Journal of King Saud University – Computer and Information Sciences 35 (2023) 101724

HOSTED BY

ELSEVIER

Contents lists available at ScienceDirect

Journal of King Saud University – Computer and Information Sciences

journal homepage: www.sciencedirect.com

A novel model based collaborative filtering recommender system via truncated ULV decomposition

Fahrettin Horasan [a,*], Ahmet Haşim Yurttakal [b], Selçuk Gündüz [c]

[a] Computer Engineering Department, Faculty of Engineering and Natural Sciences, Kırıkkale University, Kırıkkale, Turkey
[b] Computer Engineering Department, Engineering Faculty, Afyon Kocatepe University, Afyon, Turkey
[c] Internet and Information Technologies Management, Afyon Kocatepe University, Afyon, Turkey

ARTICLE INFO

Article history:
Received 21 May 2023
Revised 13 July 2023
Accepted 20 August 2023
Available online 24 August 2023

Keywords:
Recommender systems
Collaborative filtering
Information filtering
Truncated-ULV decomposition
Social recommendation

ABSTRACT

Collaborative filtering is a technique that takes into account the common characteristics of users and items in recommender systems. Matrix decompositions are one of the most used techniques in collaborative filtering based recommendation systems. Singular Value Decomposition (SVD) and Non-negative Matrix Factorization (NMF) based approaches are widely used. Although they are quite good at dealing with the scalability problem, their complexities are high. In this study, the Truncated-ULV decomposition (T-ULVD) technique was used as an alternative technique to improve the accuracy and quality of recommendations. The proposed method has been tested with Movielens 100 k, Movielens 1 M, Filmtrust, and Netflix datasets, which are widely used in recommender system researches. In order to assess the performance of the proposed model, standart metrics (MAE, RMSE, precision, recall, and F1 score) were used. It is seen that while progress was achieved in all experiments with the T-ULVD compared to the NMF, very close or better results were obtained compared to the SVD. Moreover, this study may guide T-ULVD based future studies on solving the cold-start problem and reducing the sparsity in collaborative filtering based recommender systems.

1. Introduction

Recently, advancing access to the information resources and the increase on the information resources have canalized researchers to investigate how much we have reached the accurate information. Some information resources require users to evaluate the content they access qualitatively or quantitatively to increase the access to the correct information (Anandhan et al., 2018; Ricci et al., 2011; Zhao et al., 2018). These resources use a module called recommender system, which uses feedback from users to provide better service. In the recommender systems, the feedback that users leave to the information they access is processed and the information that may be able to be liked by the users is recommended (Ekstrand et al., 2011; Zhao et al., 2018). Recommender systems typically serve users in many areas such as music, movies, videos, web pages, digital products, TV programs, search engines, e-commerce applications (Bennett et al., 2007; Chen and Fong, 2010; Healey, 2009; Linden et al., 2003). In addition, considering the financial gain to be provided, the developed recommender systems have not been able to fully meet the needs of the users yet. Therefore, researchers continue to work on the development of the recommender systems.

Recommendation systems are usually classified into thee categories: content-based filtering, collaborative filtering, and hybrid recommendation approaches (Adomavicius and Tuzhilin, 2005; Anandhan et al., 2018; Ricci et al., 2011). Each filtering method has some advantages and disadvantages. Disadvantages of the collaborative filtering are scalability, sparsity and cold start problems (Herce-Zelaya et al., 2020; Kuo et al., 2021; Singh, 2020). The content-based filtering technique is a method that proposes new items based on the items that the user liked in the past. This method examines the items that the active user has previously rated in the gigantic item pool. Then the new items that the user has never faced before are recommended. These items are similar to the items previously rated by the user. The problem mentioned is called the overspecialized recommendations (Aggarwal, 2016a; Rana et al., 2020). Coming to the advantages of the proposed

* Corresponding author.
E-mail address: fhorasan@kku.edu.tr (F. Horasan).

Peer review under responsibility of King Saud University.

ELSEVIER Production and hosting by Elsevier

https://doi.org/10.1016/j.jksuci.2023.101724

collaborative filtering techniques, the following issues can be considered. In collaborative filtering, only the rating matrix is needed during the training of the model. In the model, there is no need for any other data about the user and the item. It is also a successful method of presenting the right content that will be of interest to users, since similar users are taken into account. The Collaborative filtering technique examines all users and users' feedbacks on the items. Thus, new items could be recommended to the active user by taking into consideration the similar characteristics of the other users. This method uses a matrix called user-item matrix which contains the users' feedbacks. It can reduce the success of the recommender system because the user-term matrix is very sparse. This disadvantage is called as sparsity problem (Kuo et al., 2021). The cold start problem is that there is no any rating yet when a new user or item is included in the system (Herce-Zelaya et al., 2020; Wahab et al., 2016; Wahab et al., 2020). Thus, an item that any user has not rated yet is represented in the user-item matrix. This may reduce the accuracies of possible predictions for the added newly user/item. Another problem is scalability of recommender systems. The recommender systems process a large matrix of users and items in real time and recommend new items for each user. Since this system produces outputs according to increasing and gigantic input, this scalability problem arises (Jie et al., 2015; Singh, 2020). Researchers have proposed low-rank matrix decomposition techniques such as the SVD, regularized SVD, NMF, and probablistic matrix factorization to deal with the scalability problem (Jalili et al., 2018; Koren et al., 2009; Ma et al., 2008; Mehta and Rana, 2017). Social regulation is applied to reduce the losses appeared after dimension reduction which decreased time complexity. In this social regularization stage, the efficiency of the recommendation system is increased Jalili et al. (2018), Ma et al. (2008).

The general focus of the research is to reduce the scalability problem and reduce the computational cost. In addition, the aim is to investigate the usability of the alternative matrix decomposition technique T-ULVD in recommender systems. Finally, a more efficient model is expected to be obtained.

Matrix decompositions are mathematical model-based techniques used in the collaborative recommender systems. With this research, the ULVD technique is proposed as a mathematical model for use in the collaborative recommender systems. The T-ULVD technique is used as alternative to the SVD and NMF. In the literature, the T-ULVD is used to some fields including text analysis (Horasan et al., 2019; Lopez et al., 2018), data science (Jalili et al., 2018; Zhang et al., 2020), and signal processing (Kaloorazi and de Lamare, 2018a). In these areas T-ULVD has shown promising performance. In this context, use of T-ULVD is proposed to solve the scalability problem in the recommender systems. In addition, it's effect on the performance of the system is examined. The main contributions of this study are as follows:

- A novel collaborative recommendation system model is developed using the T-ULVD
- The performance of the developed model according to the matrix decompositions, which are frequently used in the literature, has been compared and assessed.
- The benchmarked and the proposed models have been tested with real datasets and their performance is assessed with commonly used metrics in recommendation systems.
- A dimension reduction approach is applied and time complexity.
- Better results is obtained in almost all tests with the proposed model.

In the continuation of this paper, Section 2 presents a review of the past work in the field. Section 3 explains the preliminaries on the subject and the details of the techniques mentioned in the study. The test results performed on the data sets, which are commonly used in the recommendation systems are presented in Section 4. Finally, the study is discussed and concluded in Section 5.

2. Related work

In this section, studies involving model-based collaborative filtering techniques using matrix decompositions are explained.

These systems, which are used to make recommendations based on users' purchasing habits or interests, are constantly being developed to improve the commercial gain and solve the problem of access to the correct information resources. The first idea about the collaborative filtering was proposed by Goldberg in 1992 in a study called Tapestry (Goldberg et al., 1992). Then, according to Breese et al., the collaborative recommender systems have been investigated in two groups as model based and memory based (Breese et al., 2013; Horasan, 2022). The Memory-based collaborative recommendation systems use user-item matrix to calculate the similarity between users/items. The main target of these systems is to calculate similarities of users or items and find the most similar to recommend new items to active users. However, the scalability problem emerges because they have to keep the entire user-item matrix in the memory. In addition, the cold start and the sparsity problems are also encountered (Kuo et al., 2021; Li et al., 2021; Tahmasebi et al., 2021; Symeonidis et al., 2006). The model-based recommender systems require the training of known ratings for the estimation of an unknown rating. Well known models are matrix factorization, bayesian classifiers, genetic algorithms and fuzzy systems. The sparsity and scalability problems are also observed in model-based systems (Bokde et al., 2015; Breese et al., 2013).

In order to overcome the scalability problem, matrix decomposition is usually applied. In this way, the dimensions of user and item vectors are reduced Aghdam et al. (2015), Sarwar et al. (2002), Luo et al. (2014). In the studies on collaborative filtering-based recommender systems, the SVD is generally used as matrix decomposition. Due to the facts that the computation of the SVD's time complexity is very high, there are studies on alternative matrix decompositions for the recommender systems (Aghdam et al., 2015; Himabindu et al., 2018; Koren et al., 2009; Luo et al., 2014).

The first study to use the SVD to the recommendation system was carried out by the authors in Sarwar et al. (2000). They adapted the technique of latent semantic analysis, which is usually applied in text mining, to the collaborative filtering (Deerwester et al., 1990; Valdez et al., 2018). Moreover, the fact that the SVD is a dimension reduction method has a significant effect on the solution of scalability problems (Jiao et al., 2019; Koren et al., 2009; Ma et al., 2008; Sarwar et al., 2000; Yuan et al., 2019).

The model based recommendation approaches using the matrix decompositions have become very popular because of their high accuracy and scalability (Jiao et al., 2019; Mehta and Rana, 2017; Parvin et al., 2019). The probabilistic latent semantic analysis based collaborative filtering (Hofmann, 2004; Huang et al., 2019; Li et al., 2018), the regularized singular value decomposition for collaborative filtering (Ma et al., 2008; Li et al., 2018; Paterek, 2007) and the expectation maximization algorithms for matrix decomposition (Kurucz et al., 2007; Luo et al., 2015) are some of the most well-known approaches.

Model-based collaborative filtering studies using matrix decompositions have increased further after Netflix prize competitions. This is so because the high performance of a study based on the regularized matrix decomposition in these challenges (Takács et al. (2008).

In recent years, another purpose of using matrix decompositions is to find a solution to the sparsity and cold start problems (Herce-Zelaya et al., 2020; Ran et al., 2022; Liu et al., 2011; Singh, 2020; Yuan et al., 2019). In the previous studies, it is seen that the SVD is generally used in model-based collaborative recommendation systems (Aggarwal, 2016b; Koren et al., 2009; Paterek, 2007; Ricci et al., 2011). There are also studies in which the NMF is used as an alternative to the SVD or to obtain more efficient results (Gouvert et al., 2020; Luo et al., 2014; Parvin et al., 2019). In addition to these, there are also studies in which matrix decomposition methods are used and new models are proposed (Bin and Sun, 2021; He et al., 2018; Jakomin et al., 2020; Wang et al., 2019; Hao et al., 2018).

In order to examine it more clearly, the models using the matrix decomposition techniques for the collaborative filtering-based recommender systems and their features are listed in Table 1. Some of the techniques using the SVD or NMF do not include the regularization process. The proposed study includes the regularization process after the dimension reduction process. Five of the listed studies used data imputation methods to deal with the sparsity problem in their recommendation system. Another important feature that draws attention in the collaborative recommendation systems is the way of feedback received from the users. There are two types of feedback: explicit and implicit. In systems that use explicit feedback, users are asked to rate items explicitly. In the other, the system itself generates feedback based on the user's actions and interest. Researchers have obtained implicit feedback by taking into account the neighborhood of users and items with the technique called social regularization. As can be seen from the table, it is seen that both types of feedback are used in most studies. Since the proposed study includes this social regularization process alongside the rating matrix, both explicit and implicit feedback is taken into account. Although the matrix decompositions are generally used in the model-based collaborative recommendation systems, it has been observed that they are also used in the memory-based collaborative recommendation studies. As is the case with most, a novel model-based approach is suggested in this study.

When the above-mentioned studies are examined, the following features have been remarkable. The dimension reduction is applied to the user-item matrix by means of the matrix decomposition. In order to obtain the k-dimensional vector space after the matrix decomposition process, a dimension reduction approach called rank-*k* approximation is applied. The rank-*k* approximation has the ability to remove erroneous and ineffective features in the user-item matrix, so it can be considered as a feature selection method. In this way, it is aimed to develop a system having low computational complexity and high efficiency.

Unlike the well-known matrix decompositions, a new technique is used in the proposed work. As a result of the tests, it was observed that more efficient results were obtained. In addition, the ability of the proposed T-ULVD technique to perform operations without the need for recalculation in cases where the matrix is updated is better than other techniques, which is an important reason for preference. Since the T-ULVD decomposition has not been used in recommender systems before, it is foreseen that it will shed light on new studies after this study. In addition, the parameters used to regularize the matrix can be determined by new optimization algorithms in the literature (Abualigah et al., 2021a; Abualigah et al., 2021b; Ezugwu et al., 2022).

It is seen that matrix decompositions are used in a very wide area. Some of these areas are information retrieval systems (Horasan et al., 2019), sparse matrix problems (Kaloorazi and Chen, 2019), recommender systems (He et al., 2018), scientific computing (Kaloorazi and Chen, 2021) and image processing (Rasti et al., 2018), etc. In this study, the T-ULVD is used as an alternative matrix decomposition technique to the SVD and NMF. In previous studies, the T-ULVD has provided efficient results in the signal processing (Kaloorazi and de Lamare, 2018a; Kaloorazi and Chen, 2021), data science (Jalili et al., 2018; Zhang et al., 2020), and text mining (Horasan et al., 2019) studies. The T-ULVD includes the dimension reduction process of the vector space obtained with ULV Decomposition (ULVD) (Horasan et al., 2019). The ULVD technique, which is less costly and efficient than the SVD, was proposed by Barlow and Erbay (2009). Researchers generally use the T-ULVD technique for noise reduction, dimension reduction and vectorization (Barlow and Erbay, 2009; Horasan et al., 2019; Kaloorazi and de Lamare, 2018b; Rebrova et al., 2018). In particular, the vector space model obtained with the T-ULVD is updated faster than other matrix decompositions.Therefore, this technique is recommended due to its simplicity in updating the vector space and low computational complexity (Horasan et al., 2019). This study reveals that the T-ULVD technique can be used as a good alternative method in recommender systems. Thus, it is aimed to be a study that gives direction to further studies.

3. Methods

In this part of the article, first of all, the user-item (rating) matrix and its properties are mentioned in Section 3.1. In the following sections, the SVD (in Section 3.2), NMF (in Section 3.3), and T-ULVD (in Section 3.4), are described, respectively. Section 3.5 describes dimension reduction and obtaining reduced vector space. Finally, regularization of the reduced vector space and performing the rating predictions are described in Section 3.6. In addition, the

Table 1
Feature comparison of the matrix decomposition based collaborative recommender systems.

	Techniques	Regularization	Dimension Reduction	Data Imputation	Explicit rating	Implicit rating	Approach
Ref. Ma et al. (2008)	SVD	Yes	Yes	No	Yes	Yes	Model Based
Ref. Horasan (2022)	SVD	No	Yes	No	Yes	No	Memory Based
Ref. Symeonidis et al. (2006)	SVD	No	Yes	No	Yes	No	Memory Based
Ref. Aghdam et al. (2015)	NMF	Yes	Yes	No	Yes	Yes	Model Based
Ref. Luo et al. (2014)	NMF	Yes	Yes	No	Yes	Yes	Model Based
Ref. Sarwar et al. (2000)	SVD	No	Yes	Yes	Yes	No	Model Based
Ref. Jiao et al. (2019)	SVD	Yes	Yes	No	Yes	Yes	Model Based
Ref. Yuan et al. (2019)	SVD	Yes	Yes	Yes	Yes	Yes	Model Based
Ref. Parvin et al. (2019)	NMF	Yes	Yes	No	Yes	Yes	Model Based
Ref. Li et al. (2018)	SVD	Yes	Yes	No	Yes	Yes	Model Based
Ref. Paterek (2007)	SVD	Yes	Yes	No	Yes	No	Model Based
Ref. Kurucz et al. (2007)	SVD	Yes	Yes	Yes	Yes	Yes	Model Based
Ref. Ran et al. (2022)	NMF	Yes	Yes	Yes	Yes	Yes	Model Based
Ref. Yuan et al. (2021)	SVD	Yes	Yes	Yes	Yes	Yes	Model Based
Proposed	T-ULVD	Yes	Yes	No	Yes	Yes	Model Based

flowchart of regularized matrix factorization based recommendation systems is shown in Fig. 1.

Algorithm 1. Regularized T-ULVD based collaborative filtering Algorithm

---

<u>function Regularized TUL VD</u> $(A, k, \alpha, \lambda_1, \lambda_2, UB, IB)$
Input Rating Matrix (A), $Rank - k$ value (k), learning rate parameters (α), regularization parameters (λ_1, λ_2), user and item bias (UB, IB)
Output Regularized singular vector matrices $(\widetilde{K_k}, \widetilde{M_k^T})$
Start
$[K, L, M] = ULVD(A)$
$\left[K_k, L_k, M_k^T\right] = DimensionReduction(K, L, M, k)$
$\left[\widetilde{K_k}, \widetilde{M_k^T}\right] = RegularizeTheReducedVectorSpace\left(K_k, L_k, M_k^T, UB, IB\right)$
Return $[\widetilde{K_k}, \widetilde{M_k^T}]$
End.

It was mentioned in the previous section that T-ULVD is a dimension reduction based matrix decomposition. Therefore, the proposed model both reduces the computational cost and provides more efficient results since it does not include noise-like features. It is used as an alternative matrix decomposition model to deal with the scalability problem. Moreover, social regularization is implemented after T-ULVD process. In this way, it is ensured that the recommendations are more accurate by considering both the rating matrix as well as the neighborhood of the users and items. The flowchart where we can see the stages of the proposed model is illustrated in Fig. 2 and Fig. 3.

The theme of the model is based on the TULV decomposition, dimension reduction, and regulation process, respectively. In addition, the proposed model is explained in detail in Algorithm 1. As seen in the algorithm, the dimension reduction was performed after applying the ULVD to the rating matrix. In order to minimize the semantic losses in the low-dimensional vectors obtained, regularization process has been carried out. Thus, $\widetilde{K_k}$ and $\widetilde{M_k^T}$ are obtained as user and item vectors, respectively. In this section, the steps of the algorithm are explained in detail.

3.1. User-item matrix

The datasets (The Rating Matrices) used in recommender systems are generally expressed in a very sparse matrix. This situation decreases the efficiency as well as negatively affects the processing time (Aggarwal, 2016a). The rating matrices (Table 2) are already in the form of the sparse matrix (Table 3). Each indices of 0 in the rating matrices is not included in the sparse matrix. In Table 2the U_i and I_j represents the i-th user and the j-th item, respectively.

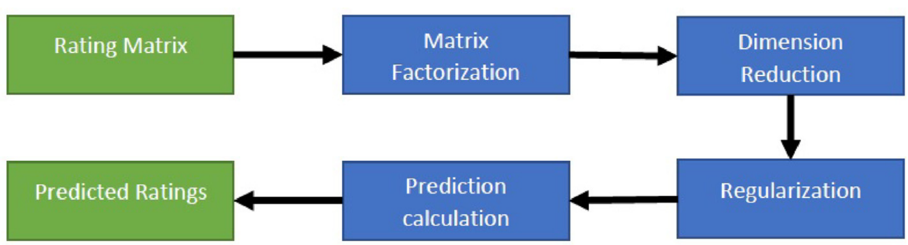


Fig. 1. The flowchart of regularized matrix factorization based collaborative filtering.

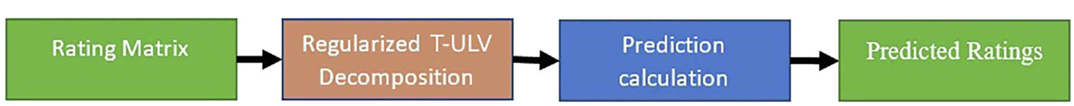


Fig. 2. The flowchart of regularized T-ULVD based collaborative filtering.

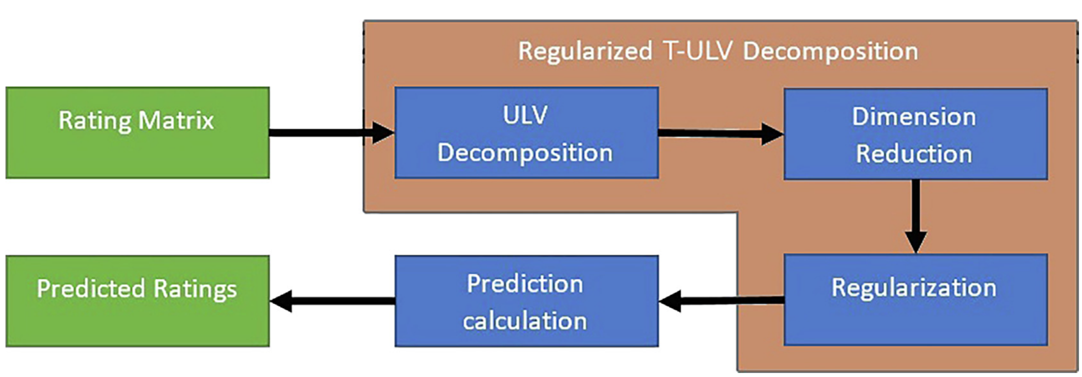


Fig. 3. The detailed flowchart of regularized T-ULVD based collaborative filtering.

Table 2
Rating matrix.

	I1	I2	I3	I4
U1	3	0	5	0
U2	0	1	4	0
U3	0	4	0	3
U4	2	0	5	0

Table 3
Sparse rating matrix.

User-Id	Item-Id	Rating
1	1	3
1	3	5
2	2	1
2	3	4
3	2	4
3	4	3
4	1	2
4	3	5

The rating matrix is symbolized as $A \in \mathbb{R}^{m \times n}$ and this matrix is represented as

$$A = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \ldots & a_{mn} \end{pmatrix} \tag{1}$$

. Here a_{ij} represents the rating score that the i-th user made to the j-th item.

$$X_i = \begin{pmatrix} a_{i1} & a_{i2} & \cdots & a_{in} \end{pmatrix} \tag{2}$$

$$Y_j = \begin{pmatrix} a_{1j} \\ a_{2j} \\ \vdots \\ a_{mj} \end{pmatrix} \tag{3}$$

.

In the above equations, the user vector for the i-th user is denoted with X_i, while the item vector for the j-th item is represented by Y_j. So the matrix A can be represented by

$$A = [A_{ij}] = \begin{pmatrix} X_1 \\ X_2 \\ \vdots \\ X_m \end{pmatrix} = \begin{pmatrix} Y_1 & Y_2 & \cdots & Y_n \end{pmatrix} \tag{4}$$

.

3.2. Singular value decomposition

The SVD is the most used matrix decomposition technique in model-based collaborative recommender systems. The latent semantic structure between the users and the items is extracted thanks to the SVD. Thus, it becomes possible to calculate the similarities between the users/items more accurately. The SVD processing is applied to the user-item matrix called A ($A \in \mathbb{R}^{m \times n}$ and $m \geqslant n$) in the recommendation systems as in Eq. 5.

$$A = U \Sigma V^T. \tag{5}$$

As can be seen from here, U, Σ and V^T matrices are obtained from the matrix A after the SVD processing. $U \in \mathbb{R}^{m \times n}$ and $V \in \mathbb{R}^{m \times n}$ matrices are orthogonal matrices ($UU^T = U^T U = I_m$ and $VV^T = V^T V = I_n$), Σ is the diagonal matrix where the singular values of the A matrix are located in descending order ($\Sigma = \mathrm{diag}(\sigma_1, \sigma_2, \cdots, \sigma_n)$).

3.3. Non-negative matrix factorization

The NMF processing of A with numerical rank k is performed by the Eq. 6, provided that $A \in \mathbb{R}^{m \times n}$ and $m \geqslant k \geqslant n$

$$A = WH. \tag{6}$$

The matrices in the equation are of the form $W \in \mathbb{R}^{m \times k}$ and $H \in \mathbb{R}^{k \times n}$. As a result of this decomposition, there is approximation, not equality. Therefore, in order to minimize the difference between A and $W.H$, the optimum W and H are found after the optimization process in Eq. 7.

$$\min_{W,H} f(W,H) = \sum_{i=1}^{n} \sum_{j=1}^{m} \left(A_{i,j} - (WH)_{i,j} \right)^2 \\ \text{Subject to } W_{i,k} \geqslant 0 H_{k,j} \geqslant 0, \forall_i, j, k. \tag{7}$$

3.4. Truncated-ULV decomposition

Provided that $A \in \mathbb{R}^{m \times n}$ and $m \geqslant n$, the T-ULVD of A is performed with

$$A = KLM^T + E. \tag{8}$$

Here $K \in \mathbb{R}^{m \times n}$ and $M \in \mathbb{R}^{n \times n}$. Also K and M are orthogonal matrices ($KK^T = K^T K = I_m$ and $MM^T = M^T M = I_n$). $L \in \mathbb{R}^{m \times n}$ is the non-singular lower triangular matrix and $E \in \mathbb{R}^{m \times n}$ is an error matrix.

3.5. Rank-k approximation and obtaining vector space

Some features of the rating matrix negatively affect the success of the recommendation systems. One of them is that the rating matrix is too sparse. In other words, the number of items rated by each user is very low compared to the total number of items. The inclusion of data that does not affect the meaning and is called noise also reduces efficiency. In addition, the large number of users and items also causes scaling problem. In order to overcome these problems and reduce their effects, the dimension reduction is applied after the matrix decomposition processing.

The dimension reduction called rank-k approximation is applied to the components obtained after the A matrix is factored, provided that $m \geqslant k \geqslant n$. In the SVD applied model, dimension-reduced A_k matrix is obtained according to Eq. 9.

$$A_k = U_k \Sigma_k V_k^T \tag{9}$$

$\Sigma_k \in \mathbb{R}^{k \times k}$ and $\Sigma_k = \mathrm{diag}(\sigma_1, \sigma_2, \cdots, \sigma_k)$ in Eq. 9 indicates a diagonal matrix, $U_k \in \mathbb{R}^{m \times k}$ indicates the first k columns of U, and $V_k^T \in \mathbb{R}^{k \times n}$ represents the first k rows of V^T. In the model using the T-ULVD, dimension-reduced A_k matrix could be calculated through Eq. 10.

$$A_k = K_k L_k M_k^T. \tag{10}$$

$L_k \in \mathbb{R}^{k \times k}$ is the non-singular lower triangular matrix, $K_k \in \mathbb{R}^{m \times k}$ represents the first k columns of K, and $M_k^T \in \mathbb{R}^{k \times n}$ indicates the first k rows of M^T. In the NMF, matrix decomposition is already performed in a reduced dimension. Therefore, there is no need for the dimension reduction after the NMF processing.

The optimum k value is obtained when the difference between the singular values in the Σ matrix is maximum. That is, the difference between the k-th singular value and the $k+1$-th singular value is expected to be higher than the difference between the

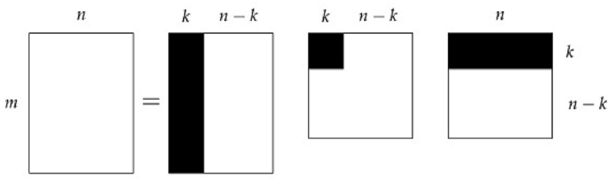


Fig. 4. Rank-k approach.

Table 4
Statistics of the datasets.

Dataset	Num of Ratings	Num. of User	Num. of Items
Movielens 100 k	943	1682	100000
Movielens 1 M	6040	3900	1000209
Filmtrust	1508	2071	35497
Netflix	979	1813	184923

other singular values. A_k, which is the rank-k approximation of A, is used instead of A. The dimendison reduction process applied to an $m \times n$ dimensional matrix with the rank-k approach is schematized in Fig. 4. For the SVD, NMF, and T-ULVD, U_k, W, and K_k represent vectors of users, respectively. Likewise, V_k^T, H, and M_k^T are vectors of the items.

3.6. Regularizing the reduced vector space and predicting the ratings

In the vector space-based recommendation systems, $R \in \mathbb{R}^{m\times n}$ matrix obtained from Eq. 11 is generally used for predictions after the matrix decomposition process.

$$R \approx XY. \tag{11}$$

Here X and Y are determined according to the matrix decomposition technique used. If the SVD is used, then $X = U_k, Y = V_k^T$, For the NMF, $X = W, Y = H$. Finally, it is determined as $X = K_k, Y = M_k^T$ in the T-ULVD technique. In the R matrix, there may be data loss due to the dimension reduction in X and Y matrices. For this reason, X and Y are regulated by using the A matrix in order to reduce losses and increase the prediction accuracy. Thus, we obtain the rating $(P_{i,j})$ given to the j-th item by the i-th user with the following Eq. 12.

$$P_{i,j} = X_i \cdot Y_j \tag{12}$$

For this purpose, the following loss function (Eq. 13) should be minimized.

$$\min_{X,Y} \|A - P\|_F. \tag{13}$$

Since matrix A is very sparse, this Equation is updated as follows:

$$\min_{X,Y} \sum_{i=1}^{m}\sum_{j=1}^{n} I(i,j).(r_{i,j} - P_{i,j})^2 \tag{14}$$

where $I(i,j)$ is determined by whether the i-th user has made a rating to the j-th item. It takes the value 1 if it has made a rating, and 0 otherwise. To avoid over-fitting, the Eq. 14 is converted to the following form (Eq. 15). Thus, regularization of the matrices is carried out.

$$in_{X,Y} \sum_{i=1}^{m}\sum_{j=1}^{n} I(i,j).(r_{i,j} - P_{i,j})^2 + \frac{\lambda_1}{2}\|X_i\|_F^2 + \frac{\lambda_2}{2}\|Y_i\|_F^2 \tag{15}$$

In addition, if biases (UB, IB) are added for both the users and the items, the prediction and optimization Equations will be as in Eq. 16 and Eq. 17, respectively. λ_1 and λ_2 usually have the same value, so a single variable a was used.

$$P_{i,j} = \mu + UB_i + IB_j + X_i Y_j \tag{16}$$

$$\min_{X,Y} \sum_{i=1}^{m}\sum_{j=1}^{n} I(i,j).(r_{i,j} - P_{i,j})^2 + \frac{\lambda}{2}\left(\|X_i\|_F^2 + \|Y_i\|_F^2 + UB_i + IB_j\right) \tag{17}$$

Here μ represents the overall average rating. Usually, gradient descent is used to minimize the loss function. Thus the multipliers of the matrix are regularized. Where $e_{i,j} = r_{i,j} - X_i Y_j, X_i$ and Y_i are updated using the Eq. 18 and Eq. 19 for the SVD and T-ULVD, respectively.

$$X_i = X_i + \alpha(e_{i,j} Y_i - \lambda_1 X_i) \tag{18}$$

$$Y_j = Y_j + \alpha(e_{i,j} X_i - \lambda_2 Y_j) \tag{19}$$

F_U represents the items that the active user gives feedback, and F_I represents the users that give feedback to the active item. So for the NMF, X_i and Y_i are updated using Eq. 20 and Eq. 21, respectively.

$$X_i = X_i \frac{\sum_{i\in F_U} Y_j.P_{i,j}}{\sum_{i\in F_U} Y_j.r_{i,j} + \lambda_1 |F_U| X_i} \tag{20}$$

$$Y_j = Y_j \frac{\sum_{i\in F_I} X_j.P_{i,j}}{\sum_{i\in F_I} X_i.r_{i,j} + \lambda_2 |F_I| Y_j} \tag{21}$$

The biases (UB, IB) used for all three techniques are updated by the following Equations.

$$UB_i = UB_i + \alpha(e_{i,j} - \lambda_1 UB_i) \tag{22}$$

$$IB_j = IB_j + \alpha(e_{i,j} - \lambda_2 IB_j) \tag{23}$$

In the Equations, λ_1 and λ_2 are the regularization parameters and α is the learning rate. After the specified number of iterations, the prediction of the rating of the j-th item made by the i-th user $(P_{i,j})$ is calculated by Eq. 16.

4. Experimental analyzes

This section describes the experiments performed for the proposed and compared models. The datasets used in the experiments are Movielens 100 k, Movielens 1 M[1], Filmtrust[2], and Netflix[3]. The number of users, items and total ratings in the Netflix dataset is very large. Therefore, researchers usually work on a subset of the Netflix dataset (Hao et al., 2021; Yuan et al., 2019; Yuan et al., 2021). In this study, a subset of user-item matrix containing randomly selected users and items was obtained. Each user in this matrix has at least 20 ratings. Details of the datasets are presented in Table 4. For each dataset, 90% and 10% of the user item matrix are randomly determined as training and test data, respectively. Moreover, a ten-fold cross-validation method was utilized for the verification of the models. All experiments were conducted with Matlab 2018b on a computer that has an Intel processor (2.5 GHz), 32G RAM, and a Win10 operating system.

The computational complexity of the techniques proposed and compared in the study is the same in each method. Accordingly, when the rank-k approach is not applied, it is $O(mn^2)$. However,

[1] http://grouplens.org/datasets/movielens.
[2] https://www.librec.net/datasets.html.
[3] https://www.netflix.com.

when the rank-k approach is applied, it becomes $O(mnk)$. Here the condition $k < min(m, n)$ is provided. Since k is very low compared to m and n, a significant gain is achieved in performance and runtime. Therefore, it can be said to be $O(mn) \approx O(mnk)$. It is also stated that if the decomposed matrix is very sparse, the complexity of the T-ULVD is O(m + n) (Horasan et al., 2019). In this way, considering the sparsity of matrices used in the collaborative filtering based recommender systems, the proposed algorithm can be explained as O(m + n).

4.1. Metrics

Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) metrics are commonly utilized to measure the performance of the recommender systems (Jalili et al., 2018). MAE and RMSE are calculated with Eq. 24 and Eq. 25, respectively.

$$MAE = \frac{\sum_{i,j \in T} |A_{i,j} - r_{i,j}|}{|T|} \tag{24}$$

$$RMSE = \sqrt{\frac{\sum_{i,j \in T} (A_{i,j} - r_{i,j})^2}{|T|}} \tag{25}$$

Here, $A_{i,j}$ represents the i-th user's rating for the j-th item. $r_{i,j}$ is used to predict the i'th user's degree for the j-th item. $|T|$ is the total number of tests. It is understood that the lower the results of both metrics, the better the performance.

On the other hand, Precision, Recall, and F1 score metrics are used to see how accurately the recommended items are predicted in the recommendation systems. In these metrics, the best N items recommended to the user by the system are taken into account. Accordingly, the Precision (P_N) and the Recall(R_N) metrics are calculated with Eq. 26 and Eq. 27, respectively. Here N indicates the number of items recommended to users.

$$P_N = \frac{TP}{TP + FP} \tag{26}$$

$$R_N = \frac{TP}{TP + FN} \tag{27}$$

Here, TP is equal to the number of correctly recommended items, FP represents the number of incorrectly recommended items, and FN denotes the number of non-recommended items. The F1 score metric, in which both Precision and Recall metrics are taken into account, is calculated as follows.

$$F1 = \frac{2 * P_N * R_N}{P_N + R_N} \tag{28}$$

4.2. Performance analysis

The information of the parameters and inputs used in each method in the study are as follows.

(-) The number of iterations to fairly compare techniques was determined as 50.

(-) The regularization parameters (λ_1, λ_2) and the learning rate (α) usually take very small values. You can see the performance

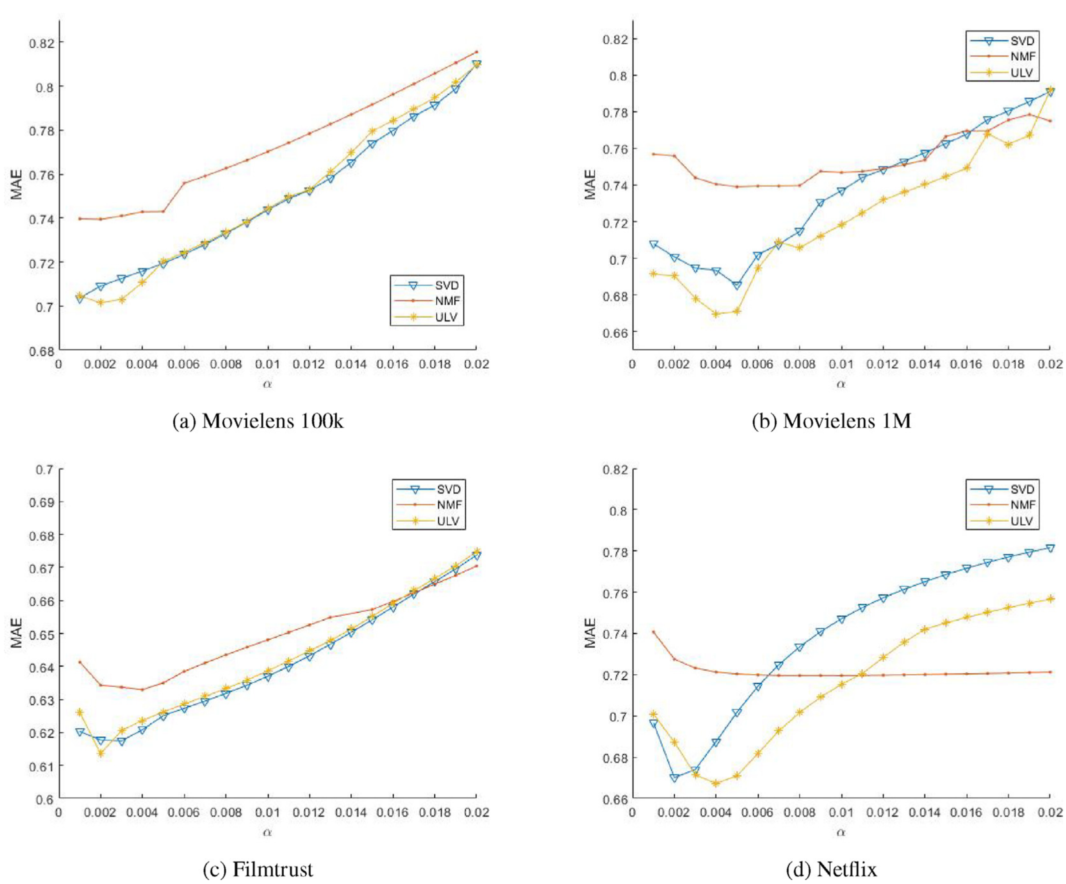
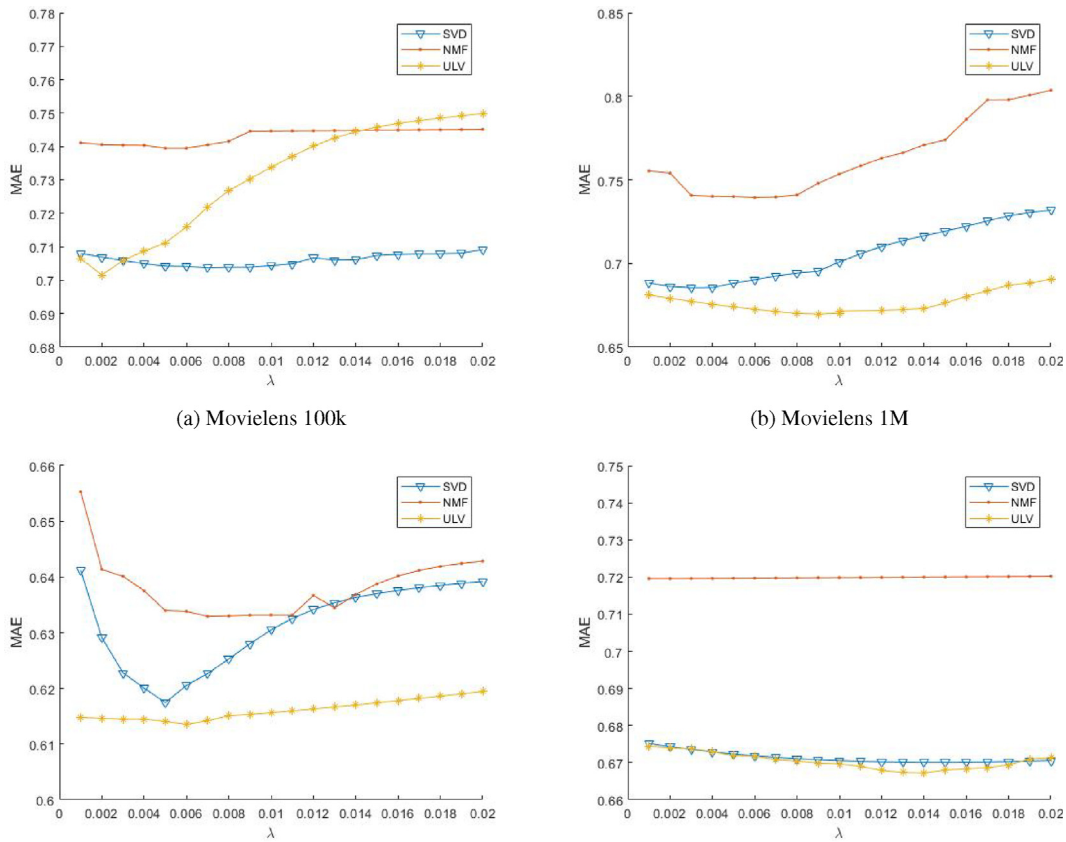


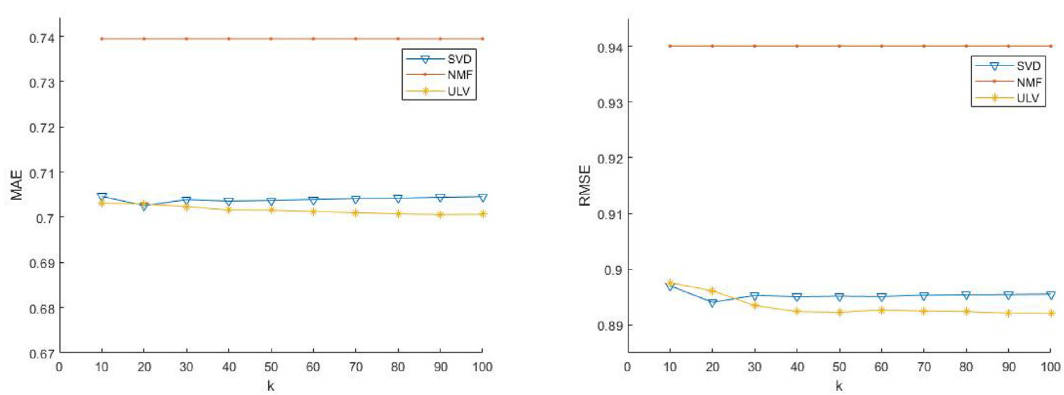
Fig. 5. Selection of α values.



(a) Movielens 100k (b) Movielens 1M

(c) Filmtrust (d) Netflix

Fig. 6. Selection of λ values.



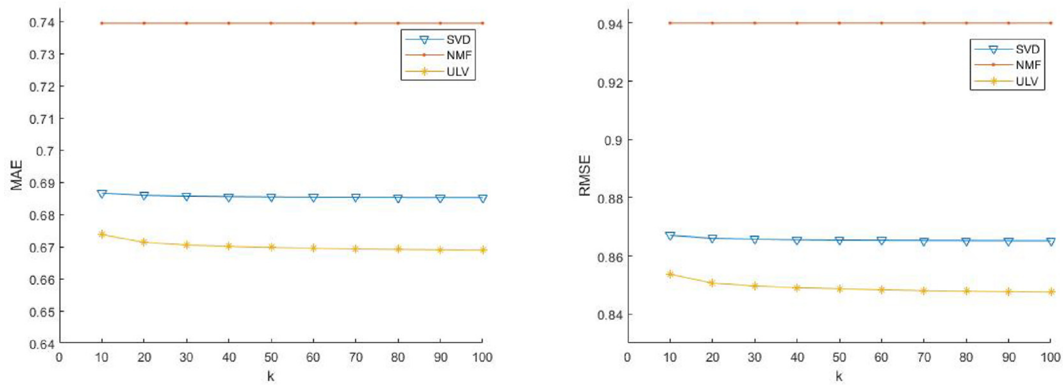
(a) (b)

Fig. 7. MAE and RMSE results according to different rank-k values for the Movielens 100 k dataset.

according to the change of the learning rate variable for an experiment where the rank-k number is determined as 50 in Fig. 5. Likewise, its performance results according to the change of Regularization parameter variables can be seen in Fig. 6.

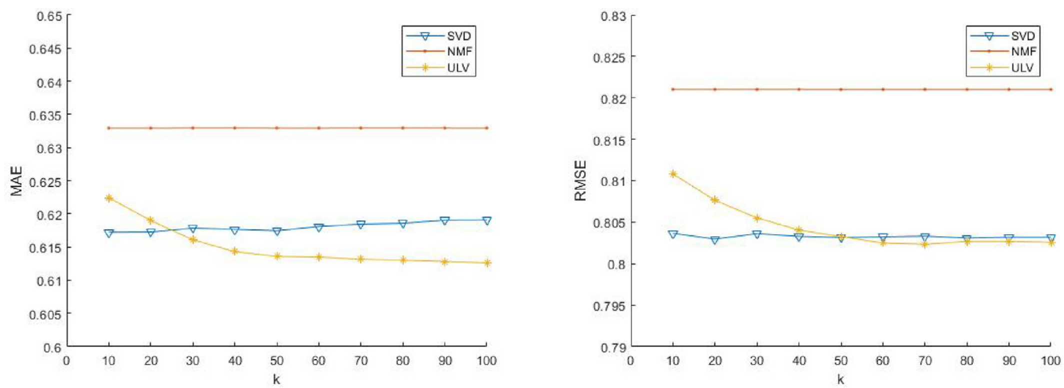
(-) Number of reduced features: Matrix decompositions are a feature extraction method. The extracted features are ranked from the most significant to the most meaningless. For this reason, the dimension reduction is made with the $rank - k$ approach mentioned in Section 3.5. The number of reduced features represents the actual size of the rating matrix (the number of noise-free features). Therefore, experiments were carried out to determine the correct number of features. The k used in the dimension reduction method ($rank - k$) indicates the number of features. After the tests for all the four datasets, you can see the effect of the number of dimensions in Figs. 7–10. It was found that RMSE and MAE metrics did not change much when $k > 50$ on average in all datasets.

Considering the changes in Fig. 5, it was set that $\alpha = 0.001$ for the SVD, $\alpha = 0.002$ for the T-ULVD and $\alpha = 0.002$ for the NMF in



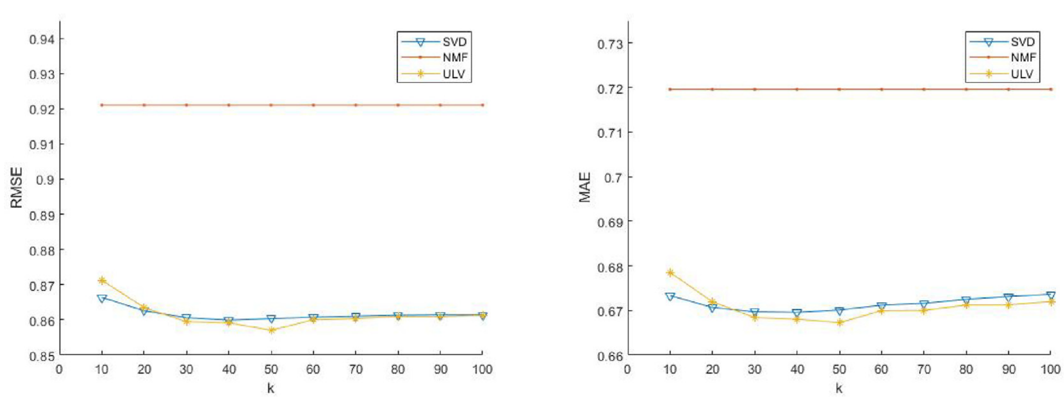
(a) (b)

Fig. 8. MAE and RMSE results according to different rank-*k* values for the Movielens 1 M dataset.



(a) (b)

Fig. 9. MAE and RMSE results according to different rank-*k* values for the Filmtrust dataset.



(a) (b)

Fig. 10. MAE and RMSE results according to different rank-*k* values for the Netflix dataset.

Movielens 100 k. For Movielens 1 M, α was chosen as 0.005 for the SVD, 0.004 for the T-ULVD and 0.007 for the NMF. α was set as 0.003 for the SVD, 0.002 for the T-ULVD and 0.004 for the NMF for Filmtrust. Finally, α was set as 0.002 for the SVD, 0.004 for the T-ULVD and 0.008 for the NMF for the Netflix.

The changes in Fig. 6 were observed. Thus, in Movielens 100 k, theλ value for the SVD, the T-ULVD, and the NMF was set to 0.007, 0.002 and 0.005, respectively. For Movielens 1 M, λ was chosen as 0.003 for the SVD, 0.009 for the T-ULVD and 0.006 for the NMF. For Filmtrust, λ was set as 0.005 for the SVD, 0.006 for the T-ULVD and

Table 5
α and λ values used in the experiments.

Parameters	Dataset	SVD	NMF	T-ULVD
α	Movielens 100 k	0.001	0.002	0.002
	Movielens 1 M	0.005	0.007	0.004
	Filmtrust	0.003	0.004	0.002
	Netflix	0.002	0.008	0.004
λ_1, λ_2	Movielens 100 k	0.007	0.005	0.002
	Movielens 1 M	0.003	0.006	0.009
	Filmtrust	0.005	0.007	0.006
	Netflix	0.016	0.001	0.014

Table 6
Performance comparisons.

Dataset	k	Metrics	SVD	NMF	T-ULVD
Movielens 100 k	10	MAE	0.7046	0.7346	0.7031
		RMSE	0.8970	0.9400	0.8975
		P@10	0.8937	0.8359	0.8992
		R@10	0.7502	0.7415	0.7504
		F1	0.8157	0.7859	0.8181
	50	MAE	0.7037	0.7346	0.7016
		RMSE	0.8952	0.9400	0.8923
		P@10	0.8976	0.8360	0.9005
		R@10	0.7512	0.7417	0.7525
		F1	0.8179	0.7860	0.8199
Movielens 1 M	10	MAE	0.6866	0.7395	0.6737
		RMSE	0.8670	0.9400	0.8537
		P@10	0.9126	0.8923	0.9133
		R@10	0.7949	0.7826	0.7981
		F1	0.8497	0.8339	0.8518
	50	MAE	0.6854	0.7394	0.6697
		RMSE	0.8654	0.9400	0.8487
		P@10	0.9136	0.8926	0.9207
		R@10	0.7964	0.7829	0.8124
		F1	0.8510	0.8342	0.8632
Filmtrust	10	MAE	0.6172	0.6330	0.6224
		RMSE	0.8037	0.8210	0.8108
		P@10	0.8902	0.7863	0.8891
		R@10	0.3628	0.3633	0.3628
		F1	0.5155	0.4970	0.5153
	50	MAE	0.6174	0.6330	0.6135
		RMSE	0.8031	0.8210	0.8031
		P@10	0.8858	0.7863	0.8923
		R@10	0.3631	0.3633	0.3635
		F1	0.5151	0.4970	0.5166
Netflix	10	MAE	0.6734	0.7196	0.6785
		RMSE	0.8664	0.9210	0.8713
		P@10	0.9025	0.8762	0.9092
		R@10	0.7401	0.7704	0.7388
		F1	0.8133	0.8199	0.8152
	50	MAE	0.6701	0.7196	0.6673
		RMSE	0.8603	0.9210	0.8571
		P@10	0.9188	0.8762	0.9281
		R@10	0.7519	0.7704	0.7533
		F1	0.8270	0.8199	0.8316

0.007 for the NMF. Finally, λ was set as 0.016 for the SVD, 0.014 for the T-ULVD and 0.001 for the NMF for the Netflix. Initial bias values for the users and the items were determined as 0.02. The α and λ values chosen for the datasets and techniques used in the experiments are shown in Table 5.

According to Figs. 7–10), the T-ULVD technique gives better results than the SVD and the NMF techniques at almost every condition. While the k value does not have much influence for the NMF, it has significant influence for the SVD and the T-ULVD. For example, for the Movielens 1 M dataset, it can be seen that the performance is better as k increases in the SVD and the T-ULVD methods. In some cases, the best results are obtained at lower k values. For example, in the Movielens 100 k, the best result for the SVD was obtained when $k = 40$. The reason why there is no or very little change is that the regularization process already exists at the base of the NMF technique. In other words, after the current regularization process in the NMF method, a second regularization process does not change the results.

In order to show the performances of the models, the performance results are presented in Table 6 for all datasets, respectively. Accordingly, it can be seen that all results for the Movielens 100 k and the Movielens 1 M are better in the T-ULVD than the SVD and NMF. For the Filmtrust and the Netflix, when the k value is 50, the T-ULVD technique is better than the SVD, and better results are obtained for all k values than the NMF technique. As seen in Table 6, the proposed technique achieved the best performance for all datasets (when k = 50). The performance results obtained from all datasets are acceptable. However, it was observed that the recall and the f1 score results for the Filmtrust could be improved. In addition, it is seen that all metrics are not affected by the number of features (k) in the NMF technique. Therefore, there is no significant change in the results obtained when the number of features is 10 or 50. However, the best results are observed when k = 50 in both SVD and T-ULVD methods.

We can draw the following inferences from Table 6: The T-ULVD technique yielded much better results than the NMF technique. It has been observed that almost all of them are better than the SVD. Those that were not better than the SVD also produced very close results to the SVD. Therefore, it is understood that the T-ULVD based model can be used successfully for collaborative recommendation systems.

5. Discussion and conclusions

In this study, a model-based matrix decomposition using the T-ULVD is proposed as an alternative to the SVD and NMF techniques, which are frequently encountered in the literature. After the experiments;

- In the technique in which the T-ULVD is used, it is seen that better results are obtained than the SVD and NMF used in many studies before.

- The T-ULVD can be used in many models and techniques where the SVD and NMF are used.

- The usability of the T-ULVD in the collaborative recommender systems has been tested and shown to be applicable to other models similar to the proposed model.

The T-ULVD draws attention with its efficiency in information retrieval studies when there are updates in the data set. For this reason, this paper could be guide further studies, such as solving the cold start problem or reducing the sparsity problem. In addition, as in the collaborative recommender systems, the T-ULVD technique can be used as a dimension reduction method in machine learning algorithms using matrix decompositions.

The findings of this work could direct many researches, especially due to its low time complexity and efficiency. This situation becomes even more remarkable when the areas where the SVD and NMF are used are also taken into account. The T-ULVD is similar to these matrix decompositions in many ways. For example, left and right eigenvector matrices obtained after decomposition have similar properties as in other matrix decompositions. The singular value matrix obtained in the SVD method is a diagonal matrix. In the ULVD, the lower left triangle matrix is obtained instead of this matrix. In collaborative filtering-based recommender systems using SVD, left and right singular vector matrices are regulated. The eigenvector matrices are not included in any process. Similarly, the singular vector matrices obtained with T-ULVD are regulated. As a result, the ULVD based model produced more efficient results rather than being a good alternative method. This shows that T-ULVD could be a technique that will find application in many areas where the popular matrix decompositions are used. Therefore, the T-ULVD could be used as an alternative method in many areas such as signal processing, information retrieval, information accessing, data mining, text mining, content based recommender systems, data hiding and digital watermarking. In general, it is predicted that the proposed the T-ULVD could be used in many areas where matrix decompositions are used.

Acknowledgments

This study is supported by Afyon Kocatepe University Scientific Research Projects Coordination Unit. Project Number:23. FENBİL.22.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

Abualigah, Laith, Diabat, Ali, Mirjalili, Seyedali, Elaziz, Mohamed Abd, Gandomi, Amir H., 2021a. The arithmetic optimization algorithm. Comput. Methods Appl. Mech. Eng. 376, 113609.
Abualigah, Laith, Yousri, Dalia, Elaziz, Mohamed Abd, Ewees, Ahmed A, Al-Qaness, Mohammed A.A., Gandomi, Amir H., 2021b. Aquila optimizer: a novel meta-heuristic optimization algorithm. Comput. Ind. Eng. 157, 107250.
Adomavicius, Gediminas, Tuzhilin, Alexander, 2005. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. IEEE Trans. Knowledge Data Eng. 6, 734–749.
Aggarwal, Charu C, 2016a. Content-based recommender systems. In: Recommender Systems. Springer, pp. 139–166.
Aggarwal, Charu C., 2016b. Model-based collaborative filtering. In: Recommender Systems. Springer, pp. 71–138.
Aghdam, Mehdi Hosseinzadeh, Analoui, Morteza, Kabiri, Peyman, 2015. A novel non-negative matrix factorization method for recommender systems. Appl. Mathe. Infr. Sci. 9 (5), 2721.
Anandhan, Anitha, Shuib, Liyana, Ismail, Maizatul Akmar, Mujtaba, Ghulam, 2018. Social media recommender systems: review and open research issues. IEEE Access 6, 15608–15628.
Barlow, Jesse L., Erbay, Hasan, 2009. Modifiable low-rank approximation to a matrix. Num. Linear Algebra Appl. 16 (10), 833–860.
Bennett James, Lanning Stan, et al., 2007. The netflix prize. In: Proceedings of KDD Cup and Workshop, vol. 2007, New York, NY, USA., pp 35.
Bin, Sheng, Sun, Gengxin, 2021. Matrix factorization recommendation algorithm based on multiple social relationships. Mathe. Probl. Eng. 2021.
Bokde, Dheeraj, Girase, Sheetal, Mukhopadhyay, Debajyoti, 2015. Matrix factorization model in collaborative filtering algorithms: A survey. Proc. Comput. Sci. 49, 136–146.
Breese, John S., Heckerman David, Kadie Carl, 2013. Empirical analysis of predictive algorithms for collaborative filtering. arXiv preprint arXiv:1301.7363.
Chen, Wei, Fong, Simon, 2010. Social network collaborative filtering framework and online trust factors: A case study on facebook. In: 2010 Fifth International Conference on Digital Information Management (ICDIM). IEEE, pp. 266–273.
Deerwester, Scott, Dumais, Susan T., Furnas, George W., Landauer, Thomas K., Harshman, Richard, 1990. Indexing by latent semantic analysis. J. Am. Soc. Infr. Sci. 41 (6), 391–407.
Ekstrand, Michael D., Riedl, John T., Konstan, Joseph A., et al., 2011. Collaborative filtering recommender systems. Found. Trends Human-Comput. Interact. 4 (2), 81–173.
Ezugwu, Absalom E., Agushaka, Jeffrey O., Abualigah, Laith, Mirjalili, Seyedali, Gandomi, Amir H., 2022. Prairie dog optimization algorithm. Neural Comput. Appl. 34 (22), 20017–20065.
Goldberg, David, Nichols, David, Oki, Brian M, Terry, Douglas, 1992. Using collaborative filtering to weave an information tapestry. Commun. ACM 35 (12), 61–70.
Gouvert, Olivier, Oberlin, Thomas, Févotte, Cédric, 2020. Ordinal non-negative matrix factorization for recommendation. In: International Conference on Machine Learning. PMLR, pp. 3680–3689.
Hao, Wu., Zhang, Zhengxin, Yue, Kun, Zhang, Binbin, He, Jun, Sun, Liangchen, 2018. Dual-regularized matrix factorization with deep neural networks for recommender systems. Knowledge-Based Syst. 145, 46–58.
Hao Qianxiu, Xu Qianqian, Yang Zhiyong, Huang Qingming, 2021. Pareto optimality for fairness-constrained collaborative filtering. In: Proceedings of the 29th ACM International Conference on Multimedia, pp. 5619–5627.
Healey, J., 2009. Grooveshark and virtual music collections. Los Angeles Times.
He, Yuan, Wang, Cheng, Jiang, Changjun, 2018. Correlated matrix factorization for recommendation with implicit feedback. IEEE Trans. Knowledge Data Eng. 31 (3), 451–464.
Herce-Zelaya, Julio, Porcel, Carlos, Bernabé-Moreno, Juan, Tejeda-Lorente, A., Herrera-Viedma, Enrique, 2020. New technique to alleviate the cold start problem in recommender systems using information from social media and random decision forests. Infr. Sci. 536, 156–170.

Himabindu, Tadiparthi V.R., Padmanabhan, Vineet, Pujari, Arun K., 2018. Conformal matrix factorization based recommender system. Infr. Sci. 467, 685–707.
Hofmann, Thomas, 2004. Latent semantic models for collaborative filtering. ACM Trans. Infr. Syst. (TOIS) 22 (1), 89–115.
Horasan, Fahrettin, 2022. Latent semantic indexing-based hybrid collaborative filtering for recommender systems. Arabian J. Sci. Eng., 1–15.
Horasan, Fahrettin, Erbay, Hasan, Varçın, Fatih, Deniz, Emre, 2019. Alternate low-rank matrix approximation in latent semantic analysis. Sci. Programm. 2019.
Huang, Li, Tan, Wenan, Sun, Yong, 2019. Collaborative recommendation algorithm based on probabilistic matrix factorization in probabilistic latent semantic analysis. Multimedia Tools Appl. 78 (7), 8711–8722.
Jakomin, Martin, Bosnić, Zoran, Curk, Tomaž, 2020. Simultaneous incremental matrix factorization for streaming recommender systems. Expert Syst. Appl. 160, 113685.
Jalili, Mahdi, Ahmadian, Sajad, Izadi, Maliheh, Moradi, Parham, Salehi, Mostafa, 2018. Evaluating collaborative filtering recommender algorithms: a survey. IEEE Access 6, 74003–74024.
Jiao, Jiangli, Zhang, Xueying, Li, Fenglian, Wang, Yan, 2019. A novel learning rate function and its application on the svd++ recommendation algorithm. IEEE Access 8, 14112–14122.
Jie, Lu., Dianshuang, Wu., Mao, Mingsong, Wang, Wei, Zhang, Guangquan, 2015. Recommender system application developments: a survey. Decis. Support Syst. 74, 12–32.
Kaloorazi, Maboud F., Chen, Jie, 2019. Randomized ulv decomposition for approximating low-rank matrices. In: 2019 IEEE International Conference on Signal Processing, Communications and Computing (ICSPCC). IEEE, pp. 1–5.
Kaloorazi, Maboud F., Chen, Jie, 2021. Projection-based qlp algorithm for efficiently computing low-rank approximation of matrices. IEEE Trans. Signal Process. 69, 2218–2232.
Kaloorazi, M., de Lamare, Rodrigo C., 2018a. Compressed randomized utv decompositions for low-rank approximations and big data applications. arXiv preprint arXiv:1810.07323.
Kaloorazi, Maboud F., de Lamare, Rodrigo C., 2018b. Compressed randomized utv decompositions for low-rank matrix approximations. IEEE J. Sel. Top. Signal Process. 12 (6), 1155–1169.
Koren, Yehuda, Bell, Robert, Volinsky, Chris, 2009. Matrix factorization techniques for recommender systems. Computer 42 (8), 30–37.
Kuo, R.J., Chen, Cheng-Kang, Keng, Shao-Hong, 2021. Application of hybrid metaheuristic with perturbation-based k-nearest neighbors algorithm and densest imputation to collaborative filtering in recommender systems. Infr. Sci..
Kurucz Miklós, Benczúr András A., Csalogány Károly, 2007. Methods for large scale svd with missing values. In: Proceedings of KDD Cup and Workshop, vol. 12, Citeseer, pp. 31–38.
Li, Ning, Luo, Wenjuan, Yang, Kun, Zhuang, Fuzhen, He, Qing, Shi, Zhongzhi, 2018. Self-organizing weighted incremental probabilistic latent semantic analysis. Int. J. Mach. Learn. Cybernet. 9 (12), 1987–1998.
Li, Zhoujun, Zhang, Haijun, Wang, Senzhang, Huang, Feiran, Li, Zhenping, Zhou, Jianshe, 2018. Exploit latent dirichlet allocation for collaborative filtering. Front. Comput. Sci. 12 (3), 571–581.
Li, Shijun, Lei, Wenqiang, Wu, Qingyun, He, Xiangnan, Jiang, Peng, Chua, Tat-Seng, 2021. Seamlessly unifying attributes and items: Conversational recommendation for cold-start users. ACM Trans. Infr. Syst. (TOIS) 39 (4), 1–29.
Linden, Greg, Smith, Brent, York, Jeremy, 2003. Amazon. com recommendations: Item-to-item collaborative filtering. IEEE Internet Comput. 1, 76–80.
Liu, Nathan N., Meng Xiangrui, Liu Chao, Yang Qiang, 2011. Wisdom of the better few: cold start recommendation via representative based rating elicitation. In: Proceedings of the fifth ACM Conference on Recommender Systems, pp. 37–44.
Lopez, Daniel Enrique Tamara, Blanquicett, Estefania Julieth Guevara, Medina, Carlos Daniel Acosta, 2018. Cross-language information retrieval using two methods: Lsi via sdd and lsi via svd. In: Multilingualism and Bilingualism. IntechOpen.
Luo, Xin, Zhou, Mengchu, Xia, Yunni, Zhu, Qingsheng, 2014. An efficient non-negative matrix-factorization-based approach to collaborative filtering for recommender systems. IEEE Trans. Ind. Infr. 10 (2), 1273–1284.
Luo, Xin, Zhou, MengChu, Xia, Yunni, Zhu, Qingsheng, Ammari, Ahmed Chiheb, Alabdulwahab, Ahmed, 2015. Generating highly accurate predictions for missing qos data via aggregating nonnegative latent factor models. IEEE Trans. Neural Networks Learn. Syst. 27 (3), 524–537.
Ma Hao, Yang Haixuan, Lyu Michael R., King Irwin, 2008. Sorec: social recommendation using probabilistic matrix factorization. In: Proceedings of the 17th ACM Conference on Information and Knowledge Management, pp. 931–940.
Mehta, Rachana, Rana, Keyur, 2017. A review on matrix factorization techniques in recommender systems. In: 2017 2nd International Conference on Communication Systems, Computing and IT Applications (CSCITA). IEEE, pp. 269–274.
Parvin, Hashem, Moradi, Parham, Esmaeili, Shahrokh, Qader, Nooruldeen Nasih, 2019. A scalable and robust trust-based nonnegative matrix factorization recommender using the alternating direction method. Knowledge-Based Syst. 166, 92–107.
Paterek Arkadiusz, 2007. Improving regularized singular value decomposition for collaborative filtering. In: Proceedings of KDD Cup and Workshop, vol. 2007, pp. 5–8.
Rana Pooja, Jain Nishi, Mittal Usha, 2020. An introduction to basic concepts on recommender systems. In: Recommender System with Machine Learning and Artificial Intelligence: Practical Tools and Applications in Medical, Agricultural and Other Industries, pp. 1–25.
Ran, Xun, Wang, Yong, Zhang, Leo Yu, Ma, Jun, 2022. A differentially private nonnegative matrix factorization for recommender system. Infr. Sci. 592, 21–35.
Rasti, Behnood, Scheunders, Paul, Ghamisi, Pedram, Licciardi, Giorgio, Chanussot, Jocelyn, 2018. Noise reduction in hyperspectral imagery: Overview and application. Remote Sens. 10 (3), 482.
Rebrova, Elizaveta, Chávez, Gustavo, Liu, Yang, Ghysels, Pieter, Li, Xiaoye Sherry, 2018. A study of clustering techniques and hierarchical matrix formats for kernel ridge regression. In: 2018 IEEE international Parallel and Distributed Processing Symposium Workshops (IPDPSW). IEEE, pp. 883–892.
Ricci, Francesco, Rokach, Lior, Shapira, Bracha, 2011. Introduction to recommender systems handbook. In: Recommender Systems Handbook. Springer, pp. 1–35.
Sarwar, Badrul, Karypis, George, Konstan, Joseph, Riedl, John, 2000. Application of Dimensionality Reduction in Recommender System-A Case Study. Technical report. Minnesota Univ Minneapolis Dept of Computer Science. Technical report.
Sarwar, Badrul, Karypis, George, Konstan, Joseph, Riedl, John, 2002. Incremental singular value decomposition algorithms for highly scalable recommender systems. Fifth International Conference on Computer and Information Science, vol. 1. Citeseer, pp. 27–28.
Singh, Monika, 2020. Scalability and sparsity issues in recommender datasets: a survey. Knowledge Infr. Syst. 62 (1), 1–43.
Symeonidis Panagiotis, Nanopoulos Alexandros, Papadopoulos Apostolos, Manolopoulos Yannis, 2006. Scalable collaborative filtering based on latent semantic indexing. In: Proc. 21st Assoc. for Advancement of Artificial Intelligence (AAAI) Workshop Intelligent Techniques for Web Personalization (ITWP'06), pp. 1–9.
Tahmasebi, Faryad, Meghdadi, Majid, Ahmadian, Sajad, Valiallahi, Khashayar, 2021. A hybrid recommendation system based on profile expansion technique to alleviate cold start problem. Multimedia Tools Appl. 80 (2), 2339–2354.
Takács Gábor, Pilászy István, Németh Bottyán, Tikk Domonkos, 2008. Matrix factorization and neighbor based algorithms for the netflix prize problem. In: Proceedings of the 2008 ACM Conference on Recommender Systems, pp. 267–274.
Valdez, Danny, Pickett, Andrew C., Goodson, Patricia, 2018. Topic modeling: latent semantic analysis for the social sciences. Social Sci. Quart. 99 (5), 1665–1679.
Wahab, Omar Abdel, Bentahar, Jamal, Otrok, Hadi, Mourad, Azzam, 2016. Towards trustworthy multi-cloud services communities: A trust-based hedonic coalitional game. IEEE Trans. Services Comput. 11 (1), 184–201.
Wahab, Omar Abdel, Cohen, Robin, Bentahar, Jamal, Otrok, Hadi, Mourad, Azzam, Rjoub, Gaith, 2020. An endorsement-based trust bootstrapping approach for newcomer cloud services. Infr. Sci. 527, 159–175.
Wang, Ruiqin, Cheng, Hsing Kenneth, Jiang, Yunliang, Lou, Jungang, 2019. A novel matrix factorization model for recommendation with lod-based semantic similarity measure. Expert Syst. Appl. 123, 70–81.
Yuan, Xiaofeng, Han, Lixin, Qian, Subin, Guoxia, Xu., Yan, Hong, 2019. Singular value decomposition based recommendation using imputed data. Knowledge-Based Syst. 163, 485–494.
Yuan, Xiaofeng, Han, Lixin, Qian, Subin, Zhu, Licai, Zhu, Jun, Yan, Hong, 2021. Preliminary data-based matrix factorization approach for recommendation. Infr. Process. Manage. 58 (1), 102384.
Zhang Yan, Zhang Zhao, Zhang Zheng, Zhao Mingbo, Zhang Li, Zha Zhengjun, Wang Meng, 2020. Deep self-representative concept factorization network for representation learning. In: Proceedings of the 2020 SIAM International Conference on Data Mining, SIAM, pp. 361–369.
Zhao Qian, Maxwell Harper, F., Adomavicius Gediminas, Konstan Joseph A., 2018. Explicit or implicit feedback? engagement or satisfaction? a field experiment on machine-learning-based recommender systems. In: Proceedings of the 33rd Annual ACM Symposium on Applied Computing, pp. 1331–1340.