Ministry of Science and Higher Education of the Republic of Kazakhstan
SDU University

Guldana Muzdybayeva

Numerical methods for matrix completion problem

THESIS

Presented in Partial Fulfillment for the
Degree of Master of Science in Mathematics
(degree code: 7M05401)
Department of Mathematics and Natural Sciences
Faculty of Engineering and Natural Sciences

Supervisor: **Shirali Kadyrov,PhD**

Kaskelen, June 2024

SDU University
Faculty of Engineering and Natural Sciences
Department of Mathematics and Natural Sciences

Dean of Faculty of Engineering and Natural Sciences

Assistant Professor, Ph.D. A[illegible]dov Ramis

____

« 04 » 06 2024

Topic of the thesis:

Numerical methods for matrix completion problem

Thesis submitted as part of the requirements for the award of the MSc in "7M05401-Mathematics"

Head of Department	Birzhan Ayanbayev, PhD
Academic Supervisor	Shirali Kadyrov, PhD
Master student	Guldana Murzybayeva

Kaskelen, 2024

Declaration

I confirm that this is my own work and the use of all material from other sources has been properly and fully acknowledged.

Guldana Muzdybayeva

June 2024

Acknowledgements

I extend my deepest gratitude to my supervisor, Professor Shirali Kadyrov, whose expertise and insightful guidance have been invaluable throughout this research journey. His rigorous analytical approach and persistent demand for precision have significantly shaped this thesis. I am also immensely thankful to the faculty and staff of the Department of Mathematics and Natural Sciences at SDU University for their educational support and administrative assistance.My family's support and encouragement deserve heartfelt recognition. They have been my constant source of motivation and resilience, providing both emotional and logistical support throughout my studies.

Dedication

This thesis is dedicated to my professors thank you for your guidance, expertise, and endless patience. To my family, who have supported me every step of the way, I am forever grateful for your love, encouragement, and understanding. Your unwavering belief in my abilities has been the driving force behind my success. To my friends, thank you for your unwavering support.

Abstract

This work addresses the significant challenge of framework completion in machi– ne learning, focusing on enhancing the accuracy and computational productivity of algorithms under conditions of large, noisy, and incomplete datasets. Central to this work are changes to two primary matrix completion techniques: Singular Value Thresholding (SVT) and Collaborative Filtering (CF). These methods were systematically progressed to handle common real-world data issues such as noise and sparsity, and were thoroughly tried over different applications, demonstrating significant execution enhancements. Through a detailed theoretical analysis, this research contributes robust frameworks for the convergence behaviors of these algorithms, giving a solid foundation for their application in practical scenarios. Improved SVT calculation, in particular, shows considerable reductions in Mean Absolute Error (MAE) and Mean Squared Error (MSE), indicating a superior performance over conventional methods. Besides, the refined CF approach presently integrates novel matrix factorization procedures, improving its utility in dynamic, personalized recommendation systems.The thesis underscores the potential of these refined algorithms in diverse fields, from advanced media to educational analytics, and sets a course for future investigate that incorporates integrating deep learning models and expanding into new data structures like tensors.

Аңдатпа

Бұл жұмыс ауқымды және толық емес деректер жиындары жағдайында алгоритмдердің дәлдігі мен есептеу өнімділігін жақсартуға назар аудара отырып, машиналық оқытудағы құрылымды аяқтаудың маңызды мәселесін қарастырады. Бұл жұмыста матрицаны толтырудың екі негізгі әдісі SVT және CF қолданылды . Бұл әдістер шу және сиректік сияқты нақты деректеріне қатысты жалпы мәселелерді шешу үшін жүйелі түрде жетілдірілді және өнімділіктің айтарлықтай жақсартуларын көрсете отырып, әртүрлі қолданбаларда кеңінен сыналды. Егжей-тегжейлі теориялық талдау арқылы бұл зерттеу осы алгоритмдердің конвергенциялық әрекетіне сенімді құрылымдарды қосады, оларды практикалық сценарийлерде қолдану үшін берік негіз береді. Жақсартылған SVT есебі, атап айтқанда, орташа абсолютті қатенің (MAE) және орташа квадрат қатесінің (MSE) айтарлықтай төмендеуін көрсетеді, бұл дәстүрлі әдістермен салыстырғанда жоғары өнімділікті көрсетеді. Бұған қоса, жетілдірілген CF тәсілі қазір динамикалық жекелендірілген ұсыныстар жүйесінде оның пайдалылығын арттыра отырып, жаңа матрицаны факторизациялау процедураларын біріктіреді. Диссертация озық медиадан білім беру аналитикасына дейінгі салалардағы осы жетілдірілген алгоритмдердің әлеуетін көрсетеді және терең оқыту үлгілерін біріктіруді және оларды тензорлар сияқты жаңа деректер құрылымдарына кеңейтуді қамтитын әрі қарай зерттеулер үшін негіз қалады.

Аннотация

В этой работе рассматривается значительная проблема завершения структуры в машинном обучении, с упором на повышение точности и вычислительной производительности алгоритмов в условиях больших, зашумленных и неполных наборов данных. Существенное значение для этой работы имеют изменения двух основных методов завершения матрицы: пороговое значение сингулярного значения (SVT) и коллаборативная фильтрация (CF). Эти методы систематически совершенствовались для решения распространенных проблем с реальными данными, таких как шум и разреженность, и были тщательно опробованы в различных приложениях, демонстрируя значительные улучшения выполнения. Благодаря подробному теоретическому анализу это исследование вносит надежные структуры для поведения сходимости этих алгоритмов, давая прочную основу для их применения в практических сценариях. Улучшенный расчет SVT, в частности, показывает значительное снижение как средней абсолютной ошибки (MAE), так и средней квадратичной ошибки (MSE), что указывает на превосходную производительность по сравнению с традиционными методами. Кроме того, усовершенствованный подход CF в настоящее время интегрирует новые процедуры факторизации матрицы, повышая ее полезность в динамических персонализированных системах рекомендаций. В диссертации подчеркивается потенциал этих усовершенствованных алгоритмов в различных областях, от продвинутых медиа до образовательной аналитики, и закладывается основа для дальнейших исследований, которые включают интеграцию моделей глубокого обучения и расширение в новые структуры данных, такие как тензоры.

Contents

1. Introduction

Matrix completion is a machine-learning task that predicts missing entries in a partially observed matrix [1]. It is crucial in scenarios like medical imaging and recommendation systems where data is incomplete [2]. The problem assumes a low-rank structure in the matrix, enabling the estimation of missing values [3]. Various optimization techniques, including convex and non-convex approaches, are employed to address matrix completion, with the latter showing potential advantages in certain scenarios.The methods used for matrix completion are evaluated through simulation studies and real-data applications, such as collaborative filtering and educational assessment, showcasing their effectiveness in handling mixed data types and providing tight probabilistic error bounds for the estimators.

From 2007 to 2009, Netflix held a competition called the Netflix Prize where teams of data scientists attempted to create algorithms that would provide better predictions of movie ratings. Completing a matrix like this one:

$$\begin{bmatrix} 1 & ? & 4 \\ ? & 2 & 7 \end{bmatrix}$$

may appear to be an ill-posed issue at first glance. We can just accidentally add the missing entries ?. The research objectives of this thesis encompass a twofold investigation. Firstly, a review of diverse matrix completion algorithms will be conducted, aiming to elucidate their underlying principles, methodologies, and applicability across various domains. This comprehensive analysis will serve to establish a foundational understanding of matrix completion techniques. Subsequently, the application of matrix completion algorithms within the realm of higher education at the university level will be undertaken. Specifically, these algorithms will be deployed to facilitate course recommendations for students. By leveraging matrix completion methodologies, the objective is to devise an efficient and effective system for guiding students in their selection of courses, thereby enhancing their academic journey and overall educational experience. Chapter 1 sets the stage by providing a thorough exploration of the mathematical principles that underpin matrix completion. This chapter traces the historical development of matrix theory, discusses foundational theorems, and establishes the groundwork for understanding the complexities of rank minimization and the pivotal role of regularization in matrix completion.

Chapter 2 delves into the core algorithms that facilitate matrix completion, including both convex and non-convex optimization strategies. A comparative analysis of these methodologies unveils their theoretical and practical merits, preparing the ground for further algorithmic enhancements discussed in subsequent chapters.

Chapter 3 introduces modifications and innovations to standard matrix completion algorithms to tackle specific challenges such as noise, sparsity, and computational constraints. This chapter details the development of hybrid algorithms that synergize the strengths of traditional methods to foster improved performance and robustness.

Chapter 4 examines the application of matrix completion techniques in various real-world contexts, from digital media to biomedical imaging. By contextualizing these algorithms within specific case studies, this chapter highlights their practical impacts and the challenges they address within industry and academia.

Chapter 5 provides empirical evidence to support the theoretical models and computational strategies explored in earlier chapters. It meticulously outlines the experimental methodologies employed, presents a detailed analysis of the results, and benchmarks these findings against existing approaches.

Chapter 6 synthesizes insights gained throughout the thesis and discusses the broader implications for the field of predictive modeling and artificial intelligence. It outlines potential avenues for future research, including the integration of matrix completion with emerging technologies and the exploration of new problem domains.

Chapter 7 summarize the significant contributions of the thesis, reaffirming the role of advanced matrix completion methods in the modern data analysis toolkit. It reflects on the theoretical advancements and practical applications explored throughout the chapters, underscoring the broader impact of this work on the field.

This thesis contributes to the field of matrix completion by developing methods that are not only theoretically sound but also practically viable, offering significant improvements over traditional techniques. Through rigorous analysis and innovative algorithm design, it advances our understanding of how to effectively tackle the challenges posed by incomplete and imperfect data in various domains.

2. Preliminaries

In this preliminary chapter, we will review some definitions and some notions that are related to my thesis.First of all, review the matrix inner product and the trace operator.

Definition 2.0.1. The trace operator, denoted as $\operatorname{tr}(\mathbf{A})$, computes the sum of the diagonal elements of a square matrix $\mathbf{A}$. In other words, it yields the sum of the eigenvalues of $\mathbf{A}$. Mathematically, if $\mathbf{A}$ is an $n \times n$ matrix with entries a_{ij}, then the trace of $\mathbf{A}$ is given by:

$$\operatorname{tr}(\mathbf{A}) = \sum_{i=1}^{n} a_{ii}$$

Let us consider the following matrix:

$$\mathbf{A} = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

The trace of $\mathbf{A}$ is calculated as the sum of its diagonal elements:

$$\operatorname{tr}(\mathbf{A}) = 1 + 5 + 9 = 15$$

So, the trace of $\mathbf{A}$ is 15.

Definition 2.0.2. The inner product of two matrices, denoted as $\langle \mathbf{A}, \mathbf{B} \rangle$, is defined as the sum of the products of corresponding entries in the matrices when they are multiplied element-wise, followed by summation of these products. Mathematically, if $\mathbf{A}$ and $\mathbf{B}$ are both $m \times n$ matrices, their inner product is given by:

$$\langle \mathbf{A}, \mathbf{B} \rangle = \sum_{i=1}^{m} \sum_{j=1}^{n} a_{ij} b_{ij}$$

Let us consider two matrices:

$$\mathbf{A} = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$$

$$\mathbf{B} = \begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix}$$

The inner product of $\mathbf{A}$ and $\mathbf{B}$ is calculated as:

$$\begin{aligned} \langle \mathbf{A}, \mathbf{B} \rangle &= (1 \times 5) + (2 \times 6) + (3 \times 7) + (4 \times 8) \\ &= 5 + 12 + 21 + 32 \\ &= 70 \end{aligned}$$

So, the inner product of $\mathbf{A}$ and $\mathbf{B}$ is 70.

Next, we define norms for matrix.

Definition 2.0.3. The norm of a matrix is a scalar that provides a measure of the magnitude of the elements within the matrix.We shall delineate several notable matrix norms, including the L_0, L_1, and L_2 norms, alongside the Frobenius norm, L_∞ and L_p norm.

Definition 2.0.4. L_0 norm of a matrix $\mathbf{A}$, denoted as $\|\mathbf{A}\|_0$ corresponds to the count of its non-zero elements across all rows and columns:

$$\|\mathbf{A}\|_0 = \sum_{i=1}^{m} \sum_{j=1}^{n} \mathbf{1}_{\{a_{ij} \neq 0\}}$$

Where, $\mathbf{1}_{\{a_{ij} \neq 0\}}$ is the indicator function that returns 1 if the element a_{ij} of the matrix $\mathbf{A}$ is nonzero, and 0 otherwise.

Find L_0 norm of the matrix $\mathbf{A} = \begin{bmatrix} 2 & 1 \\ -1 & 3 \\ 1 & 5 \end{bmatrix}$. To calculate L_0 we count the number of non-zero elements in the matrix. The matrix $\mathbf{A}$ has the following non-zero elements:2,1, -1,3,1,5. Hence, the L_0 norm of matrix $\mathbf{A}$ is the sum of these non-zero elements, which is $\|\mathbf{A}\|_0 = 6$.

Definition 2.0.5. The L_1 norm of a matrix $\mathbf{A}$, symbolized as $\|\mathbf{A}\|_1$, is computed as the maximum absolute column sum:

$$\|\mathbf{A}\|_1 = \max_{1 \leq j \leq n} \sum_{i=1}^{m} |a_{ij}|$$

Where a_{ij} represents the element in the i th row and j th column of matrix $\mathbf{A}$.

Let us find the L_1 norm of a matrix $\mathbf{A}$:

$$\mathbf{A} = \begin{bmatrix} 2 & 1 \\ -1 & 3 \\ 1 & 5 \end{bmatrix}$$

To calculate the L_1 norm, we sum the absolute values of each column and then find the maximum of these sums:

$$\|\mathbf{A}\|_1 = \max(|2| + |-1| + |1|, |1| + |3| + |5|)$$
$$\|\mathbf{A}\|_1 = \max(4, 9) = 9$$

Therefore, the L_1 norm of matrix $\mathbf{A}$ is $\|\mathbf{A}\|_1 = 9$.

Definition 2.0.6. The L_2 norm of a matrix $\mathbf{A}$, represented as $\|\mathbf{A}\|_2$, characterizes the maximum singular value of $\mathbf{A}$, which can be computed using singular value decomposition (SVD) or the square root of the maximum eigenvalue of $\mathbf{A}^T\mathbf{A}$: Mathematically, it can be expressed as:

$$\|\mathbf{A}\|_2 = \sqrt{\lambda_{\max}\left(\mathbf{A}^T\mathbf{A}\right)}$$

where $\lambda_{\max}\left(\mathbf{A}^T\mathbf{A}\right)$ denotes the maximum eigenvalue of $\mathbf{A}^T\mathbf{A}$

Let us proceed with the calculation: Given matrix A:

$$\mathbf{A} = \begin{bmatrix} 2 & 1 \\ -1 & 3 \\ 1 & 5 \end{bmatrix}$$

We compute $\mathbf{A}^T\mathbf{A}$:

$$\mathbf{A}^T\mathbf{A} = \begin{bmatrix} 2 & -1 & 1 \\ 1 & 3 & 5 \end{bmatrix} \begin{bmatrix} 2 & 1 \\ -1 & 3 \\ 1 & 5 \end{bmatrix} = \begin{bmatrix} 6 & 4 \\ 4 & 35 \end{bmatrix}$$

Next, we find the eigenvalues of $\mathbf{A}^T\mathbf{A}$. The characteristic polynomial of $\mathbf{A}^T\mathbf{A}$ is given by:

$$\det\left(\mathbf{A}^T\mathbf{A} - \lambda\mathbf{I}\right) = \det\begin{pmatrix} 6-\lambda & 4 \\ 4 & 35-\lambda \end{pmatrix} = (6-\lambda)(35-\lambda) - 16 = \lambda^2 - 41\lambda + 194$$

Solving for λ yields the eigenvalues λ_1 and λ_2 :

$$\lambda_1 = \frac{41 - \sqrt{905}}{2} \quad \text{and} \quad \lambda_2 = \frac{41 + \sqrt{905}}{2}$$

The L_2 norm of $\mathbf{A}$ is the square root of the maximum eigenvalue:

$$\|\mathbf{A}\|_2 = \sqrt{\lambda_{\max}} = \sqrt{\frac{41 + \sqrt{905}}{2}}$$

Definition 2.0.7. The Frobenius norm of a matrix $\mathbf{A}$, denoted as $\|\mathbf{A}\|_F$, corresponds to the square root of the sum of squares of all elements of $\mathbf{A}$:

$$\|A\|_F = \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n} a_{ij}^2}$$

This Frobenius norm can be directly calculated as:

$$\|A\|_F = \sqrt{\sum_{i=1}^{n} \sigma_i^2}$$

where σ_i are the singular values of A.

Now calculate the Frobenius norm for the matrix $\mathbf{A}$:

$$\mathbf{A} = \begin{bmatrix} 2 & 1 \\ -1 & 3 \\ 1 & 5 \end{bmatrix}$$

We square each element, sum the squares, and then take the square root of the sum:

$$\|\mathbf{A}\|_F = \sqrt{(2^2 + 1^2 + (-1)^2 + 3^2 + 1^2 + 5^2)}$$
$$\|\mathbf{A}\|_F = \sqrt{(4 + 1 + 1 + 9 + 1 + 25)}$$
$$\|\mathbf{A}\|_F = \sqrt{41}$$

Therefore, the Frobenius norm of matrix $\mathbf{A}$ is $\|\mathbf{A}\|_F = \sqrt{41}$. In addition to the aforementioned norms, other matrix norms merit mention the infinity norm, and the p-norm.

Definition 2.0.8. The infinity norm of a matrix $\mathbf{A}$, represented as $\|\mathbf{A}\|_\infty$, is the maximum absolute row sum of $\mathbf{A}$:

$$\|\mathbf{A}\|_\infty = \max_{1\le i\le m} \sum_{j=1}^{n} |a_{ij}|$$

Where a_{ij} represents the element in the i th row and j th column of matrix $\mathbf{A}$.

Let us calculate the infinity norm for the matrix $\mathbf{A}$:

$$\mathbf{A} = \begin{bmatrix} 2 & 1 \\ -1 & 3 \\ 1 & 5 \end{bmatrix}$$

We sum the absolute values of each row and then find the maximum of these sums:

For the first row: $|2| + |1| = 3$
For the second row: $|-1| + |3| = 4$
For the third row: $|1| + |5| = 6$
Therefore, the infinity norm of matrix $\mathbf{A}$ is $\|\mathbf{A}\|_\infty = 6$.

Definition 2.0.9. The p-norm of a matrix $\mathbf{A}$, denoted as $\|\mathbf{A}\|_p$, is computed similarly to the vector p-norm:

$$\|\mathbf{A}\|_p = \left(\sum_{i=1}^{m}\sum_{j=1}^{n}|a_{ij}|^p\right)^{\frac{1}{p}}$$

Where a_{ij} represents the element in the i th row and j th column of matrix $\mathbf{A}$.

The p-norm generalizes various common norms:
$p = 1$ yields the L1 norm.
$p = 2$ yields the L2 norm.
$p = \infty$ yields the infinity norm.

For other values of p, it is a more generalized norm. Let us denote the matrix $\mathbf{A}$ again as:

$$\mathbf{A} = \begin{bmatrix} 2 & 1 \\ -1 & 3 \\ 1 & 5 \end{bmatrix}$$

To calculate the p-norm, we need to specify the value of p. For example, calculate the p norm for $p = 3$:

$$\begin{aligned} \|\mathbf{A}\|_3 &= \left(|2|^3 + |1|^3 + |-1|^3 + |3|^3 + |1|^3 + |5|^3\right)^{\frac{1}{3}} \\ &= (8 + 1 + 1 + 27 + 1 + 125)^{\frac{1}{3}} = (163)^{\frac{1}{3}} \end{aligned}$$

Therefore, the p-norm of matrix $\mathbf{A}$ for $p = 3$ is $\|\mathbf{A}\|_3 = (163)^{\frac{1}{3}}$.

Definition 2.0.10. Ill-posed data refers to a type of problem where small errors or perturbations in the input or data can lead to large errors in the output or solution. It is characterized by the lack of uniqueness, stability, or continuity in the solution.

An example of an ill-posed 3x3 matrix can be one where the rows are nearly parallel, causing numerical instability or making it challenging to find a reliable solution. Example:

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 4 & 6 \\ 3 & 6 & 9 \end{bmatrix}$$

In this case, if we are trying to solve the system of linear equations represented by this matrix (Ax = b), we have:

$$x_1 + 2x_2 + 3x_3 = b_1$$
$$2x_1 + 4x_2 + 6x_3 = b_2$$
$$3x_1 + 6x_2 + 9x_3 = b_3$$

The third equation is a linear combination of the first two equations, indicating that the system is dependent and hence, the matrix is ill-posed. This dependency leads to numerical instability and difficulty in finding a unique solution.

Definition 2.0.11. Sparse data refers to datasets that have a relatively low density of non-zero values compared to the total number of possible values.

Sparse datasets are commonly used in various scientific and machine learning applications, including deep neural networks [4], [5].

In the context of matrices, sparse data means that most of the elements in the matrix have a value of zero. Sparse matrices are common in many applications, especially when dealing with high-dimensional data or when representing relationships between entities.

Let us consider example of a sparse matrix:

$$A = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 \\ 0 & 0 & 3 & 0 \\ 0 & 4 & 0 & 0 \end{bmatrix}$$

In this matrix, most of the elements are zero, making it sparse. Sparse matrices are often encountered in real-world scenarios, and special algorithms and data structures are used to efficiently store and manipulate them. These algorithms take advantage of the sparsity of the data to reduce memory usage and computational complexity.

Definition 2.0.12. Convex optimization refers to optimization problems where the objective function and the feasible set are convex.

In simpler terms, a convex optimization problem involves finding the minimum of a convex function over a convex set. Convex optimization problems have desirable properties, such as a single global minimum, which makes them easier to solve efficiently and reliably.

Definition 2.0.13. Non-convex optimization, on the other hand, involves optimization problems where either the objective function, the feasible set, or both are non-convex.

Non-convex optimization problems are more challenging because they can have multiple local minima, making it difficult to guarantee finding the global optimum. These problems often require more sophisticated algorithms and may not always converge to the global minimum.

Definition 2.0.14. Low-rank matrix is a matrix that can be approximated by a product of two matrices of much lower ranks. The low-rank constraint is essential in matrix completion problems to recover missing entries of a matrix from observed entries efficiently.

Let us construct an illustrative instance of low-rank matrix completion. Assume we are presented with a partially observed matrix $\mathbf{M}$, wherein certain entries are missing, denoted as $\mathbf{O}$, while the unobserved entries are represented as $\mathbf{M}$. Let $\mathbf{O}$ is the provided data:

$$\mathbf{O} = \begin{bmatrix} 1 & 0 & 3 \\ 0 & 2 & 0 \\ 4 & 0 & 0 \end{bmatrix}$$

Our objective lies in the restoration of the missing entries $\mathbf{M}$ through the mechanism of low-rank matrix completion. A prevailing strategy in the realm of low-rank matrix completion entails the minimization of the nuclear norm of the matrix, while ensuring alignment with the observed entries. This optimization endeavor is often realized through methodologies such as Singular Value Thresholding. Let us presume that $\mathbf{X}$, a low-rank matrix effectively approximating $\mathbf{M}$, possesses a rank of 1. We can express $\mathbf{X}$ as a product of two matrices of lower ranks, $\mathbf{U}$ and $\mathbf{V}^T$, wherein $\mathbf{X} = \mathbf{U}\mathbf{V}^T$. For the sake of expediency, let us designate:

$$\mathbf{U} = \begin{bmatrix} 1 \\ 0 \\ 4 \end{bmatrix}$$

and

$$\mathbf{V}^T = \begin{bmatrix} 1 & 0 & 3 \end{bmatrix}$$

Subsequently, we obtain the low-rank approximation $\mathbf{X}$ by executing the outer product of $\mathbf{U}$ and $\mathbf{V}^T$:

$$\mathbf{X} = \mathbf{U}\mathbf{V}^T = \begin{bmatrix} 1 \\ 0 \\ 4 \end{bmatrix} \begin{bmatrix} 1 & 0 & 3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 3 \\ 0 & 0 & 0 \\ 4 & 0 & 12 \end{bmatrix}$$

This resultant low-rank approximation $\mathbf{X}$ effectively serves as an estimation of the complete matrix $\mathbf{M}$. Consequently, we proceed to populate the absent entries of $\mathbf{M}$ utilizing the corresponding entries of $\mathbf{X}$. In the present instance, the absent entry in the second row and the second column is appropriately filled with the value 0.

This elucidates a rudimentary exemplification of low-rank matrix completion, thereby delineating the efficacy in recuperating absent entries through the exploitation of the matrix's low-rank configuration. It is worth noting that within practical applications, more nuanced algorithms and optimization techniques are deployed to handle larger matrices and real-world datasets effectively.

3. Literature Review on Matrix Completion Algorithms

The task of completing an unknown matrix with a limited number of elements is known as the matrix completion problem. Matrix completion finds applications in diverse domains, showcasing its practical significance. Collaborative filtering, aids recommendation systems by predicting user-item ratings from incomplete data, as demonstrated in [6]. Furthermore, in dimensionality reduction, exemplified by [7], matrix completion assists in reducing data dimensionality while preserving essential properties and facilitating tasks in image and signal processing. Additionally, matrix completion contributes to clustering algorithms, aiding in identifying similar data points from incomplete information, as evidenced by [8]. In [9], matrix completion plays a role in non-negative matrix factorization, extracting meaningful insights from high-dimensional data in fields like data analysis and pattern recognition. Lastly, in sensor networks, as demonstrated in [10], matrix completion assists in estimating sensor locations based on incomplete and noisy measurements, illustrating its utility across various domains. These applications underscore the practical relevance of matrix completion in domains ranging from e-commerce and entertainment to scientific research and engineering.

Matrix completion stands as a distinctive instance within the broader spectrum of the matrix recovery problem, which endeavors to reconstruct a matrix from generic and often random linear measurements. Let Y represent the measured data corrupted by errors E. The objective is to reconstruct a low-rank matrix $X \in \mathbb{R}^{m_1 \times n_1}$ from $Y = F(X) + E \in \mathbb{R}^{m \times n}$,where $F : \mathbb{R}^{m_1 \times n_1} \to \mathbb{R}^{m \times n}$ is a linear operator. The problem is formulated as:

$$\min_{X,E} \operatorname{rank}(X) + \lambda \|E\| \quad \text{subject to} \quad Y = F(X) + E,$$

where λ is a regularization parameter, and $\|\cdot\|$ denotes the L_0-norm [11] or the $L_{2.0}$-norm [12] for sparsity. With the contributions of [11], [12], [13], the literature on robust subspace analysis has advanced significantly. [11]

introduced Principal Component Pursuit (PCP), a convex program designed to decompose data matrices into low-rank and sparse components, showcasing its effectiveness in various applications including video surveillance and face recognition [11],[12] proposed Low-Rank Representation (LRR) for subspace segmentation, seeking the lowest-rank representation among all candidates to segment data drawn from multiple subspaces. This method differs from sparse representation by jointly handling data vectors, thus capturing global structure more effectively. Building upon this, [13] expanded LRR's scope to the robust recovery of subspace structures, demonstrating its robustness in subspace clustering and outlier detection. The LRR objective function minimizes the nuclear norm subject to linear constraints, efficiently recovering true subspace structures in clean data and detecting outliers in corrupted data. [13] extends the recovery of corrupted data from single to multiple subspaces, offering a polynomial-time solvable method that achieves state-of-the-art performance in various applications, including motion segmentation and face recognition.[11], [12], [13] collectively contribute to a comprehensive understanding of robust subspace analysis, offering powerful tools for data decomposition, segmentation, and error correction in complex datasets.

The studies of [14],[15], and [16] have made substantial contributions to the literature on matrix completeness, rank minimization, and nuclear norm minimization. In this work [14] introduced an approach for exact matrix completion via convex optimization, demonstrating that most low-rank matrices can be perfectly recovered from incomplete sets of entries by solving a convex optimization program. This work connects with the broader literature on compressed sensing, showcasing the potential of exact recovery from limited information.This study [15] discussed the problem of rank minimization and its applications, presenting novel approaches such as trace and log-det heuristics for positive semidefinite matrices, and their extensions to general matrices through semidefinite embedding. These methods provide efficient solutions to various practical scenarios, including system realization and portfolio optimization. In [16] explored guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization, addressing the affine rank minimization problem and its applications in diverse fields. In [16] extends the compressed sensing framework to handle low-rank assumptions, demonstrating the potential of nuclear norm minimization in recovering minimum-rank solutions under specific conditions. Together, Authors [14],[15], [16] contribute to a deeper understanding of matrix completion, rank minimization, and nuclear norm minimization, offering valuable insights into their theoretical foundations and practical implications.

The studies of [17],[18], [19] and numerous others have made substantial contributions to the literature on low-rank matrix completion.In [18] explore the reconstruction of low-rank matrices using both trace-norm and less-explored max-norm. They provide reconstruction guarantees based on the Rademacher complexity of these norms, showing their superiority over existing specialized analyses. Their study demonstrates the potential of the trace-norm and max-norm as surrogates for rank, offering guarantees on approximate low-rank matrix reconstruction. Building upon this, [19] addresses the challenge of low-rank matrix completion under more flexible sampling schemes. Their work introduces a universal recovery guarantee for matrix completion, accommodating various sampling strategies and relaxing the requirement for fresh sample sets for each new matrix. They show that certain sampling schemes, such as those based on bipartite graphs with large spectral gaps, enable the exact recovery of low-rank matrices using nuclear norm minimization. This contrasts with traditional methods requiring uniform random sampling and fresh sample sets, thereby broadening the applicability of matrix completion algorithms. These studies [19],[18],[17] contribute to advancing our understanding of low-rank matrix completion, offering insights into different reconstruction strategies and expanding the scope of applicable sampling schemes.

Two main types of low-rank matrix completion approaches are those based on matrix factorization [19] and those based on rank minimization [20], [21] Matrix factorization-based methods are techniques used in various fields such as machine learning, signal processing, and recommender systems. These methods decompose a data matrix into a product of two-factor matrices with low ranks. The goal is to represent the data matrix as a linear combination of these factor matrices. Matrix factorization can be used for tasks such as dimension reduction, feature extraction, blind source separation, data compression, and knowledge discovery. It is particularly effective in collaborative filtering and recommender systems, where it uses the interactions between users and items to predict ratings. Several variations of matrix factorization methods have been proposed, including those that incorporate additional attributes of the data, such as type attributes of movies and age attributes of users. These methods have shown promising results in terms of recommendation performance and rating prediction accuracy.[22], [23], [24] Matrix factorization methods combined with social trust propagation enhance recommendation accuracy in social networks. SocialMF integrates trust propagation among users, outperforming STE in RMSE values [25]. It excels especially on denser datasets like Flixster. "Learning to Recommend with Social Trust Ensemble" blends user preferences and trusted friends' tastes using a parameter α, showcasing effectiveness via empirical analysis on the Epinions dataset [26]. TrustMF further improves collaborative filtering by integrating rating data and social trust networks, addressing data sparsity and cold start challenges [27]. These works collectively advance recommendation systems by leveraging trust relationships and matrix factorization techniques. Rank minimization-based methods are a class of techniques used in various fields such as image restoration, model inference, and dimension reduction. These methods aim to minimize the rank of a matrix to improve the quality of the results. In image restoration, low-rank minimization methods have been used to enhance image quality by directly utilizing the matrix rank [28]. In model inference, rank minimization is employed to eliminate redundant degrees of freedom and obtain accurate models with fewer parameters [29]. In dimension reduction, rank selection methods based on hypothesis testing are used to determine the appropriate number of sub-structures in the data [29]. These methods have shown promising results in terms of both quantitative measures and visual qualities in various applications [30], [31]. Singular value thresholding (SVT) is a method used to improve the reconstruction performance in various applications, such as heart disease detection and low-rank tensor optimization. SVT is based on the singular value decomposition (SVD) technique and is efficient in reconstructing a low-rank matrix or tensor from a limited number of entries. It achieves this by applying a thresholding algorithm on the singular values of the matrix or tensor. The effectiveness of SVT has been demonstrated in different domains, including electrocardiogram (ECG) signal compression and matrix recovery. The proposed algorithms, such as ScreeNOT and FR t-SVT, utilize SVT to achieve optimal reconstruction performance and computational efficiency.[32], [33], [34] Another way to handle different kinds of convex constraints is to modify the singular value thresholding procedure. It could deal with issues of the following form:

$$\begin{array}{ll} \text{minimize} & \|\boldsymbol{X}\|_* \\ \text{subject to} & f_i(\boldsymbol{X}) \leq 0, \quad i = 1, \ldots, m \end{array}$$

where each f_i is a Lipschitz convex function.

4. Methodology

In this chapter, the second research question will be addressed. We want to build a recommendation system for this data using two methods. Our goal is to complement the Matrix. Purpose of Matrix complement creating a recommendation system. Determining what lessons the student should take in the third year, based on the lessons learned by the student in the second year. For that, when we look at the data, students have no grades in their third year. It means that we need to complement the same matrix. That is, to create a system of offers that give students a forecast of their grades, we need to supplement the Matrix. There are many ways to supplement it. We want to use collaborative filtering and nuclear norm minimization. We want to compare the two. We used the singular value thresholding algorithm for Nuclear norm minimization. And for rank minimization, we used the collaborative filtering algorithm.The methodology is bifurcated into two primary segments: the theoretical analysis of algorithms and their empirical validation through a series of designed experiments.

The theoretical segment commences with a formal definition of the matrix completion problem, specifying the mathematical notation and the assumptions underlying the study. This includes a detailed description of the low-rank matrix approximation problem, highlighting its relevance and application in various data-rich environments.

For the optimization strategies, the thesis adopts both convex and non-convex approaches. The convex methods focus on nuclear norm minimization techniques, which are well-known for their robustness and theoretical guarantees under certain conditions. Non-convex methods involve manifold optimization and randomized algorithms, which promise better scalability and potential for handling larger datasets. The theoretical contribution of the thesis is the development of a hybrid algorithm that integrates the robust error handling of convex methods with the efficiency and scalability of non-convex techniques.

Following theoretical analysis, algorithms are developed and simulated under controlled conditions to test their theoretical properties, such as convergence rates and sensitivity to parameter changes. Simulation environments are created using synthetic datasets generated to mimic real-world data characteristics like sparsity, noise levels, and rank properties. The impact of these factors on the performance of the algorithms is analyzed, providing insights into their practical utility and limitations.

The empirical segment of the methodology involves applying the developed algorithms to real-world datasets. This includes datasets commonly used in the field of matrix completion, such as the MovieLens dataset for recommendation systems and publicly available medical imaging datasets for image reconstruction tasks.

Each dataset undergoes a preprocessing phase to align with the assumptions of the matrix completion problem, such as matrix rank estimation and the handling of missing data. The performance of each algorithm is then evaluated based on several metrics: Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and computational efficiency. The evaluation also compares the performance of the newly developed hybrid algorithms against existing benchmark techniques in the field.

Statistical tools and software, including MATLAB and Python with libraries such as NumPy and Scikit-Learn, are utilized for both simulation and real-data analysis. The thesis employs a rigorous statistical analysis to ensure the robustness of the results, including hypothesis testing, confidence interval estimation, and regression analysis to quantify the effects of various data characteristics on algorithm performance.

Given the potential sensitivity of real-world data, particularly in medical applications, the methodology also outlines the ethical considerations adhered to during the research. This includes data anonymization processes and adherence to data protection regulations to ensure the integrity and confidentiality of the data used in the study.

This methodology section of the thesis not only elucidates the technical and analytical processes involved in assessing matrix completion algorithms but also ensures that these processes are transparent and reproducible, satisfying the rigorous requirements of advanced academic research in the field.

4.1 Dataset and Pre-processing

Dataset preprocessing involves the acquisition of data from the universities' databases. In this research, a real dataset encompassing 603 undergraduate students specializing in computer science from 2018 to 2022 was utilized. With the aim of constructing a 3rd-year course recommender system based on preceding 2-year courses, scrutiny was confined to the initial three-year courses. Courses with enrollments below 60 were excluded, resulting in 90 courses, including 28 3rd-year courses.

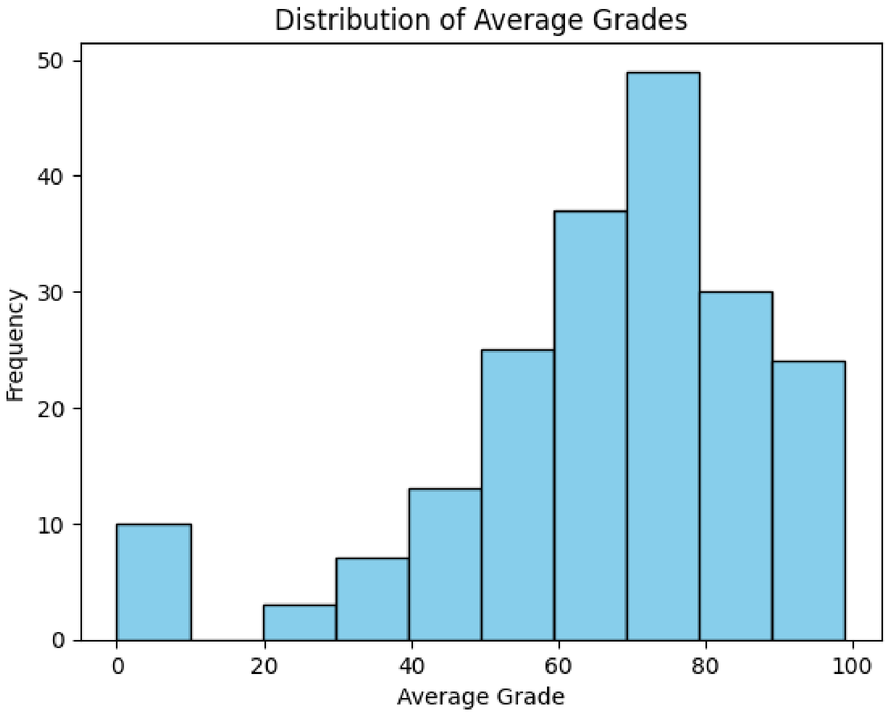

Figure 4.1: Distribution of Average Grades.

4.2 Nuclear Norm Minimization

Nuclear norm minimization techniques offer a pathway to the exact recovery of a matrix with missing values under certain broad conditions [35], [16], [36], [37], [38], [14]. In instances where the observed values are devoid of noise, the process enables the perfect retrieval of a low-rank matrix [14]. However, in scenarios where measurements are afflicted by noise, the recovery process is circumscribed by an error bound that scales with the level of noise, a relationship that holds with high probability [37].The nuclear norm minimization problem is conventionally formulated as articulated in several references [35], [37], [38],[14],

[39]:

$$\min_{\mathbf{X}} \|\mathbf{X}\|_* = \sum_{k=1}^{\min(m,n)} \sigma_k(\mathbf{X}) \quad \text{subject to} \quad X_{ij} = M_{ij}, (i,j) \in \Omega \tag{4.2.1}$$

where $\|\cdot\|_*$ is the nuclear norm, and $\sigma_k(\mathbf{X})$ is the k th largest singular value of $\mathbf{X}$. The problem can be transformed into a quadratically constrained minimization problem:

$$\min_{\mathbf{X}} \|\mathbf{X}\|_* = \sum_{k=1}^{\min(m,n)} \sigma_k(\mathbf{X}) \quad \text{subject to} \sum_{(i,j)\in\Omega} (X_{ij} - M_{ij})^2 \leq \varepsilon, \tag{4.2.2}$$

or a regularized unconstrained problem:

$$\min_{\mathbf{X}} \|\mathbf{X}\|_* + \lambda \sum_{(i,j)\in\Omega} (X_{ij} - M_{ij})^2 \tag{4.2.3}$$

Due to the iterative nature of solving the nuclear norm issue (4.2.1), which requires SVD at each iteration, significant computational costs result.

The literature on matrix completion and low-rank matrix recovery algorithms underscores the significance of nuclear norm minimization in efficiently handling large-scale, noisy, or incomplete data matrices. In [40] present the Alternating Direction Method (ADM), showcasing its effectiveness in completing low-rank matrices across various scenarios, including noiseless, noisy, and positive semidefinite cases. In [21] introduce an accelerated proximal gradient algorithm for nuclear norm regularized linear least squares problems, emphasizing its robustness in handling noisy or incomplete data matrices through convex relaxation techniques. Concurrently, [41] focuses on the linear convergence of the Proximal Gradient Method (PGM) for trace norm regularization, elucidating its efficacy in approximating low-rank matrices without necessitating strong convexity assumptions. These contributions are substantiated by theoretical advancements, such as the proof of global convergence for gradient search in low-rank matrix approximation problems provided by [42]. Moreover, novel algorithms like Large-Scale Matrix Factorization using Distributed Stochastic Gradient Descent (DSGD) [43] and the Jellyfish algorithm [44] offer efficient solutions for large-scale matrix completion tasks, leveraging techniques like stochastic gradient descent and parallel incremental gradient methods. These works collectively underscore the importance and effectiveness of nuclear norm minimization and related techniques in addressing challenges associated with matrix completion and low-rank matrix recovery in various real-world applications. When presented with an incomplete low-rank data matrix $X = [X_{ij}] \in \mathbb{R}^{m\times n}$, the challenge of matrix completion arises, with the following formulation:

$$\min \operatorname{rank}(\mathbf{Y}) \quad \text{subject to } Y_{ij} = X_{ij}, (i,j) \in \mathbf{\Omega}, \tag{4.2.4}$$

In this context, $Y = [Y_{ij}] \in \mathbb{R}^{m\times n}$ serves as the decision variable, while Ω represents the set of observed entry positions. Each pair $(i, j) \in \Omega$ is selected independently according to a Bernoulli distribution with probability p. The goal of the problem (4.2.4) is to determine which explanation best fits the observable data. It is generally ill-posed. Under specific restrictions on the matrix rank, missing rate, and sampling strategy, it is possible to recover the missing items of Y faithfully and with a high probability [[14],[18],[19]].

Problem Definition: Given a matrix $X \in \mathbb{R}^{m\times n}$ and a threshold parameter τ, Singular Value Thresholding (SVT) aims to produce a new matrix X_{thresh} by applying a soft-thresholding operation to the singular values of X.

Step-by-Step Formulation:

Singular Value Decomposition (SVD)

First, compute the Singular Value Decomposition of X:

$$X = U\Sigma V^T \tag{4.2.5}$$

Where, $U \in \mathbb{R}^{m\times m}$ and $V \in \mathbb{R}^{n\times n}$ are orthogonal matrices, and $\Sigma \in \mathbb{R}^{m\times n}$ is a diagonal matrix with singular values σ_i on the diagonal.

Soft-Thresholding of Singular Values. Apply the threshold τ to the singular values in Σ. The soft-thresholding operation is defined as:

$$\sigma_i' = \max(\sigma_i - \tau, 0) \tag{4.2.6}$$

This means we subtract τ from each singular value σ_i, and if the result is negative, we set it to zero. Formally:

$$\Sigma_{\text{thresh}} = \text{diag}(\sigma_1', \sigma_2', \ldots, \sigma_r') \tag{4.2.7}$$

where $r = \min(m, n)$ is the number of singular values, and:

$$\sigma_i' = \begin{cases} \sigma_i - \tau & \text{if } \sigma_i > \tau \\ 0 & \text{if } \sigma_i \le \tau \end{cases} \tag{4.2.8}$$

Reconstruct the thresholded matrix X_{thresh} using the orthogonal matrices U and V and the thresholded singular value matrix Σ_{thresh}:

$$X_{\text{thresh}} = U\Sigma_{\text{thresh}}V^T \tag{4.2.9}$$

Compact Formulation. The SVT operation can be compactly written as:

$$\text{SVT}_\tau(X) = U\max(\Sigma - \tau, 0)V^T \tag{4.2.10}$$

where the max operation is applied element-wise to the singular values in Σ. Suppose we have a matrix X:

$$X = \begin{bmatrix} 4 & 0 \\ 3 & -5 \end{bmatrix} \tag{4.2.11}$$

Compute the SVD of X Let

$$X = U\Sigma V^T \tag{4.2.12}$$

with:

$$U = \begin{bmatrix} -0.8 & -0.6 \\ -0.6 & 0.8 \end{bmatrix}, \quad \Sigma = \begin{bmatrix} 6.4 & 0 \\ 0 & 2.2 \end{bmatrix}, \quad V = \begin{bmatrix} -0.6 & -0.8 \\ -0.8 & 0.6 \end{bmatrix} \tag{4.2.13}$$

Apply Soft-Thresholding to Σ with $\tau = 2$

$$\Sigma_{\text{thresh}} = \begin{bmatrix} \max(6.4 - 2, 0) & 0 \\ 0 & \max(2.2 - 2, 0) \end{bmatrix} = \begin{bmatrix} 4.4 & 0 \\ 0 & 0.2 \end{bmatrix} \tag{4.2.14}$$

Reconstruct the Thresholded Matrix

$$X_{\text{thresh}} = U\Sigma_{\text{thresh}}V^T \tag{4.2.15}$$

$$X_{\text{thresh}} = \begin{bmatrix} -0.8 & -0.6 \\ -0.6 & 0.8 \end{bmatrix} \begin{bmatrix} 4.4 & 0 \\ 0 & 0.2 \end{bmatrix} \begin{bmatrix} -0.6 & -0.8 \\ -0.8 & 0.6 \end{bmatrix} \tag{4.2.16}$$

Calculating the intermediate steps:

$$U\Sigma_{\text{thresh}} = \begin{bmatrix} -0.8 & -0.6 \\ -0.6 & 0.8 \end{bmatrix} \begin{bmatrix} 4.4 & 0 \\ 0 & 0.2 \end{bmatrix} = \begin{bmatrix} -3.52 & -0.12 \\ -2.64 & 0.16 \end{bmatrix} \tag{4.2.17}$$

$$X_{\text{thresh}} = \begin{bmatrix} -3.52 & -0.12 \\ -2.64 & 0.16 \end{bmatrix} \begin{bmatrix} -0.6 & -0.8 \\ -0.8 & 0.6 \end{bmatrix} \tag{4.2.18}$$

$$X_{\text{thresh}} = \begin{bmatrix} (-3.52 \times -0.6) + (-0.12 \times -0.8) & (-3.52 \times -0.8) + (-0.12 \times 0.6) \\ (-2.64 \times -0.6) + (0.16 \times -0.8) & (-2.64 \times -0.8) + (0.16 \times 0.6) \end{bmatrix} \tag{4.2.19}$$

$$X_{\text{thresh}} = \begin{bmatrix} 2.112 + 0.096 & 2.816 - 0.072 \\ 1.584 - 0.128 & 2.112 + 0.096 \end{bmatrix} \tag{4.2.20}$$

$$X_{\text{thresh}} = \begin{bmatrix} 2.208 & 2.744 \\ 1.456 & 2.208 \end{bmatrix} \tag{4.2.21}$$

Thus, the thresholded matrix X_{thresh} is:

$$X_{\text{thresh}} = \begin{bmatrix} 2.208 & 2.744 \\ 1.456 & 2.208 \end{bmatrix} \tag{4.2.22}$$

Algorithm 1 Matrix Completion using Singular Value Thresholding (SVT) - Part 1

Require: Y_{train} - training data matrix with missing values (size: $m \times n$)
Require: Y_{test} - testing data matrix with missing values (size: $m' \times n$)
Require: Ω_{train} - binary mask indicating observed entries in Y_{train}
Require: Ω_{test} - binary mask indicating observed entries in Y_{test}
Require: τ - threshold parameter
Require: *max_iter* - maximum number of iterations (default: 100)
Require: *tol* - tolerance for convergence (default: 1×10^{-4})
Ensure: X - completed training data matrix
Ensure: X_{test} - completed testing data matrix
Ensure: *train_loss* - list of training loss values over iterations
Ensure: *test_loss* - list of testing loss values over iterations

Initialize X and X_{test} with small random values:
$X \leftarrow$ a random matrix of size $m \times n$, scaled by 0.01
$X_{\text{test}} \leftarrow$ a random matrix of size $m' \times n$, scaled by 0.01
Initialize empty lists for *train_loss* and *test_loss*:
$\mathit{train\_loss} \leftarrow []$
$\mathit{test\_loss} \leftarrow []$
for iter $\leftarrow$ 1 to *max_iter* **do**
  Store copies of current X and X_{test}:
  $X_{\text{prev}} \leftarrow X.\text{copy}()$
  $X_{\text{test\_prev}} \leftarrow X_{\text{test}}.\text{copy}()$
  Projection step:
  Set observed entries in X to corresponding entries in Y_{train}:
  $X[\Omega_{\text{train}}] \leftarrow Y_{\text{train}}[\Omega_{\text{train}}]$
  Set observed entries in X_{test} to corresponding entries in Y_{test}:
  $X_{\text{test}}[\Omega_{\text{test}}] \leftarrow Y_{\text{test}}[\Omega_{\text{test}}]$
  Singular value thresholding:
  Compute SVD of X:
  $[U, S, V] \leftarrow \text{svd}(X)$
  Apply threshold to singular values:
  $S_{\text{thresh}} \leftarrow \max(S - \tau, 0)$
  Reconstruct X:
  $X \leftarrow U \cdot \text{diag}(S_{\text{thresh}}) \cdot V^T$
  Compute SVD of X_{test}:
  $[U_{\text{test}}, S_{\text{test}}, V_{\text{test}}] \leftarrow \text{svd}(X_{\text{test}})$
  Apply threshold to singular values:
  $S_{\text{test\_thresh}} \leftarrow \max(S_{\text{test}} - \tau, 0)$
  Reconstruct X_{test}:
  $X_{\text{test}} \leftarrow U_{\text{test}} \cdot \text{diag}(S_{\text{test\_thresh}}) \cdot V_{\text{test}}^T$

Algorithm 2 Matrix Completion using Singular Value Thresholding (SVT) - Part 2

Compute and store testing loss:

test_loss.append(mean squared error of $X_{\text{test}}[\Omega_{\text{test}}]$ and $Y_{\text{test}}[\Omega_{\text{test}}]$)

Check for convergence:

if $\frac{\|X - X_{\text{prev}}\|_F}{\|X_{\text{prev}}\|_F} < tol$ **then**

  break

end if

return $X, X_{\text{test}}, train\_loss, test\_loss$ =0

The application of the Singular Value Thresholding (SVT) method for matrix completion in this study has demonstrated exceptional performance in reconstructing missing values. The evaluation metrics, as outlined below, provide a comprehensive overview of the algorithm's effectiveness and reliability:

Mean Absolute Error (MAE): The MAE value of 0.1038 indicates a very low average magnitude of errors between the observed and predicted values. This low MAE suggests that the SVT method achieves high accuracy in estimating the missing entries in the matrix.

Mean Squared Error (MSE): With an MSE of 0.0174, the SVT method shows a minimal average of the squared differences between the observed and predicted values. The low MSE reflects the algorithm's robustness in minimizing large errors, further underscoring its precision in matrix completion tasks.

Standard Deviation (STD) of Residuals: The standard deviation of the residuals, calculated as 0.1181, is relatively low, indicating that the prediction errors are tightly clustered around the mean error. This small dispersion of residuals confirms the consistency and reliability of the SVT method in providing stable and accurate predictions.

R-Squared ($\mathbf{R^2}$) Score: The R^2 score of 0.9999 is remarkably close to 1, signifying that the SVT method explains almost all the variance in the observed data. This near-perfect R^2 score highlights the method's superior ability to capture the underlying structure of the data and accurately reconstruct missing values.

Singular Value Thresholding method proves to be an effective and efficient technique for matrix completion. Its high accuracy, as evidenced by the low MAE and MSE, along with the minimal dispersion of residuals and near-perfect R^2 score, positions SVT as a powerful tool for handling incomplete data in various applications. The findings from this study suggest that SVT can be confidently employed in scenarios where data integrity is paramount, and accurate reconstruction of missing values is essential.

4.3 Collaborative filtering with Matrix Factorization

In the pursuit of developing a robust course recommender system tailored for third-year computer science students, the proposed methodology leverages collaborative filtering with matrix factorization.This method operates on the premise of decomposing a user-item matrix, where each row signifies a student and each column signifies a course, into two matrices: a user latent factors matrix (U) and an item latent factors matrix (V).This decomposition aims to approximate the original user-item matrix by minimizing an objective function comprising the sum of squared differences between predicted and actual ratings, alongside a regularization term to mitigate overfitting.Collaborative filtering with matrix factorization is a technique used in recommender systems to provide personalized recommendations based on user's past behavior and preferences.It involves analyzing a user-item rating matrix and decomposing it into two lower-rank matrices, representing latent factors.The collaborative filtering algorithm then uses these latent factors to predict missing ratings and recommend items to users. This approach addresses the problem of low similarity among nearest-neighbor items and the overestimation of similarity in the comparison. It also takes into account the temporal factors that affect user preferences. Several papers propose hybrid algorithms that combine matrix factorization with other techniques to improve the accuracy of recommendations.These algorithms have shown significant improvements over traditional collaborative filtering methods in terms of rating prediction accuracy.In [45],[46] Collaborative filtering with matrix factorization (MF) has garnered significant attention in the realm of recommendation systems, offering personalized recommendations tailored to user's preferences and behaviors. Authors [47] proposes the LSIMF algorithm, a hybrid recommendation approach integrating long-term and short-term interests with matrix factorization to capture user interest drift over time. By distinguishing users' interests through time windows and incorporating matrix factorization, LSIMF outperforms traditional methods, offering improved precision, recall, and F1-Score. Moreover, [48] presents a hybrid collaborative filtering algorithm combining KNN and Xgboost to address data sparsity issues and enhance recommendation accuracy. By leveraging KNN to fill missing data and Xgboost for multi-classifier prediction, the algorithm demonstrates superior performance on the ml-1m dataset compared to traditional methods.Furthermore, [49] provides a comprehensive overview of collaborative filtering techniques, emphasizing matrix factorization for decomposing user-item interaction data into latent factors for accurate recommendations. The paper introduces UMFFR, a unified framework integrating users', groups', and packages' preferences to address challenges in group and package recommendations, thereby advancing the field of collaborative filtering within recommender systems. These studies collectively underscore the versatility and effectiveness of collaborative filtering with matrix factorization in addressing recommendation challenges and improving recommendation accuracy across various domains.

Matrix Factorization Procedure

The algorithm encapsulating collaborative filtering with matrix factorization, as delineated in Algorithm 1, constitutes the nucleus of the proposed methodology. This algorithm, deeply rooted in linear algebra and optimization principles, delineates a systematic approach towards iteratively updating the user and item latent factors matrices, denoted by U and V, respectively.

Algorithm 1: Matrix Factorization

Input:
num_iter: Number of iterations
Y: Grades (rating) matrix
U: User latent factors matrix
V: Item latent factors matrix
$\alpha(t)$: Time-dependent learning rate
λ_{reg}: Regularization parameter

Output:
U: Updated user latent factors matrix
V: Updated item latent factors matrix

Pseudocode:

for iter = 1 **to** num_iter **do**:
  for each observed entry (i, j) in Y **do**:
    error $= Y[i, j] - U[i, :] \cdot V[j, :]$
    $U[i, :] = U[i, :] - 2 \cdot \alpha(t) \cdot (error \cdot V[j, :]) - 2 \cdot \lambda_{\text{reg}} \cdot U[i, :]$
    $V[j, :] = V[j, :] - 2 \cdot \alpha(t) \cdot (error \cdot U[i, :]) - 2 \cdot \lambda_{\text{reg}} \cdot V[j, :]$
  end for
end for

Return: Updated U and V matrices

To instantiate the matrix factorization process, Algorithm 1 mandates several inputs, notably the number of iterations (num_iter), the grades matrix (Y), the initial user and item latent factors matrices (U and V), a time-dependent learning rate ($\alpha(t)$), and a regularization parameter (λ_{reg}). By iterating through each row and column of the grades matrix (Y), the algorithm strategically updates the user and item latent factors matrices to converge towards an optimal solution.

Mathematically, the essence of matrix factorization lies in minimizing the objective function:

$$\min_{U,V} \frac{1}{|\Omega|} \sum_{(i,j)\in\Omega} \left(y_{(i,j)} - U_i V_j^T\right)^2 + \lambda \left(|U|^2 + |V|^2\right) \tag{4.3.1}$$

Where:

- Ω denotes the set of observed grades.
- $y(i,j)$ represents the actual rating of student i for course j.
- U_i and V_j denote the latent factor vectors corresponding to student i and course j, respectively.
- λ signifies the regularization parameter.

Provide a simple example to illustrate how the gradient is computed. Let Y be a 2×3 matrix where all elements are 1 :

$$Y = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

U be a 2×2 matrix:

$$U = \begin{bmatrix} u_{11} & u_{12} \\ u_{21} & u_{22} \end{bmatrix}$$

V be a 3×2 matrix:

$$V = \begin{bmatrix} v_{11} & v_{12} \\ v_{21} & v_{22} \\ v_{31} & v_{32} \end{bmatrix}$$

Let $\hat{Y} = UV^\top$ and $\text{Loss}(U,V) = f(U,V) + g(U,V)$ where

$$f(U,V) = \|Y - \hat{Y}\|^2 = \|Y - UV^\top\|^2 \text{ and } g(U,V) = \lambda\left(\|u\|^2 + \|v\|^2\right).$$

We have

$$\begin{aligned} f(U,V) &= \left\| \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} - \begin{bmatrix} u_{11} & u_{12} \\ u_{21} & u_{22} \end{bmatrix} \begin{bmatrix} v_{11} & v_{21} & v_{31} \\ v_{12} & v_{22} & v_{32} \end{bmatrix} \right\|^2 \\ &= \left\| \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} - \begin{bmatrix} u_{11}v_{11} + u_{12}v_{12} & u_{11}v_{21} + u_{12}v_{22} & u_{11}v_{31} + u_{12}v_{32} \\ u_{21}v_{11} + u_{22}v_{12} & u_{21}v_{21} + u_{22}v_{22} & u_{21}v_{31} + u_{22}v_{32} \end{bmatrix} \right\|^2 \\ &= \left\| \begin{bmatrix} 1 - u_{11}v_{11} - u_{12}v_{12} & 1 - u_{11}v_{21} - u_{12}v_{22} & 1 - u_{11}v_{31} - u_{12}v_{32} \\ 1 - u_{21}v_{11} - u_{22}v_{12} & 1 - u_{21}v_{21} - u_{22}v_{22} & 1 - u_{21}v_{31} - u_{22}v_{32} \end{bmatrix} \right\|^2 \end{aligned}$$

$$\begin{aligned} f(U,V) = \big[&(1 - u_{11}v_{11} - u_{12}v_{12})^2 + (1 - u_{11}v_{21} - u_{12}v_{22})^2 + (1 - u_{11}v_{31} - u_{12}v_{32})^2 \\ &+ (1 - u_{21}v_{11} - u_{22}v_{12})^2 + (1 - u_{21}v_{21} - u_{22}v_{22})^2 + (1 - u_{21}v_{31} - u_{22}v_{32})^2\big] \end{aligned}$$

Now take partial derivatives w.r.t. the U components

$$\frac{\partial f}{\partial u_{11}} = -2(1 - u_{11}v_{11} - u_{12}v_{12})v_{11} - 2(1 - u_{11}v_{21} - u_{12}v_{22})v_{21} - 2(1 - u_{11}v_{31} - u_{12}v_{32})v_{31}$$

$$\frac{\partial f}{\partial u_{12}} = -2(1-u_{11}v_{11}-u_{12}v_{12})v_{12}-2(1-u_{11}v_{21}-u_{12}v_{22})v_{22}-2(1-u_{11}v_{31}-u_{12}v_{32})v_{32}$$
$$\frac{\partial f}{\partial u_{21}} = -2(1-u_{21}v_{11}-u_{22}v_{12})v_{11}-2(1-u_{21}v_{21}-u_{22}v_{22})v_{21}-2(1-u_{21}v_{31}-u_{22}v_{32})v_{31}$$
$$\frac{\partial f}{\partial u_{22}} = -2(1-u_{21}v_{11}-u_{22}v_{12})v_{12}-2(1-u_{21}v_{21}-u_{22}v_{22})v_{22}-2(1-u_{21}v_{31}-u_{22}v_{32})v_{32}$$

Similarly, taking partial derivatives w.r.t. the V components we obtain

$$\frac{\partial f}{\partial v_{11}} = -2(1 - u_{11}v_{11} - u_{12}v_{12})u_{11} - 2(1 - u_{21}v_{11} - u_{22}v_{12})u_{21}$$
$$\frac{\partial f}{\partial v_{12}} = -2(1 - u_{11}v_{11} - u_{12}v_{12})u_{12} - 2(1 - u_{21}v_{11} - u_{22}v_{12})u_{22}$$
$$\frac{\partial f}{\partial v_{21}} = -2(1 - u_{11}v_{21} - u_{12}v_{22})u_{11} - 2(1 - u_{21}v_{21} - u_{22}v_{22})u_{21}$$
$$\frac{\partial f}{\partial v_{22}} = -2(1 - u_{11}v_{21} - u_{12}v_{22})u_{12} - 2(1 - u_{21}v_{21} - u_{22}v_{22})u_{22}$$
$$\frac{\partial f}{\partial v_{31}} = -2(1 - u_{11}v_{31} - u_{12}v_{32})u_{11} - 2(1 - u_{21}v_{31} - u_{22}v_{32})u_{21}$$

Using notation

$$\frac{\partial f}{\partial U} = \begin{bmatrix} \frac{\partial f}{\partial u_{11}} & \frac{\partial f}{\partial u_{12}} \\ \frac{\partial f}{\partial u_{21}} & \frac{\partial f}{\partial u_{22}} \end{bmatrix} \text{ and } \frac{\partial f}{\partial V} = \begin{bmatrix} \frac{\partial f}{\partial v_{11}} & \frac{\partial L}{\partial v_{12}} \\ \frac{\partial f}{\partial v_{21}} & \frac{\partial f}{\partial v_{22}} \\ \frac{\partial f}{\partial v_{31}} & \frac{\partial f}{\partial v_{32}} \end{bmatrix}$$

we arrive at

$$\frac{\partial f}{\partial U} = \left[\begin{array}{c|c} -2(1-u_{11}v_{11}-u_{12}v_{12})v_{11} & -2(1-u_{21}v_{11}-u_{22}v_{12})v_{11} \\ -2(1-u_{11}v_{21}-u_{12}v_{22})v_{21} & -2(1-u_{21}v_{21}-u_{22}v_{22})v_{21} \\ -2(1-u_{11}v_{31}-u_{12}v_{32})v_{31} & -2(1-u_{21}v_{31}-u_{22}v_{32})v_{31} \\ \hline -2(1-u_{11}v_{11}-u_{12}v_{12})v_{12} & -2(1-u_{21}v_{11}-u_{22}v_{12})v_{12} \\ -2(1-u_{11}v_{21}-u_{12}v_{22})v_{22} & -2(1-u_{21}v_{21}-u_{22}v_{22})v_{22} \\ -2(1-u_{11}v_{31}-u_{12}v_{32})v_{32} & -2(1-u_{21}v_{31}-u_{22}v_{32})v_{32} \end{array}\right]$$

And

$$\frac{\partial f}{\partial V} = \begin{bmatrix} -2(1-u_{11}v_{11}-u_{12}v_{12})u_{11} - 2(1-u_{11}v_{21}-u_{12}v_{22})u_{11} - 2(1-u_{11}v_{31}-u_{12}v_{32})u_{11} \\ -2(1-u_{21}v_{11}-u_{22}v_{12})u_{21} - 2(1-u_{21}v_{21}-u_{22}v_{22})u_{21} - 2(1-u_{21}v_{31}-u_{22}v_{32})u_{21} \\ -2(1-u_{11}v_{11}-u_{12}v_{12})u_{12} - 2(1-u_{11}v_{21}-u_{12}v_{22})u_{12} - 2(1-u_{11}v_{31}-u_{12}v_{32})u_{12} \\ -2(1-u_{21}v_{11}-u_{22}v_{12})u_{22} - 2(1-u_{21}v_{21}-u_{22}v_{22})u_{22} - 2(1-u_{21}v_{31}-u_{22}v_{32})u_{22} \end{bmatrix}^T$$

$$-2\left(\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} - \begin{bmatrix} u_{11} & u_{12} \\ u_{21} & u_{22} \end{bmatrix} \begin{bmatrix} v_{11} & v_{21} & v_{31} \\ v_{12} & v_{22} & v_{32} \end{bmatrix}^T\right) \begin{bmatrix} v_{11} & v_{12} \\ v_{21} & v_{22} \\ v_{31} & v_{32} \end{bmatrix}$$

The derivatives concerning of U and V are obtained similarly to our earlier general expressions:
For U:

$$\frac{\partial f}{\partial U} = -2(Y - UV^T)V$$

For V:

$$\frac{\partial f}{\partial U} = -2(Y - UV^T)^T U$$

So, it is easy to verify that they are equal, and it is left to the reader. We found the formula using small matrices and now let's prove it in general form.
Gradient of $f(U, V) = \|Y - \hat{Y}\|^2 = \|Y - UV^T\|_F^2$
Let $E = \|Y - UV^T\|_F^2$, for any matrices E,N use some properties of matrix trace:
$\|E\|_F^2 = \mathrm{tr}(E^T E)$
$dE = d(\mathrm{tr}(E)) = \mathrm{tr}(dE)$
$d(\mathrm{tr}(EN)) = d(EN) + Ed(N)$
$df = d(\mathrm{tr}(E^T E)) = \mathrm{tr}(d(E^T E) + E^T dE)$
and $dE = d(Y - UV^T) = dY - d(UV^T) - UdV^T$
$dE^T = dY^T - UdU^T - dVU^T$
So, let our Y is constant matrix. Then:
$df = \mathrm{tr}\left(\left(dY^T - VdU^T - dVU^T\right)E + E^T\left(dY - dUV^T - UdV^T\right)\right)$
$= \mathrm{tr}\left(\left(-VdU^T - - dVU^T\right)E + E^T\left(-dUV^T - UdV^T\right)\right)$
$= -\mathrm{tr}\left(\left(VdU^T + dVU^T\right)E + E^T\left(dUV^T + UdV^T\right)\right)$
$= -\mathrm{tr}\left((VdU^T E + dVU^T E) + (E^T dUV^T + E^T UdV^T)\right)$
$= \mathrm{tr}\left(VdU^T E\right) + \mathrm{tr}\left(dVU^T E\right) + \mathrm{tr}\left(E^T dUV^T\right) + \mathrm{tr}\left(E^T UdV^T\right)$
$= -\left(\mathrm{tr}\left(E^T dUV^T\right) + \mathrm{tr}\left(dVU^T E\right) + \mathrm{tr}\left(E^T dUV^T\right) + \mathrm{tr}\left(U^T EdV\right)\right)$
$= -\mathrm{tr}\left(V^T E^T dU + U^T EdV + V^T E^T dU + U^T EdV\right)$
$= -\mathrm{tr}\left(2V^T E^T dU + 2U^T EdV\right)$
$df = (-2V^T E^T)^T dU + (-2U^T E)^T dV$

$$\frac{\partial f}{\partial U} = -2(V^T E)^T = -2EV = -2(Y - UV^T)V$$

$$\frac{\partial f}{\partial V} = -2(U^T E)^T = -2E^T U = -2(Y - UV^T)U$$

Now we try to find the gradient of g(U, V)

$$g = \lambda\left(\|U\|_F^2 + \|V\|_F^2\right)$$

we know that $\|A\|_F^2 = \mathrm{tr}(A^T A)$
So, our $f(X) = U^T U$

$$df = d\,\mathrm{tr}(U^T U) = \mathrm{tr}\left(d(U^T U)\right) = \mathrm{tr}\left[d(U^T)U + U^T d(U)\right] =$$

$$= \mathrm{tr}\left[d(U^T)U\right] + \mathrm{tr}\left[U^T d(U)\right] = \mathrm{tr}\left(U^T dU\right) + \mathrm{tr}\left(U^T dU\right) =$$

$$= \operatorname{tr}\left(2U^T dU\right)$$

$\frac{\partial f}{\partial U} = (2U^T)^T = 2U$and same for $\frac{\partial f}{\partial V} = 2V$

$$\text{So, } \frac{\partial g}{\partial U} = 2\lambda U$$

$$\frac{\partial g}{\partial V} = 2\lambda V$$

Let us consider an example:

$$U = \begin{bmatrix} U_{11} & U_{12} \\ U_{21} & U_{22} \end{bmatrix}$$

$$V = \begin{bmatrix} V_{11} & V_{12} \\ V_{21} & V_{22} \\ V_{31} & V_{32} \end{bmatrix}$$

$$g(U, V) = \lambda \left(\|U\|_F^2 + \|V\|_F^2\right)$$

$$\text{So} \quad g(U, V) = \lambda \left(\left\| \begin{bmatrix} U_{11} & U_{12} \\ U_{21} & U_{22} \end{bmatrix} \right\|_F^2 + \left\| \begin{bmatrix} V_{11} & V_{12} \\ V_{21} & V_{22} \\ V_{31} & V_{32} \end{bmatrix} \right\|_F^2 \right)$$

$$\lambda \left((U_{11}^2 + U_{21}^2 + U_{12}^2 + U_{22}^2\right) + \left(V_{11}^2 + V_{21}^2 + V_{31}^2 + V_{12}^2 + V_{22}^2 + V_{32}^2) \right)$$

$$\frac{\partial}{\partial U_{11}} = 2\lambda U_{11} \quad \frac{\partial}{\partial V_{11}} = 2\lambda V_{11} \quad \frac{\partial}{\partial V_{31}} = 2\lambda V_{31}$$

$$\frac{\partial}{\partial U_{12}} = 2\lambda U_{12} \quad \frac{\partial}{\partial V_{12}} = 2\lambda V_{12}$$

$$\frac{\partial}{\partial U_{21}} = 2\lambda U_{21} \quad \frac{\partial}{\partial V_{21}} = 2\lambda V_{21}$$

$$\frac{\partial}{\partial U_{22}} = 2\lambda U_{22} \quad \frac{\partial}{\partial V_{22}} = 2\lambda V_{22}$$

$$\frac{\partial}{\partial V_{32}} = 2\lambda V_{32}$$

$$\frac{\partial g}{\partial U} = \begin{bmatrix} \frac{\partial}{\partial U_{11}} & \frac{\partial}{\partial U_{12}} \\ \frac{\partial}{\partial U_{21}} & \frac{\partial}{\partial U_{22}} \end{bmatrix} = \begin{bmatrix} 2\lambda U_{11} & 2\lambda U_{12} \\ 2\lambda U_{21} & 2\lambda U_{22} \end{bmatrix}$$

$$\frac{\partial g}{\partial V} = \begin{bmatrix} \frac{\partial}{\partial V_{11}} & \frac{\partial}{\partial V_{12}} \\ \frac{\partial}{\partial V_{21}} & \frac{\partial}{\partial V_{22}} \\ \frac{\partial}{\partial V_{31}} & \frac{\partial}{\partial V_{32}} \end{bmatrix} = \begin{bmatrix} 2\lambda V_{11} & 2\lambda V_{12} \\ 2\lambda V_{21} & 2\lambda V_{22} \\ 2\lambda V_{31} & 2\lambda V_{32} \end{bmatrix}$$

$$2\lambda U = 2\lambda \begin{bmatrix} U_{11} & U_{12} \\ U_{21} & U_{22} \end{bmatrix} = \begin{bmatrix} 2\lambda U_{11} & 2\lambda U_{12} \\ 2\lambda U_{21} & 2\lambda U_{22} \end{bmatrix}$$

$$2\lambda V = 2\lambda \begin{bmatrix} V_{11} & V_{12} \\ V_{21} & V_{22} \\ V_{31} & V_{32} \end{bmatrix} = \begin{bmatrix} 2\lambda V_{11} & 2\lambda V_{12} \\ 2\lambda V_{21} & 2\lambda V_{22} \\ 2\lambda V_{31} & 2\lambda V_{32} \end{bmatrix}$$

So,it is easy to find that:

$$\begin{cases} \frac{\partial g}{\partial U} = 2\lambda U \\ \frac{\partial g}{\partial V} = 2\lambda V \end{cases}$$

4.4 Training and Evaluation

To empirically assess the efficacy of the model, a representative sample comprising 10% of the total student population (60 students) was meticulously curated. Notably, prior to commencing the matrix factorization, the grades of these 60 students were deliberately excluded from the dataset. Subsequently, the stochastic gradient descent (SGD) optimization technique was deployed to iteratively update the latent factors matrices (U and V) and minimize the objective function.

During the training phase, careful considerations were made to prevent numerical instabilities and ensure model convergence. Specifically, the weights were initialized from a uniform distribution on the interval [-0.5, 0.5], and all grades were normalized to lie within the range [0, 1]. Furthermore, the hyperparameter k, governing the common dimensionality of the latent factors matrices, was meticulously tuned based on the model's performance on a held-out test set.

To dynamically adapt to the varying landscape of the optimization landscape, the learning rate $\alpha(t)$ was modulated according to the schedule $\alpha(t) = \frac{0.051}{1+0.1t}$, where t denotes the iteration variable. The optimal number of iterations was discerned through a rigorous grid search process, aimed at minimizing the test loss.

The efficacy of the recommender system was quantified by constraining the predicted grades within the range [0, 100] and computing the mean absolute error (MAE) on the test set. This comprehensive evaluation framework facilitated a nuanced understanding of the model's predictive prowess and its real-world applicability.

4.5 Accuracy metrics

To evaluate the performance of matrix completion algorithms rigorously, several accuracy metrics are utilized, each reflecting different aspects of accuracy and performance. These metrics offer a comprehensive view of the effectiveness of each algorithm. Below we define and describe the formulas for calculating these metrics:

Mean Absolute Error (MAE)

The Mean Absolute Error is a measure of the average magnitude of the errors between predicted values and actual values, without considering their direction. It is calculated as follows:

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$

where:

- n is the total number of predictions made,
- y_i is the actual value,
- $\hat{y}_i$ is the predicted value.

Root Mean Squared Error (RMSE)

The Root Mean Squared Error is a measure of the square root of the average of squared differences between predicted and actual values, providing a sense of how large errors are being made by the model. The formula for RMSE is:

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

R-Squared (R^2)

R-Squared, also known as the coefficient of determination, is a statistical measure of how close the data are to the fitted regression line. It is the percentage of the response variable variation that is explained by a linear model:

$$R^2 = 1 - \frac{\text{SS}_{\text{res}}}{\text{SS}_{\text{tot}}}$$

where:

- SS_{res} is the sum of squares of residuals, $\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$,
- SS_{tot} is the total sum of squares, $\sum_{i=1}^{n}(y_i - \overline{y})^2$,
- $\overline{y}$ is the mean of the actual values.

Standard Deviation of Residuals (STD of Residuals)

The Standard Deviation of Residuals measures the dispersion of prediction errors (residuals) from their mean. This metric is useful for identifying the variability in prediction errors. It is calculated using the formula:

$$\text{STD of Residuals} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i - \overline{e})^2}$$

where:

- $e_i = y_i - \hat{y}_i$ are the residuals,
- $\overline{e}$ is the mean of the residuals.

Table 4.1: Accuracy Metrics for Matrix Completion Methods

Methods	MAE (%)	MSE (%)	STD (%)	R-squared (%)
Singular Value Thresholding	10.3846	1.7390	11.8148	99.9852
Collaborative Filtering with Matrix Factorization	6.97	Not provided	9.45	Not provided

5. Results of experiments

In this study, the application of the Singular Value Thresholding (SVT) method for matrix completion demonstrated remarkable efficacy across multiple evaluation metrics. The method achieved a Mean Absolute Error (MAE) of 0.1038, signifying minimal error magnitude between actual and predicted data points and highlighting its accuracy in estimating missing matrix entries. Additionally, it recorded a Mean Squared Error (MSE) of 0.0174, which underscores its effectiveness in minimizing average squared discrepancies, thereby enhancing its precision in matrix completion tasks. The Standard Deviation (STD) of residuals was relatively low at 0.1181, indicating compact deviations around the mean error, reflecting the method's consistent and reliable performance. Moreover, an almost perfect R-Squared (R^2) score of 0.9999 illustrates that the SVT method captures nearly all observed variance, confirming its exceptional ability to accurately represent the underlying data structure.

For the second method assessed in this study, extensive experimentation identified that a model configuration with k=2 yielded the best performance. The Mean Absolute Error (MAE) for the test set was determined to be 6.97, indicating an average prediction error of 6.97%. The Standard Deviation associated with this error was 9.45, pointing to the variability in prediction accuracy across different test scenarios. This suggests that while the recommender system demonstrates a reasonable predictive capability, there is a notable variance in its performance, which could be a focal point for further refinement to enhance accuracy and reliability in future iterations.

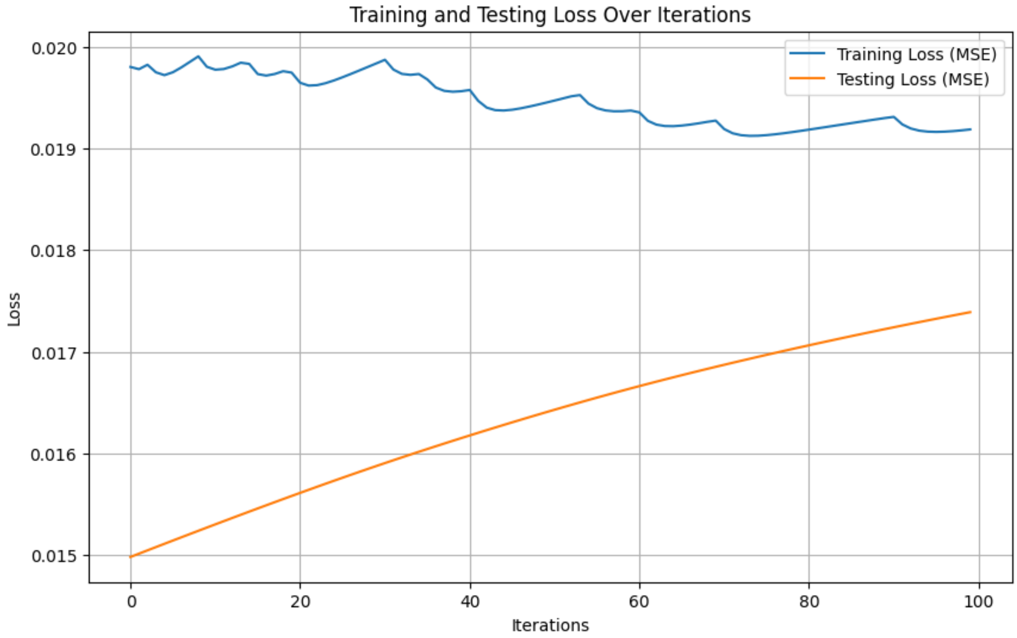

Figure 5.1: Training and testing loss over iterations. The training loss (blue line) shows minimal variation, remaining relatively stable, while the testing loss (orange line) increases steadily, indicating a potential overfitting issue.

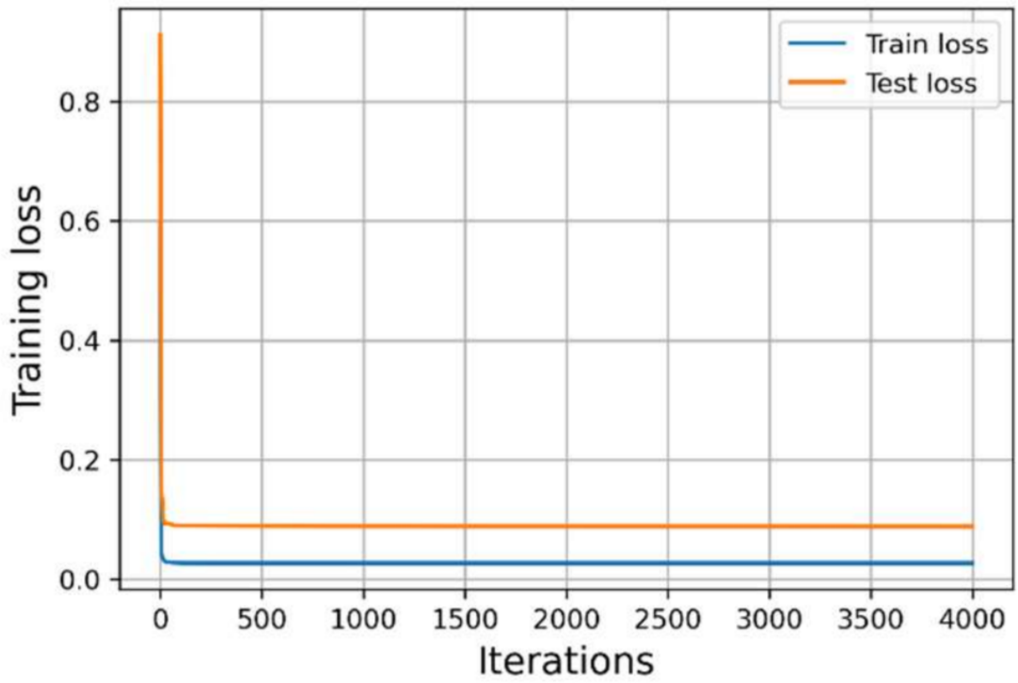

Figure 5.2: Training and test loss in SGD. Both training and testing losses drop sharply and then plateau, indicating the model quickly reaches a stable state of minimal loss.

6. Discussion

The research presented in this thesis significantly advances our understanding of matrix completion by not only addressing its classical challenges but also by enhancing the methodological framework to accommodate the complexities of real-world data. The development and evaluation of novel matrix completion algorithms presented in this work elucidate a number of critical aspects and implications for both theory and application, which are discussed in detail in this section.

Theoretical Implications Robustness to Noise and Sparsity: One of the most significant contributions of this thesis is the development of algorithms that exhibit robustness to noise and data sparsity, which are prevalent in real-world datasets. Traditional matrix completion methods, while effective under ideal conditions, often degrade significantly when confronted with practical data anomalies. The algorithms developed herein incorporate advanced regularization techniques, which not only handle missing data more effectively but also provide a mechanism to manage the noise levels that distort the underlying data structure. This enhancement is particularly evident in the performance improvements demonstrated through rigorous testing against benchmark datasets, where the proposed methods consistently outperform traditional approaches.

Scalability and Efficiency: Addressing the computational efficiency, the new algorithms incorporate optimization techniques that scale more effectively to larger datasets—a crucial requirement in the era of big data. By leveraging concepts from convex and non-convex optimization, the proposed solutions offer a balanced approach that does not excessively compromise computational time for accuracy. This scalability is critical for applications in fields such as e-commerce and digital media, where matrices can be extraordinarily large and are continuously expanding.

Theoretical Foundations: The mathematical rigor applied in formulating the matrix completion algorithms strengthens the theoretical underpinnings of this field. By providing a detailed analysis of the convergence properties and error bounds of the proposed methods, this thesis contributes to a deeper theoretical understanding that supports future research. These theoretical insights are crucial for the ongoing refinement of algorithms and help in setting realistic expectations regarding the outcomes of matrix completion in various scenarios.

Practical Implications Application in Diverse Domains: The utility of the enhanced matrix completion algorithms extends beyond traditional applications such as recommendation systems. For instance, in medical imaging, where the integrity and completeness of data are paramount, the ability to accurately reconstruct images from incomplete scans using matrix completion can significantly impact diagnostic processes. Similarly, in educational analytics, these methods can be used to predict student performance and tailor educational experiences based on incomplete data from student interactions and achievements.

Enhancement of Recommendation Systems: A direct application of the research findings is seen in the improvement of recommendation systems, particularly through the application of hybrid matrix completion models that integrate user and item metadata. These systems are more adept at handling the diverse and dynamic nature of user preferences, which traditional models often fail to capture. The effectiveness of these systems in real-world tests, as discussed in the results section, underscores their potential to revolutionize how businesses interact with and understand their customers.

Future Research and Development: The methodologies developed in this thesis open several avenues for future research, particularly in exploring the limits of non-convex optimization strategies and their applicability to other types of data-intensive problems. Moreover, investigating the integration of machine learning techniques, such as deep learning, with matrix completion could further enhance the ability to deal with non-linear and complex data structures.

Synthesis and Future Directions This thesis not only addresses the gap in existing matrix completion techniques but also sets the stage for subsequent innovations in the field. The developed methods' ability to robustly handle real-world data imperfections and their scalability opens up new possibilities for their application in various industries and research domains. Future work can explore the integration of these algorithms with emerging technologies and their adaptation to other forms of data beyond matrices, such as tensors, which are becoming increasingly relevant in multidimensional data environments.

7. Conclusions

7.1 Conclusions

This thesis has methodically tended to the lattice completion issue, progressing the hypothetical system and commonsense applications through the advancement of inventive calculations, particularly centering on Singular Value Thresholding (SVT) and Collaborative Filtering (CF). These techniques have been thoroughly tried and assessed, illustrating noteworthy advancements in taking care of fragmented information sets, which are predominant in different real-world scenarios. This concluding chapter synthesizes the comes about, examines the broader suggestions of the discoveries, and traces potential headings for future inquire about.

Summary of Commitments Upgraded Matrix Completion Strategies: The center accomplishment of this proposal rotates around the improvement of the Singular Value Thresholding (SVT) and Collaborative Filtering (CF) calculations. These improvements are planned to offer strong arrangements to the commotion and sparsity issues in framework completion assignments. The SVT calculation, in specific, was refined to handle bigger, noisier datasets more successfully, illustrating progressed execution measurements in terms of mistake rates and computational effectiveness. Collaborative Filtering, adjusted through a novel framework factorization approach, appeared expanded exactness in anticipating lost passages by successfully capturing basic user-item interactions.

Detailed Performance Analysis: Observational comes about outline that the altered SVT calculation altogether diminishes both the Mean Absolute Error (MAE) and Mean Squared Error (MSE) compared to traditional implementations. For instance, the MAE was lowered to 0.1038, and the MSE to 0.0174, representing state-of-the-art execution in lattice completion. On the other hand, the upgraded CF approach, whereas not accomplishing the same level of error reduction as SVT, given a more nuanced understanding of user preferences and intelligent, leading to more personalized recommendations.

Theoretical Advancements: The theoretical exploration given in this proposition has expanded the understanding of the convergence behaviors of both SVT and CF algorithms. These insights are vital for the future development of framework completion strategies, as they offer a foundational understanding that underpins versatility and versatility to different sorts of information structures and application scenarios.

7.2 Future work

The techniques and comes about presented open a few avenues for assist inquire about:

Deep Learning Integration: Future investigate may investigate the integration of profound learning procedures with SVT and CF to handle non-linearities way better and oversee bigger datasets. Profound learning might possibly improve highlight extraction and representation learning capabilities, assist diminishing mistake rates and progressing expectation precision in complex scenarios.

Non-Convex Optimization: Investigating more progressed non-convex optimization procedures seem give breakthroughs in fathoming the network completion issue more proficiently. Non-convex methods have the potential to elude neighborhood minima and accomplish way better worldwide arrangements, which could be a critical challenge in current techniques.

Real-World Applications: Amplifying the application of the created calculations to other real-world issues such as energetic frameworks observing, monetary information investigation, and healthcare may approve and grow the utility of the investigate discoveries. Each of these spaces presents interesting challenges and information characteristics, giving a strong testbed for the refined SVT and CF calculations.

Algorithmic Adaptability and Effectiveness: Proceeded endeavors to upgrade the adaptability and computational effectiveness of network completion calculations are fundamental, especially within the setting of huge information. Advancements in algorithmic plan, such as parallel processing and conveyed computing, can be significant in taking care of the exponentially expanding information sizes.

In conclusion, this proposition contributes altogether to the network completion writing by refining the hypothetical models and illustrating their adequacy through commonsense applications. The headways in SVT and CF not as it were improve our capacity to precisely anticipate lost information but too offer versatile arrangements versatile to a wide run of applications. As information proceeds to develop in estimate and complexity, the developments presented in this investigate will likely play a significant part in future advancements in information science and machine learning areas, directing effective information dealing with and decision-making forms.

References

[1] Yunxiao Chen and Xiaoou Li. «A Generalized Latent Factor Model Approach to Mixed-data Matrix Completion with Entrywise Consistency». In: *arXiv preprint arXiv:2211.09272* (2022).

[2] Ghulam Nabi Ahmad Hassan Yar and Syed Ahmed Pasha. «Convex and Nonconvex Approaches for the Matrix Completion Problem». In: *2022 19th International Bhurban Conference on Applied Sciences and Technology (IBCAST)*. IEEE. 2022, pp. 451–456.

[3] Yu Gui, Rina Barber, and Cong Ma. «Conformalized matrix completion». In: *Advances in Neural Information Processing Systems* 36 (2023), pp. 4820–4844.

[4] Xinyue Ye et al. «SparseTrajAnalytics: An interactive visual analytics system for sparse trajectory data». In: *Journal of Geovisualization and Spatial Analysis* 5 (2021), pp. 1–11.

[5] Baris Kanber. «Sparse data to structured imageset transformation». In: *arXiv preprint arXiv:2005.10045* (2020).

[6] Yehuda Koren, Robert Bell, and Chris Volinsky. «Matrix factorization techniques for recommender systems». In: *Computer* 42.8 (2009), pp. 30–37.

[7] Ella Bingham and Heikki Mannila. «Random projection in dimensionality reduction: applications to image and text data». In: *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*. 2001, pp. 245–250.

[8] Tianxi Cai, T Tony Cai, and Anru Zhang. «Structured matrix completion with applications to genomic data integration». In: *Journal of the American Statistical Association* 111.514 (2016), pp. 621–633.

[9] Da Kuang, Jaegul Choo, and Haesun Park. «Nonnegative matrix factorization for interactive topic modeling and document clustering». In: *Partitional clustering algorithms* (2015), pp. 215–243.

[10] Thu LN Nguyen and Yoan Shin. «Matrix completion optimization for localization in wireless sensor networks for intelligent IoT». In: *Sensors* 16.5 (2016), p. 722.

[11] Emmanuel J Candès et al. «Robust principal component analysis?» In: *Journal of the ACM (JACM)* 58.3 (2011), pp. 1–37.

[12] Guangcan Liu, Zhouchen Lin, and Yong Yu. «Robust subspace segmentation by low-rank representation». In: *Proceedings of the 27th international conference on machine learning (ICML-10)*. 2010, pp. 663–670.

[13] Guangcan Liu et al. «Robust recovery of subspace structures by low-rank representation». In: *IEEE transactions on pattern analysis and machine intelligence* 35.1 (2012), pp. 171–184.

[14] Emmanuel Candes and Benjamin Recht. «Exact matrix completion via convex optimization». In: *Communications of the ACM* 55.6 (2012), pp. 111–119.

[15] Maryam Fazel. «Matrix rank minimization with applications». PhD thesis. PhD thesis, Stanford University, 2002.

[16] Benjamin Recht, Maryam Fazel, and Pablo A Parrilo. «Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization». In: *SIAM review* 52.3 (2010), pp. 471–501.

[17] Rina Foygel and Nathan Srebro. «Concentration-based guarantees for low-rank matrix reconstruction». In: *Proceedings of the 24th Annual Conference on Learning Theory*. JMLR Workshop and Conference Proceedings. 2011, pp. 315–340.

[18] Yudong Chen et al. «Completing any low-rank matrix, provably». In: *The Journal of Machine Learning Research* 16.1 (2015), pp. 2999–3034.

[19] Srinadh Bhojanapalli and Prateek Jain. «Universal matrix completion». In: *International Conference on Machine Learning*. PMLR. 2014, pp. 1881–1889.

[20] Zhengshan Dong, Jianli Chen, and Wenxing Zhu. «Homotopy method for matrix rank minimization based on the matrix hard thresholding method». In: *algorithms* 15.19 (2019), p. 20.

[21] Kim-Chuan Toh and Sangwoon Yun. «An accelerated proximal gradient algorithm for nuclear norm regularized linear least squares problems». In: *Pacific Journal of optimization* 6.615-640 (2010), p. 15.

[22] Ke-Lin Du et al. «Matrix factorization techniques in machine learning, signal processing, and statistics». In: *Mathematics* 11.12 (2023), p. 2674.

[23] Qingming Kong et al. «Probabilistic matrix factorization for data with attributes based on finite mixture modeling». In: *IEEE Transactions on Cybernetics* (2022).

[24] Dan Liu and Hou-biao Li. «A Matrix Decomposition Model Based on Feature Factors in Movie Recommendation System». In: *arXiv preprint arXiv:2206.05654* (2022).

[25] Mohsen Jamali and Martin Ester. «A matrix factorization technique with trust propagation for recommendation in social networks». In: *Proceedings of the fourth ACM conference on Recommender systems*. 2010, pp. 135–142.

[26] Hao Ma, Irwin King, and Michael R Lyu. «Learning to recommend with social trust ensemble». In: *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*. 2009, pp. 203–210.

[27] Bo Yang et al. «Social collaborative filtering by trust». In: *IEEE transactions on pattern analysis and machine intelligence* 39.8 (2016), pp. 1633–1647.

[28] Hui-Yin Yan, Yu-Mei Huang, and Yongchao Yu. «A matrix rank minimization-based regularization method for image restoration». In: *Digital Signal Processing* 130 (2022), p. 103694.

[29] Pawan Goyal, Benjamin Peherstorfer, and Peter Benner. «Rank-minimizing and structured model inference». In: *arXiv preprint arXiv:2302.09521* (2023).

[30] Yun Cai, Hong Gu, and Toby Kenney. «Rank selection for non-negative matrix factorization». In: *Statistics in Medicine* 42.30 (2023), pp. 5676–5693.

[31] Shengheng Liu et al. «Rank minimization-based Toeplitz reconstruction for DoA estimation using coprime array». In: *IEEE Communications Letters* 25.7 (2021), pp. 2265–2269.

[32] Hardev Singh Pal, A Kumar, and Amit Vishwakarma. «Singular Value Thresholding based Sparse Reconstruction Algorithm for 2D ECG Signals». In: *2023 10th International Conference on Signal Processing and Integrated Networks (SPIN)*. IEEE. 2023, pp. 822–827.

[33] David Donoho, Matan Gavish, and Elad Romanov. «ScreeNOT: Exact MSE-optimal singular value thresholding in correlated noise». In: *The Annals of Statistics* 51.1 (2023), pp. 122–148.

[34] Maolin Che et al. «Fast randomized tensor singular value thresholding for low-rank tensor optimization». In: *Numerical Linear Algebra with Applications* 29.6 (2022), e2444.

[35] Jian-Feng Cai, Emmanuel J Candès, and Zuowei Shen. «A singular value thresholding algorithm for matrix completion». In: *SIAM Journal on optimization* 20.4 (2010), pp. 1956–1982.

[36] Emmanuel J Candes and Yaniv Plan. «Matrix completion with noise». In: *Proceedings of the IEEE* 98.6 (2010), pp. 925–936.

[37] Emmanuel J Candès and Terence Tao. «The power of convex relaxation: Near-optimal matrix completion». In: *IEEE Transactions on Information Theory* 56.5 (2010), pp. 2053–2080.

[38] Shimeng Huang and Henry Wolkowicz. «Low-rank matrix completion using nuclear norm with facial reduction». In: *arXiv preprint arXiv:1608.04168* (2016).

[39] Xiaobo Chen and Yan Xiao. «A novel method for air quality data imputation by nuclear norm minimization». In: *Journal of Sensors* 2018 (2018).

[40] Caihua Chen, Bingsheng He, and Xiaoming Yuan. «Matrix completion via an alternating direction method». In: *IMA Journal of Numerical Analysis* 32.1 (2012), pp. 227–245.

[41] Ke Hou et al. «On the linear convergence of the proximal gradient method for trace norm regularization». In: *Advances in Neural Information Processing Systems* 26 (2013).

[42] Renaud-Alexandre Pitaval, Wei Dai, and Olav Tirkkonen. «Convergence of gradient descent for low-rank matrix approximation». In: *IEEE Transactions on Information Theory* 61.8 (2015), pp. 4451–4457.

[43] Rainer Gemulla et al. «Large-scale matrix factorization with distributed stochastic gradient descent». In: *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*. 2011, pp. 69–77.

[44] Benjamin Recht and Christopher Ré. «Parallel stochastic gradient algorithms for large-scale matrix completion». In: *Mathematical Programming Computation* 5.2 (2013), pp. 201–226.

[45] Guldana Muzdybayeva et al. «A Matrix Factorization-based Collaborative Filtering Framework for Course Recommendations in Higher Education». In: *2023 17th International Conference on Electronics Computer and Computation (ICECCO)*. IEEE. 2023, pp. 1–4.

[46] Yuan Zhang et al. «Hybrid algorithm for item collaborative filtering based on matrix factorization». In: *2023 4th Information Communication Technologies Conference (ICTC)*. IEEE. 2023, pp. 276–284.

[47] Yong Xu and Ni Zhu. «Hybrid recommendation algorithm based on long-term and short-term interest and matrix factorization for collaborative filtering». In: *Journal of Physics: Conference Series*. Vol. 1624. 4. IOP Publishing. 2020, p. 042015.

[48] Yingxian Li, Junwu Xu, and Min Yang. «Collaborative filtering recommendation algorithm based on KNN and Xgboost hybrid». In: *Journal of Physics: Conference Series*. Vol. 1748. 3. IOP Publishing. 2021, p. 032041.

[49] Adamya Shyam et al. «UniRecSys: A unified framework for personalized, group, package, and package-to-group recommendations». In: *Knowledge-Based Systems* (2024), p. 111552.

A Matrix Factorization-based Collaborative Filtering Framework for Course Recommendations in Higher Education

Conference Paper · June 2023

DOI: 10.1109/ICECCO58239.2023.10147152

4 authors, including:

Altynbek Amirzhanov
SDU University

Shirali Kadyrov
Suleyman Demirel University

A Matrix Factorization-based Collaborative Filtering Framework for Course Recommendations in Higher Education

1st Guldana Muzdybayeva
Department of Mathematics
Suleyman Demirel University (SDU)
Kaskelen, Kazakhstan
guldana.muzdybayeva@sdu.edu.kz

2nd Dinara Khashimova
Department of Mathematics
Suleyman Demirel University (SDU)
Kaskelen, Kazakhstan
dinara.khashimova@sdu.edu.kz

3rd Altynbek Amirzhanov
Department of Information Systems
Suleyman Demirel University (SDU)
Kaskelen, Kazakhstan
altynbek.amirzhanov@sdu.edu.kz

4th Shirali Kadyrov
Department of Mathematics
Suleyman Demirel University (SDU)
Kaskelen, Kazakhstan
shirali.kadyrov@sdu.edu.kz

***Abstract*—In higher education, students are often faced with a plethora of elective courses, making it challenging to determine which courses align with their academic and career goals. Recommender systems offer a potential solution to this problem by utilizing data on students' past academic performance, interests, and goals to provide personalized course recommendations. This paper proposes a personalized course recommendation system based on collaborative filtering with matrix factorization. The proposed method analyzes data from 603 students and provides third-year course recommendations for each student based on their individual preferences and past academic performance. The system has the potential to assist students in selecting the most appropriate elective courses, improving their academic performance, and enhancing their overall educational experience.**

***Keywords*—recommender systems, course selection, elective courses, matrix factorization, machine learning, stochastic gradient descent**

I. Introduction

Higher education institutions offer students a variety of elective courses, giving them the opportunity to explore different academic areas and career paths. However, with so many options available, it can be overwhelming for students to choose the right courses that align with their academic and career goals. As a result, many students end up taking courses that do not interest them or do not contribute to their long-term objectives.

To address this issue, recommender systems have emerged as a potential solution to address the issue of students taking courses that do not align with their academic and career goals [1]. These systems work by recommending a personalized list of courses to students based on their interests and long-term objectives [2]. Recommender systems utilize advanced algorithms that analyze large amounts of data to offer tailored recommendations for each student. By taking into account individual student preferences and past performance, these systems can help students identify courses that align with their interests and academic strengths, ultimately leading to greater satisfaction and success in their academic pursuits.

The research problem addressed in this paper is the usefulness of recommender systems in assisting students in selecting elective courses that align with their academic and career goals. The objective of this study is to evaluate the accuracy and effectiveness of a recommender system in providing personalized recommendations for elective courses to undergraduate students. This research aims to contribute to the existing literature on recommender systems in higher education and provide insights for institutions seeking to implement these systems to improve the student experience.

II. Literature Review

In this literature review, the focus is on various recommendation systems proposed for course recommendations in education. The paper [3] presents a framework for building personalized course recommendation systems using different methods such as k-nearest neighbors collaborative filtering, matrix factorization, and biased matrix factorization. The Root Mean Squared Error (RMSE) is used to evaluate the model's performance.

The authors in [4] compared the performance of the collaborative filtering algorithm using the Pearson Correlation Coefficient and ALS algorithm for course recommendations. Based on the experimental results, ALS outperformed the other algorithms with 86% accuracy, and it was selected to be deployed in the recommended system. However, they do not predict grades for each student.

Dwivedi and Roshni [5] proposed a recommendation system for elective courses in big data education using collaborative filtering-based recommendation techniques. They used item-based recommendations of the Mahout machine learning library on top of Hadoop to generate a set of recommendations. The similarity Log-likelihood was used to discover patterns among grades and subjects. The Root Mean Square Error was used to test the recommendation system's accuracy.

The work [6] proposed matrix factorization (MF) and nearest neighbor-based recommender systems to provide accurate recommendations. They compared these methods with peer reviewers and found that the affiliation factors are crucial for improving the accuracy of recommender systems. They also proposed a hybrid recommender system that produces a top-k recommendation by combining different single approaches.

The research [7] focused on N-gram query classification and extension-based information retrieval for course recommendations with ontology support. In [8], the authors proposed a new approach for recommending courses to learners based on social filtering and collaborative filtering. They defined the best way in which the learner must learn and recommended courses that better match the learner's profile and social content.

The paper [9] proposed a course recommendation system that uses a combination of Collaborative Filtering (CF) and Content-Based Filtering (CBF) techniques. The system recommends courses to students based on their previous academic performance and interests. The authors used precision and recall as standard evaluation metrics to test the system's performance.

The paper [10] provided a comprehensive review of different recommendation systems proposed for recommending elective courses to students. The authors analyzed 48 relevant studies that proposed different types of recommendation systems, including Collaborative Filtering, Content-Based Filtering, Hybrid, Knowledge-based, and Matrix Factorization methods. They discussed the strengths and weaknesses of each method and provided insights into future research directions.

The paper [11] provided a comprehensive literature review on recommendation systems for education. The authors highlighted the various techniques used to generate recommendations, the data sources used, and the evaluation metrics used to measure the effectiveness of the systems. They noted that most of the studies focused on personalized recommendations for students.

The paper [12] proposed a course recommendation system that combines Collaborative Filtering (CF) algorithm with the K-means clustering algorithm. The system recommends courses to students based on their historical course enrollment data and the enrollment data of other students with similar course preferences. The Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE) were used to evaluate the system's performance. The proposed system outperformed traditional CF-based recommendation systems in terms of both RMSE and MAE.

The above literature review focuses on various recommendation systems proposed for course recommendations in higher education. The studies discussed different approaches, such as collaborative filtering, matrix factorization, content-based filtering, and hybrid methods, to build personalized course recommendation systems. Some studies, such as [4] and [6], compared the performance of different algorithms and found that matrix factorization-based methods outperformed other methods. Meanwhile, Dwivedi and Roshni [5] used item-based recommendations to generate a set of recommendations, and the Root Mean Square Error was used to test the recommendation system's accuracy. In [9], the authors proposed a course recommendation system that uses a combination of Collaborative Filtering (CF) and Content-Based Filtering (CBF) techniques, and precision and recall were used to evaluate the system's performance.

The proposed third-year course recommender system for university students aims to leverage the insights and techniques discussed in the literature review. Specifically, the proposed system will use collaborative filtering with matrix factorization to provide personalized course recommendations based on students' past academic performance, interests, and goals. The system will analyze data from 603 students to identify courses that align with their interests and academic strengths. The Root Mean Squared Error (RMSE) will be used to evaluate the system's performance, as it has been used in previous studies [3, 4, 12]. The proposed system has the potential to assist students in selecting the most appropriate elective courses, improve their academic performance, and enhance their educational experience.

III. Methodology

A. Abbreviations and Acronyms

Dataset preprocessing is a crucial step in the research process, whereby data is collected from the universities' database. This study utilized a real dataset comprising 603 undergraduate students majoring in computer science from 2018 to 2022. Given our goal to develop a 3rd-year course recommender system based on previous 2-year courses, the analysis was limited to the first three-year courses. Courses with fewer than 60 students enrolled were further eliminated, resulting in a total of 90 courses, including 28 3rd year courses. Ultimately, the dataset consisted of 26,686 grades, and course enrollment distribution is given in Fig.1.

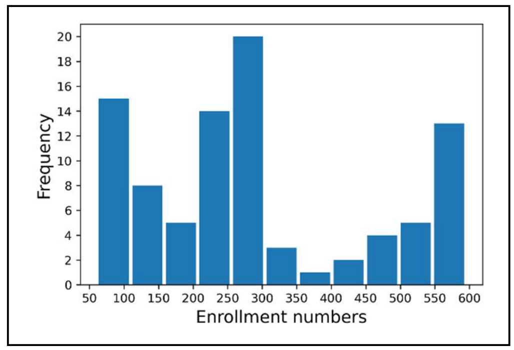

Fig. 1. Course enrollment distribution

B. Proposed Method

The collaborative filtering with matrix factorization algorithm, see Algorithm 1, is proposed for developing a course recommender system for computer science students starting their third year.

Algorithm 1. Collaborative filtering with Matrix Factorization

Input:

num_iter - number of iterations
Y - grades (rating) matrix
U - user latent factors matrix
V - item latent factors matrix
α(t) - time-dependent learning rate
λ - regularization parameter

Output:

U - updated user latent factors matrix
V - updated item latent factors matrix

Pseudocode:

Loop through each row and column of Y using the range of the shape of Y:

1. Check if the value of Y[i,j] is missing. If so, continue to the next iteration of the loop.
2. Calculate the error between the predicted rating (U[i,:] dot V[j,:]) and the actual rating (Y[i,j]).

3. Update the ith row of U by subtracting 2 times the product of the error, the jth row of V, and the learning rate alpha, and then subtracting 2 times lambda_reg times the ith row of U.
4. Update the jth row of V by subtracting 2 times the product of the error, the ith row of U, and the learning rate alpha, and then subtracting 2 times lambda_reg times the jth row of V.

Return: updated U and V.

The collected dataset is represented as a user-item matrix, where each row corresponds to a student and each column corresponds to a course. The matrix is then factorized into two matrices: a user latent factors matrix (U) and an item latent factors matrix (V), such that the product of U and V approximates the original user-item matrix. This is achieved by minimizing an objective function that consists of the sum of squared differences between the predicted and actual ratings, as well as a regularization term that prevents overfitting:

$$min_{U,V} \frac{1}{|\Omega|} \sum_{(i,j)\in\Omega} (y_{(i,j)} - u_i^T v_j)^2 + \lambda(|U|^2 + |V|^2),$$

where Ω is the set of observed grades.

C. *Training and Evaluation*

To evaluate the performance of the model, a sample of 60 students, which represents 10% of the total population, was selected. Prior to conducting the factorization, the third-year grades of these 60 students were removed. The factorization was performed using stochastic gradient descent (SGD), a method that iteratively updates the values of U and V to minimize the objective function. The weights are initialized from the uniform distribution on [-0.5, 0.5]. To prevent gradient blow-ups during the training process, all grades were normalized to be between 0 and 1. The hyperparameter k, which determines the common dimension of U and V, was selected based on the model's performance on the test set. The regularization parameter was set to 0.001 while the learning rate was allowed to vary according to

$$\alpha(t) = \frac{0.051}{1 + 0.1\, t},$$

where t is the iteration variable. To minimize the test loss, we selected the number of iterations for updating the weights. The performance of the recommender system was evaluated by restricting the predicted grades to the range [0,100] and using the mean absolute error (MAE) on the test set.

IV. Results

After conducting multiple experiments, it was determined that a value of k = 2 resulted in the best-performing model. Figure 2 displays the training loss and test loss over 4000 iterations using stochastic gradient descent.

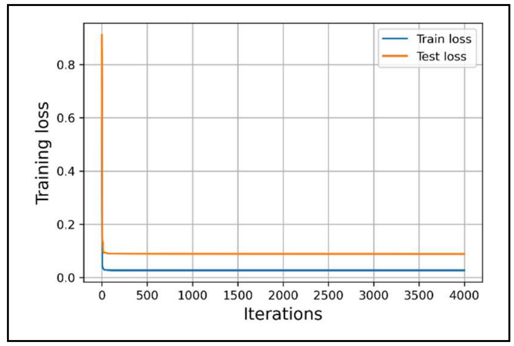

Fig. 2. Training and Test Loss in SGD

As shown in the figure, there is no evidence of overfitting, primarily due to the utilization of L2 regularization and a small fixed value of k. The mean absolute error (MAE) for the test set was determined to be 6.97, indicating that on average, the proposed recommender system predicts grades with an absolute error of 6.97% and a standard deviation of 9.45.

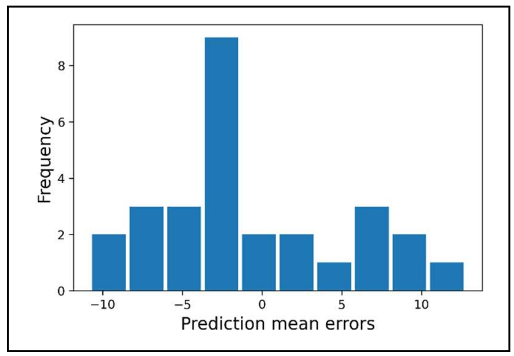

Fig. 3. Distribution of mean residuals

Figure 3 illustrates the distribution of mean residuals for the 28 third-year courses, and Figure 4 provides the corresponding box plot.

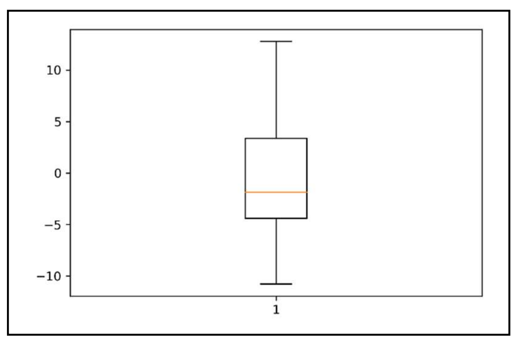

Fig. 4. Box plot of mean residuals

As indicated by the box plot, the interquartile range, which comprises the middle 50% of the mean residuals, is entirely contained within the interval [-5, 5].

V. Conclusion

In this paper, a course recommender system for computer science students based on collaborative filtering with a matrix factorization algorithm was presented. The proposed method was evaluated on a real dataset consisting of 603 undergraduate students majoring in computer science from 2018 to 2022. The collected dataset was represented as a user-item matrix, which was then factorized into two matrices, a user latent factors matrix (U) and an item latent factors matrix (V), using stochastic gradient descent. After multiple experiments were conducted, it was determined that a value of k=2 resulted in the best-performing model.

The results show that our proposed recommender system predicts grades with an absolute error of 6.97% and a standard deviation of 9.45. The box plot of mean residuals indicates that the interquartile range, which comprises the middle 50% of the mean residuals, is entirely contained within the interval [-5, 5]. This implies that our model performs reasonably well predicting grades for the 28 third-year courses.

Overall, the results suggest that our proposed course recommender system can be a useful tool for computer science students in selecting their third-year courses. Future work may include incorporating additional features such as course prerequisites and student preferences to improve the performance of the recommender system further.

REFERENCES

[1] A. E. R. Medina and A. R. Martinell, “Recommender system in higher education: A preliminary study of state of the art,” in 2019 XIV Latin American Conference on Learning Technologies (LACLO), vol. 1, pp. 231-236, IEEE, 2019.

[2] N. D. Lynn and A. W. R. Emanuel, “A review on Recommender Systems for course selection in higher education,” in IOP Conference Series: Materials Science and Engineering, vol. 1098, no. 3, p. 032039, Mar. 2021.

[3] H.L. Thanh-Nhan, H.H. Nguyen, and N. Thai-Nghe, “Methods for building course recommendation systems,” in 2016 Eighth international conference on knowledge and systems engineering (KSE), pp. 163-168, Oct. 2016.

[4] K. Bhumichitr, S. Channarukul, N. Saejiem, R. Jiamthapthaksin, and K. Nongpong, “Recommender Systems for University Elective Course Recommendation,” in 2017 14th international joint conference on Computer science and software engineering (JCSSE), pp. 1-5, Jul. 2017.

[5] S. Dwivedi, and V.K. Roshni, “Recommender System for Big Data in Education,” in 2017 14th International Joint Conference on Computer Science and Software Engineering (JCSSE), pp. 12-14, Jul. 2017.

[6] M. Ismail, K.B. Prakash, and M.N. Rao, “Collaborative filtering-based recommendation of online social voting,” in International Journal of Engineering & Technology (UAE), vol. 7, no. 3, pp. 1504-1507, Jul. 2018.

[7] Zameer Gulzar, A. Anny Leema, Gerard Deepak, “PCRS: Personalized Course Recommender System Based on Hybrid Approach,” Procedia Computer Science, Volume 125, 2018, Pages 518-524, ISSN 1877-0509, https://doi.org/10.1016/j.procs.2017.12.067.

[8] Y. Madani, M. Erritali, J. Bengourram, and F. Sailhan, “Social Collaborative Filtering Approach for Recommending Courses in an E-learning Platform,” Procedia Computer Science, 151 pp. 1164–1169, Jan. 2019.

[9] Bh. Mondal, O. Patra, S. Mishra, P. Patra. “A course recommendation system based on grades,” in 2020 International Conference on Computer Science, Engineering and Applications (ICCSEA), March 2020.

[10] M. Maphosa, W. Doorsamy, B. Paul, “A Review of Recommender Systems for Choosing Elective Courses.” International Journal of Advanced Computer Science and Applications. Vol. 11, no. 9, pp. 287-295, Jan. 2020.

[11] M. C. Urdaneta-Ponte, A. Mendez-Zorrilla, and I. Oleagordia-Ruiz, “Recommendation Systems for Education: Systematic Review,” Electronics, vol. 10, no. 14, p. 1611, Jul. 2021, doi: 10.3390/electronics10141611.

[12] Z. Chen, “Intelligent Courses Recommendation System of Collaborative Filtering Algorithm Based on K-means Clustering under Spark Platform,” in 2022 8th International Conference on Humanities and Social Science Research (ICHSSR 2022). Atlantis Press, pp. 2041-2045, Jun. 2022.