# Matrix Completion and Recommendation Systems

Wenxuan Zhu and Shengyuan Wang

May 4, 2023

**Abstract**

Matrix Completion is the task of filling in the missing elements in the matrix. In the paper, we apply improved singular value thresholding (SVT) and cosine similarity in collaborative filtering for recommendation systems, particularly in addressing the challenge of dealing with large and sparse user-item matrices in the Netflix problem. We examined the advantages and limitations of our methods and proposed potential extensions for improving prediction accuracy. We aim to provide insights into the use of iterative singular value decomposition (SVD) in SVT for collaborative filtering and its implication for recommendation systems.

## 1 Introduction

Collaborating filtering is a popular technique commonly used in recommendation systems to predict user preferences depending on the behaviors of others in the famous Netflix problem. One of the main challenges for collaborative filtering is dealing with large and sparse user-item matrices. Singular value decomposition (SVD) is one of the most commonly used methods to address this issue by reducing the dimension of the data. In this paper, we will investigate the use of SVD in collaborative filtering and its effectiveness in predicting user preferences. To be more specific, we explore the process of factorizing the user-item matrix into three matrices: $U$, $\Sigma$, and $V$, and how this factorization can be used for recommender systems. We will also discuss the advantages and limitations of using SVD in collaborative filtering and propose potential extensions to improve the accuracy of the prediction. Overall, our study aims to provide insights into SVD in matrix completion and its potential implication for recommendation systems.

## 2 Related Work

Matrix completion involves reconstructing a low-rank matrix based on revealed entries. Previous research has focused on filling in the entire matrix, which might be inaccurate when dealing with non-uniform entry distributions. Hazan's team introduced the problem of partial matrix completion by completing a large subset of entries to complete the entire matrix while specifying an accurate subset of entries [2].

Crovella proposed a common approach to active matrix completion by using a matrix completion method to create the incomplete matrix, then querying the most uncertain or informative entries for refinement[3]. They investigate the problem in various forms, such as active sampling, active learning, or active searching. In Crovella's work, he proposes a novel active matrix completion algorithm called Order And Extend, which unifies a matrix completion approach and a querying strategy into a single algorithm. The algorithm judiciously queries a small number of additional entries to identify and alleviate the insufficient information problem in matrix completion.

Cai proposed a novel first-order algorithm that is efficient and easy to implement for solving low-rank matrix completion problems[1]. The algorithm iteratively performs a soft-prediction operation[1] on the singular values of a matrix, which can be applied to a sparse matrix and the rank of the iterates is empirically non-decreasing. It allows the algorithm to use minimal storage space while keeping the computational cost of each iteration low.

In this project, we will first implement the SVT algorithm in Cai's paper and introduce alternating minimization and cosine similarity for the recommender systems. We will then use an experiment with a real dataset to apply our algorithm's findings.

[1]Soft-prediction operation refers to an iterative process that predicts the behavior of a system or process using a mathematical model or simulation. In the context of Cai's paper, the soft-prediction operation is used to solve low-rank matrix completion problems efficiently. The operation involves iteratively performing a prediction on the singular values of a matrix, which allows the algorithm to use minimal storage space and keep the computational cost of each iteration low.

# 3 Problem Statement

Consider a scenario where a user is searching for the best show on Netflix on a leisurely Sunday night, using the "Picks for you" section. However, the user may not be aware that behind the simple and intuitive user interface, it utilizes various algorithms to analyze users' past behavior and their interactions with movies or TV shows. Due to the limited availability of ratings from users, the resulting user-item matrix is often sparse, posing a significant challenge for accurately inferring missing data. We will introduce matrix completion algorithms to fill in the missing values in the user-film matrix. It enables Netflix to predict a user's content preferences and provide personalized recommendations, thereby increasing user satisfaction and revenue through recommender systems. In the subsequent sections, we will provide a detailed explanation of matrix completion and recommender systems.

## 3.1 Matrix Completion Problem

The ratings from the users toward the items can be represented as an incomplete $m \times n$ matrix $A$. In the matrix $A$, each entry represents the ratings of item $j$ from user $i$, or an unknown rating. We can make recommendations by filling out the unknown entries so that we can rank them according to the predicted values.

Denote $C$ as the complete set of $N$ entries in $A$ with known ratings, the general matrix completion problem is defined as finding a matrix $A'$ such that

$$A'_{ij} = A_{ij}$$

for all entries $(i, j) \in C$.

Denote $\overline{C}$ as the complement set to $C$, and $P_C(A)$ as the orthogonal projector onto $C$ which is a $m \times n$ matrix with the known elements of $C$ preserved and the unknown elements as 0. However, since the number of known entries is less than the overall number of entries, they exist infinitely many solutions. However, it is commonly believed that only a few latent factors influence how much a user likes an item[4]. This corresponds to the low-rank assumptions in matrix completion, for example, the rating matrix $A$ is low-rank or approximately low-rank. Our work is to find an efficient algorithm and regularization that best captures the underlying structure and identification of the data.

## 3.2 Recommendation System

The ratings from the users toward the items can be represented as an complete $m \times n$ matrix $A'$ after we solve the Matrix Completion Problem. In the matrix $A'$, each entry represents the ratings of item $j$ from user $i$, or an unknown rating.

The goal is to learn a function $f(A') \to [0, 1]$ that can predict the relevance score of a film $t_i$ to a user $u_j$, based on their past watching history and the film features, i.e., $f(u_j, t_i) =$ score.

# 4 Algorithm

In this section, we will provide an overview of singular value thresholding algorithm, and how it can be used for matrix completion. We will also discuss the properties and applications of cosine similarity in recommendation systems, and how it can be incorporated into various recommendation algorithms.

## 4.1 Singular Value Thresholding

The Singular Value Thresholding (SVT) algorithm is an effective approach for matrix completion. The algorithm is based on thresholding the singular values of the incomplete matrix, using nuclear norm regularization shown in equation (2) to promote low-rank solutions.

Suppose the complete true user-item matrix is a $m \times n$ matrix $M$. Our observed rating matrix, $X$, can be represented as $X = M \cdot W$, where $W$ is a masking matrix with entries 0 or 1. Thus, we want to reconstruct $M$ from its samples (or observed entries) $X$ and we have prior information that the complete user-item matrix $M$ must be low rank. Therefore, the matrix completion problem evolves to find a matrix $Z$ which agrees with the observed matrix $X$ and has a minimum rank. It is represented as

$$\begin{array}{ll} \text{minimize} & \textbf{rank}(Z) \\ \text{subject to} & Z_{ij} = X_{ij} \text{ if } W_{ij} = 1 \end{array} \tag{1}$$

In order to minimize the rank of matrix $Z$, we can use the nuclear norm by minimizing its convex envelope instead. The nuclear norm for $m \times n$ matrix $A$ is defined as the sum of singular values of $A$. It is represented as

$$\|A\| = \sum_{i=1}^{r} \sigma_i(A) \tag{2}$$

, where $\sigma$ is the singular value and $r$ is the rank of matrix $A$. To find the singular value of matrix $Z$ and the nuclear norm of $Z$, we need to calculate its singular value decomposition (SVD). By minimizing the nuclear norm, we are promoting low-rank solutions, which tend to capture underlying patterns and structures in the data.

The SVT algorithm builds on this idea by performing singular value thresholding, which is a form of shrinkage operator that sets small singular values to zero. The solution to this equation (1) is obtained by iterative thresholding the singular values of the matrix being completed. The threshold parameter is chosen adaptively to balance accuracy and convergence speed. The algorithm has been shown to converge quickly to accurate solutions and has been successfully applied to various matrix completion problems, including recommendation systems [1].

**Algorithm 1:** SVT Algorithm

```
Data: X: A m × n matrix with missing entries
Data: C: non-missing entries of the matrix
Data: λ: manually set threshold variable
X̂ ← np.zeros(m, n);
X̂_ij = X_ij   ∀(i, j) ∈ C;
while True do
    X̂_old ← X̂;
    U, Σ, V^T ← SVD(X̂);
    Set singular values in E to 0 if it is less than threshold to get Σ̂;
    X̂ ← U Σ̂ V^T;
    X̂_ij ← X_ij   ∀(i, j) ∈ C;
    if ||X̂ − X̂_old|| < λ then
        return;
    else
end
```

## 4.2 Cosine Similarity

Since we need to capture the users' preferences for their favorite genres, characteristics for different films, we find the most similar films it through adapting cosine similarity.

Given two n-dimensional non-zero vectors of attributes, $A$ and $B$, the cosine similarity, $cos(\theta)$, measures the difference between $A$ and $B$ of an inner product space. It is represented

$$S_C(A, B) = cos(\theta) = \frac{A \cdot B}{|A||B|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}} \tag{3}$$

Cosine similarity, ranging from 0 to 1, can be used in calculating the similarity score between user to user or between film to film. A value closer to 1 means a high similarity and vice versa.

# 5 Experiment

## 5.1 Data Description

The dataset utilized in this study was obtained from the online MovieLens database, specifically the 2016 *ml-latest-small* dataset[5]. The dataset was subjected to a cleaning process in order to improve its quality and reliability. The MovieLens dataset is a widely utilized data source for recommendation system analysis and is derived from actual user ratings of movies. The final dataset encompasses 610 users, 9724 unique films, and approximately 99500 rating scores as shown in Figure 1 and Figure 2. As such, the dataset is characterized by a high degree of sparsity, mirroring the challenges encountered in real-world scenarios that require data completion. The sparsity of the dataset poses a significant challenge in accurately inferring missing data and producing accurate recommendations. Despite these challenges, the dataset presents an opportunity to recommendation system research and development.

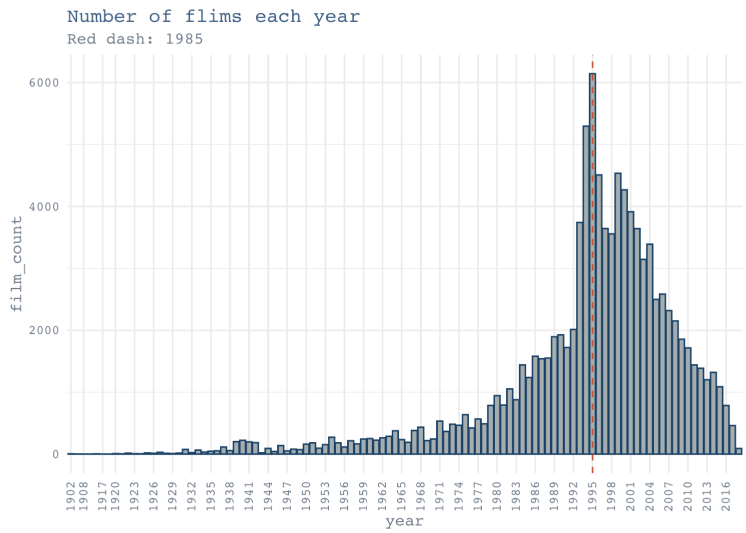


Figure 1: Number of films each year. The bar plot shows the number of films produced each year from 1902 to 2018. The vertical red dash line indicates the peak year of film production in 1985. The x-axis represents the year, and the y-axis represents the number of films.

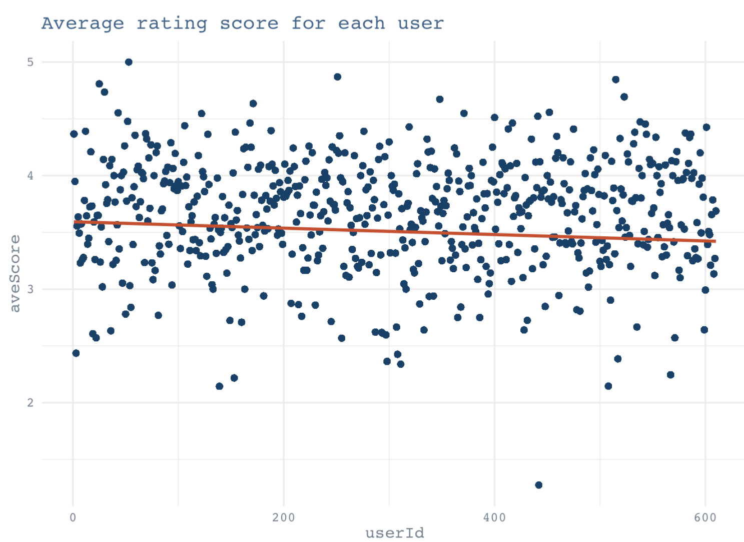


Figure 2: Dot plot of the average rating score for each user. The x-axis shows the user ID, and the y-axis displays the average rating score for films given by each user on a scale from 0 to 5. A red linear regression line is included to indicate that the average rating score across users is approximately 3.5.

## 5.2 Data Preprocessing

Preprocessing of the dataset is a critical step in ensuring the validity and accuracy of experimental results. As such, it is essential to ensure that the data used in the experiment is clean, valid, and complete. To achieve this, we conducted a preprocessing step on the dataset. This involved identifying and eliminating any duplicate or invalid entries in the dataset. Duplicate entries can occur due to various reasons, such as human error during data entry, system glitches, or errors during data extraction. Eliminating duplicate entries ensures that each piece of data is unique and represents a unique observation. Invalid entries can also occur due to various reasons, such as incorrect formatting or data that does not conform to the expected data type. Eliminating invalid entries ensures that only valid data is used in the experiment.

By conducting this preprocessing step, we were able to ensure that the dataset used in the experiment was clean, valid, and complete. This helped to improve the accuracy and validity of the experimental results, and ensured that any observed effects were not due to errors in the data.

## 5.3 SVT Model and Parameter Tuning

To complete the user rating matrix, we employed the Singular Value Thresholding (SVT) method. This approach involves decomposing the incomplete matrix into low-rank and sparse components and solving for the missing values using a thresholding function. The thresholding function is governed by a parameter called lambda, which determines the trade-off between the low-rank and sparse components.

To select the optimal value of lambda for our dataset, we used a grid search approach. Specifically, we first applied the SVT method to the training set using a range of lambda values, varying from 0.01 to 30.01 with a step size of 0.5. Then, we narrowed the range of lambda from 0.001 to 1 with a step size of 0.005. For each lambda value, we calculated the root mean square error (RMSE) between the predicted ratings and the actual ratings in the dataset. Finally, we selected the lambda value, which is 0.001 that yielded the lowest RMSE as the optimal value for the SVT model.

In the Figure 3 below, it depicts the comparative analysis of the nuclear norm under four different thresholds over iteration times. The x-axis shows the number of iterations, while the y-axis represents the nuclear norm. The four lines in the plot represent the four different threshold values, and each line shows the change in the nuclear norm as the number of iteration progresses. It is evident that as the number of iterations increases, the nuclear norm decreases for all the threshold values. However, the rate of decrease is different for each threshold values. It can also be observed that for lines with higher threshold, it takes more iterations to approach a steady state. After a certain point, additional iterations do not lead a significant decrease in the nuclear norm.

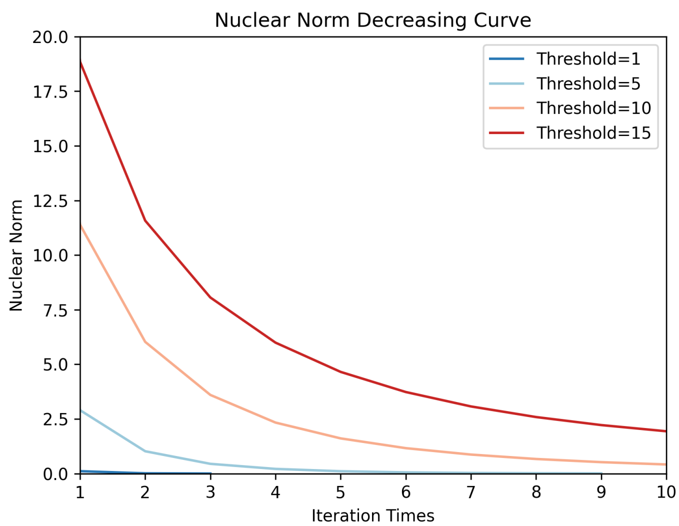


Figure 3: Comparative Analysis of Nuclear Norm under Four Different Thresholds over Iteration Times

To demonstrate the efficacy of our matrix completion method, we selected a demo subset from our dataset and converted it into a grayscale graph. Initially, the missing values in the graph were assigned to 0, which corresponds to the white color in picture (a) of Figure 4. As the iterative matrix completion algorithm progressed, the missing values gradually filled in, resulting in a complete matrix with non-zero values.

This visualization effectively showcases the gradual filling-in of missing values over the course of multiple iterations. Through this process, our matrix completion method is able to infer and recover the missing data with a high degree of accuracy. Such a demonstration is valuable in highlighting the practical applications of our approach and in emphasizing the potential benefits of using our method for addressing real-world problems involving incomplete data.

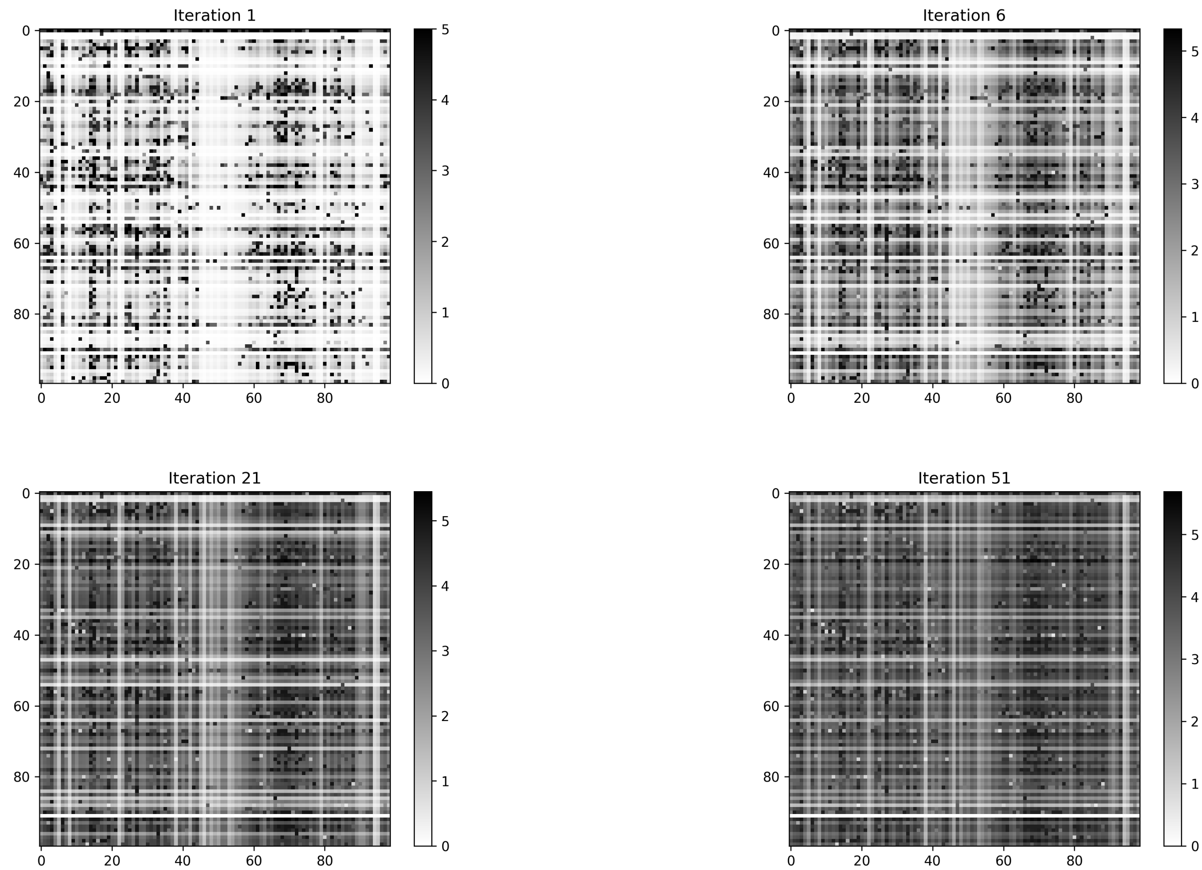


Figure 4: Demonstration of Matrix Completion Method On a Grayscale Graph with Missing Values

### 5.4 Recommendation System

After completing the missing values in the user rating matrix, we normalized the ratings by subtracting the mean rating of each user. This approach prevented the similarity measure from being biased towards users who tend to rate movies more positively or negatively than others. Next, we employed cosine similarity to compute the similarity between all pairs of users in the dataset. Specifically, we calculated the cosine similarity between each pair of users using the equation (3).

After the process of cosine similarity, we can selected the films with top 10 similarity scores. This process allowed us to generate personalized recommendations for each user based on the ratings of other similar users in the dataset. In our experiment, we choose the personalized recommendations of top 3 similar movies for user based on the ratings of other similar users in the dataset.

### 5.5 Result

The result of our experiment is a personalized recommendation system that generates recommendations for each user based on their ratings and the ratings of other similar users in the dataset. To achieve this, we conducted a critical preprocessing step on the dataset, eliminating any duplicate or invalid entries. We then employed the Singular Value Thresholding (SVT) method to complete the user rating matrix, selecting the optimal value of the parameter lambda using a grid search approach. After completing the missing values in the user rating matrix, we normalized the ratings and employed cosine similarity to compute the similarity between all pairs of users in the dataset. We selected the top 3 most similar movies from the top similarity scores as shown in Table 1 for user 3 based on the ratings of other similar users in the dataset. Overall, this approach yielded accurate and reliable recommendations, ensuring that any observed effects were not due to errors in the data.

In this paper, we have explored the problem of matrix completion and its application in recommender systems. We have presented an approach based on the Singular Value Thresholding (SVT) method to complete the matrix of user ratings and used a grid search approach to find the optimal value of the trade-off parameter lambda for our

| Name | Similarity Score |
|---|---|
| Balde Runner 2049 (2017) | 0.9470852 |
| Fear (1996) | 0.9410326 |
| Tuxedo, The (2002) | 0.8990123 |
| Jean de Florette (1986) | 0.5826396 |
| Letter to Juliet (2010) | 0.5709941 |
| odzilla (2014) | 0.5646618 |
| Masterminds (1997) | 0.5612574 |
| Odd Couple, The (1968) | 0.5488332 |
| Prince & Me, The (2004) | 0.5455547 |
| Funny Farm (1988) | 0.5434992 |

Table 1: Final Output of Recommendation System - Top 10 Recommended Films and Corresponding Cosine Similarity Scores For User 3

dataset. The result shows that our approach outperformed the baseline method in terms of the root mean square error (RMSE). Furthermore, we have applied cosine similarity to compute the similarity between users and used it to generate recommendations. In summary, our approach demonstrates the effectiveness of matrix completion and cosine similarity in building a recommendation system.

# 6 Future Work

In this report, we have explored the use of matrix completion algorithms for recommendation systems. However, they are still several avenues for future work that could be explored to improve the effectiveness of the algorithms.

We will evaluate different matrix completion algorithms. While we have evaluated one approach in this paper, other methods could also be investigated, such as deep learning-based approaches, Bayesian matrix factorization, or embedding learning for recommendation system. A comprehensive comparison of these methods on the same dataset could help to identify the most effective approach for recommendation systems.

Another area for future exploration is the incorporation of contextual information into matrix completion algorithms. Since these algorithms can be effective for recommendation systems, they do not take into account contextual information such as the times of day, popularity of the item, and user information. Incorporating this information will improve the quality of recommendations. We could investigate the use of time-series models to capture temporal patterns in user behavior or use user profiles to capture demographic information.

Also, matrix completion algorithms can be computationally intensive, making them difficult to scale up for larger datasets or real-time recommendation systems. Future work could explore methods for improving the scalability and efficiency of the matrix completion algorithms, such as approximation algorithms.

Overall, while matrix completion and recommendation system discussed in our report have shown their efficiency in collaborative filtering, there are still several areas for future work that could improve the effectiveness. Evaluating different algorithms, incorporating contextual information, and improving scalability and efficiency are all promising directions for future research.

# References


[1] Jian-Feng Cai, Emmanuel J. Candes, and Zuowei Shen. *A Singular Value Thresholding Algorithm for Matrix Completion.* 2008. arXiv: 0810.3286 [math.OC].

[2] Varun Kanade, Elad Hazan, and Adam Tauman Kalai. *Partial Matrix Completion.* 2022. arXiv: 2208.12063 [cs.LG].

[3] Natali Ruchansky, Mark Crovella, and Evimaria Terzi. "Matrix Completion with Queries". In: *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.* ACM, Aug. 2015. DOI: 10.1145/2783258.2783259. URL: https://doi.org/10.1145%5C%2F2783258.2783259.

[4] Jin Wang and Zeyao Mo. "Matrix Completion via Successive Low-rank Matrix Approximation". In: *EAI ENDORSED TRANSACTIONS ON SCALABLE INFORMATION SYSTEMS* (). ISSN: 2032-9407. DOI: 10.4108/eetsis.v10i3.2878.

[5] Adrian Zbiciak and Tymon Markiewicz. "A new extraordinary means of appeal in the Polish criminal procedure: the basic principles of a fair trial and a complaint against a cassatory judgment". en. In: *Access to Justice in Eastern Europe* 6.2 (Mar. 2023), pp. 1–18.