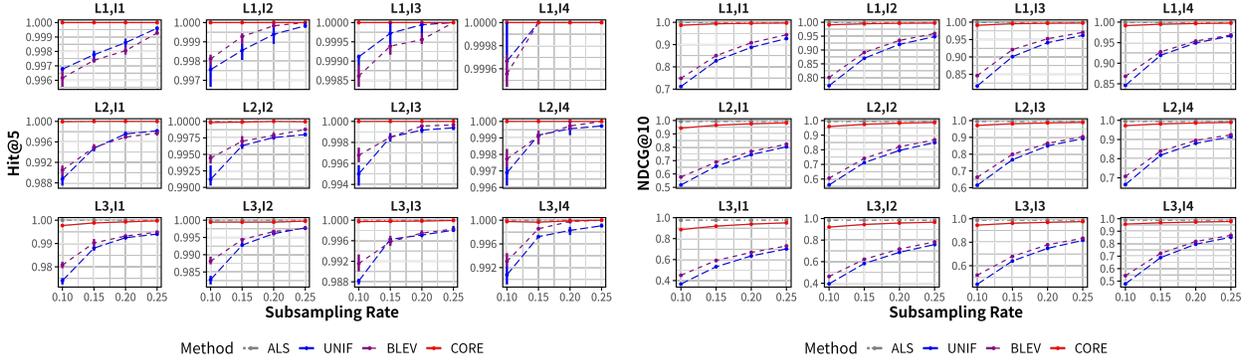
To further test the effectiveness of our algorithm, we fix the distribution of \mathbf{U} and \mathbf{M} , as well as the density of the rating matrix \mathbf{R} . Let $\lambda \in \{0.05, 0.1, 0.15\}$, referred to **L1** – **L3**. Note that **L1** corresponds to relatively small penalty terms, and **L3** corresponds to relatively large penalty terms. Let implicit vector dimension $n_f \in \{40, 50, 60, 70\}$, referred to **I1** – **I4**. We investigate the performance of the algorithm under different regularization parameters and different implicit vector dimensions in Fig. 5. The corresponding running times and memory consumption are provided in the Appendix.

As shown in Fig. 5, the CORE method still substantially outperforms other sampling



(a) ReMSE for different L and I.

(b) P-ReMSE for different L and I.



(c) Hit@5 for different L and I.

(d) NDCG@10 for different L and I.

Figure 5: Performance of four methods under different implicit vector dimensions and regularizations.

methods across all metrics under different implicit vector dimensions and regularizations, and it matches the performance of the full-sample method even at a small subsampling rate.

5.2 Convergence Performance

In addition to estimation accuracy, we also focus on the convergence speed, as faster convergence requires fewer iterations and thus less computational time. We define convergence as the point where the iteration error falls below 0.01. Throughout this experiment, the rating matrix dimensionality is kept fixed, the sparsity is set to 20%, λ is fixed at 0.2, and the implicit vector dimension is set to 60. Figure 6 shows how the ReMSE changes w.r.t the number of iterations for different algorithms under four subsampling rates.

As shown in Fig. 6, with the increase of the sampling rate, the convergence speed of all

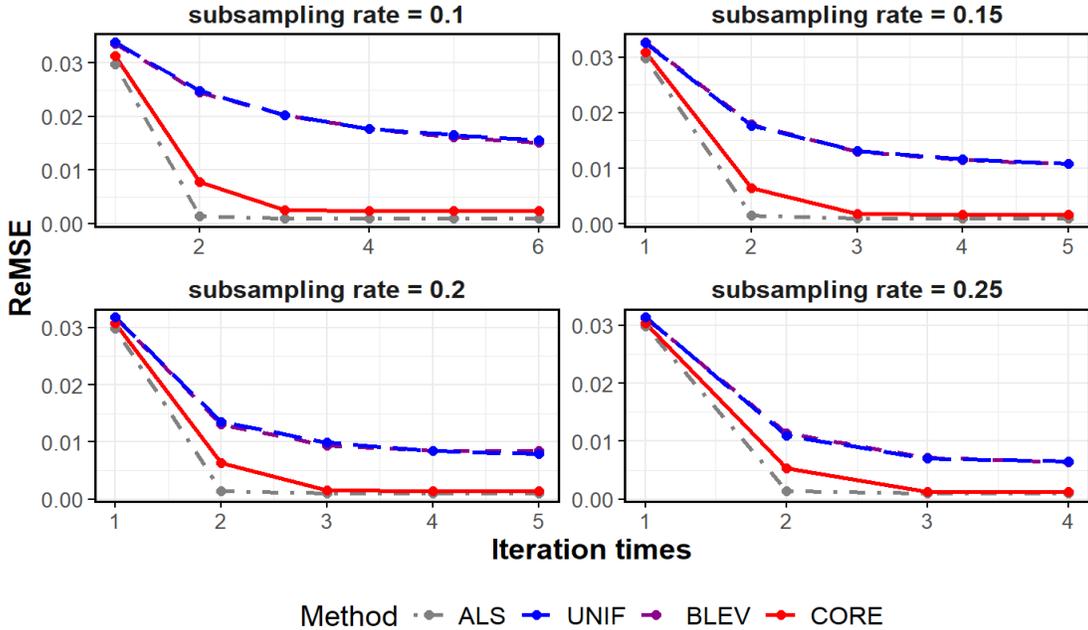


Figure 6: ReMSE with respect to the number of iterations.

methods accelerates, aligning well with practical expectations. Additionally, we observe that the convergence speed of the CORE method is consistently close to that of the full-sample ALS, and it also requires a similar number of iterations.

5.3 Computing Time

Next, we provide the specific runtime of the algorithms. The size of the rating matrix is set to 5000×5000 , with the implicit vector dimension being set to 50, and the other parameters are consistent with those of **L1, I1**. Because BLEV incurs significant overhead in sampling processes, it does not lead to a notable acceleration of ALS. Therefore, we only present the runtime of UNIF here, as its sampling overhead is ignorable. Thus, for the same sampling ratio, it serves as a lower bound for the algorithm’s runtime. Results are shown in Table 1.

As shown in Table 1, CORE significantly reduces the runtime of ALS at various sampling ratios, with speeds only slightly slower than UNIF. Due to hardware constraints, we were unable to scale the matrix size indefinitely. Additionally, to obtain more stable results, we did

Table 1: Computing time for different methods

Subsampling rate	FULL	0.1	0.15	0.2	0.25
CORE	-	50.50s	65.82s	75.68s	83.00s
UNIF	-	36.65s	50.40s	59.63s	65.14s
ALS	222.03s	-	-	-	-

not extract additional samples, since we needed to ensure that the number of sampled rows exceeded the latent dimension n_f , thus avoiding potential singularities in the calculations. In fact, for very large matrices, the sampling overhead of the CORE method becomes negligible, while the speedup achieved through sparse matrix multiplication becomes increasingly significant.

6 Real Data Analysis

6.1 Netflix Competition Data

The Netflix Competition Data (Netflix, 2006) stemmed from a large-scale data mining competition organized by Netflix to identify the most accurate recommender system for predicting user movie ratings. The training data comprises more than 100 million user ratings, contributed by over 480,000 users for 17,700 movies. Each record is stored as a quadruple (user, movie, date, rating), where the rating is an integer from 1 to 5. We employ this dataset to assess the performance of our Core-ALS method.

6.1.1 Estimation Accuracy

Given the huge scale of the original dataset and the memory constraints of our computing infrastructure, we constructed a reduced yet representative subset by selecting the top 10,000

users and the top 10,000 movies based on interaction frequency. For each run of our repeated experiments, 80% of the ratings were randomly sampled to form the training set, while the remaining 20% constituted the test set. Subsequently, we treat the items in the test set with ratings greater than or equal to 4 as the true recommendation targets. We evaluated the performance of all compared methods using four commonly adopted metrics: Relative Mean Square Error (ReMSE), Prediction ReMSE (P-ReMSE), Hit Ratio at rank 5 (Hit@5), and Normalized Discounted Cumulative Gain at rank 10 (NDCG@10).

We also evaluate each method under different subsampling rates, regularization parameters, and implicit vector dimensions. We set $\lambda = \{0.1, 0.15, 0.2\}$, referred to **L1** – **L3** and set $n_f = \{20, 25, 30, 35\}$, referred to **I1** – **I4**. Figure 7 shows the relationships between four metrics and the subsampling rate under different regularization parameters and implicit vector dimensions.

As shown in Fig. 7, the ReMSE and P-ReMSE of all methods decrease as the subsampling rate increases, which is consistent with expectations. Among all methods, CORE consistently remains closest to the full-sample method. Moreover, as the subsampling rate increases, CORE exhibits an overall upward trend in both Hit and NDCG metrics, demonstrating its ability to achieve recommendation performance comparable to the full-sample method. We have also included additional experiments on the full Netflix dataset in the Appendix to demonstrate that CORE can effectively handle 100 million interactions.

6.1.2 Computing Time

For a comparison of runtime, we selected ratings from 40,000 users for 15,000 movies. This dataset is highly sparse, with only about 1% of non-missing values. The implicit vector dimension is set to 80, with the remaining parameters consistent with the settings mentioned earlier.

Although the rating matrix seems large, its extreme sparsity makes its effective size

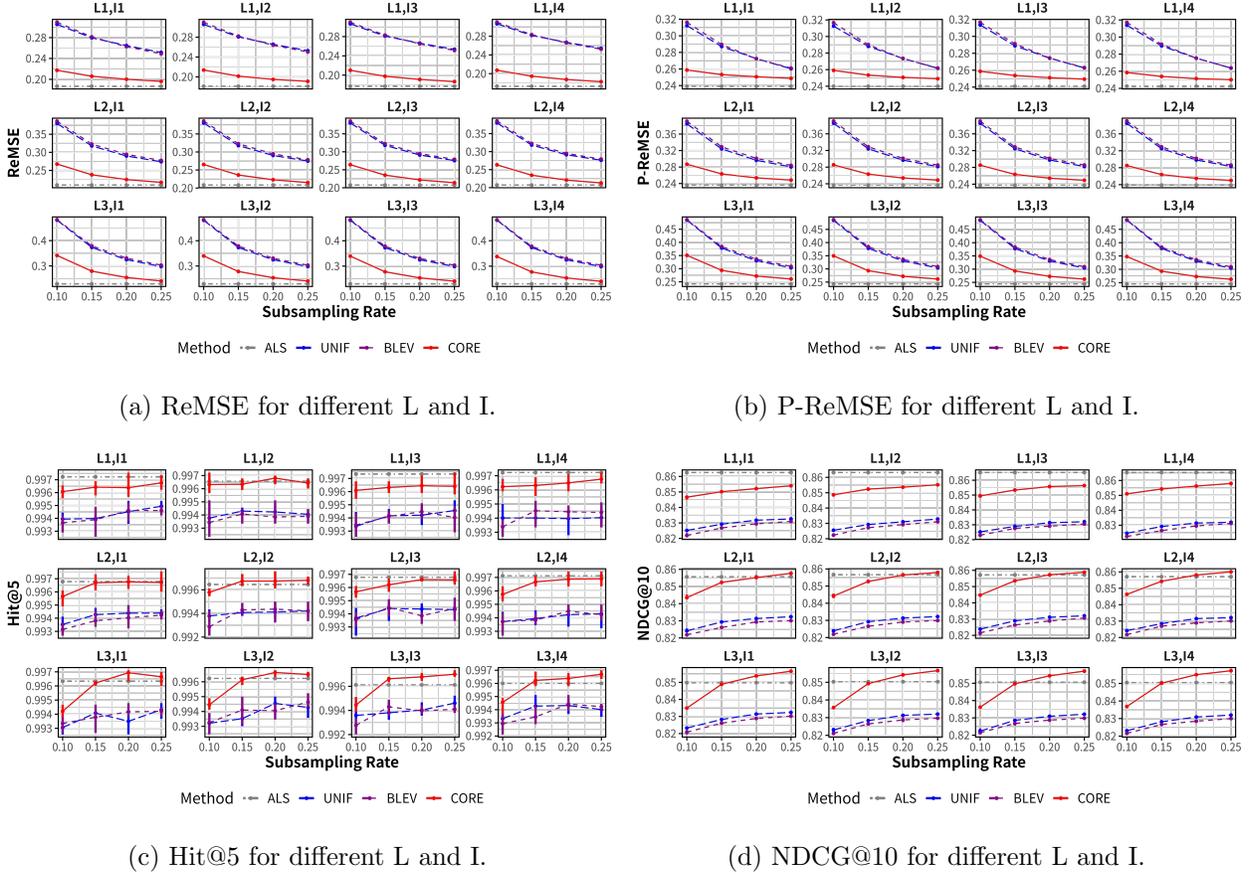


Figure 7: Performance of four methods under Netflix data.

comparable to that of the simulation. Consequently, the results obtained by Table 2 are consistent with those of Table 1.

6.2 Bark Texture Image Data

To illustrate the performance of the Core-ALS method more clearly, we applied it to an image restoration task. We observed that the ALS algorithm demonstrated superior performance in restoring low-rank images. Therefore, we selected several images from the bark texture image dataset (Truong Hoang, 2020) and randomly occluded a subset of pixels in each image. Subsequently, the masked images were restored using both ALS and CORE algorithms. The restoration results from ALS and CORE were then compared to assess the accuracy and effectiveness of the CORE algorithm in the context of image restoration.

Table 2: Computing time for different methods

Subsampling rate	FULL	0.1	0.15	0.2	0.25
CORE	-	137.35s	162.51s	185.72s	196.95s
UNIF	-	115.37s	145.79s	155.33s	168.70s
ALS	606.15s	-	-	-	-

In this experiment, 60% of the pixels in each image were randomly masked. For the ALS algorithm, the dimensionality of the implicit vector was set to 50, with a regularization parameter (λ) of 0.01 and 5 iterations. The CORE algorithm used a sampling proportion of 0.15, with all other parameters aligned with those of the ALS algorithm. Figure 8 displays the restoration results for three selected images from the dataset.

As shown in Fig. 8, even with a sampling rate as low as 0.15, the CORE method achieves image recovery performance almost identical to that of ALS.

In conclusion, for the real data, the test results are in good agreement with our simulation results, which reflects the effectiveness of our proposed core-elements algorithm.

7 Conclusion

This study has introduced an innovative core-elements method for enhancing the efficiency of the alternating least squares (ALS) algorithm used in matrix factorization for recommender systems. The method strategically selects a subset of elements, optimizing computational resources while maintaining the integrity and accuracy of the model predictions.

Our theoretical analyses establish strong guarantees for the approximation and convergence of the proposed method. These results confirm that the core-elements method not only retains the predictive power of the full dataset but does so with significantly reduced

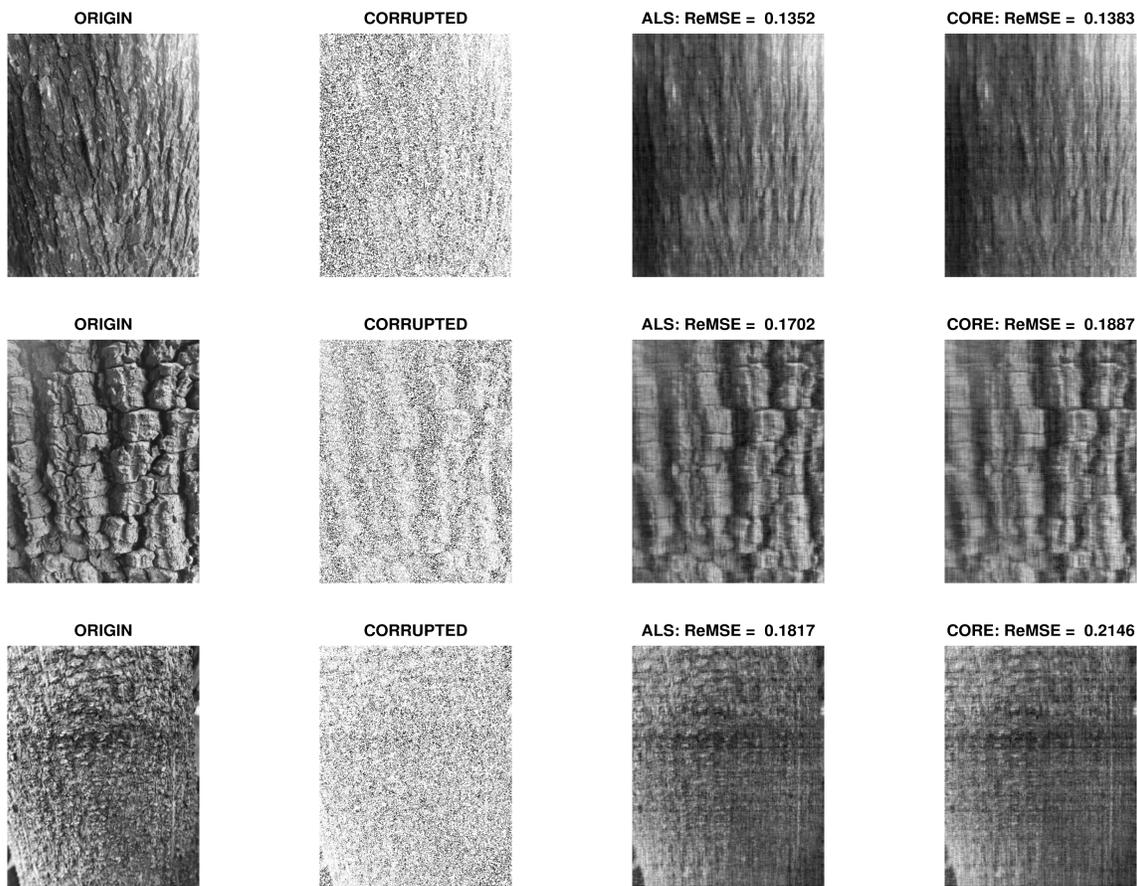


Figure 8: Comparison of bark texture image restoration tasks.

computational overhead. This makes the method particularly useful in settings where computational resources are limited or when dealing with extremely large datasets.

Practical applications using both simulated and real-world datasets have demonstrated the effectiveness of the core-elements method. In all tested scenarios, the method performed on par or better than traditional subsampling methods for ALS, especially in terms of computational efficiency. This was evident in our experiments with the Netflix Prize dataset, where the core-elements method consistently showed superior performance compared to other subsampling techniques.

This accelerated algorithm has also prompted several potential avenues for future research. Specifically, determining the optimal hyperparameters that yield the best lower bound for

acceleration, analyzing the convergence rates under various sampling probabilities, and exploring richer sampling strategies such as block or row-column hybrid sampling are aspects that remain unexplored in this study. We anticipate that further investigation will enable the development of a robust theoretical framework. Additionally, our current method focuses solely on accelerating ALS in terms of runtime, without improvements in memory usage. Reducing memory overhead is an important direction, and we plan to explore memory-efficient implementations in future work. Moreover, we aim to extend the accelerated algorithm to low-rank tensor decomposition. [Cheng et al. \(2016\)](#) applied leverage score-based sampling methods to this area, but experimental results indicate that such methods are less effective in matrix factorization, and the computation of leverage scores imposes a considerable computational burden. Therefore, we contend that accelerating core element sampling presents a promising alternative.

Furthermore, with the advancement of computational tools, we are keen to explore the application of tensor decomposition to large-scale, real-world data sets, addressing critical scientific challenges, and furthering our understanding across various research domains.

Acknowledgments

This work is supported by Beijing Municipal Natural Science Foundation No. 1232019, National Natural Science Foundation of China Grant No. 12301381, and Renmin University of China research fund program for young scholars.

Supplementary Material

Appendix: The Appendix contains additional content of the main text and complete proofs of theoretical results, technical details of the theoretical results. Additional numerical

experiments and results of various extended versions of Core-ALS are also provided.
(appendix.pdf, a pdf file)

Code: The zip file contains the R code that implements the proposed method and reproduces the numerical results. A README file is included to explain the contents. (code.zip)

Disclosure Statement

The authors report there are no competing interests to declare.

References

- Abdi, H. (2007). Singular value decomposition (SVD) and generalized singular value decomposition. *Encyclopedia of measurement and statistics 907*, 44.
- Ai, M., F. Wang, J. Yu, and H. Zhang (2021). Optimal subsampling for large-scale quantile regression. *Journal of Complexity 62*, 101512.
- Ai, M., J. Yu, H. Zhang, and H. Wang (2021). Optimal subsampling algorithms for big data regressions. *Statistica Sinica 31*(2), 749–772.
- Balle, B., G. Barthe, and M. Gaboardi (2020). Privacy profiles and amplification by subsampling. *Journal of Privacy and Confidentiality 10*.
- Bi, X., A. Qu, J. Wang, and X. Shen (2017). A group-specific recommender system. *Journal of the American Statistical Association 112*(519), 1344–1353.
- Cheng, D., R. Peng, Y. Liu, and I. Perros (2016). Spals: Fast alternating least squares via implicit leverage scores sampling. *Advances in neural information processing systems 29*.
- Drineas, P., M. W. Mahoney, and S. Muthukrishnan (2006). Sampling algorithms for l2 regression and applications. In *Proceedings of the seventeenth annual ACM-SIAM symposium on Discrete algorithm*, pp. 1127–1136.
- Han, Y., J. Yu, N. Zhang, C. Meng, P. Ma, W. Zhong, and C. Zou (2025). Leverage classifier: Another look at support vector machine. *Statistica Sinica 35*(3).

- Hastie, T., R. Mazumder, J. D. Lee, and R. Zadeh (2015). Matrix completion and low-rank svd via fast alternating least squares. *The Journal of Machine Learning Research* 16(1), 3367–3402.
- Hu, Y., M. Li, X. Liu, and C. Meng (2024). Sampling-based methods for multi-block optimization problems over transport polytopes. *Mathematics of Computation*.
- Jain, P., P. Netrapalli, and S. Sanghavi (2013). Low-rank matrix completion using alternating minimization. In *Proceedings of the forty-fifth annual ACM symposium on Theory of computing*, pp. 665–674.
- Klema, V. and A. Laub (1980). The singular value decomposition: Its computation and some applications. *IEEE Transactions on automatic control* 25, 164–176.
- LeBlanc, P. M., D. Banks, L. Fu, M. Li, Z. Tang, and Q. Wu (2024). Recommender systems: A review. *Journal of the American Statistical Association* 119(545), 773–785.
- Lee, D. and H. S. Seung (2000). Algorithms for non-negative matrix factorization. *Advances in neural information processing systems* 13.
- Lee, K. and D. Stöger (2023). Randomly initialized alternating least squares: Fast convergence for matrix sensing. *SIAM Journal on Mathematics of Data Science* 5(3), 774–799.
- Li, M., J. Yu, T. Li, and C. Meng (2024). Core-elements for large-scale least squares estimation. *Statistics and Computing* 34(6), 1–16.
- Li, M., J. Yu, H. Xu, and C. Meng (2023). Efficient approximation of Gromov-Wasserstein distance using importance sparsification. *Journal of Computational and Graphical Statistics* 32(4), 1512–1523.
- Li, M., J. Zhang, and C. Meng (2024). Nonparametric additive models for billion observations. *Journal of Computational and Graphical Statistics* 33(4), 1397–1412.
- Li, T. and C. Meng (2020). Modern subsampling methods for large-scale least squares regression. *International Journal of Cyber-Physical Systems (IJCPS)* 2(2), 1–28.
- Lü, L., M. Medo, C. H. Yeung, Y.-C. Zhang, Z.-K. Zhang, and T. Zhou (2012). Recommender systems. *Physics reports* 519, 1–49.
- Ma, P., Y. Chen, X. Zhang, X. Xing, J. Ma, and M. W. Mahoney (2022). Asymptotic analysis of sampling estimators for randomized numerical linear algebra algorithms. *Journal of Machine Learning Research* 23(177), 1–45.

- Ma, P., J. Z. Huang, and N. Zhang (2015). Efficient computation of smoothing splines via adaptive basis sampling. *Biometrika* 102(3), 631–645.
- Ma, P., M. Mahoney, and B. Yu (2014). A statistical perspective on algorithmic leveraging. In *Proceedings of the 31st International Conference on Machine Learning*, pp. 91–99. PMLR.
- Ma, P. and X. Sun (2015). Leveraging for big data regression. *Wiley Interdisciplinary Reviews: Computational Statistics* 7, 70–76.
- Martinez, C. (2004). Partial quicksort. In *Proc. 6th ACM-SIAM Workshop on Algorithm Engineering and Experiments*, pp. 224–228.
- Meng, C., Y. Wang, X. Zhang, A. Mandal, W. Zhong, and P. Ma (2017). Effective statistical methods for big data analytics. In *Handbook of Research on Applied Cybernetics and Systems Science*, pp. 280–299. IGI Global.
- Meng, C., R. Xie, A. Mandal, X. Zhang, W. Zhong, and P. Ma (2021). Lowcon: A design-based subsampling approach in a misspecified linear model. *Journal of Computational and Graphical Statistics* 30, 694–708.
- Meng, C., J. Yu, Y. Chen, W. Zhong, and P. Ma (2022). Smoothing splines approximation using hilbert curve basis selection. *Journal of Computational and Graphical Statistics* 31(3), 802–812.
- Meng, C., X. Zhang, J. Zhang, W. Zhong, and P. Ma (2020). More efficient approximation of smoothing splines via space-filling basis selection. *Biometrika* 107(3), 723–735.
- Netflix (2006). Netflix prize dataset. <http://www.netflixprize.com>. Available at <https://www.kaggle.com/datasets/netflix-inc/netflix-prize-data>.
- Pan, R., Y. Zhou, B. Cao, N. N. Liu, R. Lukose, M. Scholz, and Q. Yang (2008). One-class collaborative filtering. In *2008 Eighth IEEE international conference on data mining*, pp. 502–511.
- Pilászy, I., D. Zibriczky, and D. Tikk (2010). Fast ALS-based matrix factorization for explicit and implicit feedback datasets. In *Proceedings of the fourth ACM conference on Recommender systems*, pp. 71–78.
- Shimizu, A., X. Cheng, C. Musco, and J. Weare (2023). Improved active learning via dependent leverage score sampling. *arXiv preprint arXiv:2310.04966*.
- Sun, R. and Z.-Q. Luo (2016). Guaranteed matrix completion via non-convex factorization. *IEEE Transactions on Information Theory* 62(11), 6535–6579.

- Takács, G. and D. Tikk (2012). Alternating least squares for personalized ranking. In *Proceedings of the sixth ACM conference on Recommender systems*, pp. 83–90.
- Tikhonov, A. N. (1977). Solutions of ill-posed problems. *VH Winston and Sons*.
- Ting, D. and E. Brochu (2018). Optimal subsampling with influence functions. *Advances in neural information processing systems* 31.
- Truong Hoang, V. (2020). Barkvn-50. Mendeley Data, V1, doi:10.17632/gbt4tdmtn.1. Available at <https://www.kaggle.com/datasets/saurabhshahane/barkvn50/data>.
- Wang, H. and Y. Ma (2021). Optimal subsampling for quantile regression in big data. *Biometrika* 108(1), 99–112.
- Wang, H., M. Yang, and J. Stufken (2019). Information-based optimal subdata selection for big data linear regression. *Journal of the American Statistical Association* 114, 393–405.
- Wang, H., R. Zhu, and P. Ma (2018). Optimal subsampling for large sample logistic regression. *Journal of the American Statistical Association* 113(522), 829–844.
- Wang, L., J. Elmstedt, W. K. Wong, and H. Xu (2021). Orthogonal subsampling for big data linear regression. *The Annals of Applied Statistics* 15(3), 1273–1290.
- Winlaw, M., M. B. Hynes, A. Caterini, and H. De Sterck (2015). Algorithmic acceleration of parallel als for collaborative filtering: Speeding up distributed big data recommendation in spark. In *2015 IEEE 21st International Conference on Parallel and Distributed Systems (ICPADS)*, pp. 682–691. IEEE.
- Wu, X., Y. Huo, H. Ren, and C. Zou (2024). Optimal subsampling via predictive inference. *Journal of the American Statistical Association* 119(548), 2844–2856.
- Xie, R., S. Bai, and P. Ma (2023). Optimal sampling designs for multidimensional streaming time series with application to power grid sensor data. *The Annals of Applied Statistics* 17(4), 3195–3215.
- Xie, R., Z. Wang, S. Bai, P. Ma, and W. Zhong (2019). Online decentralized leverage score sampling for streaming multidimensional time series. In *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*, pp. 2301–2311. PMLR.
- Yi, S.-Y. and Y.-D. Zhou (2023). Model-free global likelihood subsampling for massive data. *Statistics and Computing* 33(1), 9.
- Yu, J., M. Ai, and Z. Ye (2024). A review on design inspired subsampling for big data. *Statistical Papers* 65(2), 467–510.

- Yu, J., J. Liu, and H. Wang (2023). Information-based optimal subdata selection for non-linear models. *Statistical Papers* 64(4), 1069–1093.
- Yu, J., H. Wang, and M. Ai (2024). A subsampling strategy for aic-based model averaging with generalized linear models. *Technometrics*, 1–11.
- Yu, J., H. Wang, M. Ai, and H. Zhang (2022). Optimal distributed subsampling for maximum quasi-likelihood estimators with massive data. *Journal of the American Statistical Association* 117(537), 265–276.
- Zhang, J., C. Meng, J. Yu, M. Zhang, W. Zhong, and P. Ma (2023). An optimal transport approach for selecting a representative subsample with application in efficient kernel density estimation. *Journal of Computational and Graphical Statistics* 32, 329–339.
- Zhang, M., Y. Zhou, Z. Zhou, and A. Zhang (2023). Model-free subsampling method based on uniform designs. *IEEE Transactions on Knowledge and Data Engineering*.
- Zhang, Y., L. Wang, X. Zhang, and H. Wang (2024). Independence-encouraging subsampling for nonparametric additive models. *Journal of Computational and Graphical Statistics* 33(4), 1424–1433.
- Zhao, F. (2024). *Distributed Singular Value Decomposition Through Least Squares*. Ph. D. thesis, Massachusetts Institute of Technology.
- Zhong, W., Y. Liu, and P. Zeng (2023). A model-free variable screening method based on leverage score. *Journal of the American Statistical Association* 118(541), 135–146.
- Zhou, Y., D. Wilkinson, R. Schreiber, and R. Pan (2008). Large-scale parallel collaborative filtering for the Netflix prize. In *Proceedings of the fourth AAIM Conference*, pp. 337–348.