Cluster Them Softly: A Source Separation Approach for Shared Account Problem in Movie Recommendation Systems

Working with noisy, incomplete, and mixed observation is inevitable in the realm of data. A severe problem happens in streaming platforms when a single account is usually shared among household members, and the user's interaction with the system will be mixed. Consequently, recommendations would severely fail to follow each user's preferences. Hence, it is crucial to decompose the mixed data from both user and business perspectives. This paper explores a Blind Source Separation technique to tackle such problems. We propose a novel approach utilizing Semi-Binary, Nonnegative Matrix Factorization to identify and separate users' ratings within a shared movie account. Our methodology solely leverages movie genres as auxiliary information to articulate the movie features. Furthermore, a shared account dataset was generated from MovieLens ratings based on CAMRa2011 statistical features to evaluate the method. The effectiveness of our approach is rigorously validated by identifying more than 75 % of users and reconstructing more than 90 % of the true data, even with a few training data. The results show a marked improvement in recommendation accuracy, evidenced by up to 40% reduction in Root Mean Squared Error (RMSE) recommendations after demixing. To tackle the data decomposition problem in shared accounts, we offer a fast yet efficient demixing approach that hinges on learning a latent representation of mixed data.

Keywords:

Blind Source Separation, Mixture Data Decomposition, Clustering, Feature representation, Movie Recommendation System

1 INTRODUCTION

Data is abundant everywhere as sensory devices develop with ever higher resolution. This data is noisy, high dimensional, or mixed. Nevertheless, to understand and gain insights from it, feature extraction, dimensionality reduction, and decomposition techniques become increasingly important in machine learning [1]. In an exploratory approach, mixture data is regarded as observations generated from some latent sources via an unknown mapping. This inverse problem of recovering sources from an observed mixture without prior knowledge of the mixing algorithm or sources is called Blind Source Separation [2]. The celebrated Principal, Independent Component Analysis, and Nonnegative Matrix Factorization are among the leading solutions to BSS. The areas of application are commonly audio signal processing, biomedical signal analysis, and, in our case, recommender systems.

A Recommender System is vital in e-commerce, media streaming, and social platforms, creating substantial value for customers and businesses [3], [4]. In such systems, a single user interacts with available items via an account, and the most desirable items for the user are predicted based on the user's preferences. The preference data of users are generally collected explicitly expressed, e.g., as a point picked from a finite scale by users, or are inferred by interpreting the actions of the users [5]. Either rating data are arranged in a user-item matrix, in which each row contains a user's preferences for interacted items or missing values. The main problem in RS is finding these missing values [6].

Various approaches have been developed to solve this problem. These approaches can be categorized as Content-Based and Collaborative Filtering. The CB approach generates recommendations based on similarities between active user preferences and item attributes. The CF methods extract the user's preference pattern from past ratings to estimate future preferences; the two major approaches are Neighborhood-Based and Model-Based [7]. Matrix Factorization and Matrix Completion are among the most acclaimed methods in Model-Based approaches due to their capacity to incorporate auxiliary information, be scalable, and avoid the cold start problem [8].

Besides, there is a common belief that the user's and the item's features correlate, and only a few latent features influence how much a user likes an item [9]. Mathematically, the user-item matrix can be seen as a high-dimensional vector space where users and items can be linearly represented in fairly lower-dimensional subspaces [10]. This amounts to the fundamental modeling assumption of CF scenarios. In the MF method, we assume ratings follow a linear model over items' and users' features [9], and equivalently, in the MC, we presume the rating matrix is low-rank [11].

Nevertheless, the user who was supposed to be single often shares her account. In fact, the user-item matrix will be an account-item matrix. This is particularly true for subscription media streaming platforms like Netflix and Spotify [12]. In 2015, Netflix introduced account sharing as an open-key problem [13]. They also allude to the fact that there is a need for research to figure out how to credit viewing to the proper profile automatically. Account sharing has become a prominent social phenomenon that hinders personalization. Sharing practice immediately affects the rating matrix so that its rows may contain a mixture of different individual preferences. In this situation, simply applying a recommendation algorithm leads to inferior results.

Consequently, the abovementioned fundamental assumption will no longer be valid, i.e., the ratings would not follow a linear model over users' and items' features. As a result, MF methods would fail [14]. Also, the rating matrix would be a high rank that cannot be completed via conventional methods [15]. This problematic situation is called the shared account problem. It incurs a loss of value for users and businesses by polluting recommendations [12], [16], [17].

This problem has been tackled mainly through the User Identification task in the RS literature. The UI goal is to attribute the items a particular user consumes to the user while the mapping is unknown. Proper recommendations can be generated by knowing just a few of each user's favorite items and desirably diminishing the negative effect of account sharing. As we elaborate in the next section, most studies addressed this task in content-based and session feed settings by excessively exploiting geo-temporal users' data. These data are not always available, and incorporating them in the algorithms leads to over complexity.

This work focuses on movie recommender systems since sharing movie accounts is more prevalent among other fields [12]. In the presence of a shared account, we set out to separate ratings of individual users in the account-movie matrix, leading to an estimate of the user-movie matrix solely utilizing the genre of movies as auxiliary data. Herein, the users are sources, and the rating of each account can be seen as an observation. Therefore, we regard the problem as a BSS or simply data demixing when it is underdetermined and not fully observed. In other words, there are more sources than observations and too many missing values in the data matrix. Motivated by soft clustering and interpretability of Non-negative Matrix Factorization, we cast the problem into an amplified semi-binary NMF to decompose ratings.

In Section 2, after reviewing shared account problem literature, we briefly discuss three main data decomposition techniques. The first two sections will establish the foundation of our novel solution. Next, in Section 3, the proposed method will be explained. Finally, the numerical results and evaluations will provide a tangible understanding and performance of the method in Section 4.

2 LITERATURE REVIEW

This review is twofold. First, we look at the solutions developed for the shared account problem in RS. Then, motivated by leveraging data decomposition to identify users in a shared account, the second part aims to shed light on such methods. But before starting the review, an analogy and explanation need to be made. A Group Recommender System seeks to aggregate the preferences of individual users within a group and generate recommendations for a group of users collectively [18]. In contrast, herein, we separate the preferences of individual users within a group and create recommendations for a group of users individually. Besides, the practice of account sharing has also been investigated in [12], [19]–[21], answering questions like why people share their accounts. Notably, these studies reported that a couple, a family, or roommate friends share most accounts. These small social groups could be referred to as a household. However, a gap within these studies is investigating the demographic distribution of account users.

2.1 Shared Account problem in RS review

The problem of identifying users within a shared account was first addressed in a supervised manner at the RecSys 2011 ACM conference [22]. However, the first formal attempt to realistically model and solve the problem was [14]. Their method merely takes the user-movie rating matrix into account. The item's feature vector is available via rating matrix factorization. However, they proposed a mean square error minimization program based on some observed user-related points to estimate the user profile (the number of users within an account and who rated which movies). Each point was made by extending the rating given to a film with one and its feature vector. Geometrically, these points lie on a union of some hyperplanes, called subspace arrangement. The number of hyperplanes in each arrangement and the closest hyperplane to each point and its normal vector would be equivalent to the number of users in an account, the users who rated this movie, and the user's feature vector, respectively. This insight enabled them to solve the problem in a subspace clustering manner via Spectral Clustering and Expectation Maximization methods. They evaluated their approach on the CAMRa2011 dataset, focusing on 272 rows of its rating matrix, which entails two-user household accounts. To measure their proposal performance, they introduced a similarity criterion, which is a weighted sum. If each movie's identified user matched the ground truth map, one would be added to the sum. Similarity near 1 shows excellent performance, while similarity near 0 reflects poor performance.

In another pioneering work, Verstrepen et al. [17] have considered a Top-N Item-Based collaborative recommendation system with binary rating input and no available contextual information. They devised a solution by tackling three problems arising from account sharing. First, the K Nearest Neighbor items for the preferred items of an account are determined. Then, the recommendation score for those K items is computed. The generality problem arises when the system cannot distinguish between a score that is the sum of a few significant or many small contributions. This problem will be solved by a length adjustment procedure, in which the recommendation score for all possible subsets of preferred items in each account will be approximated. Then, it will be divided by the size of each subset. This procedure will balance all scores for users within an account. The second problem happens because the system does not consider that a specific user may have more available ratings than others. To solve this dominance problem, a disambiguating procedure is proposed, which is vague itself. The performance of the approach is measured by a fraction of users who do not get any individualized recommendations. Their approach is limiting and only accountable in an Item-Base Top-N RS with binary input.

Another idea to solve the SA problem is to look for different consumption patterns, such as geographical and temporal. Wang et al. in [23] proposed a method that decomposes users based on mining other preferences over different periods. Similarly, Lesaege et al. [27] tried to differentiate the active user by learning the periodic temporal consumption pattern, such as news on weekday afternoons. To do so, they assumed that movie consumption followed a Dirichlet distribution and maximized the likelihood of observed data. They paired users of MovieTweetings to synthesize an SA dataset. They also used the similarity criterion as in [14].

Jiang et al. [24] proposed an interesting, rather too complex method[25]. Initially, a graph of items is constructed from their metadata, such as genre, artist, and publication year. Then, a feature representation of nodes will be learned by maximizing the likelihood of observations. Unlike the previously mentioned study [26], Jiang et al. assumed that the user's behavior follows the Poisson distribution. They also defined a function to capture each session's feature representation based on the occurrence of each item in a given session. Finally, user identification is promised via clustering and grouping these session features.

Lian et al. [27] look for interest incoherence in an account by computing the average of the item's similarity preferred therein. Yang et al. [28] took a broader approach, assuming users sharing an account have different tastes in movie features like genre, which vary depending on user-related contextual information such as device and location.

Accordingly, present personas in each account will be learned as distinct user preference patterns. To do so, the user's consumption vectors are projected into a lower dimensional subspace by PCA. If these lower dimensional vectors were not close to each other, it reflects the presence of different personas. Like Jiang et al. [25], clustering these vectors will determine personas. Eventually, mapping the active user to the predicted personas will be done via learning a regression model.

There is also a line of solutions that leverage the perceptional aspect of the user. Mao et al. [29] developed a probabilistic method that models users' patience in an account. Furthermore, they devised a recommendation system that makes a tradeoff between recommending discriminating items for user identification and preferable ones to keep the user interested. Wen et al. [30] proposed using the human attention mechanism.

Pimentel [15] was the first to consider the rating matrix of a shared account as a mixture data structure. Although the main application of his study is image processing, he has proposed an alternating cluster-complete method to solve such a problem. It considers a mixture matrix with missing entries that need to be completed. This matrix is a high rank. First, it needs to be decomposed into low-rank partitions. Then, each part can be completed by conventional methods. This method is rather too complex to be applied in real-world RS. It is worth noting that most studies compared their proposition with a simple RS algorithm, where the presence of multiple users in an account has been neglected. All in all, there is no thorough evaluation between developed methods.

2.2 Decomposition Methods Review

When analyzing complex systems, dealing with mixed data is unavoidable. Hence, decomposing a data into multiple components has become a fundamental technique across various fields, allowing researchers and analysts to uncover hidden patterns and gain deeper insights. For a successful separation, we must seek structure or independent directions in data [2]. These correspond to component analysis, each of which is a base of a subspace that can reproduce the whole space. In the following, we concisely review three main data decomposition techniques that aim to learn the unknown sources that produced the data.

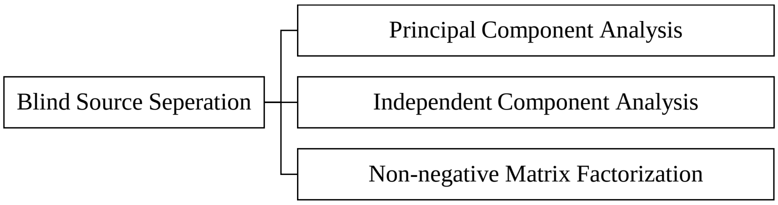

Blind Source Seperation
Principal Component Analysis
Independent Component Analysis
Non-negative Matrix Factorization

Figure 1: A BSS Categorization

Basically, projecting data into other axes or space can be used for separating data out into sources. In other words, data decomposition is the task of finding bases that can be used to span the whole space where the high-dimensional data lives. Classical decomposition methods, such as Fourier and Wavelet transforms, assume that the projections onto each component are independent of the other components and that the number of components equals the space dimension, meaning those bases span the whole space [31]. Principal Component Analysis (PCA) decorrelate the data by projecting it onto a new coordinate system where the axes (principal components) are aligned with the directions of maximum variance in the data. The central assumption behind this task is the Gaussian distribution of the data [32].

Independent Component Analysis projects the data onto a new coordinate system where the components are as statistically independent as possible. Unlike PCA, which decorrelates the data, ICA goes further to make the projected components statistically independent. The standard ICA assumes the sources have non-Gaussian distribution and their number is equal to the number of observations [33]. However, to handle the latter, rarely true assumption remedies have been proposed [34], [35].

Another BSS solution is Non-negative Matrix Factorization. Given the data matrix, $V \in \mathbb{R}_{+}^{m\times n}$ contains m observation in n dimensions, NMF factorizes V into two non-negative components, namely the basis ($W \in \mathbb{R}_{+}^{m\times k}$) and the coefficient ($H \in \mathbb{R}_{+}^{k\times n}$) matrices such that $V \cong \langle W, H\rangle$, when k is the number of latent features and priori known. This method can also be seen as an unsupervised clustering method in terms of machine learning literature [36]. It projects the original data into a lower-dimensional space where both the basis and the coefficient are non-negative, assuming the data and sources are non-negative and additively mixed. The clustering effect of NMF is based on the idea that H contains the vectors of cluster centroids, and the rows of W contain the soft clustering membership of data points for clusters [37]. The largest array of each row can be chosen for hard clustering outcomes. A salient application of this is document clustering with K priori known topics. Where terms of each document are placed in rows of data matrix V, rows of W provide a topic-wise representation, and the columns of H provide a keyword-wise topic representation of each document [38].

The abovementioned methods have some presumptions on the data to be decomposed, such as knowledge of the statistical distribution of sources and the equal number of sources and observations. Conversely, in our problem, prohibitively, no distribution can be assumed, and the number of observations is less than the sources; thus, another approach will be imperative. Therefore, we propose a soft decomposition algorithm for learning sources by forcing the coefficient component of the factorization process to be in a desirable feature-wise representation and extracting out the basis component as a membership matrix to each source.

3 FORMULATION AND MODELLING

Consider a collaborative filtering movie recommender system with a 5-star user feedback setting; realistically, each account could be used by multiple users. Let n_A and n_M denote the number of accounts and the number of movies in the system. Although we had a user-movie rating conventionally, herein, the available data, $R^{\mathrm{sa}} \in \mathbb{R}_{+}^{n_A\times n_M}$ is an account-movie rating matrix. The missing entries are filled with zeros. The ratings in each account are indeed a mixture of ratings of users within the account.

Our only assumption is common in shared account studies [14], [17], [20]. Each account is being used by people with different tastes within a household [13]. Then, the rating matrix is a mixture of different household members' preferences. Now, demixing the shared rating matrix will identify users. This demixing problem is rather too complex. We look for a lower dimensional user-related common subspace between accounts and movies to avoid over-complexity and make user identification scalable.

Based on the fundamental assumption of CF, the account-movie rating matrix (R^{sa}) cannot be described over users. But, it can be factorized in an account feature $A \in \mathbb{R}^{n_A\times K}$ and a movie feature matrix $M \in \mathbb{R}^{n_M\times K}$, given by the following equation:

$$R^{\mathrm{sa}} \cong \langle A, M\rangle \qquad (1)$$

Where the symbol$\langle , \rangle$ denotes the inner matrix product. Suppose we choose the K latent features to be representative of users' central taste patterns. If we articulate the coefficient matrix M with these K latent features, factoring out the basis matrix A aids a desirable user–wise separation of the rating matrix. Our algorithm can be summarized in three steps:

1. Determining the coefficient component by representing the features in a desirable latent manner
2. Semi-binary non-negative factorization of observation matrix with the predetermined coefficient
3. Demixing observations by the weight of elements of components

The following subsection will propose a quantitative approach for describing the movies in that user-wise articulation.

3.1 Coefficient Representation

According to the literature, the movie genre is the most inclusive feature on which the user's preference depends. Let $G \in \mathbb{R}^{n_M \times g}$ represent movie genre features when g is the number of movie genres in the system. If we transform the movie-genre matrix into an account's interest-genre matrix by $\langle R^{sa}, G \rangle$, then the centroids resulted from clustering such transformation can capture central taste patterns or different user segments. Since users with varying taste patterns have been mixed in accounts, identifying such patterns in each account will amount to identifying different users. This task will be formulated in the next subsection. Let matrix $C \in \mathbb{R}^{g \times K}$ contain those centroids.

$$C = Centroids\ of\ applying\ Kmeans\ clustering\ on\ \langle R^{sa}, G \rangle \qquad (2)$$

After that, we can project the result back into the original space by the following equation, which provides a taste pattern-wise representation of the data.

$$M = \langle G, C \rangle \qquad (3)$$

Notice that all elements of $R^{sa}, G,$ and C are nonnegative. Thus, M values will desirably become nonnegative $\mathbb{R}_{+}^{n_M \times K}$, too.

3.2 Modelling

Consider Equation (1), given R^{sa} along with insight and calculation of coefficient matrix M via Equation (3). If we specify the basis matrix A as a binary matrix, each array shows the presence or absence of each taste pattern in a given account. Specifying A can be done by minimizing the Frobenius norm error over a binary domain, as follows:

$$\min_{\{0,1\}^{n_A \times K}} \| R^{sa} - X.M \|_F^2 \qquad (4)$$

Since Equation (4) elements are nonnegative, it actually denotes a semi-binary nonnegative matrix factorization, similar to the data decomposition technique in [37] by Zdunek. Notice that R^{sa}, the data to be demixed, has overlapping clusters in it. In this respect, we relax the binary constraint to achieve a softly clustered result. Zhang proposed a computational solution for this problem [39], which we use to solve Equation (4). It makes the arrays of the solution near {0, 1} by penalizing the objective function with a $\lambda \| X^2 - X \|_F^2$ term. Thus, Equation (4) will become a convex program where the regularization parameter λ controls the effect of the penalty term.

So far, this formulation is completely aligned with the BSS effect of NMF, as explained in Subsection 2.2. It can be interpreted that with priori known K clusters or rank of the mixture data if each row R^{sa} represents a single observation, a binary estimation row vector of A indicates the weights of cluster's membership, and the corresponding column of M indicates the direction of centroids of that cluster. Still, we need to strictly specify the cluster's membership using a controllable hard clustering auxiliary approach, which will be described in the following section.

3.3 Auxiliary Hard Clustering

As the binary format of X was desirable, inspired by the Optimization with Bounded ℓ_∞-Norm of [40], we propose using an indicator function after approximation of X so that its arrays become exactly {0, 1}. Since at least one user is in each account, at least one array in each row of A must be 1. As a result, we define the indicator function Π as follows:

$$\Pi(x) = \begin{cases} 1 & x \geq l * argmax(x) \\ 0 & x < l * argmax(x) \end{cases} \qquad (5)$$

In this respect, binarizing X will identify the presence or absence of each user's taste pattern in each account . According to what has been put, Algorithm 1 will accomplish the user identification mission.

Algorithm 1: Source Identification

```
Input: $R^{sa}$ Mixture Data Matrix, $\lambda$: Binary Factorization Parameter, $l$: Hard Clustering Parameter
Output: $\hat{A}$ Basis Matrix, $\widehat{M}$ Updated Coefficient Matrix, $\hat{A}B$ Membership Matrix
Initialize $X = 1^{n_A \times K}$
Normalize columns of $M$      # so that each movie has the same total weight.
Solve Equation (4) and return $\hat{A}$ and $\widehat{M}$
Calculate $\Pi(\hat{A})$ and return $\hat{A}B$
```

For the case of user identification, this algorithm can specify how many users are present in each account as rows of $\hat{A}B$ and how much of the mixture data has been generated by each of them as rows of $\hat{A}$ and $\widehat{M}$. Now, we need to demix the ratings. That is, separating those arrays of each row of R^{sa} that were generated by different users. To do so, we propose the following algorithm.

3.4 Demixing Algorithm

As described in Section 2, NMF captures the additive behavior of data and puts it into components in terms of weights. To reverse the process of values getting merged, we can multiply each merged row to the corresponding basis and coefficient components so that it breaks out to its sources. This process is depicted in the following pseudo-code.

Algorithm 2: Source Separation

```
Input: $R^{sa}$ Mixture Data Matrix, $\hat{A}$ Basis Matrix, $\widehat{M}$ Updated Coefficient Matrix, $\hat{A}B$ Membership Matrix
Output: $R^r$ Reconstructed Data Matrix
Initialized $R^u$and $R^r$ as empty matrices to store separated and the final values
For $i = 0, \ldots, n_A$:
        Remove rows of $\hat{A}_i^{\,T}$ where $\hat{A}B_i^{\,T} = 0$
        $R^u = \hat{A}_i^{\,T} * \widehat{M}$
        Normalize columns of $R^u$ with $l_1$ Norm.
        For each row of $R^u$:
                Append $(R_i^{sa} * R^u)$ to $R^r$
Round values of $R^r$ to integer and return it
```

To illuminate the algorithm, an example of rating demixing is given. Suppose from Algorithm 1: Source Identification outputs, the first row of $\hat{A}\_B$ is estimated as [1, 0, 0, 1]. It shows two distinct patterns or two users are sharing the first account. Thus, taste patterns with users' indices $k = \{0, 3\}$ should be separated from the first row of the rating matrix. We do this separation by multiplying the transposed vector of the first row of $\hat{A}$ to the whole coefficient matrix after removing rows corresponding to $k = \{1, 2\}$. The resulting matrix contains the mixing map relative taste weights of users' indices $k = \{0, 3\}$ towards each movie in the dataset. Finally, multiplying the ratings of this first account by these relative weights demixes the ratings. Continuing this process for all accounts and appending the resulting demixed ratings will reconstruct the true taste values belonging to all users. A further illustration will be presented in Figure 3 of Section 4.2.

4 NUMERICAL RESULTS AND EVALUATION

In this section, first, we introduce a mixture dataset in the context of recommendation systems. Then, after applying our algorithms to the dataset, the numerical results will be presented. We also provide a macro and micro view of the results for showcasing the mechanics of the proposed method. In the following, a thorough analysis will be performed via computing the confusion matrix and similarity metrics, comparing real users with identified ones. Eventually, we will showcase how our demixing procedure positively affects the recommendation performance. Note that codes for all implementations are available at this repository[1].

4.1 Shared Account Dataset Generation

A standard shared account dataset is needed to evaluate the performance of our proposed method. The CAMRa2011 dataset was released at the Context-Aware Movie Recommendation challenge. Although the publisher removed it from public access, we extracted and summarized its statistical characteristics in Table 1 based on [14]. Notably, the maximum number of users in an account is 4. Regarding the BSS approach, herein, the number of sources is 4. In this case, another important note to be considered is the severe lack of training data, especially for accounts comprising four users.

Table 1: Statistical Characteristics of CAMRa2011

Ratio	Amount
Total number of shared account members per total users	0.33
Shared account of size 2 per total number of shared accounts	0.94
Shared account of size 3 per total number of shared accounts	0.05
Shared account of size 4 per total number of shared accounts	0.01

MovieLens is the most renowned dataset in the recommendation research community. Consider MovieLens 100K rating in a user-movie matrix format, named $R \in \mathbb{R}_{+}^{943\times 1682}$. Furthermore, we use the movie-genre data named $G \in \mathbb{R}_{+}^{1682\times 18}$, available in the MovieLens 100K package, to produce the coefficient component described in Subsection 3.1. Since two movies have unknown genres in the dataset, we first scope out these movies. Then, we generate a shared account dataset by randomly merging rows of R based on ratios in Table 1 above. Recall that our shared account assumption was "users with distinct tastes in an account". To implement this assumption, rows to be merged are chosen from different clusters, resulting from applying k-means on rows of R with *k=4*. It leads to the promised account-movie dataset, namely $R^{sa} \in \mathbb{R}_{+}^{784\times 1680}$ Figure 2 compares the density of MovieLens 100K and the generated shared account dataset. All in all, 784 accounts of R^{sa} comprise 943 users of the MovieLens 100K dataset.

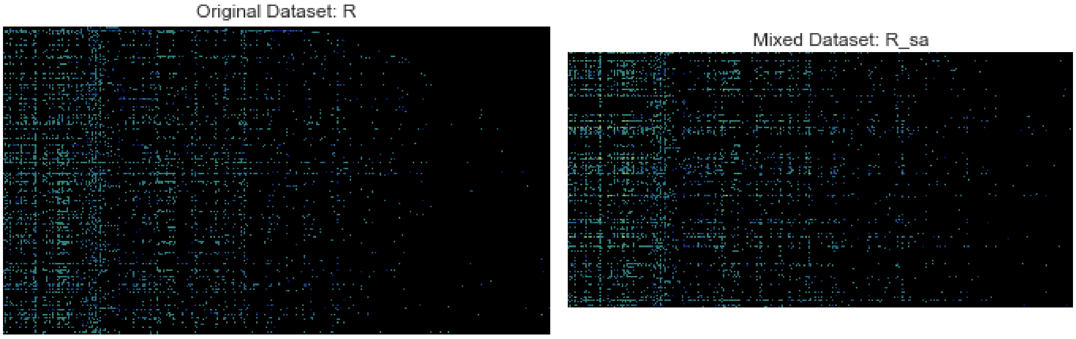


Figure 2: Shared account generation from MovieLens100K. Left: before, Right: After sharing accounts.

4.2 Numerical results

The problem at hand is severe to the extent that even prominent studies such as Guess Who Rate This [14] have merely evaluated their method on 600 users that were in two-user accounts. Also, Verstrepen in [17] has changed the ratings to a setting of 0,1 only. However, we stick to the actual settings of the reported shared account dataset.

[1] https://github.com/mkhademali

We use the NIMFA library [41] to evaluate our proposed method to solve Equation (4) in Algorithm 1 with $\lambda = 1.55$. Also, we modify the BMF algorithm by fixing the regularization parameter for the binarizing coefficient component to be 1. Experimentally, we tune the parameter l of the indicator function depicted by Equation (5) to be 0.9. After applying Algorithm 1: Source Identification to the abovementioned problem, results obtained as follows:

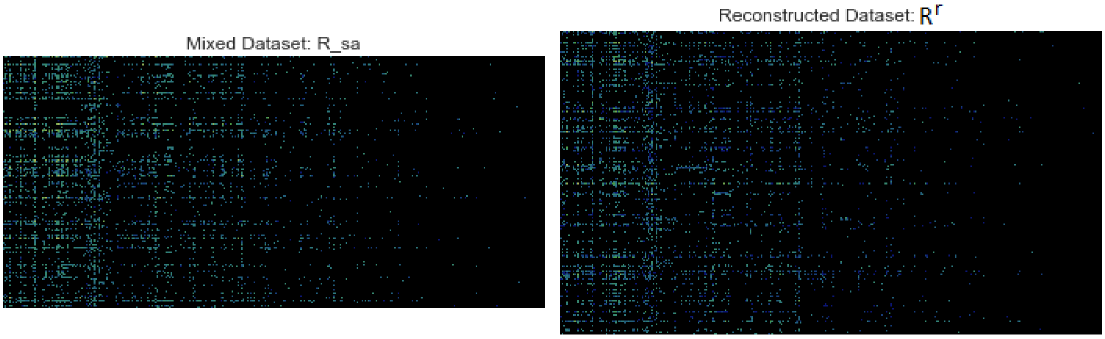
Table 2 Part of $\hat{A}$

Account number	Pattern_0	Pattern_1	Pattern_2	Pattern_3
0	0.85	0.08	0.41	0.01
1	0	0	0	0.83
...	...	...	...	...
783	0.14	0.92	0	0.03

Table 3: Part of $\hat{A}B$

Account number	Pattern_0	Pattern_1	Pattern_2	Pattern_3
0	1	0	0	0
1	0	0	0	1
...	...	...	...	...
783	0	1	0	0

Each array in each row of $\widehat{A}$ represents the amount of taste pattern trace in each account. For instance, the prevalence of Pattern_0 in the first account is evident from Table 2. To explicitly determine the identification of each source in each observation, the procedure for hard clustering, explained in Subsection 3.3, will be applied, which leads to Table 3. Now, the value 1 in the first row identifies the source that has produced Pattern_0. Likewise, the sum of $\widehat{A}B$ arrays is 943, which replies that 943 users are identified from 784 accounts. Based on this identification, the demixing process will be done via Algorithm 2: Source Separation, which reconstructs the rating matrix. The result is shown in Figure 3.

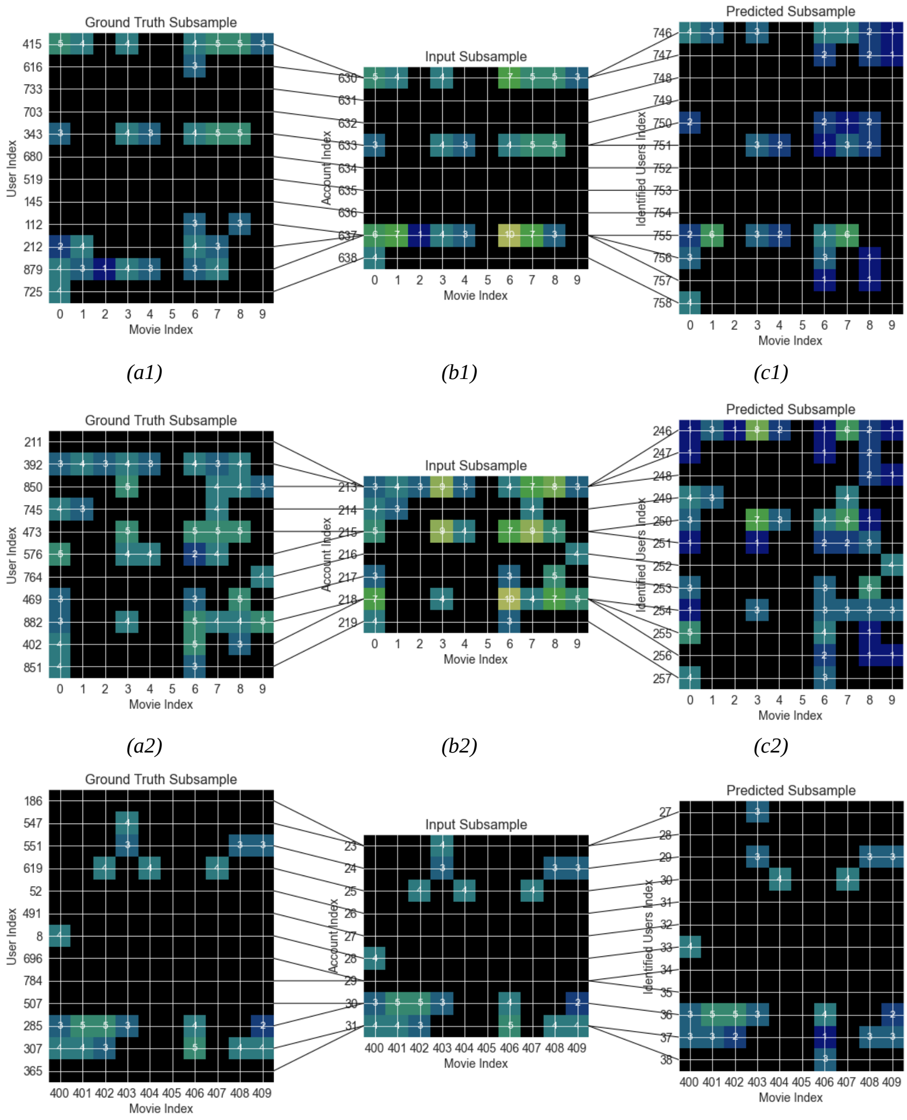


Mixed Dataset: R_sa

Reconstructed Dataset: R^r

Figure 3: Account decomposition. Left: Shared account dataset, Right: After separation.

To see, we need to look closer. Thus, we choose three random samples of size almost 10 by 10 of the above matrices to illuminate how data mixing happens and how our proposed algorithm demixes such amalgam. Each row of Figure 4 shows one of the samples. The middle blocks entitled (b) are taken from the mixture data, R^{sa}, which are inputted into the algorithm. Being demixed, the rightmost (c) blocks are the algorithm's output. The left blocks under rubric (a) are the true data corresponding to the mixture shown in (b) and are used solely for the evaluation exhibition. Finally, the black lines between the blocks show the mapping. For example, a1, b1, and c1 blocks represent the ratings for 0 to 9-th movies. The 415-th and 616-th users are mixed (a1) and placed into the 630-th account (b1), then the algorithm demixes this account’s rating and places them in the 746-th and 747-th rows of its output.



(a1) *(b1)* *(c1)*

(a2) *(b2)* *(c2)*

(a3) *(b3)* *(c3)*

Figure 4: A closer view of the matrices and Source Separation process.

4.3 Evaluation

Our evaluation is twofold in terms of recommendations and source separation performance. First, we apply a basic collaborative filtering algorithm equipped with Mean Squared Difference similarity between all pairs of users (or items) to generate the recommendation based on the previously mentioned data sets. Then, via the Surprise library [42], we run a 5-fold cross-validation to compare the Root Mean Squared Errors of the recommendations. Table 4 reports the increase in RMSE from R to R^{sa}, highlighting the severity of the shared account problem. At the same time, the subsequent decrease after applying our algorithm demonstrates its effectiveness in improving recommendation accuracy.

Table 4- The improvement of the proposed algorithm on recommendations

Dataset	**Original Ratings: R**	**Shared Accounts Ratings: R^{sa}**	**Reconstructed Ratings: R^{r}**	**Improvement**

User Base RMSE	0.9784	1.2908	1.1919	**37%**
Item Base RMSE	0.9742	1.3071	1.2013	**39%**

We also evaluate the algorithm's source separation performance, which involves identifying and separating individual user contributions within shared accounts. This is assessed using precision, accuracy, and confusion matrices. Table 5 presents the precision and accuracy of the predictions. Table 6 provides the numerical confusion matrix. Correspondingly, Figure 5 visualizes the confusion matrix in percentage, showing the proportion of correctly and incorrectly classified accounts across different categories. For example, it can be seen from the third row of Table 6 and Figure 5 that 3 out of 5 accounts were correctly found to be three-user accounts, implying that 60 percent of the three-user accounts have been correctly identified.

Table 5- Precision and Accuracy of the Prediction

Identification of	**Single-User Accounts**	**Two-User Accounts**	**Three-User Accounts**	**Four-User Accounts**
Precision	0.89	0.5	0.23	0
Accuracy	0.83	0.82	0.98	1

Table 6-Confusion Matrix in Numbers

Predicted **True**	**Single-User Accounts**	**Two-User Accounts**	**Three-User Accounts**	**Four-User Accounts**
Single-User Accounts	568	63	1	0
Two-User Accounts	70	67	9	0
Three-User Accounts	0	2	3	0
Four-User Accounts	0	1	0	0

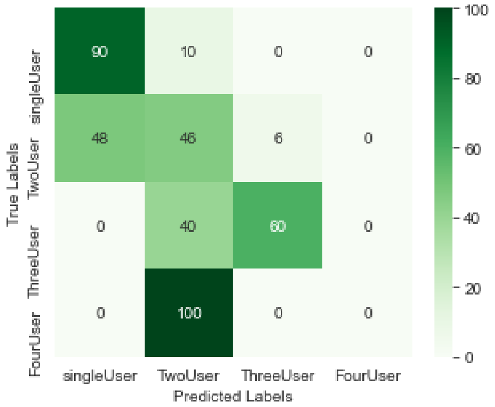


Figure 5-Confusion Matrix in Percentage

To further evaluate the performance of our algorithm, we employed two similarity metrics: Elementwise Similarity and Structural Similarity Index (SSIM). These metrics offer insightful evaluations of how well the reconstructed data aligns with the ground truth data. Elementwise Similarity adopted from [14] quantifies the agreement between reconstructed data elements (identified ratings) and ground truth data elements (actual ratings), considering possible permutations. This metric focuses on the individual rating values, providing a granular view of the reconstruction accuracy. As illustrated in Figure 6-Left, the distribution of Elementwise Similarities across all observations shows a high concentration of similarity values close to 1, indicating a strong agreement between the reconstructed and ground truth ratings for most data points.

SSIM, derived from the image processing literature[43], measures the similarity between two datasets by evaluating the structural information rather than just numerical differences. This is particularly valuable in recommendation systems where the overall pattern or structure (e.g., user preferences) is crucial. Figure 6 displays the distribution of Structural Similarities across all observations. The majority of the similarity values are close to 1, demonstrating that our algorithm effectively preserves the structure of the ground truth matrix in the reconstructed matrix. Table 7 reports the mean value of similarity metrics.

Table 7-Mean Value of Similarity over All Rows of Datasets

Similarity	Structural	Elementwise
Mean	0.90	0.96

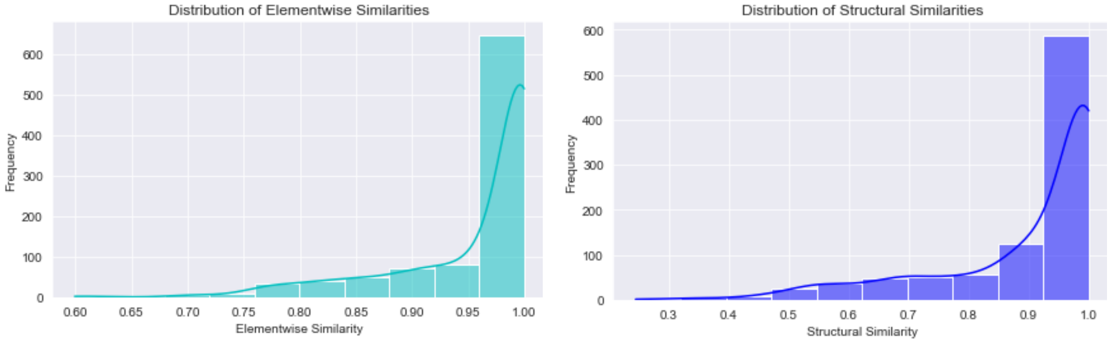


Figure 6-Similarity Metrics

Also, the Pareto diagram below analyzes the RMSE of predictions by visualizing their distribution and cumulative impact. The blue histogram illustrates the RMSE frequency distribution of reconstructed data, in which values range from 0 to 2, with most values concentrated towards the lower end, indicating that most predictions are relatively accurate. The red line curve demonstrates how the RMSE values accumulate, showing that a small number of users contribute significantly to the total RMSE. Overall, the mean value of 0.41 of RMSEs across all observations is a reference point for evaluating the predictive performance.

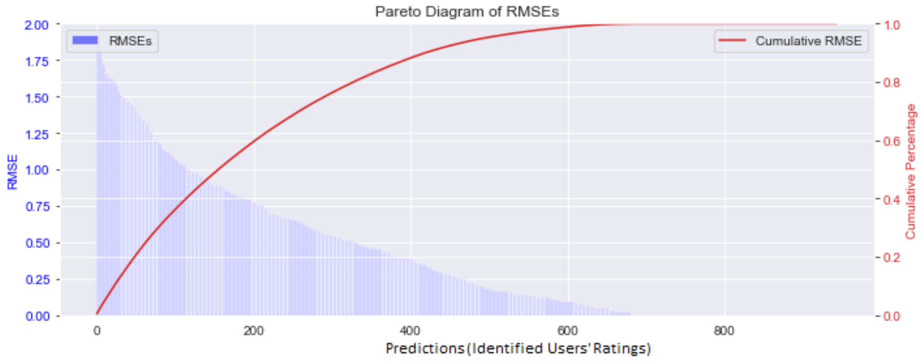


Figure 7-Pareto Diagram of RMSEs

5 CONCLUSION

In conclusion, this research underscores the efficacy of advanced Blind Source Separation (BSS), specifically the Semi-binary Non-negative Matrix Factorization (SBMF), in addressing the shared account problem within movie recommendation systems. This problem is prohibitively severe in ubiquitous collaborative filtering recommender systems. In the setting that no prior assumption on the data distribution or number of users can be made, our approach successfully identify the presence of different users and demixes individual user ratings, significantly improving the accuracy of personalized recommendations. Our method demonstrates a substantial reduction in RMSE and high precision in user identification, validating its robustness and scalability. This study illustrates the broader applicability of BSS and data decomposition methods beyond traditional signal processing domains, offering a compelling solution to the shared account challenge in recommender systems. Future research could extend this framework by incorporating additional auxiliary data sources and exploring its application in other domains with similar complexities. Our findings contribute to advancing recommendation system methodologies enhancing user experience and business value in subscription-based media streaming platforms.

REFERENCES

[1] J. Arenas-Garcia, K. B. Petersen, G. Camps-Valls, and L. K. Hansen, "Kernel Multivariate Analysis Framework for Supervised Subspace Learning: A Tutorial on Linear and Kernel Multivariate Methods," *IEEE Signal Process. Mag.*, vol. 30, no. 4, pp. 16–29, 2013, doi: 10.1109/MSP.2013.2250591.

[2] E. Roussos, "Data decomposition: from independent component analysis to sparse representations," *PeerJ Prepr.*, vol. 6, p. e27456, 2018, [Online]. Available: https://api.semanticscholar.org/CorpusID:58013381.

[3] N. Ghanem, S. Leitner, and D. Jannach, "Balancing consumer and business value of recommender systems: A simulation-based analysis," *arXiv Prepr. arXiv2203.05952*, 2022.

[4] D. Jannach and M. Jugovac, "Measuring the business value of recommender systems," *ACM Trans. Manag. Inf. Syst.*, vol. 10, no. 4, pp. 1–23, 2019.

[5] F. Ricci, "Recommender Systems: Models and Techniques," *Encycl. Soc. Netw. Anal. Min.*, pp. 2147–2159, 2018, doi: 10.1007/978-1-4939-7131-2_88.

[6] D. Roy and M. Dutta, "A systematic review and research perspective on recommender systems," *J. Big Data*, vol. 9, no. 1, p. 59, 2022, doi: 10.1186/s40537-022-00592-5.

[7] X. Su and T. M. Khoshgoftaar, "A Survey of Collaborative Filtering Techniques," *Adv. Artif. Intell.*, vol. 2009, p. 421425, 2009, doi: 10.1155/2009/421425.

[8] R. Mehta and K. Rana, "A review on matrix factorization techniques in recommender systems," in *2017 2nd International Conference on Communication Systems, Computing and IT Applications (CSCITA)*, 2017, pp. 269–274, doi: 10.1109/CSCITA.2017.8066567.

[9] M. Udell, C. Horn, R. Zadeh, and S. Boyd, "Generalized Low Rank Models," 2015.

[10] M. D. Ekstrand, J. T. Riedl, and J. A. Konstan, "Collaborative filtering recommender systems," *Found. Trends Human-Computer Interact.*, vol. 4, no. 2, pp. 81–173, 2010, doi: 10.1561/1100000009.

[11] A. Ramlatchan, M. Yang, Q. Liu, M. Li, J. Wang, and Y. Li, "A survey of matrix completion methods for recommendation systems," *Big Data Min. Anal.*, vol. 1, no. 4, pp. 308–323, 2018, doi: 10.26599/bdma.2018.9020008.

[12] N. Sailaja and A. Fowler, "An Exploration of Account Sharing Practices on Media Platforms," in *ACM International Conference on Interactive Media Experiences*, 2022, pp. 141–150, doi: 10.1145/3505284.3529974.

[13] C. A. Gomez-Uribe and N. Hunt, "The Netflix Recommender System: Algorithms, Business Value, and Innovation," *ACM Trans. Manag. Inf. Syst.*, vol. 6, no. 4, Dec. 2016, doi: 10.1145/2843948.

[14] A. Zhang, N. Fawaz, S. Ioannidis, and A. Montanari, "Guess Who Rated This Movie: Identifying Users through Subspace Clustering," in *Proceedings of the Twenty-Eighth Conference on Uncertainty in Artificial Intelligence*, 2012, pp. 944–953.

[15] D. L. Pimentel-Alarcón, "Mixture matrix completion," *arXiv Prepr. arXiv1808.00616*, 2018.

[16] W. Zhang and C. Challis, "Towards addressing unauthorized sharing of subscriptions," *Appl. Intell.*, 2021, doi: 10.1007/s10489-021-02812-6.

[17] K. Verstrepen and B. Goethals, "Top-N Recommendation for Shared Accounts," in *Proceedings of the 9th ACM Conference on Recommender Systems*, 2015, pp. 59–66, doi: 10.1145/2792838.2800170.

[18] S. Dara, C. R. Chowdary, and C. Kumar, "A survey on group recommender systems," *J. Intell. Inf. Syst.*, vol. 54, no. 2, pp. 271–295, 2020, doi: 10.1007/s10844-018-0542-3.

[19] M. Jacobs, H. Cramer, and L. Barkhuus, "Caring About Sharing: Couples' Practices in Single User Device Access," in *Proceedings of the 2016 ACM International Conference on Supporting Group Work*, 2016, pp. 235–243, doi: 10.1145/2957276.2957296.

[20] B. Obada-Obieh, Y. Huang, and K. Beznosov, "The Burden of Ending Online Account Sharing," in *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, 2020, pp. 1–13, doi:

10.1145/3313831.3376632.

[21] C. Y. Park, C. Faklaris, S. Zhao, A. Sciuto, L. Dabbish, and J. Hong, “Share and share alike? An exploration of secure behaviors in romantic relationships,” in *Fourteenth Symposium on Usable Privacy and Security ({SOUPS} 2018)*, 2018, pp. 83–102.

[22] A. Said, S. Berkovsky, E. W. De Luca, and J. Hermanns, “Challenge on Context-Aware Movie Recommendation: CAMRa2011,” in *Proceedings of the Fifth ACM Conference on Recommender Systems*, 2011, pp. 385–386, doi: 10.1145/2043932.2044015.

[23] Z. Wang, Y. Yang, L. He, and J. Gu, “User Identification within a Shared Account: Improving IP-TV Recommender Performance BT - Advances in Databases and Information Systems,” 2014, pp. 219–233.

[24] J.-Y. Jiang, C.-T. Li, Y. Chen, and W. Wang, “Identifying Users behind Shared Accounts in Online Streaming Services,” in *The 41st International ACM SIGIR Conference on Research &amp; Development in Information Retrieval*, 2018, pp. 65–74, doi: 10.1145/3209978.3210054.

[25] J.-Y. Jiang, C.-T. Li, Y. Chen, and W. Wang, “Identifying Users behind Shared Accounts in Online Streaming Services,” in *The 41st International ACM SIGIR Conference on Research &amp; Development in Information Retrieval*, 2018, pp. 65–74, doi: 10.1145/3209978.3210054.

[26] C. Lesaege, F. Schnitzler, A. Lambert, and J. Vigouroux, “Time-Aware User Identification with Topic Models,” in *2016 IEEE 16th International Conference on Data Mining (ICDM)*, 2016, pp. 997–1002, doi: 10.1109/ICDM.2016.0126.

[27] T. Lian, Z. Li, Z. Chen, and J. Ma, “The Impact of Profile Coherence on Recommendation Performance for Shared Accounts on Smart TVs BT - Information Retrieval,” 2017, pp. 30–41.

[28] S. Yang, S. Sarkhel, S. Mitra, and V. Swaminathan, “Personalized Video Recommendations for Shared Accounts,” in *2017 IEEE International Symposium on Multimedia (ISM)*, 2017, pp. 256–259, doi: 10.1109/ISM.2017.43.

[29] K. Mao, J. Niu, X. Liu, S. Tang, L. Liao, and T.-S. Chua, “A patience-aware recommendation scheme for shared accounts on mobile devices,” *IEEE Trans. Knowl. Data Eng.*, p. 1, 2021, doi: 10.1109/TKDE.2021.3069002.

[30] X. Wen, Z. Peng, S. Huang, S. Wang, and P. S. Yu, “MISS: A Multi-user Identification Network for Shared-Account Session-Aware Recommendation BT - Database Systems for Advanced Applications,” 2021, pp. 228–243.

[31] C. F. Caiafa, J. Solé-Casals, P. Marti-Puig, S. Zhe, and T. Tanaka, “Decomposition Methods for Machine Learning with Small, Incomplete or Noisy Datasets,” *Appl. Sci.*, vol. 10, no. 23, 2020, doi: 10.3390/app10238481.

[32] K. P. F.R.S., “LIII. On lines and planes of closest fit to systems of points in space,” *Philos. Mag. Ser. 1*, vol. 2, pp. 559–572, 1901, [Online]. Available: https://api.semanticscholar.org/CorpusID:125037489.

[33] G. D. Clifford, “Chapter 15 - BLIND SOURCE SEPARATION: Principal \& Independent Component Analysis,” 2005, [Online]. Available: https://api.semanticscholar.org/CorpusID:36711606.

[34] T.-W. Lee, M. S. Lewicki, M. A. Girolami, and T. J. Sejnowski, “Blind source separation of more sources than mixtures using overcomplete representations,” *IEEE Signal Process. Lett.*, vol. 6, pp. 87–90, 1999, [Online]. Available: https://api.semanticscholar.org/CorpusID:6298687.

[35] M. S. Lewicki and T. J. Sejnowski, “Learning Overcomplete Representations,” *Neural Comput.*, vol. 12, pp. 337–365, 2000, [Online]. Available: https://api.semanticscholar.org/CorpusID:6254191.

[36] C. Ding, X. He, and H. D. Simon, “On the equivalence of nonnegative matrix factorization and spectral clustering,” in *Proceedings of the 2005 SIAM international conference on data mining*, 2005, pp. 606–610.

[37] R. Zdunek, “Data Clustering with Semi-binary Nonnegative Matrix Factorization BT - Artificial Intelligence and Soft Computing – ICAISC 2008,” 2008, pp. 705–716.

[38] D. Kuang, J. Choo, and H. Park, “Nonnegative Matrix Factorization for Interactive Topic Modeling and Document Clustering BT - Partitional Clustering Algorithms,” M. E. Celebi, Ed. Cham: Springer International Publishing, 2015, pp. 215–243.

[39] Z. Zhang, T. Li, C. Ding, and X. Zhang, “Binary Matrix Factorization with Applications,” in *Seventh IEEE International Conference on Data Mining (ICDM 2007)*, 2007, pp. 391–400, doi: 10.1109/ICDM.2007.99.

[40] M. Jaggi, “Convex Optimization without Projection Steps,” pp. 1–61, 2011, [Online]. Available: http://arxiv.org/abs/1108.1170.

[41] B. Zitnik, Marinka and Zupan, “Nimfa: A Python Library for Nonnegative Matrix Factorization,” *J. Mach. Learn. Res.*, vol. 13, pp. 849–853, 2012, [Online]. Available: https://github.com/mims-harvard/nimfa?tab=readme-ov-file#cite.

[42] N. Hug, “Surprise: A Python library for recommender systems,” *J. Open Source Softw.*, vol. 5, no. 52, p. 2174, 2020, doi: 10.21105/joss.02174.

[43] M. Mirbod, A. R. Ghatari, S. Saati, and M. Shoar, “Industrial parts change recognition model using machine vision, image processing in the framework of industrial information integration,” *J. Ind. Inf. Integr.*, vol. 26, p. 100277, 2022, doi: https://doi.org/10.1016/j.jii.2021.100277.