Enhancing Group Recommendation using Soft Impute Singular Value Decomposition

Mubaraka Sani Ibrahim

Department of Computer Science,
African University of Science and Technology,
Abuja, Nigeria
msani@aust.edu.ng

Isah Charles Saidu

Department of Computer Science,
Baze University,
Abuja, Nigeria
charles.isah@bazeuniversity.edu.ng

Lehel Csató

Faculty of Mathematics and Computer Science,
Babes-Bolyai University,
Cluj-Napoca, Romania
lehel.csato@ubbcluj.ro

Abstract: The growing popularity of group activities increased the need to develop methods for providing recommendations to a group of users based on the collective preferences of the group members. Several group recommender systems have been proposed, but these methods often struggle due to sparsity and high-dimensionality of the available data, common in many real-world applications. In this paper, we propose a group recommender system called Group Soft-Impute SVD, which leverages soft-impute singular value decomposition to enhance group recommendations. This approach addresses the challenge of sparse high-dimensional data using low-rank matrix completion. We compared the performance of Group Soft-Impute SVD with Group MF based approaches and found that our method outperforms the baselines in recall for small user groups while achieving comparable results across all group sizes when tasked on Goodbooks, Movielens, and Synthetic datasets. Furthermore, our method recovers lower matrix ranks than the baselines, demonstrating its effectiveness in handling high-dimensional data.

Keywords: collaborative filtering; matrix completion; matrix factorisation; nuclear norm; rank; recommendation system; sparsity.

1 Introduction

Recommendation system is an algorithm that provides personalized recommendations to users based on preferences, past behaviour, and similarities to other users.

In general, there are two main types of recommendation systems: content-based filtering and collaborative filtering (CF). Content-based filtering (Lops et al., 2011) uses features to recommend items based on their similarity to items that the user has liked in the past. For instance, when recommending books, various features of the book such as its publication year, genre, average rating, plot and characters are taken into account to create a recommendation list tailored to a user.
On the other hand, collaborative filtering (Cacheda et al., 2011) suggests items to users by analyzing the preferences of other users with similar interests. Collaborative filtering can be classified as memory-based CF and model-based CF:

1 Memory-based CF aims to predict a target user's preferences for an item based on the preferences of similar users or a set of items (Sarwar et al., 2001). Typically, similarity measures are used to identify a

neighborhood of users (or items) that are similar to the target user (or item). Subsequently, the target user's interest in an item is determined based on the ratings of these neighboring users.

2 Model-based CF involves developing predictive machine learning models to forecast a user's rating for items they have not yet rated (Al-Mani et al., 2022). The model uses observed values in the user-item matrix as the training dataset and generates predictions for the missing values using the trained model. Model-based approaches include factorization methods, such as singular value decomposition (SVD) (Rodpysh et al., 2023), Bayesian networks (García et al., 2007) and clustering algorithms (Xue et al., 2017), such as k-means clustering.

A group recommender system generates personalized recommendations for a collection of users by aggregating individual preferences to formulate a suggestion that meets the needs of the group as a whole (Jameson and Smyth, 2007). In contrast to traditional recommender systems, which focus on individual preferences, group recommenders must address the collective preferences, interactions, and potential conflicts within the group. These systems take into account various factors, including the diversity of preferences, the need for fairness in balancing different user inputs, and the dynamics within the group (Li et al., 2024), such as majority rule, or consensus-based decision-making strategies.

In recent times, the popularity of group activities has increased the need to develop methods to cater to a group of users collectively, given their preferences. For instance, visiting a restaurant with friends, watching a movie with family, or trending content on social media websites are examples of activities well suited for groups of people.

There are several reasons for conducting group recommendation studies. First, most people engage in social activities with a group of people who are close acquaintances because humans are social by nature (Masthoff, 2011). Secondly, depending on technique, group recommendation may be more time efficient than personalized recommendation because estimating the individual preferences of users requires more computation time (Ntoutsi et al., 2012).

In recommender systems, users generally express their preferences for items by providing ratings. However, not all items receive ratings from users, resulting in a partially observed high-dimensional user-rating matrix with missing entries corresponding to unrated items. To address this, standard recommender systems aim to predict and impute these missing values, a process referred to as matrix completion (Candès and Tao, 2009).

In the context of group recommendation, we not only deal with a partially observed user-item matrix but also aim to generate recommendations at the group level, despite the sparse nature of the user-item interactions. This introduces an additional layer of complexity to the standard matrix completion problem, which is well-known to be NP-hard.

Two techniques commonly used to address group recommendation problems (Felfernig et al., 2018): aggregated predictions and aggregated models. The first approach involves generating preferences for each group member individually and then combining them to produce a group recommendation. In contrast, the second approach combines the preferences of individual group members into an extended dataset, which is subsequently utilised to generate recommendations for the entire group. Both methods rely on the chosen aggregation strategy.

In this study, we utilise the soft-impute algorithm (Mazumder et al., 2010), an efficient convex optimization technique for large-scale matrix completion. This approach operates under the assumption that group-level information is implicitly captured through the low-rank approximation of the partially observed user-item matrix. In our aggregated model-based approach, we leverage the extended dataset to ensure that items with more frequent ratings, as well as those exhibiting low variability in ratings, have a greater influence on the reconstruction process.

The remainder of the paper is structured as follows, Section 2 reviews the related work and discusses group recommendation sec. 2.1.1; matrix completion sec. 2.2, and the soft-impute algorithm sec. 2.3. Section 3.1 explains the proposed methodology. Section 4, presents the comparative evaluation results. Finally, Section 5 provides a summary of this study, including conclusions and future research directions.

2 Related Works

In this section, we provide a background to our work, and review the relevant works. Additionally, we present key definitions and concepts. We then provide a brief introduction to the soft-impute algorithm, which inspired our proposed method.

Several approaches in literature have been proposed to address the high-dimensional and sparse nature of recommender system data. One approach integrates dimensionality reduction techniques, using both supervised and unsupervised learning to efficiently address the curse of dimensionality problem, detect user groups, and improve prediction quality (Ait Hammou et al., 2019).

In another research, the authors propose a generalized nonparametric additive model to analyse high-dimensional group testing data to efficiently identify

defective items, enabling efficient identification of defective items. The authors utilise the EM algorithm and stochastic gradient descent algorithm to enhance computational efficiency of the group testing procedure (Zuo et al., 2024). Another author Li et al. (2024) utilised a stack ensemble learning framework for processing high-dimensional group structure data. This work combines the strength of multiple algorithms and group structure regularization to tackle variable selection in high-dimensional data effectively.

Rodpysh et al. (2023) addressed these issues by initially creating matrices based on item-user properties, which were then combined to form the similarity singular value decomposition matrix (SSVD) (Rodpysh et al., 2023). The authors then developed a context matrix using the contextual weighted property clustering similarity criterion applied to contextual data. The context matrix is then merged with the SSVD matrix to improve recommendation performance and reduce the challenge of cold start and sparse data. Li et al. (2009) proposed a cross-domain collaborative filtering method to transfer user-item rating patterns from an auxiliary rating matrix to another domain, alleviating the sparsity of the target domain's rating matrix. The aggregation for these approaches can be achieved using various techniques such as average without misery (Masthoff, 2004), average (McCarthy, 2002; Seo et al., 2021), plurality voting (Seo et al., 2021), least misery (Ortega et al., 2016), and most pleasure strategies (Senot et al., 2010).

Several studies have focused on improving group recommendation and tackling challenges within the group recommendation context. Shi et al. (2015) proposes a latent group model approach to enhance the accuracy and efficiency of group recommendation. This method involves identifying clusters by applying k-means algorithms on a user-latent factor matrix. The resulting user-factor groups are then combined to create the group profile, which generates group preferences for items using matrix multiplication. Hu et al. (2013) focus on SVD-based group recommendation methods, which involve aggregating ratings from group members using various decision strategies. The authors further evaluate the effectiveness of SVD-based aggregation methods, including aggregation profiles and prediction techniques. Another approach (Ba et al., 2013) combines the clustering algorithm with SVD to enhance the collaborative filtering method, address data sparsity and cold start issues as well. Qin et al. (2020) applied the collaborative filtering method on interest subgroups by taking into account user preferences. They then combine recommendation list from all subgroups to form a final recommendation for a large user group. Another paper presents a graph clustering algorithm that leverages pairwise preference data to generate recommendation to user groups (Abolghasemi et al., 2024).

2.1 Notation

We provide a brief summary of notations used throughout the paper, the notations are similar to (Cai et al., 2010; Mazumder et al., 2010). Define a matrix $P_\Omega(X)$

$$P_\Omega(X) = \begin{cases} X_{i,j} & if\ (i,j) \in \Omega \\ 0 & if\ (i,j) \notin \Omega \end{cases}$$

as the projection of the matrix $X_{m\times n}$ onto the observed entries. Here, $\Omega \subset \{1,\ldots,m\} \times \{1,\ldots,n\}$ represents the set of indices corresponding to observed entries in X. For $(i,j) \in \Omega$, the observed rating is denoted by X_{ij}, while $X_{ij} = 0$ for unobserved entries. Additionally, from Mazumder et al. (2010) the complementary projection $P^{\perp}_{\Omega}(X)$ is the projection of the matrix $X_{m\times n}$ onto the unobserved entries such that, $P^{\perp}_{\Omega}(X) + P_\Omega(X) = X$, where, $\Omega^{\perp}_{ij} = 1 - \Omega ij$ is the complement of Ω.

2.1.1 Problem Statement: Group Recommendation

Let $\mathcal{U} = \{u_i\}_{i=1}^m$ denote the set of m users and $\mathcal{V} = \{v_j\}_{j=1}^n$ represent the set of n items. The interaction between users and items is captured by the partially observed user-item rating matrix $X_{i,j} \in \mathbb{R}^{m\times n}$, where X_{ij} corresponds to the rating provided by user u_i for item v_j.

Given a group of users G, such that $G \subset \mathcal{U}$, the task is to recommend a list of k items, denoted by $V_G = \{v_j\}_{j=1}^k$, where $V_G \subset \mathcal{V}$. These recommendations are expected to satisfy the preferences of the group of users G.

2.2 Matrix completion and low rank assumption

Matrix completion aims to fill in missing entries of a partially observed matrix using its observed entries so that the reconstructed matrix approximates the original complete matrix. Obviously, this is an ill-posed problem and usually requires additional assumptions to the problem.

To obtain a well-posed problem, it is necessary to make low-rank assumptions on the underlying matrix (Laurent, 2001). For instance, the Netflix data matrix, which contains user ratings for movies, is believed be low rank because user's preference for movies is influenced by a small number of latent factors such as movie genre and audience type. Low rank techniques are effective for solving problems of matrix completion by exploiting redundant information in interaction matrices and compressing it for more efficient computations. Low-rank models have shown their effectiveness in addressing

sparsity problems, imputing missing data, denoising noisy data, and performing feature extraction (Udell et al., 2016; Udell and Townsend, 2019).

Matrix factorisation techniques are a widely adopted class of algorithms. These methods learn the factors of two or more low-rank matrices by reformulating the problem as an optimization task. However, the rank constraints lead to non-convex objective functions, making the problem inherently NP-hard and computationally intractable. As a result, these problems are typically addressed using approximation techniques that involve various relaxations of the objective functions.

In the following, we present two widely used formulations and algorithms - soft-impute and MF - for addressing the matrix completion problem.

2.3 Matrix Completion using Soft-Impute Algorithm

A fundamental approach to the matrix completion problem is given in 1, where the rank constraint is relaxed through the nuclear norm regularization term $||\mathbf{Z}||_*$ (Fazel et al., 2004). This term, defined as the sum of the singular values of $\mathbf{Z}$, serves as a convex surrogate for the rank function, enabling a tractable optimization formulation.

$$\min_{\mathbf{Z}} \quad ||P_\Omega(\mathbf{X}) - P_\Omega(\mathbf{Z})||_F^2 + \lambda||\mathbf{Z}||_* \tag{1}$$

where:

- $\mathbf{X} \in \mathbb{R}^{m\times n}$ is the partially observed data matrix;
- $\mathbf{Z} \in \mathbb{R}^{m\times n}$ is the completed matrix to be optimized;
- $P_\Omega(\cdot)$ is a projection operator that restricts the comparison to the observed entries in Ω;
- $||\cdot||_F$ denotes the Frobenius norm;
- $||\mathbf{Z}||_*$ represents the nuclear norm, defined as the sum of the singular values of $\mathbf{Z}$;
- $\lambda > 0$ is a regularization parameter that controls the trade-off between matrix completion accuracy and low-rank structure.

An iterative approach, known as soft-Impute algorithm and based on the SVD of $\mathbf{Z}$, was proposed by Mazumder et al. (2010). The soft-impute algorithm operates similarly to an expectation-maximization (EM) procedure, where missing values are iteratively updated using the current estimate. The optimization problem is then solved on the completed matrix by applying a soft-thresholded SVD, as described in the following section.

The soft-thresholded SVD is governed by the thresholding operator, defined as follows:

$$\widetilde{\mathbf{X}} = \mathbf{U}D_\lambda(\mathbf{\Sigma})\mathbf{V}^T \tag{2}$$

where

$$D_\lambda(\mathbf{\Sigma}) = \text{diag}[(\sigma_i - \lambda)_+],$$

$$t_+ = \max(0, t).$$

The operator $D_\lambda(\cdot)$ applies soft-thresholding to the singular values σ_i of $\mathbf{X}$, shrinking them toward zero. Singular values smaller than λ are set to zero, while those greater than λ are reduced by λ.

By utilizing the soft-thresholding operator in combination with nuclear norm minimization, the soft-impute algorithm iteratively computes a low-rank approximation of $\mathbf{X}$. In addition, Mazumder et al. (2010) demonstrated that decreasing the regularization parameter λ leads to an increase in the rank of the resulting matrix. Consequently, they suggested initializing each iteration with the solution from the previous step, denoted as $Z_{\lambda_{i-1}}$, where $\lambda_{i-1} > \lambda_i$. The initial value λ_1 is set as the maximum singular value of the observed matrix $P_\Omega(X)$, i.e.,

$$\lambda_1 = \sigma_{\max}(P_\Omega(X)),$$

providing a warm start for the iterative process.

2.4 Matrix Factorisation and Alternating Least Square

MF method addresses the matrix completion problem by learning two distinct low-rank matrices of rank r. This rank-constrained formulation is obtained by solving the following optimization problem:

$$\begin{aligned}\min_{\mathcal{U},\mathcal{V}} \quad & ||P_\Omega(\mathbf{X}) - P_\Omega(\mathcal{U}\mathcal{V}^T)||_F^2 \\ & + \lambda(||\mathcal{U}||_F^2 + ||\mathcal{V}||_F^2)\end{aligned} \tag{3}$$

where:

- $\mathcal{U}$ and $\mathcal{V}$ are latent factor matrices of dimensions $m \times r$ and $n \times r$, respectively, where m represents the number of users and n represents the number of items.

Although the MF formulation is not convex in $\mathcal{U}$ and $\mathcal{V}$ simultaneously, it is *bi-convex*. Specifically, for a fixed $\mathcal{U}$, the objective function is convex in $\mathcal{V}$, and for a fixed $\mathcal{V}$, the function is convex in $\mathcal{U}$. As a result, an alternating least squares technique is commonly employed to minimize (3). Consider the case where $\mathcal{U}$ is fixed, and we seek to optimize $\mathcal{V}$. In this scenario, the problem decomposes into n separate ridge regression problems, where each column $\mathbf{x}_j$ of $P_\Omega(\mathbf{X})$ serves as the response variable, and the r columns of $\mathcal{U}$ act as predictors. Since some entries in $\mathbf{x}_j$ are missing, these missing observations are ignored, and the corresponding rows of $\mathcal{U}$ are removed for the j-th regression. Consequently, each regression problem operates on a different subset of data, despite all regressions being derived from

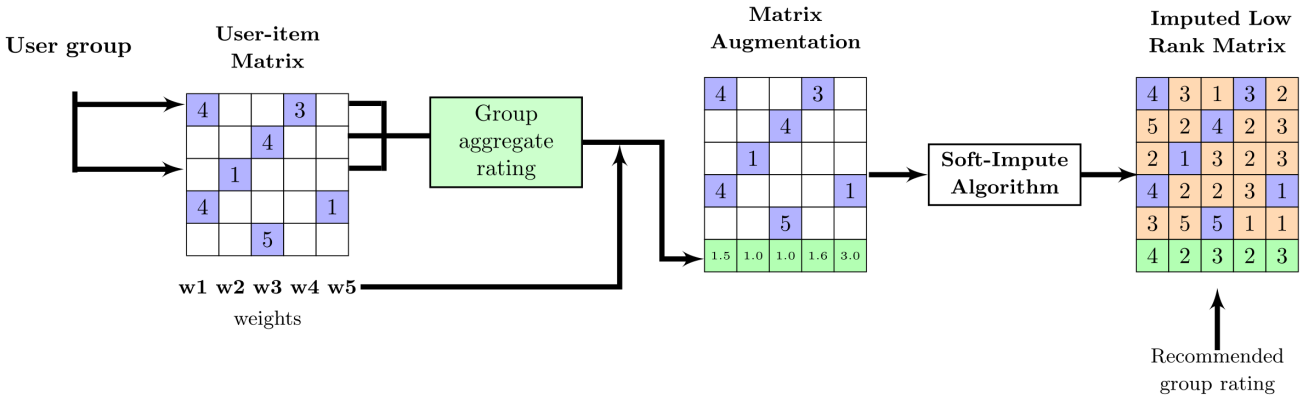

Figure 1: Proposed Methodology

$\mathcal{U}$. This formulation ensures that the optimization procedure remains computationally feasible and scalable across large datasets. However, unlike the soft-impute algorithm, which automatically promotes a low-rank solution through nuclear norm regularization, the rank constraint in the MF approach is imposed through the fixed dimensions of the factorized matrices $\mathcal{U}$ and $\mathcal{V}$, which limits the model's flexibility in adapting to the intrinsic rank structure of the data.

3 Methodology

3.1 Proposed Approach

In this section, the proposed framework for the group recommender system that forms the basis of our algorithm is illustrated in Fig. 1 and described below. Additionally, the details of our algorithm, Group soft-impute SVD (GSI-SVD) are presented in Algorithm 1.

The first step begins with a user-item matrix, which represents the interactions between users and items. This matrix is a partially observed matrix due to sparse user feedback, where blank squares represent unobserved entries, and blue squares represent observed entries as shown in Fig. 1.

To incorporate group-based information into the user-item matrix for group recommendation, we compute a group aggregate rating that reflects the combined preferences of the users. This is achieved using a weighted aggregation approach, where each user's contribution is adjusted by a corresponding weight. To compute the weighted average ratings for group G, only items rated by group members are considered. The average user rating for each item j in the aggregated group entry is given by (4), while the weights for each item in the aggregated group entry are given by (5), where $\sigma_{G,i}$ is the standard deviation of item i in the group G. The weights are based on the proportion of group members who rated the items and the similarity between items rated by the group members, following a similar approach to Ortega et al. (2016). Next, the soft-impute algorithm (Mazumder et al., 2010) is iteratively applied to the augmented matrix to estimate unknown group preferences by leveraging patterns in the observed data. After the soft-impute algorithm converges, the result is a low-rank matrix with the missing values predicted. The final step produces a recommended group rating for each item.

3.2 Computational Complexity and Convergence

The proposed GSI-SVD algorithm achieves efficient computation by combining iterative singular value thresholding with a dynamic low-rank matrix reconstruction strategy that retains only the non-zero singular components at each step.

At each iteration, the algorithm performs soft-thresholding on the singular values. For an input matrix $X \in \mathbb{R}^{n\times m}$, the SVD is applied to the matrix

$$Y = P_\Omega(X) + P_{\Omega^\perp}(Z_{\text{old}}),$$

which has a worst-case computational cost of $\mathcal{O}(nm^2)$. However, after thresholding, only a small number

Algorithm 1 Group Soft-Impute SVD

Require: Input matrix $X \in \mathbb{R}^{n \times m}$, user group $i \in G$, number of lambda values k, $\lambda_{max} = \sigma_{max}\left(P_\Omega\left(X\right)\right)$, $\lambda_{min} = 1$, convergence threshold $\epsilon > 0$

Randomly initialize Z^{old}

$\lambda_{grid} = [\lambda_{max}, ..., \lambda_{min}]$

Calculate the group average ratings for observed entries:

$$r_{G,j} = \frac{1}{|G|} \sum_{i \in G} x_{i,j}, \quad \text{where } x_{i,j} > 0. \tag{4}$$

Compute weights for each item j as:

$$w_{G,i} = \frac{\#\{u \in G | u_{u,i} \neq 0\}}{|G|} \frac{1}{1 + \sigma_{G,i}} \tag{5}$$

Merge weighted group rating $r_{G,j}$ as an additional row in X:

$$X_{\text{new}} = \begin{bmatrix} X \\ r_{G,j} w_{G,j} \end{bmatrix}$$

for each $\lambda_i \in \lambda_{grid}$ **do**

  Compute $U, D, V^T \leftarrow \text{SVD}\left(P_\Omega(X_{\text{new}}) + P_{\frac{1}{\Omega}}(Z^{\text{old}})\right)$

  Compute $D_{\lambda_i} \leftarrow \max\left(D - \lambda_i, 0\right)$

  $X^{new} \leftarrow U D_{\lambda_i} V^T$

  if $\frac{\|Z^{new} - Z^{old}\|_F^2}{\|Z^{old}\|_F^2} < \epsilon$ **then**

    break

  end if

  $Z^{old} \leftarrow Z^{new}$

end for

$Z_{\lambda_\kappa} \leftarrow Z^{old}$

$r_{G_j} \leftarrow Z_{\lambda_\kappa}[-1]$

Output: Final reconstructed matrix Z_{λ_κ}, Predicted group rating r_{G_j}

of singular components $r \ll \min(n, m)$ are typically retained. Reconstructing the matrix using only these r components reduces the per-iteration cost to:

$$\mathcal{O}(nmr).$$

Additionally, applying the mask over the observed entries Ω introduces a further cost of $\mathcal{O}(|\Omega|)$. Therefore, the total computational cost per iteration becomes:

$$\mathcal{O}(|\Omega| + nmr).$$

Following the convergence analysis in (Mazumder et al., 2010), the algorithm guarantees that the relative error between consecutive iterates,

$$\frac{\left\|Z^{(k+1)} - Z^{(k)}\right\|_F}{\left\|Z^{(k)}\right\|_F}, \tag{6}$$

tends to zero as the number of iterations k increases. Under mild assumptions, this results in geometric convergence (See Appendix for the convergence proof), with the number of iterations required to reach a desired accuracy ϵ scaling as:

$$k = \mathcal{O}(\log(\epsilon^{-1})).$$

This makes the GSI-SVD algorithm well-suited for large-scale, sparse matrix completion in group recommendation scenarios.

3.3 Baseline

We evaluate our proposed method by comparing it against two established techniques: the weighted before factorisation method and the after factorisation method (Ortega et al., 2016). Unlike the soft-impute algorithm, these group recommendation methods are derived MF approaches.

The MF-based group recommendation framework is extended in two ways:

1. Ratings for a selected group of users are first aggregated using an aggregation function **h** to construct a pseudo-group user u_g, before applying factorization to the rating matrix.

2. Ratings are computed after aggregating predictions derived from the latent factor matrices.

These two techniques serve as strong baselines for evaluating group recommendation systems within collaborative filtering frameworks. By effectively capturing and integrating user preferences, they provide a robust benchmark for assessing the performance of the proposed approach.

In the following sections, we present a detailed description of these methods.

After Factorisation Method

The after factorisation (AF) approach proposed in the literature (Ortega et al., 2016) recommends items to a group of users by leveraging the latent factors learned through MF. Initially, MF model is trained to obtain latent factor representations for both users and items, effectively capturing their underlying preferences and characteristics.

To model group preferences, the individual latent factors of the group members are aggregated into a group profile $\mathbf{u}_G$ and item profile $\mathbf{v}_G$, respectively, using a predefined aggregation function **h**.

Commonly used aggregation functions include the average, weighted average, minimum aggregation and maximum aggregation.

Weighted Before Factorization Method

The weighted before factorization (WBF) approach Ortega et al. (2016) focuses on computing a weighted aggregation of the individual preferences in the user rating matrix before applying MF. By emphasizing the importance of certain entries prior to factorization, this

method constructs a collective preference profile for the group while ensuring a degree of fairness among its members.

For a group G with a set of items, weights $w_{G,j}$ are defined as in equation 5 The group user profile $\mathbf{u}_G$ is computed by solving the objective function described in equation 3.

4 Experiments

4.1 Datasets

The experiments were performed on the well-known real-world Goodbooks dataset (Zajac, 2017), Movielens dataset Harper and Konstan (2015) and a synthetic dataset.

Table 1 Goodbooks dataset

Dataset	Detail
#number of users	53K
#number of books	10K
#number of ratings	6 million

The Goodbooks dataset (Table 1) contains ratings on a scale from 1 to 5, provided by various users for different books. For computational purposes, we select a sample of 2000 users and 200 books from the dataset. The data has a sparsity of 90%, indicating the proportion of unobserved entries. The Movielens dataset contains 100,000 ratings

Table 2 Movielens dataset

Dataset	Detail
#number of users	943
#number of movies	1682
#number of ratings	100K

provided by 943 users for 1,682 items. Each user has rated at least 20 movies. The ratings span a scale from 1 to 5 and have a sparsity of 86%. For this study, we selected a sample of 943 users and 500 items from the data set. For the matrix completion experiments, we first use the KNN imputer algorithm to estimate the missing entries. We then create a partially observed matrix by randomly selecting 75% of the observed ratings for training and masking the remaining 25% for testing.

For the synthetic data, we generated a synthetic user-item rating dataset. The dataset consisted of 2,000 users and 200 items, with ratings drawn from a bounded normal distribution centred around a mean rating of 3.5 and

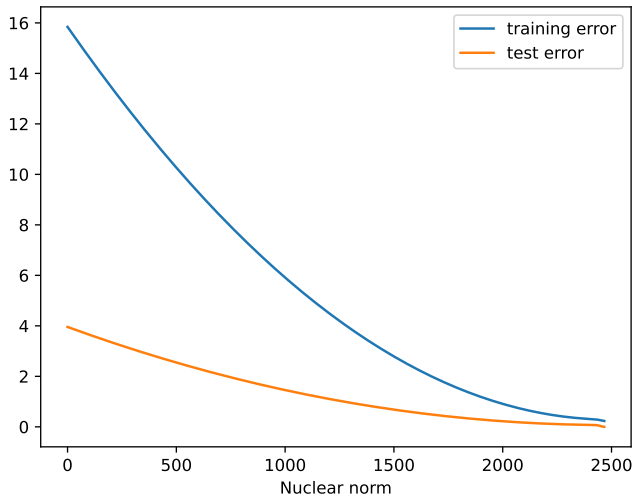

Figure 2: Test error-Training error vs Nuclear norm.

a standard deviation of 0.65. The ratings were clipped to fall within the appropriate range, ensuring alignment with typical rating scales used in real-world recommender systems. To introduce sparsity, a binary mask was applied, with 25% of entries observed and 75% missing entries, simulating real-world missing data scenarios.

4.2 Experimental Results

In the first phase of the experiment, we employ the soft-impute algorithm to demonstrate the effectiveness of the nuclear norm as a convex surrogate for recovering low-rank matrices.

All matrix completion computations are implemented using the PyTorch library, which provides efficient support for tensor operations and GPU acceleration.

Soft-impute iteratively fills in the missing entries in the partially observed matrix X using the current iterate Y_t, evaluated on a grid of 10 values of the regularization parameter λ. Convergence is achieved when the relative change in Frobenius norm between successive iterates falls below 10^{-5}.

To reduce computational complexity and align with the efficient *soft-impute* framework proposed by Mazumder et al. (2010), we modify the standard SVD-based reconstruction in our GSI-SVD implementation. Rather than reconstructing the entire matrix using all singular vectors and a full-size diagonal matrix, we retain only the singular values $\sigma_i' > 0$ that remain positive after soft-thresholding. The low-rank matrix is then reconstructed using only the reduced set of singular vectors:

$$Z_{\text{new}} = U_r \cdot \Sigma_r \cdot V_r^\top,$$

where $U_r \in \mathbb{R}^{n\times r}$, $\Sigma_r \in \mathbb{R}^{r\times r}$, and $V_r \in \mathbb{R}^{m\times r}$ correspond to the singular vectors and values associated with the retained r non-zero components.

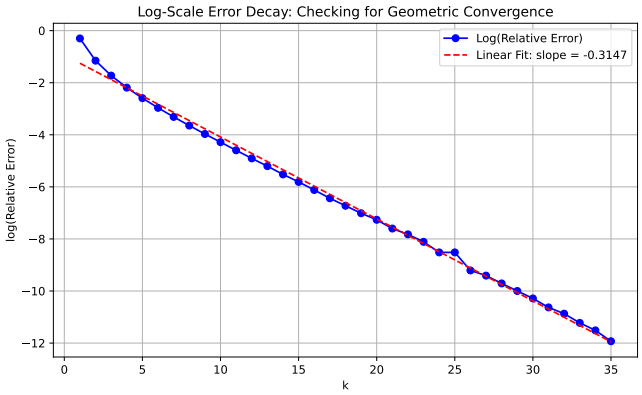

Figure 3: Log-scale plot of the relative error across iterations.

For performance evaluation,

- We use the mean squared error(MSE) for training error and test error defined as:

$$\text{Training error} = \frac{P_\Omega(X-Y)^2}{|\Omega|}$$
$$\text{Test error} = \frac{P_{\frac{1}{\Omega}}(X-Y)^2}{|\frac{1}{\Omega}|}$$

Figure 2 displays plots of the training and test error for the GSI-SVD algorithm as a function of nuclear norm over the grid of $\lambda \in [\lambda_{max} - |n|,0]$, where λ_{max} corresponds to the largest singular value obtained from the SVD of $P_\Omega(X)$ and $|n|$ is the cardinality of the items in X.

The results in Figure 2 show the performance of the algorithm for datasets with 75% missing entries. The results also indicate that soft-impute effectively recovers the matrix with high nuclear norm.

To evaluate the performance of the group recommendation model, we compute precision, recall and F1 score that compare the aggregated individual-rating predictions $r(G,i)$ with the aggregated group predictions $\hat{r}(G,i)$, following a similar approach to prior work Felfernig et al. (2018).

$$r(G,i) = \frac{\sum_{u\in G} x(i,j)}{|G|} \tag{7}$$

$$\hat{r}(G,i) = \frac{\sum_{u\in G} \hat{x}(i,j)}{|G|} \tag{8}$$

Precision (Ting, 2016) is the proportion of recommended items that are relevant, calculated as the ratio of relevant recommended items to the total number of recommended items.

$$\text{precision} = \frac{\text{TP}}{\text{TP}+\text{FP}} \tag{9}$$

Recall Ting (2016) is the ratio of the number of relevant recommended items to the total number of relevant items in the dataset.

$$\text{recall} = \frac{\text{TP}}{\text{TP}+\text{FN}} \tag{10}$$

where TP, FP, and FN denote true positives, false positives, and false negatives, respectively.

F1 score (Hand et al., 2021) is the harmonic mean of precision and recall given by:

$$\text{F1 score} = 2\cdot\frac{\text{precision}\cdot\text{recall}}{\text{precision}+\text{recall}} \tag{11}$$

These metrics are typically evaluated at a predefined recommendation list size k. The results show that the proposed framework is effective in capturing the preferences of the group members and providing accurate group-level recommendations.

To evaluate the proposed method (GSI-SVD), we compared it against the baselines (WBF and AF approach). The comparison was based on precision, recall and F1 score for the top K recommended items. Figure 4 present comparisons for different user groups of sizes 5, 10, 15, 20 and 25 against k number of item recommendation, where (k=20) for this experiment.

For the Goodbooks data, the experimental results show that the WBF and AF methods outperform the proposed method in terms of precision across all user groups. In general, AF and WBF demonstrate better precision, indicating their superior ability to recommend relevant books with fewer false positives. In terms of recall, GSI-SVD outperforms WBF for user group 5, but shows similar performance to baseline methods for group sizes 15 and 25, as shown in Figure 4a . AF maintains a perfect F1 score in all groups, indicating high precision and recall. In contrast, GSI-SVD exhibits the highest variation in F1 score, peaking at around ∼0.98 for group 15 before dropping to approximately ∼0.92 for group 20. For the Movielens dataset, AF and WBF outperform GSI-SVD in terms of recommendation accuracy for some user groups, although the differences are relatively small. GSI-SVD shows lower precision for smaller group sizes of 5 and 10, but its performance improves for larger group sizes. While GSI-SVD achieves a higher recall than WBF and AF for user group 5, the latter two methods outperform it for user groups 10 and 15. Additionally,

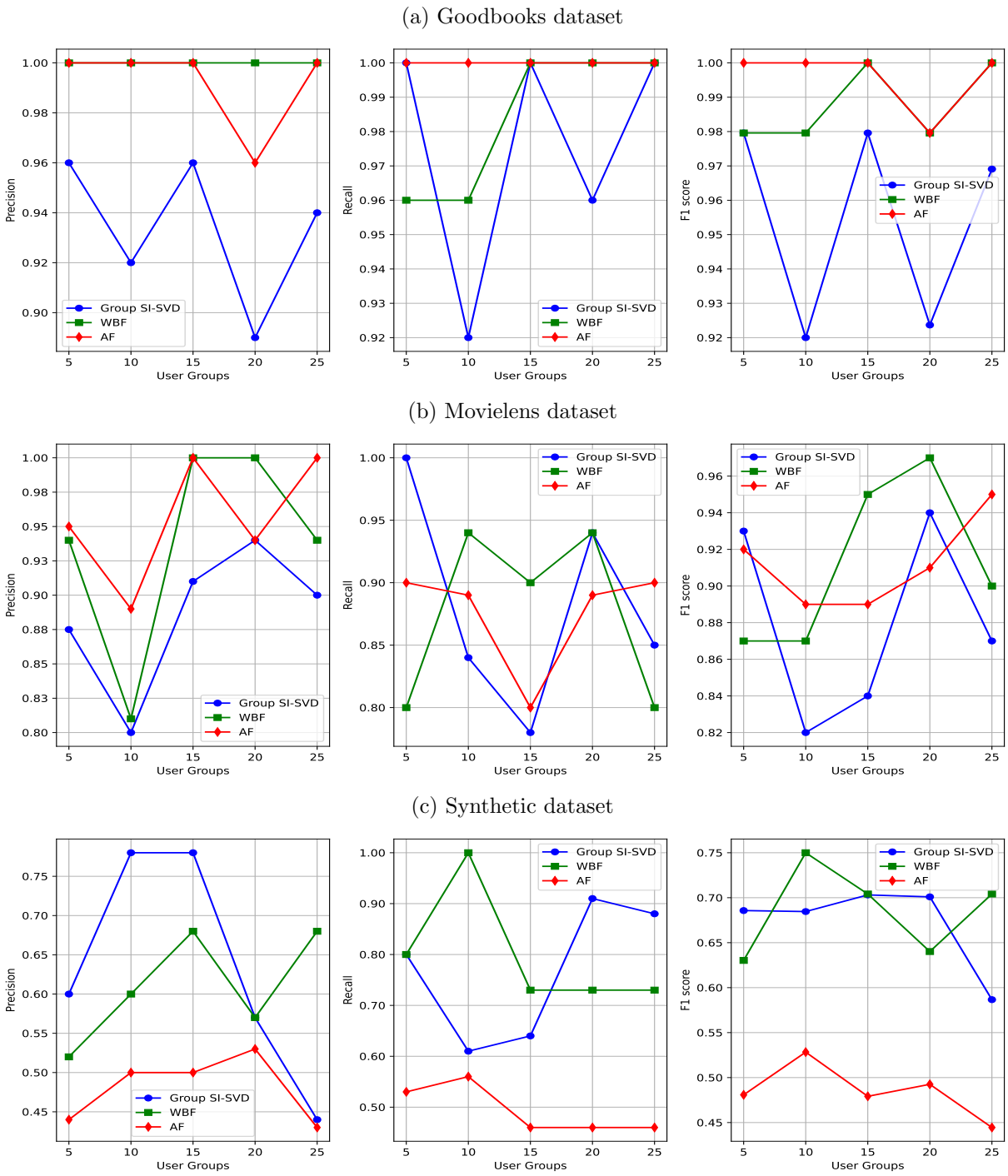

Figure 4: Performance comparison of GSI-SVD, AF, and WBF for different user group sizes across (a) Goodbooks, (b) Movielens, and (c) Synthetic datasets.

GSI-SVD has the highest F1 score for user group 5, but is outperformed by WBF and AF in the mid-sized groups as shown Figure 4b .

For the Synthetic data set, GSI-SVD maintains a consistently higher precision than the other methods broadly across different group sizes as shown in Figure 4c. Both GSI-SVD and WBF outperform AF, which shows the lowest precision across all groups. In terms of recall, GSI-SVD demonstrates improved performance as the group size increases. Conversely, AF consistently

Table 3 Recovered ranks for WBF, AF, and GSI-SVD at varying λ values across datasets.

λ	GoodBooks			Movielens			Synthetic		
	WBF	AF	GSI	WBF	AF	GSI	WBF	AF	GSI
0.001	199	199	200	498	497	491	44	44	200
0.01	200	199	176	498	498	270	199	199	182
0.1	197	196	174	496	496	255	198	196	182
1	198	197	166	497	498	237	198	198	175
10	198	199	72	499	498	82	198	199	111

shows low recall, suggesting it is missing many relevant recommendations. Both GSI-SVD and WBF demonstrate more competitive performance in terms of F1 score compared to AF. This suggests that GSI-SVD and WBF are the most effective models for achieving high recall, while AF is the least effective for this data set.

We also compare GSI-SVD, WBF, and AF in terms of rank recovery across different λ values. WBF and AF seem to recover approximately the full rank ~ 200 and ~ 497 for Movielens across all λ values whereas GSI-SVD shows a strong dependence on λ, with ranks decreasing significantly as λ increases as shown in Table 3.

To empirically verify the convergence behavior of the proposed method, we analyze the decay of the relative error 6 between successive iterates k, as shown in Figure 3. Specifically, we plot the logarithm of the relative error, against the iteration count. The observed near-linear trend with a negative slope on the log-scale plot indicates exponential decay of the error, thereby validating the geometric convergence behavior of the GSI SVD algorithm. Therefore, the method used in this paper significantly enhances the quality of recommendations in terms of precision and recall for small, mid size and large user groups, as shown in Figure 4. Furthermore, the incorporating weights to group recommendation significantly enhances the quality of recommendations in terms of precision and recall showing that the predictions are aligned with the group's collective preferences.

5 Conclusion

In this paper, a novel recommendation method for collaborative filtering that imputes missing entries in a matrix and offers recommendation to a given user group leveraging soft-impute algorithm. The results show that the soft-impute algorithm is effective in recovering the underlying low-rank structure of the partially observed matrix and providing group recommendation that satisfy the preferences of the small size, mid-size and large number of group members. The results indicate that the recommendation process improves both precision and recall across varying user group sizes (5, 10, 15 20 and 25) and different numbers of item recommendations. In future studies, we aim to investigate the performance of the proposed GSI-SVD algorithm for even larger group sizes, evaluating its effectiveness and scalability. Additionally, we plan to develop optimization techniques to address the computational challenges of large group recommendations while maintaining accuracy.

References

Ait Hammou, B., Ait Lahcen, A. and Mouline, S. (2019) 'A distributed group recommendation system based on extreme gradient boosting and big data technologies', *Applied Intelligence*, Vol. 49, No. 12, pp. 4128–4149.

Abolghasemi, R., Viedma, E.H., Engelstad, P.E., Djenouri, Y. and Yazidi, A. (2024) 'A graph neural approach for group recommendation system based on pairwise preferences', *Information Fusion*, Vol. 107, Article 102343.

Al-Mani, I.A., Al-Sabaawi, A.M.A., and Hussien, M. H. (2022) 'A Review Paper of Model Based Collaborative Filtering Techniques', *Proceedings of the 2022 International Conference on Data Science and Intelligent Computing (ICDSIC)*, pp.52–57.

Ba, Q., Li, X., and Bai, Z. (2013) 'Clustering collaborative filtering recommendation system based on SVD algorithm', *Proceedings of the 2013 IEEE 4th International Conference on Software Engineering and Service Science*, pp.963–967.

Cacheda, F., Carneiro, V., Fernández, D. and Formoso, V. (2011) 'Comparison of collaborative filtering algorithms: Limitations of current techniques and proposals for scalable, high-performance recommender systems', *ACM Transactions on the Web (TWEB)*, Vol. 5, No. 1, pp.1–33.

Cai, J.-F., Candès, E.J. and Shen, Z. (2010) 'A Singular Value Thresholding Algorithm for Matrix Completion', *SIAM Journal on Optimization*, Vol. 20 No. 4, pp. 1956–1982.

Candès, E.J. and Tao, T. (2009) 'The Power of Convex Relaxation: Near-Optimal Matrix Completion', *IEEE*

Transactions on Information Theory, Vol. 56, no. 5, pp. 2053-2080.

Fazel, M., Hindi, H.A. and Boyd, S.P. (2004) 'Rank minimization and applications in system theory', *Proceedings of the 2004 American Control Conference*, IEEE, Boston, MA, USA, Vol. 4, pp.3273–3278.

Felfernig, A., Boratto, L., Stettinger, M. and Tkalčič, M. (2018) 'Evaluating Group Recommender Systems', in *Group Recommender Systems: An Introduction*, SpringerBriefs in Electrical and Computer Engineering, Springer, Cham, pp. 59–71.

Felfernig, A., Müssler, T., Atas, M., Helic, D., Tran, T.N. and Samer, R. (2018) 'Algorithms for Group Recommendation', in *Group Recommender Systems: An Introduction*, SpringerBriefs in Electrical and Computer Engineering, Springer, Cham, pp.27–58.

García, P., Amandi, A., Schiaffino, S. and Campo, M. (2007) 'Evaluating Bayesian networks precision for detecting students' learning styles', *Computers & Education*, Vol. 49, No. 3, pp.794–808.

Hand, D.J., Christen, P. and Kirielle, N. (2021) 'F*: An interpretable transformation of the F-measure', *Machine Learning*, Vol. 110, pp. 451–456, DOI:10.1007/s10994-021-05964-1.

Harper, F.M. and Konstan, J.A. (2015) 'The MovieLens Datasets: History and Context', *ACM Transactions on Interactive Intelligent Systems (TiiS)*, Vol. 5, No. 4, pp. 1-19 , DOI:http://dx.doi.org/10.1145/2827872.

Hu, X., Meng, X., and Wang, L. (2011) 'SVD-based group recommendation approaches: an experimental study of Moviepilot', *Proceedings of the 2nd Challenge on Context-Aware Movie Recommendation*, Association for Computing Machinery, New York, NY, USA, pp.23–28.

Jameson, A. and Smyth, B. (2007) 'Recommendation to Groups', in Brusilovsky, P. et al (Eds.), *The Adaptive Web: Methods and Strategies of Web Personalization*, Springer, Berlin, Heidelberg, pp.596–627.

Laurent, M. (2001), 'Matrix Completion problems', in Floudas, C.A., Pardalos, P.M. (Eds.), *Encyclopedia of Optimization*, Springer, Boston, MA, pp. 1311-1319.

Li, B., Yang, Q., and Xue, X. (2009) 'Can movies and books collaborate? Cross-domain collaborative filtering for sparsity reduction', *Proceedings of the 21st International Joint Conference on Artificial Intelligence*, Pasadena, CA, pp.2052–2057.

Li, D., Pan, C., Zhao, J. and Luo, A. (2024) 'A penalized variable selection ensemble algorithm for high-dimensional group structured data', *PLoS ONE*, Vol. 19 No. 2, Article e0296748, DOI:10.1371/journal.pone.0296748.

Li, Y., Liu, K., Satapathy, R., Wang, S., and Cambria, E. (2024) 'Recent Developments in Recommender Systems: A Survey [Review Article]', *IEEE Computational Intelligence Magazine*, Vol. 19, No. 2, pp. 78-95.

Lops, P., de Gemmis, M. and Semeraro, G. (2011) 'Content-based Recommender Systems: State of the Art and Trends', in Ricci, F. et al (Eds.), *Recommender Systems Handbook*, Springer, Boston, MA, pp. 73–105.

Masthoff, J. (2004) 'Group Modeling: Selecting a Sequence of Television Items to Suit a Group of Viewers', *User Modeling and User-Adapted Interaction*, Vol. 14, No. 1, pp.37–85.

Masthoff, J. (2011) 'Group Recommender Systems: Combining Individual Models', in Ricci, F. et al (Eds.), *Recommender Systems Handbook*, Springer, Heidelberg, pp.677–702.

Mazumder, R., Hastie, T. and Tibshirani, R. (2010) 'Spectral Regularization Algorithms for Learning Large Incomplete Matrices', *Journal of Machine Learning Research*, Vol. 11, pp. 2287-2322.

McCarthy, J.F. (2002) 'Pocket RestaurantFinder: A Situated Recommender System for Groups', *Proceedings of the ACM Conference on Human Factors in Computer Systems (CHI)*. 20 April 2002, Minneapolis, US.

Ntoutsi, E., Stefanidis, K., Nørvåg, K. and Kriegel, H.P. (2012) 'Fast Group Recommendations by Applying User Clustering', *Conceptual Modeling (ER 2012)*, Springer, Berlin, Heidelberg, pp.126–140.

Ortega, F., Hernando, A., Bobadilla, J. and Kang, J.H. (2016) 'Recommending items to groups of users using Matrix Factorization-based Collaborative Filtering', *Information Sciences*, Vol. 345, pp.313–324.

Qin, D., Zhou, X., Chen, L., Huang, G., and Zhang, Y. (2020) 'Dynamic Connection-Based Social Group Recommendation', *IEEE Transactions on Knowledge and Data Engineering*, Vol. 32, No. 3, pp.453–467.

Rodpysh, K. V., Mirabedini, S. J., and Banirostam, T. (2023) 'Employing Singular Value Decomposition and Similarity Criteria for Alleviating Cold Start and Sparse Data in Context-Aware Recommender Systems', *Electronic Commerce Research*, Vol. 23, No. 2, pp.681–707, DOI:10.1007/s10660-021-09488-7.

Sarwar, B., Karypis, G., Konstan, J. and Riedl, J. (2001) 'Item-based collaborative filtering recommendation algorithm', *Proceedings of the 10th International Conference on World Wide Web*, ACM, New York, USA, pp.285–295.

Senot, C., Kostadinov, D., Bouzid, M., Picault, J., Aghasaryan, A. and Bernier, C. (2010) 'Analysis of Strategies for Building Group Profiles', in De Bra, et al (Eds.), *User Modeling, Adaptation, and Personalization*, Lecture Notes in Computer Science, Springer, Berlin, Heidelberg, pp.40–51.

Seo, Y.D., Kim, Y.G., Lee, E. and Kim, H. (2021) 'Group recommender system based on genre preference focusing on reducing the clustering cost', *Expert Systems with Applications*, Vol. 183, Article 115396.

Shi, J., Wu, B., and Lin, X. (2015) 'A Latent Group Model for Group Recommendation', *2015 IEEE International Conference on Mobile Services*, New York, NY, USA, 2015, pp. 233-238.

Ting, K.M. (2016) 'Precision and Recall ', in Sammut, C. and Webb, G.I. (Eds.), *Encyclopedia of Machine Learning and Data Mining*, Springer, Boston, MA, pp. 1, DOI: 10.1007/978-1-4899-7502-7_659-1.

Udell, M., Horn, C., Zadeh, R., and Boyd, S. (2016) 'Generalized Low Rank Models', *Foundations and Trends in Machine Learning*, Vol. 9, No. 1, pp.1–118, DOI:10.1561/2200000055.

Udell, M., and Townsend, A. (2019) 'Why Are Big Data Matrices Approximately Low Rank?', *SIAM Journal on Mathematics of Data Science*, Vol. 1, No. 1, pp.144–160, DOI:10.1137/18M1183480.

Xue, G.R., Lin, C., Yang, Q., Xi, W., Zeng, H.J., Yu, Y. and Chen, Z. (2005) 'Scalable collaborative filtering using cluster-based smoothing', *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, ACM, New York, NY, USA, pp. 114–121.

Zajac, Z. (2017) 'Goodbooks-10k: A New Dataset for Book Recommendations', *FastML*. http://fastml.com/goodbooks-10k.

Zuo, X., Ding, J., Zhang, J. and Xiong, W. (2024) 'Nonparametric Additive Regression for High-Dimensional Group Testing Data ', *Mathematics*, Vol. 12 No. 5, p. 686, DOI:10.3390/math12050686.

Appendix : Convergence

While the theoretical convergence of the original Soft-Impute algorithm is generally sublinear (Mazumder et al., 2010), modifications introduced in Group Soft-Impute SVD, combined with empirical observations, support the assumption of linear(geometric) convergence rate. The soft-thresholded SVD operator, which is fundamental to the algorithm, is known to be non-expansive in the Frobenius norm (Mazumder et al., 2010):

$$\|\mathrm{D}_\lambda(A) - \mathrm{D}_\lambda(B)\|_F \leq \|A - B\|_F.$$

However, when the iterates approach a stable low-rank solution manifold, soft-thresholded SVD can exhibit *contractive* behavior, satisfying:

$$\|Z^{(k+1)} - Z^*\|_F \leq \rho \|Z^{(k)} - Z^*\|_F, \quad \text{for some } 0 < \rho < 1,$$

where Z^* denotes the optimal low-rank solution. This contraction behaviour implies that the iterates converge exponentially toward the solution. Prior studies on low-rank matrix projection and proximal methods confirm that such contraction is expected when the iterates remain within a stable low-rank subspace, thus supporting the empirical linear convergence observed in our results.

Convergence Derivation

Assume that the iterates Z_k of the Group Soft-Impute SVD algorithm converge geometrically to the optimal solution Z_∞, that is,

$$\|Z_k - Z_\infty\|_F \leq \rho^k \|Z_0 - Z_\infty\|_F, \quad \text{for some } 0 < \rho < 1.$$

We aim to find the number of iterations k such that the reconstruction error is bounded by a desired accuracy $\epsilon > 0$:

$$\|Z_k - Z_\infty\|_F \leq \epsilon.$$

From the geometric decay:

$$\rho^k \|Z_0 - Z_\infty\|_F \leq \epsilon.$$

Dividing both sides and taking the logarithm:

$$k \log(\rho) \leq \log\left(\frac{\epsilon}{\|Z_0 - Z_\infty\|_F}\right),$$

which gives:

$$k \geq \frac{\log\left(\frac{\|Z_0 - Z_\infty\|_F}{\epsilon}\right)}{-\log(\rho)} = \mathcal{O}(\log(1/\epsilon)).$$

Thus, the number of iterations grows logarithmically with the inverse of the target accuracy, consistent with the geometric convergence we observe in practice.