Expert Systems With Applications 228 (2023) 120487

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

TrustDL: Use of trust-based dictionary learning to facilitate recommendation in social networks

Navid Khaledian [a,*], Amin Nazari [b], Keyhan Khamforoosh [a], Laith Abualigah [c], Danial Javaheri [d,*]

[a] Department of Computer Engineering, Sanandaj Branch, Islamic Azad University, Sanandaj, Iran
[b] Department of Computer Engineering, Bu-Ali Sina University, Hamedan, Iran
[c] Hourani Center for Applied Scientific Research, Al-Ahliyya Amman University, Amman 19328, Jordan
[d] Department of Computer Engineering, Chosun University, Gwangju 61452, Republic of Korea

ARTICLE INFO

Keywords:
Recommender systems
Dictionary learning
Trust relationship
Sparse representation
Machine learning

ABSTRACT

Collaborative filtering (CF) is a widely applied method to perform recommendation tasks in a wide range of domains and applications. Dictionary learning (DL) models, which are highly important in CF-based recommender systems (RSs), are well represented by rating matrices. However, these methods alone do not resolve the cold start and data sparsity issues in RSs. We observed a significant improvement in rating results by adding trust information on the social network. For that purpose, we proposed a new dictionary learning technique based on trust information, called TrustDL, where the social network data were employed in the process of recommendation based on structural details on the trusted network. TrustDL sought to integrate the sources of information, including trust statements and ratings, into the recommendation model to mitigate both problems of cold start and data sparsity. It conducted dictionary learning and trust embedding simultaneously to predict unknown rating values. In this paper, the dictionary learning technique was integrated into rating learning, along with the trust consistency regularization term designed to offer a more accurate understanding of the feature representation. Moreover, partially identical trust embedding was developed, where users with similar rating sets could cluster together, and those with similar rating sets could be represented collaboratively. The proposed strategy appears significantly beneficial based on experiments conducted on four frequently used datasets: Epinions, Ciao, FilmTrust, and Flixster.

1. Introduction

Nowadays, with the increasing growth of online services, a large amount of data is generated. It can be highly time-consuming and expensive to search this vast network, aiming to find the required information. Therefore, using smart tools such as recommender systems (RSs) seems very necessary. These systems seek to provide useful suggestions to users by understanding their interests. Generally, RSs can be classified into three types: collaborative filtering (CF), content-based (CB), and hybrid methods. The CB approach provides suggestions by exploring and extracting the characteristics of items and users. However, lack of scalability, data fragmentation, and cold start are common issues faced by these methods. CF derives users' preferences by analyzing their past interests and data. Then, it groups users based on their similarities and provides recommendations to each based on the group it belongs to. CF severely suffers from data sparsity and cold start problems; however, it is more scalable than CB. There are two main categories of CF methods: model-based and memory-based (Khaledian & Mardukhi, 2022; Papadakis, Papagrigoriou, Panagiotakis, Kosmas, & Fragopoulou, 2022). Memory-based methods use measures of similarity between users to predict ranking values and are divided into item-based and user-based filtering. To deal with the problems of dispersion and scalability, a combination of item-based and user-based methods can be employed. In contrast, model-based approaches apply learning techniques to generate models that predict the users' recommendations (Behera & Nain, 2022b).

Although model-based methods allow more accuracy compared to memory-based ones, they still suffer from data sparsity and cold start

* Corresponding authors.
E-mail addresses: navid.khaledian@iausdj.ac.ir (N. Khaledian), aminnazari91@gmail.com (A. Nazari), k.khamforoosh@iausdj.ac.ir (K. Khamforoosh), aligah.2020@gmail.com (L. Abualigah), javaheri@chosun.ac.kr (D. Javaheri).

https://doi.org/10.1016/j.eswa.2023.120487
Received 20 January 2023; Received in revised form 1 May 2023; Accepted 11 May 2023
Available online 16 May 2023

problems (Rashidi, Khamforoosh, & Sheikhahmadi, 2021). Various methods have been proposed in the literature to solve these problems, which we categorize into several groups. They include unsupervised learning (UL), supervised learning (SL), semi supervised learning, and reinforcement learning (RL), as well as metaheuristic algorithms (MA) as an optimization approach (Javaheri, Gorgin, Lee, & Masdari, 2022). SL approaches leverage classification algorithms such as random forest, neural networks, and deep learning for learning the model (Kordabad, Nazari, & Mansoorizadeh, 2022). These methods are able to offer personalized recommendations according to the user's past behavior and preferences. In addition, they are scalable and can handle large amounts of data (up to millions of users and items). However, the time complexity (computational complexity) of such methods is the underlying challenge; besides, they require large amounts of data for training to avoid overfitting. A UL RS can improve the accuracy of recommendations by grouping users and items based on similarity (Nazari, Kordabadi, & Mansoorizadeh, 2023). Moreover, it is highly scalable because cluster-based recommender systems can handle large datasets. In contrast, cluster-based recommender systems have difficulty handling new users or items since they are difficult to group. Furthermore, cluster-based recommender systems may be too specialized, recommending only a limited set of items to users, thereby preventing the introduction of new, diverse content. RL systems can adapt to changes over time based on user feedback, but they require large amounts of data to train effectively and may be biased toward certain products or users, thus leading to incorrect or unfair recommendations. MA RSs can be customized to match different types of recommendation problems and user preferences. Moreover, these methods can be used to manage noisy or incomplete data because they make more powerful recommendations. On the other hand, these methods are computationally expensive. Furthermore, it may be challenging to interpret and understand how recommendations are made (Alhijawi & Kilani, 2020; Kuo, Chen, & Keng, 2021).

Recently, a new technique in machine learning called dictionary learning (DL) has been introduced. It aims to learn a dictionary that considers a sparse representation of data. Therefore, each example can be considered a linear combination of several examples from the dictionary. This technique is used in various fields, including signal processing, computer vision, medical screening, and natural language processing. Due to the success of this method in other fields, its application to RSs has been considered recently. One of the most important ways to address the data sparsity and cold start issues is to use additional information, including trust data, item contents, and user profiles (Ahmadian, Joorabloo, Jalili, & Ahmadian, 2022; Behera & Nain, 2022a). The authors (Geluvaraj & Sundaram, 2022) demonstrated the effectiveness of using the trust to mitigate cold start. However, the combination of rating and trust to improve recommendations calls for greater attention because the methods that can combine these data and represent them in a common space are strongly felt. Therefore, we attempt in this study to map the data in a common space using DL and use this new representation to improve recommendations. Our main motivation for conducting this research was that learning methods face the problem of curse of dimensionality and that DL can reduce the amount of data.

In this paper, we use DL to embed the trust relationship to deal with the issues of cold start and fragmentation. DL can learn the user's similarity based on the rating or trust matrix. Simultaneous learning of both can lead to the construction of a robust, accurate model that can resolve the cold start problem since the data deficiencies in the rating model can be eliminated to some extent using trust data. This trust-based recommendation system utilizes social network data to mitigate the negative impacts of the data sparsity and cold start issues on recommendation performance. Our approach applies dictionary learning and trusts embedding simultaneously. The dictionary learning technique is incorporated to learn rating and trust regularization. Therefore, the rating feature space is designed so that the feature representation is learned more appropriately. In order to make use of the local manifold data structure, the objective function considered in this study incorporates a manifold regularization featuring a Laplacian graph. Thus, the proposed method provides high prediction accuracy for different data views due to the use of both trust and the DL method in the recommendation process. The main contributions of our research can be summarized as below.

1) A novel trust-based approach is proposed to incorporate trust in dictionary learning with rating, and we can thus overcome the data sparsity and cold start issues.
2) We use the dictionary learning approach in the rating feature space to form an over-complete dictionary with an appropriate representation of the supplied input data from the training rating space.
3) In our proposed method, trust embedding is designed in the trust feature space. The trust embedding with fair collaborative representation is based on users with similar trust sets. As a result, these users can be clustered together to collectively represent one another. A joint objective function is designed to perform identical trust embedding and dictionary learning at the same time.
4) The cold start and data sparsity challenges may be resolved using social network data in the process of recommendation by modeling user roles concerning the structural data over the trusted network. Because social network trust data can significantly improve recommendation performance when it appropriately was added into the DL model.

The rest of the paper is organized as follows. Section 2 presents a review of a number of related works in the field of recommender systems. In Section 3, the TrustDL algorithm is analyzed and discussed in detail, along with the pseudo-code. In Section 4, the performance of the proposed methods is evaluated using various comparisons to state-of-the-art related works. Eventually, Section 5 concludes the paper.

2. Related works

In this section, we present and discuss the methods related to social-based and DL-based RSs for CF.

2.1. CF and matrix factorization

RSs can be categorized into three groups: content-based (CB), CF-based, and hybrid. CB systems seek to match content with user interests (Livne et al., 2022). CF is the most popular, widely-used technique in the recommendation environment. A CF technique is more beneficial for a personalization recommendation system than a content-based one, as the former approach receives explicit ratings directly from the user (Behera & Nain, 2022b). A CF system predicts user interests by applying the opinions of a group of users all in the same cluster (Papadakis et al., 2022). A model-based CF develops a model using the perceived ratings and then makes predictions using the generated model. Matrix factorization (MF) is a model-based approach that maps the items and the users into a lower-dimensional space so that the interactions between users and items are modeled in the latent space (Tahmasbi, Jalali, & Shakeri, 2021). Moreover, hybrid methods combine CF with CB in the recommendation process for higher efficiency. Given a set of products labeled I and a group of users labeled U, a rating matrix of a user-item, called R, is a $|U| \times |I|$ matrix in which each element $r_{u,i}$ indicates the rating given by user u to item i. In the MF-based model, the rating matrix is disintegrated into two latent factors P and Q, where P is $|U| \times f$, Q is $|I| \times f$, and $f \leq min(|U|, |I|)$ involves the dimensions of the latent space (Li et al., 2019a). Matrix factorization approaches have proven effective and dependable for the development of recommender systems, where data sparsity difficulties may be eased indirectly through the incorporation of various heterogeneous data sources (Jakomin, Bosnić, & Curk, 2020). In (Qi et al., 2021), a graph embedding method was introduced for a Bayesian network based on matrix factorization. Therefore, the authors suggested a non-negative matrix completion method (named NMC) to predict unknown ratings and introduce an intermediate, complete matrix to which the DL process is applied to predict missing entries. Besides, we are generally faced with extremely sparse, high-dimensional rating matrices in many industrial applications. Therefore, scalable DL methods need to be proposed.

In order to overcome the gaps in current collaborative filtering systems and reach the highest possible level of prediction accuracy using artificial neural networks and deep learning techniques, was proposed a new hybrid deep learning recommender system (Kiran, Kumar, & Bhasker, 2020). Node clustering in networks is applied to address the data sparsity issue in recommendations (Zhang, Qi, Liu, Mao, & Zeng, 2020). The authors employed node clustering to reconstitute denser bipartite user-item networks. A new non-negative matrix factorization is introduced using asymmetric link clustering and the PageRank model, referred to as NMF-AP (Chen, Xu, Wang, Feng, & Feng, 2020). The impact score of a node was calculated using the PageRank algorithm, which collects information about the global network structure. Khaledian et al. introduced a method called CFMT, which integrated non-negative matrix factorization and trust relationships in social networks for unknown prediction ratings (Khaledian & Mardukhi, 2022). They proposed a new hybrid approach based on directed trust and probabilistic matrix factorization.

2.2. Dictionary learning in RSs

Sparse representation-based techniques have yielded intriguing results in RSs. Dictionary learning seeks to learn an over-complete dictionary with a correct representation of the input data provided from the training sample space (Li, Tao, Ding, Zhang, & Meng, 2022). It has proven effective in several fields of application, including recommender systems, software defect prediction, image clustering and classification, and pattern recognition (Du, Zhang, Ma, & Zhang, 2021). According to the recent progress in CF, approaches based on dictionary learning can efficiently predict user preferences. These approaches assume that (i) an unstructured, latent feature space (hidden representation/code) can be found behind the user ratings and (ii) an item rating equals the item multiplied by the user's feature. A formal expression of the sparse coding problem follows.

$$\min\|\boldsymbol{x}\|_0 s.t\boldsymbol{b} = \boldsymbol{A}x \tag{1}$$

The zero pseudo-norm x0 counts the non-zero $\|x\|_0$ elements in $\boldsymbol{x}$, $\boldsymbol{b}$ is a given vector for which the algorithm seeks a sparse representation, and $\boldsymbol{A}$ is a provided matrix (or dictionary), the columns of which are linearly merged by the sparse representation $\boldsymbol{x}$ for the reconstruction of $\boldsymbol{b}$. In this application, $\boldsymbol{A}$ stands for the user-item matrix, $\boldsymbol{b}$ for a rating vector with one or more missing values that requires prediction, and $\boldsymbol{x}$ for a sparse coefficient set that reconstructs $\boldsymbol{b}$ using columns from the user-item matrix. In the present application, $\boldsymbol{A}$ represents the user-item matrix, $\boldsymbol{b}$ indicates a rating vector with one or more missing values requiring prediction, and $\boldsymbol{x}$ denotes a sparse coefficient set choosing columns from the user-item matrix to rebuild $\boldsymbol{b}$. Although they may not be similar to $\boldsymbol{b}$, the columns selected from $\boldsymbol{A}$ can reconstruct $\boldsymbol{b}$ precisely in combination. That is not the case with a conventional CF recommender system, such as one based on k-NN, in which the columns of $\boldsymbol{A}$ are chosen based on their similarity to $\boldsymbol{b}$. In this application, the ratings and, as a result, the values in $\boldsymbol{b}$ and $\boldsymbol{A}$ are non-negative. Moreover, there is a constraint on the sparse coefficients, $\boldsymbol{x}$, in that they cannot be harmful since it is not logical or intuitive for one rating to be reconstructed through the subtraction of another. It has been found that this additional constraint also leads to superior performance. The issue of sparse coding was modeled in (Wang, Zhu, Dai, Xu, & Gao, 2021) as a problem of regularized linear optimization, where $\|x\|_0$ is replaced by $\|x\|_1$, correctly formulating in mathematical terms the problem of obtaining a vector with the smallest possible number of non-zero coefficients. $\|x\|_1$ is obtained so that sparse solutions can be achieved, and the process of obtaining solutions can be traced more easily. It is an empirical issue to specify a possibly better choice. Although the present paper is not concentrated on the above issue, it provides some contribution. Soft thresholding and coordinate descent were utilized in (Permiakova & Burger, 2022) for solving the optimization problem. The sparse solver, an approach based on neural networks and involving a competition within a set of neurons to represent the input signal $\boldsymbol{b}$ leading to sparse solutions, was used in (Amestoy et al., 2022) to solve sparse coding as in Eq. (1).

2.3. Trust-based recommender system

Trust information has widely been employed to increase the accuracy of prediction in a recommender system. Moreover, people need to trust information to better communicate and use each other's experiences on a social network in the real world. This propagates trust across the trusted network to include additional trusted users. Depending on the category to which the user trust data belong, there are two main trust-based methods, *i.e.* explicit and implicit. Explicit trust concerns trust values that are explicitly indicated by users, while implicit trust pertains to trust levels that may be predicted based on user similarity. In much research, explicit trust-based techniques have been proposed to overcome the data sparsity issue, where the ratings can be predicted from latent rating and trust information.

TrustSVD uses the SVD++ algorithm to evaluate both the explicit and implicit effects of trust and rating (Guo, Zhang, & Yorke-Smith, 2015). TrustMF takes into account both the trustee and the trusted to accurate to increase accuracy (Yang, Lei, Liu, & Li, 2016). CPD is a strategy proposed by Azadjalal, Moradi, Abdollahpouri, and Jalili (2017) that leverages both implicit and explicit trust relationships simultaneously. To validate implicit trust statements, CPD uses a reliability measure. The most prominent user for the target one is specified using a Pareto dominance and confidence measure, and only these prominent users are utilized to provide recommendations. The authors presented a set of new measures to evaluate trust and influence based on users' social relationships and rating information (Li, Ye, Xin, & Jin, 2017). A unique trust recommendation model based on the latent factor methodology and trusty neighborhood fitting model was presented by Li, Mo, Xin, and Jin (2018), which combines heterogeneous information over the Internet and cyberspace. CosRA + T is a trust-based recommendation approach incorporating information about trust relations into the resource-redistribution process (Chen & Gao, 2018). A recommendation method with trusted relevance and matrix factorization is proposed by Li et al. (2019b). The user's social information is integrated into the recommendation algorithm by generating an effective trust metric model. To address the issues of data sparsity and cold start, the TrustANLF technique incorporates trust statements into the recommendation model as a suitable alternative source of information apart from rating values. In addition, to minimize computational and memory costs and enhance convergence speed, the trust-based non-negative MF model is solved via alternating direction optimization (Parvin, Moradi, & Esmaeili, 2019b). The authors offered a unified method that leverages explicit trust, implicit trust, and user preference similarity to develop a unified rating profile for the target user, resulting in more effective, more accurate recommendations (Ayub et al., 2019). When only a few ratings are accessible, the proposed unified method improves the recommendation performance of a CF-based recommender system. TCFACO is a method presented to reduce sparsity and cold user concerns using an explicit trust (Parvin, Moradi, & Esmaeili, 2019a). The authors introduced a context-aware recommendation algorithm called CSSVD to increase recommendation performance and mitigate the cold start and sparse data challenges (Rodpysh, Mirabedini, & Banirostam, 2021). They propose T-MRGF, a trust-ware recommendation method based on the fusion of heterogeneous multi-relational graphs. It merges the user-related and item-related networks with the user-item interaction graph

Table 1
A comparison of related works from various perspectives.

References	Technique	Trust	Cold Start Control	Scalability
(Qi et al., 2021)	Matrix factorization & dictionary learning	×	✓	Medium
(Kiran et al., 2020)	Deep learning	×	✓	Low
(Zhang et al., 2020)	Node Clustering	×	✓	High
(Chen et al., 2020)	Non-negative matrix factorization & Link clustering	×	✓	Medium
(Khaledian & Mardukhi, 2022)	Non-negative matrix factorization	✓	✓	Medium
(Yang et al., 2016)	Matrix factorization	✓	✓	Medium
(Parvin et al., 2019a)	Ant colony optimization	✓	×	Low
(Parvin et al., 2019b)	Non-negative matrix factorization	✓	✓	Medium
(Kordabadi et al., 2022)	Machine Learning	×	×	Medium
(Forouzandeh, Berahmand, & Rostami, 2021b)	Random Walk	✓	×	Medium
(Nazari et al., 2023)	Community detection & Link prediction	×	✓	High
TrustDL 2023	Matrix factorization & dictionary learning & Manifold	✓	✓	Medium

and fully utilizes the high-level connections present in heterogeneous graphs, in contrast to other standard techniques (Guo, Zhou, Zhang, Song, & Chen, 2021). The authors presented a novel trust-based matrix factorization technique, namely CFMT, for recommender systems (Khaledian & Mardukhi, 2022). They use social network data and model the users as trustees. They claim that their proposed method resolves data sparsity and cold start problems to some extent. A method that performs dictionary learning based on social networking information has not yet been provided. Our approach conducts dictionary learning and trust embedding simultaneously to address the cold start and data sparsity challenges. Table 1 compares some of the methods.

3. The proposed method

This section delves into the details of our proposed strategy, which is based on dictionary learning and includes identical trust embedding and rating consistency regularization. We first explain discriminative dictionary learning using the rating consistency regularization term in the rating feature space. The overall objective function is then described, followed by identical trust embedding in the trust space. In order to improve prediction accuracy, our system involves social trust information. While Fig. 1 indicates the architecture and internal procedures of the proposed methods, further details on the proposed approach are provided in the following subsections. Besides all indexes and variables used in the equations were defined in Table 2.

3.1. *Dictionary learning with regularization of trust consistency*

In this work, the recommendation task involves the prediction of an unspecified rating of user (u) on item (i), which is not already known through the application of the user-item matrix (R) and the user-trust matrix (T). For this purpose, we consider a graph G= (V; E) that represents a trusted network, where V denotes a set of N users, and E indicates the set of trust relationships between them. Thus, the adjacency matrix $T = [t_{a,b}]_{N\times N}$ is applied to determine the trust relationships between the users, where $t_{a,b}$ implies the extent to which user a trusts b. Therefore, $t_{a,b}= 1$ means that user a trusts user b, whereas $t_{a,b} = 0$ denotes the distrust relationship. Furthermore, the sparsity of trust matrix T is exceptionally high. Hence, TrustDL uses a combination of the rating matrix and the trust matrix to anticipate unknown ratings.

Let $[r_1, r_2 \cdots r_N]$ be the training data from R, where r_i represents the ratings of user i on all the items. First, dictionary B should adequately represent the projected sets of features. It should be noted that these sets are those obtained after feature transformation, as introduced in the following subsection. Specifically, the over-complete dictionary $B \in R^{p\times k}$ ($k > p$, usually set to N) that we learn must be well able to represent the projected low-dimensional feature sets $P^T R \in \mathbb{R}^{p\times N}$; that is, $P^T R \approx BZ$, where P is the transformation matrix. $Z = [z_1, z_2 \cdots z_N] \in \mathbb{R}^{k\times N}$ denotes the set of representation coefficients. The objective function for the reconstruction error is defined as follows.

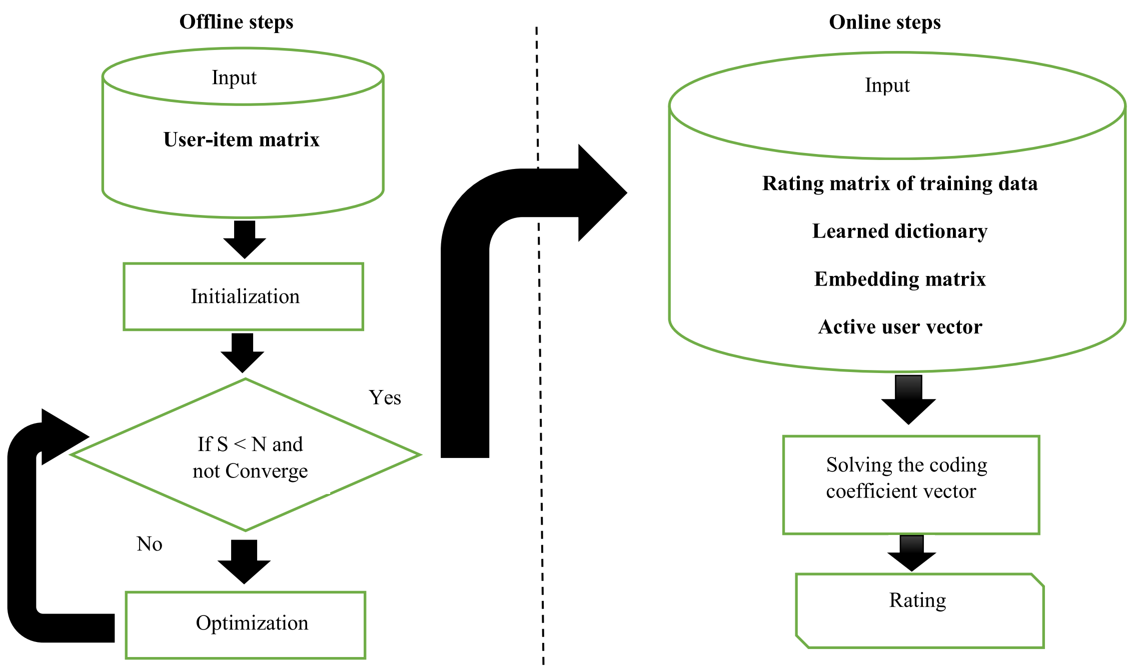

Fig. 1. The diagram of procedures in TrustDL.

Table 2
Notation.

Symbol	Description
R	User-item matrix
U	User set
I	Item set
$r_{u,i}$	Rating given by user u to item i
r_i	Ratings of user i
P	User feature matrix
Q	Item feature matrix
V	Set of N users
E	Set of trust relationships
λ, α	Regularization coefficients
$t_{a,b}$	User a trusts b
$\|x\|_0$	Zero Norm
T	Trust matrix
t_i	Sparse code
t_{ik}	r_i and d_k have similar trust sets
$\|\cdot\|_F^2$	Frobenius Norm
B	A Dictionary
Z	Set of representation coefficients
W	Trust feature matrix
d_k	Dictionary atom
S_{ij}	Similarity rate i,j
W_i^2	i^{th} row vector in W^2
ψ	Basic vector

$$\Phi(B,Z,W,P) = \min\frac{1}{2}\|P^TR - BZ\|_F^2 \tag{2}$$

The rating consistency regularization term is then introduced to obtain the set of discriminative sparse codes Z with the learned dictionary B. This gives exact sparse representations of the projected features of users with similar trust sets. Here is the definition of the objective function of dictionary learning with the designed trust consistency regularization term.

$$\Phi(B,Z,W,P) = min\frac{1}{2}\|P^TR - BZ\|_F^2 + \lambda\|T - WZ\|_F^2 + \alpha\|Z\|_1 \tag{3}$$

The relative contributions of the three terms, including the reconstruction error, trust consistency regularization, and sparsity of Z, are controlled by λ and α. The discriminative sparse codes of the rating features of samples R in the low-dimensional space are represented by $T = [t_1, t\cdots t_N] \in \mathbb{R}^{k\times N}$. $t_i = [0,..1,1\cdots0]$ indicates a discriminative sparse code responding to trust in the low-dimensional space. Let us consider the i^{th} dictionary atom learned $d_i(i = 1,\cdots,N)$ to have the same trust set as the i^{th} user vector $r_i(i = 1,\cdots,N)$. If r_i and the dictionary atom d_k have similar trust sets, $t_{ik} = 1$, and $t_{ik} = 0$ otherwise. $W \in R^{m\times m}$ indicates a linear transformation matrix. The linear transformation $g(Z;W) = WZ$ is identified here. Thus, the original set of sparse codes Z is transformed into the most discriminative within the sparse feature space $\mathbb{R}^m$. It is required by the trust consistency regularization term $\|T - WZ\|_F^2$ that an accurate approximation of the discriminative set of sparse codes T is provided by the transformed set of sparse codes WZ, which can make the learned dictionary more discriminable.

3.2. Identical trust embedding

In a real-world dataset, there are often few users with similar trust sets. It is thus not thoroughly exploited by Eq. (3) how correlated different user sets are. For that reason, the other objective term is designed, allowing users to have partially similar trust sets to represent one another collaboratively. On that basis, the definition of the objective term is provided below.

$$min\frac{1}{2}\sum_i\left\|P^Tr_i - \sum_j W_{ij}^2P^Tr_j\right\|_2^2 \tag{4}$$

Samples can be transformed through the transformation of matrix P into a more discriminative feature space. Therefore, users with similar trust values are grouped together. W^2 is a similarity graph described in the following: If users i and j have several users in common in their trust sets, *i.e.*, they partially involve identical trust sets, we set $W_{ij}^2 = \frac{1}{|W_i^2|^2}$; otherwise, we set $W_{ij}^2 = 0$, where W_i^2 implies the i^{th} row vector in W^2, and $|\cdot|$ denotes the non-zero elements in a vector.

3.3. Dictionary learning with manifold regularization

It is challenging to estimate the global manifold data space structure precisely since there are inadequate samples and the ambient space is high-dimensional. That is why many methods have been reported to be used for capturing the local manifold structure. The abundance of work put into manifold learning (Zhu, Liu, Cauley, Rosen, & Rosen, 2018) has proven that the closest neighboring graph on the sampled data points may successfully simulate the local geometric structure of the data manifold. Based on many previous related works (Talmon, Mallat, Zaveri, & Coifman, 2015; Wang, Yan, Nie, Yan, & Sebe, 2018), integration of global and local structures is most likely available on a low-dimensional sub-manifold of the high-dimensional ambient Euclidean space, is critical in data clustering or classification. As a result, the intrinsic manifold structure must be further investigated and examined to improve the recommended technique's performance.

Here, the manifold regularization to the objective function is introduced, and the improved variant is obtained to employ such manifold data. The Laplacian graph characterizes the manifold regularization and captures the geometric structure of the local data manifold, so similar low-rank representations tend to be exhibited by nearby points in the intrinsic geometry of the data space. In (Zhou, Du, Lü, & Wang, 2021), an NMF factorization with multiple regularizations is proposed and developed. Because it aims to explore both the global Euclidean and local manifold structures of the data by appending an extra manifold structure learning component to the final objective function, our recommended technique is thought to have better-discriminating power than others. As a result, we expect the intrinsic geometry structure of the data manifold to be used to improve recommendation accuracy. An assumption is made for this purpose, known as a manifold assumption (Li, Li, An, Zheng, & Li, 2019b), which is used to improve various algorithms. On that basis, it is assumed that two data points r_i and r_j are close together if the intrinsic geometry of data manifold is close. The nearest neighboring graph is constructed below to optimize the following objective function to characterize the data manifold local geometry.

$$\begin{aligned}\psi(R,P) &= min\frac{1}{2}\sum_{i,j=1}^{N}\|P^Tr_i - P^Tr_j\|_2^2S_{ij} = \sum_{i=1}^{N}r^Tr_iD_{ii} - \sum_{i,j=1}^{N}r^Tr_iS_{ij}\\ &= Tr\left(RDR^T\right) - Tr\left(RSR^T\right) = Tr\left(RLR^T\right)\end{aligned} \tag{5}$$

The i^{th} diagonal element of D is a diagonal matrix with $D_{ii} = \sum_j S_{ij}$ and $L = D - S$. Matrix L is commonly referred to as the graph Laplacian. To simplify the deductions, the constant value of 1/2 is utilized in the calculation of ψ. The objective function determines that the representation coefficients should be smooth or that, as assumed earlier, the low-dimensional representations of neighboring points r_i and r_j should be very close if they are similar (a relatively greater S_{ij}). As a result, the minimization of Eq. (5) is an effort to guarantee the manifold assumption. T is a manifold regularization algorithm that has been used to improve a variety of algorithms (Zhu et al., 2018). The absolute value of the cosine similarity of users is derived for calculating their similarity. Eq. (6) is employed to compute the cosine similarity of users i and j:

$$S_{ij} = \left|\sum_{i=1}^{P}(a_ib_i) \Big/ \left(\sqrt{\sum_{i=1}^{P}a_i^{\ 2}}\right)\left(\sqrt{\sum_{i=1}^{P}b_i^{\ 2}}\right)\right| \tag{6}$$

where i and j demonstrate two users with p-dimensional vectors ($i = [a1,$

$a2, ..., ap]$ and $j = [b1, b2, ..., bp])$. The equation demonstrates that the similarity value lies between 0 and 1. It is 1 for two fully identical users and 0 for fully dissimilar ones (Ar & Bostanci, 2016).

3.4. Objective function and optimization

Linear integration of the rating value and the trust effect in Eq. (1) is a reasonable option for merging a user's rating matrix and trust matrix to enhance the performance of the RS and alleviate the cold start and data sparsity challenges. Thus, the joint objective function of the proposed method can be formulated as follows according to the dictionary learning with partially identical trust embedding and manifold regularization.

$$\Phi(B, Z, W, P) = \min\frac{1}{2}\left(\begin{array}{c}\|P^TR - BZ\|_F^2 + \lambda\|T - WZ\|_F^2 + \alpha\|Z\|_1 + \\ \sum_i\left(\|P^Tr_i - \sum_j W_{ij}^2P^Tr_j\|_2^2\right) + \sum_{i,j=1}^N\|P^Tr_i - P^Tr_j\|_2^2S_{ij}\end{array}\right) \tag{7}$$

The optimization for the total objective function of the proposed method, Eq. (7), is provided here. Eq. (7) is convex for Z, B, P, or W, with the others being fixed, but it is not convex for them all simultaneously. The four steps listed below are taken for iterative optimization of Z, B, P, and W: (1) update coding coefficient matrix Z, (2) update the dictionary B, (3) update embedding matrix P, and (4) update the linear transformation W (Jing et al., 2016). In order to update the coding coefficient matrix, Z, B, P, and W need to be fixed. Afterward, the objective function in Eq. (7) can be reduced as follows to update Z:

$$\begin{aligned}\langle Z\rangle &= \min\|P^TR - BZ\|_F^2 + \lambda\|T - WZ\|_F^2 + \alpha\|Z\|_1 \\ &= \sum_{i=1}^N\left(\|P^Tr_i - Bz_i\|_F^2 + \lambda\|t_i - Wz_i\|_F^2 + \alpha\|z_i\|_1\right)\end{aligned} \tag{8}$$

where z_i is the i^{th} column in Z. To obtain z_i, Eq. (8) can be reformulated to solve N different problems.

$$Z = \underset{z_i}{\mathrm{argmin}} z_i^TB^TBz_i - 2r_i^TPBz_i + \lambda\left(z_i^TW^TWz_i - 2t_i^TWz_i\right) + \alpha Z_1 \tag{9}$$

The feature-sign search algorithm can be used to solve it as a sparse coding problem. With P, Z, and W assumed to be fixed, Eq. (9) can be reduced to the following.

$$B = \underset{B}{\mathrm{argmin}} P^TR - BZ_F^2 \tag{10}$$

Calculating the derivative of the $\|P^TR - BZ\|_F^2$ error with respect to B and manipulating it, we obtain the following.

$$\frac{\partial\left(\|P^TR - BZ\|_F^2\right)}{\partial_B} = 2BZZ^T - 2P^TRZ^T \tag{11}$$

Setting Eq. (11) to zero, we have the following.

$$B = P^TRZ^T\left(ZZ^T\right)^{-1} \tag{12}$$

To update the embedding matrix P, we fix B, Z, and W. Therefore, Eq. (7) can be simplified to:

$$\begin{aligned}&\underset{P}{\mathrm{argmin}}\frac{1}{2}\left(\|P^TR - BZ\|_F^2 + \sum_i\left(\|P^Tr_i - \sum_j W_{ij}^2P^Tr_j\|_2^2\right) + \sum_{i,j=1}^N\|P^Tr_i - P^Tr_j\|_2^2S_{ij}\right) \\ &= Tr\left(\left(P^TR - BZ\right)^T\left(P^TR - BZ\right)\right) + Tr\left(P^TRMR^TP\right)\end{aligned} \tag{13}$$

where $M = E - S + (I - W^2)^T(I - W^2)$ and $Tr(\cdot)$ denote a matrix trace, and E is a diagonal matrix with $E_{ii} = \sum_{j\neq i}S_{ij}$. Hence, we calculate the derivative of the above equation for transformation matrix P.

$$\begin{aligned}&\frac{\partial\left(Tr\left(\left(P^TR - BZ\right)^T\left(P^TR - BZ\right)\right) + Tr\left(P^TRMR^TP\right)\right)}{\partial_P} \\ &= 2RR^TP - 2RZ^TB^T + RMR^TP + RM^TR^TP\end{aligned} \tag{14}$$

Setting Formula (14) to zero, we have the following.

$$P = \left(RR^T + RMR^T\right)^{-1}RZ^TB^T \tag{15}$$

B, P, and Z are fixed for updating the linear transformation W. The following error is minimized as transformation matrix W is updated.

$$\|T - WZ\|_F^2 \tag{16}$$

By calculating the derivative of the error considering W and manipulating it, we obtain the following.

$$\frac{\partial\left(\|T - WZ\|_F^2\right)}{\partial_W} = 2WZZ^T - 2TZ^T \tag{17}$$

Setting (17) to zero, we have the following.

$$W = TZ^T\left(ZZ^T\right)^{-1} \tag{18}$$

3.5. The algorithm of the proposed method

A summary of the optimization process of the variables in Eq. (7), the detailed optimization procedure of which was given in the previous sections, is provided in Algorithm 1. The process of rating predictions involves the use of the training samples to assign a set of ratings to the query items for the active user. Using learned embedding matrix P and the discriminative dictionary D, we can anticipate the active user's rating vector. The variables are iteratively optimized in the proposed method. As long as the maximum number of iterations has not been obtained, the objective function values in consecutive iterations are far apart. The convergence of the approach employed in this research is empirically observed. The maximum number of iterations is a fixed number, *i.e.* 100. The algorithm involves several inputs, such as parameters λ and α, rating matrix R, and trust matrix T. It is noteworthy that the parameters included in the optimization process should be started with random positive integer values to ensure an effective and efficient update. The regularizing coefficients are set to acceptable values for running the DL algorithm.

Table 3
Statistical information on our four datasets.

Feature	*Epinions*	*FilmTrust*	*Ciao*	*Flixster*
#users	40,163	1,508	30,444	53,213
#items	139,738	2,071	72,665	18,197
#ratings	664,824	35,497	1,625,480	409,803
Density	0.051%	1.14%	0.03%	0.04%
#trusters	33,960	609	6,792	47,029
#trustees	49,288	732	7,297	47,029
#trusts	487,183	1,853	111,781	655,054
Density	0.029%	0.42%	0.23%	0.03%

Algorithm 1. *Learning in TrustDL*

Inputs: $R = rating\ matrix, U = user\ set, I = item\ set, T = trust\ network$
Outputs: *Z, B, P,* ***and*** *W***.**
Begin
Step 1: Initialization
Initialize randomly α, γ, β
Initialize n =max iteration
We initialize all the atoms in B as random vectors with unit ι_2 -norm and initialize P and W as random matrices.
Step 2: Iteratively updating *Z, B, P, W* in turn
While $s \leq n$ && not converge **do**
- **Fix** *B*, *P*, *W*, and update the sparse coding coefficients *Z* by solving Eq. (9) with the feature-sign search algorithm.
-**Fix** *Z*, *P*, W, and then update the dictionary *B* by solving Eq. (12).
-**Fix** *Z*, *B, W*, and then update the partial-identical embedding matrix *P* by solving Eq. (15).
-**Fix** *Z*, *B*, *P*, and then update the linear transformation matrix *W* by solving Eq. (18).
If $J\ (i+1) - J\ (i) \leq \varepsilon$, where $J\ (i)$ is the value of an objective function in the *ith* iteration, break.
$s = s + 1;$
End while
End

An active user vector can be mapped precisely into the low-dimensional feature space, indicated as $a^t = P^T a$. Since a discriminative dictionary with an appropriate representation ability has been learned, it can represent the features of the query user vector, *i.e.*, a^t. We can obtain the sparse coding coefficients in this case by resolving:

$$\langle z_t \rangle = \underset{z_t}{\operatorname{argmin}} \left(\|P^T a - B z_t\|_F^2 + \gamma \|z_t\|_1 \right) \tag{19}$$

where γ is a constant used to balance the terms for sparsity and reconstruction error. Rating propagation can be performed on a query user vector through employment of the obtained space coding coefficient vector z_t and the entire training data rating:

$$\hat{r}_i = z_t \times R_i \tag{20}$$

in which R_i is the i^{th} row of the rating matrix for active user i. The procedure of using the proposed method to rate predictions is summarized in Algorithm 2.

Algorithm 2. *TrustDL for rating prediction*

Begin
1. Input
The rating matrix of training data *R*, learned dictionary *B*, the embedding matrix *P*, and active user vector *a*.
2. Solving the coding coefficient vector
The coding coefficient vector *q a* of the query user vector *a* over *B* can be obtained by solving Eq. (19).
3. Rating prediction
Obtain rating vector $\hat{r}_i$ of the active user vector *a* with Eq. (20).
End

4. Experiments

Several tests were conducted, as reported in this section, to compare the recommendation quality of the techniques used in this study to those of well-known trust-based and CF-based recommendation systems in order to indicate the success of the proposed algorithm. The section is organized as follows. The datasets utilized in the study are summarized in Section 4.1. The evaluation measures are discussed in Section 4.2. While Section 4.3 examines eleven well-known, cutting-edge comparison techniques. Finally, Section 4.4 discusses the comparison to trust-based approaches, performance facing cold-start users, comparison to other models, parameter-sensitive tests, the scalability of the presented approach, and the impact of the *k* dimensions. All the experiments were run on a PC with 6 GB of RAM and a Core i7 processor using the Java programming language. Furthermore, all the techniques were developed on top of LibRec 2.0.0, which already involved the majority of the well-known methods.

4.1. Datasets

This section introduces the datasets[1] utilized to test the proposed approaches. Epinions, FilmTrust, Flixster, and Ciao were the four data sources we drew on for this study. These four datasets are probably the only ones providing access to data that include the social relationships and item evaluations provided by active users. They have been employed extensively in previous research on trust-aware RSs. The articles in FilmTrust and Flixster pertain solely to movies, but those in Epinions and Ciao are widely varied, concerning computers, sports, technology, and other fields. A rating in FilmTrust is an actual number in the range [0.5, 4.0], one in Flixster lies in [0.5, 5.0], both with increments of 0.5, and a rating in Epinions or Ciao is an integer between 1 and 5. By interacting with others with similar interests, users in these data may generate social networks by exchanging ratings of sets. Table 3 gives the statistics on the datasets. While the social relationships in Epinions and Ciao are trusted ones, the equivalent relationships in Flixster and FilmTrust can be referred to as trust-like. Users are usually specified as trustworthy in the former datasets, as suggested by evaluations of textual reviews and the quality of others' ratings. The notion of friendship *per se* is adopted in Flixster only in symmetric movie-related user relations. The notion of trust is adopted in FilmTrust, where the original values range between 1 and 10, but the version available to the public involves only binary values. Due to the tremendous amount of noise caused by such degrading, the relationships are categorized not as trust but as trust-like. Users who trust an active user (*i.e.*, trustees) and users who are trusted by the active user (*i.e.*, neighbors) were combined in this work to provide a complete picture (*i.e.*, trustees). The trust-based model was generated using the statistics from the four datasets.

4.2. Evaluation metrics

Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) are two standard metrics employed in evaluation processes to assess prediction performance (Guo et al., 2015). RMSE is always larger than MAE, and they both range in [0 ϴ].

$$\mathrm{RMSE} = \sqrt{\frac{\sum_{(u,i)} \left(\hat{r}_{u,i} - r_{u,i} \right)^2}{H}} \tag{21}$$

$$MAE = \frac{\sum_{(u,i)} \left| \hat{r}_{u,i} - r_{u,i} \right|}{H} \tag{22}$$

The results of the comparison were reported by the MAE and RMSE measures for evaluation, with the following definitions:

where $r_{u,i}$ is an actual rate, $\hat{r}_{u,i}$ is an estimated rate, and H is the set of ratings of user u on item i.

[1] https://guoguibing.github.io/librec/datasets.html

Table 4
Parameter Settings.

Methods	Parameters
SVD++	Factors = 10. Reg = 0.03. Learning-rate = 0.05. Max-iter = 100.
Ayub	Factors = 10. β = 0.5. α = 0.5. Max-iter = 100.
GA	Factors = 10. β = 0.5. α = 0.5. Max-iter = 100.
PMF	Factors = 5. Reg = 0.01. Learning-rate = 60. Moment = 0.8. Max-iter = 200.
TrustMF	Factors = 10. Reg = 0.001. Reg-Social = 1. Learning-rate = 0.001. Max-iter = 200.
LLORMA	Factors = 6. Reg-U = 0.01. Reg-I = 0.01. Model. num = 55. Max-iter = 200.
Social MF	Factors = 10. Reg = 0.001. Reg-S = 5. Learning-rate = 0.05. Max-iter = 200.
SLAM	Factors = 10. Reg = 0.001. Reg-C = 1. Learning-rate = 0.001. Max-iter = 100.
EMR	Factors = 10. λ_P = 0.06.λ_Q = 0.06. Max-iter = 150.
TrustSVD	Factors = 10. Reg = 0.6. Reg-S = 0.5. Learning-rate = 0.001. Max-iter = 130.
TrustANLF	Factors = 10. λ = 0.5. η = 0.3. β = 0.05. Max-iter = 100.
ABC-T	Factors = 10. Reg = 0.05. Reg-S = 1. Learning-rate = 0.03. Max-iter = 120.
UTV	Factors = 10. Reg = 0.005. Reg-S = 1. Learning-rate = 0.04. Max-iter = 150.
CCI	Factors = 10. Reg = 0.06. Reg-S = 1. Learning-rate = 0.05. Max-iter = 100.
CFMT	Factors = 10. β = 0.08. λ_P = 0.3.λ_Q = 0.08. Max-iter = 100.
TrustDL	Factors = 10. β = 0.08. α = 0.3. Max-iter = 100.

4.3. Baselines and parameter setting

Our proposed method was compared to the others in the conducted experiments, as briefly described here.

SVD++. SVD++ generates a more precise matrix factorization model by merging the factor and neighborhood models. The model is then extended to take advantage of implicit and explicit user feedback (Koren, 2008).

PMF. For high performance on large-scale and sparse data, PMF introduces the probabilistic matrix factorization model, which scales linearly with the number of observations (Mnih & Salakhutdinov, 2008).

SocialMF. It is a model-based approach to social network recommendation that incorporates matrix factorization methods and the trust propagation mechanism. It makes suggestions for a certain user based on the ratings of users with direct/indirect social interactions with that user (Jamali & Ester, 2010).

LLORMA. The Local Low-Rank Matrix Approximation model assumes that the observed matrix is a weighted sum of low-rank matrices (Lee, Kim, Lebanon, & Singer, 2013).

TrustSVD. On top of SVD++, TrustSVD combines implicit and explicit trust as an extra source to address the cold start and data sparsity issues (Guo et al., 2015).

GA. Through the application of genetic operators onto PCC, Cosine, and extended Jaccard similarity, GA changes the weights of the acquired similarity. Its apparent flaw is that it rejects the users' details, including trusted friends (Ar & Bostanci, 2016).

TrustMF. TrustMF is a model-based technique employing matrix factorization in two low-dimensional spaces, *i.e.*, those of the truster and the trustee, via factorization of trust networks based on the directional feature of trust. The two spaces are subsequently employed concurrently with the user and item spaces acquired from the factorization of the rating matrix for the generation of fused models and higher realistic prediction of genuine users' desires (Yang et al., 2016).

SLAM. It stands for Sparse Latent Model, which is based on the concepts of matrix factorization and sparse representation. The object and user representation vectors in the latent space are predicted to be sparse in SLAM as a result of the l1-regularization applied to those vectors (Feng, Wu, Tang, & Li, 2018).

EMR. It is a top-n recommender system that generates a list of objects most likely relevant to a certain user. The proposed approaches are compared to current collaborative filtering recommender systems using a range of measures, and they are shown to be competitive (Kartoglu & Spratling, 2018).

Ayub. It generates a unified user rating profile for the target user in order to provide more robust, more precise recommendations based on implicit and explicit trust and similarity in user preferences (Ayub et al., 2019).

TrustANLF. It is a strategy presented to reduce sparsity and cold user concerns by incorporating users' social trust information into the NMF framework and utilizing the Alternating Direction Optimization method to increase convergence time (Parvin et al., 2019b).

UTV. This study proposed a movie recommender system using ensemble learning and graph embedding. This system was trained on the MovieLens datasets. In the first step, some classes are generated using ensemble learning. Then, a UTV is generated for each user based on the extracted fuzzy rules and a combination of vectors. Finally, recommendations on the outputs of the three vectors are made to the user based on the results achieved from them (Forouzandeh, Berahmand, & Rostami, 2021a).

CCI-TrustWalker. In this paper, the items' scores that the user has not rated are predicted, and a trust-based recommender system is employed to anticipate the scores of these items. To reach this goal, a trusted network is generated that consists of users with similar behavior to the target user in selecting items and friends. Then, after the generation of the trusted network, a TrustWalker is developed, which can randomly select the network nodes by employing the BRW algorithm. In the end, before the movement of TrustWalker between users on the network, the degree of trust between them is determined and computed (Forouzandeh et al., 2021b).

ABC-T. This system was designed and implemented using

Table 5
A comparison to trust-based methods in the testing view of all[a].

Datasets	Error Metrics	Methods												
		EMR	TrustMF	SocialMF	TrustSVD	SLAM	TrustANLF	Ayub	GA	ABC-T	UTV	CCI	CFMT	**TrustDL**
FilmTrust	MAE	0.640	0.721	0.698	0.607	0.638	0.584	0.668	0.672	0.601	0.595	0.588	0.584	**0.574**
	RMSE	0.835	0.919	0.852	0.787	0.831	0.777	0.868	0.882	0.810	0.800	0.785	0.789	**0.769**
Epinions	MAE	0.958	0.877	0.862	0.834	0.884	0.785	0.944	0.953	0.802	0.782	0.779	0.775	**0.762**
	RMSE	1.278	1.184	1.104	1.094	1.142	1.063	1.307	1.302	1.025	1.018	1.016	1.031	**1.014**
Ciao	MAE	0.826	0.505	0.637	0.723	0.769	0.519	0.794	0.796	0.655	0.610	0.578	**0.504**	0.614
	RMSE	1.075	0.710	0.905	0.955	1.035	0.720	1.110	1.089	0.918	0.861	0.818	0.716	**0.704**
Flixter	MAE	0.785	0.625	0.637	0.723	0.785	–	–	–	0.777	0.764	0.758	0.631	**0.620**
	RMSE	1.012	0.710	0.905	0.955	1.025	–	–	–	1.08	1.06	1.03	0.730	**0.702**

[a] The bold values indicate the best performance among the results of experiments.

Table 6
A comparison of the proposed method with social-based methods in terms of cold-start users[a].

Datasets	Error Metrics	Methods												
		EMR	TrustMF	SocialMF	TrustSVD	SLAM	TrustANLF	Ayub	GA	ABC-T	UTV	CCI	CFMT	**TrustDL**
FilmTrust	MAE	0.680	0.619	0.589	0.650	0.648	0.607	0.633	0.652	0.642	0.62	0.611	0.632	**0.575**
	RMSE	0.884	0.882	0.818	0.845	0.848	0.784	0.824	0.868	0.887	0.868	0.855	0.854	**0.774**
Epinions	MAE	1.051	0.934	0.919	0.861	0.892	0.842	**0.502**	0.695	0.818	0.799	0.771	0.823	0.793
	RMSE	1.266	1.373	1.312	1.117	1.138	1.090	**0.878**	0.952	1.055	1.057	0.975	1.062	1.042
Ciao	MAE	0.802	1.073	1.014	0.725	0.736	0.716	0.805	0.838	0.766	0.755	0.74	0.836	**0.715**
	RMSE	1.015	1.311	1.266	0.939	1.036	0.928	1.211	1.185	0.969	0.943	0.938	1.141	**0.892**
Flixter	MAE	0.885	0.975	0.885	0.845	0.896	–	–	–	0.889	0.914	0.876	0.880	**0.832**
	RMSE	1.136	1.326	1.112	1.052	1.145	–	–	–	1.091	1.147	1.097	1.170	**1.071**

[a] The bold values indicate the best performance among the results of experiments.

Table 7
A comparison to rating-based methods[a].

Datasets	Error Metrics	Methods								
		UAvg	IAvg	PMF	SVD++	RSDL	LLORMA	TrustANLF	CFMT	**TrustDL**
FilmTrust	MAE	0.729	0.696	0.753	0.699	0.661	0.848	0.584	0.582	**0.574**
	RMSE	0.943	0.925	1.02	0.891	0.884	1.041	0.777	0.789	**0.787**
Epinions	MAE	0.925	0.823	1.35	0.912	0.973	1.62	0.785	0.775	**0.762**
	RMSE	1.19	1.09	1.81	1.205	1.101	2.03	1.063	1.031	**1.014**
Ciao	MAE	0.456	0.255	**0.24**	0.745	0.841	0.803	0.519	0.506	0.614
	RMSE	0.678	0.480	**0.42**	1.00	0.659	1.014	0.720	0.768	0.904
Flixter	MAE	**0.725**	0.856	0.765	0.825	0.802	–	–	0.755	0.752
	RMSE	**0.978**	1.085	1.002	1.096	1.062	–	–	1.01	1.00

[a] The bold values indicate the best performance among the results of experiments.

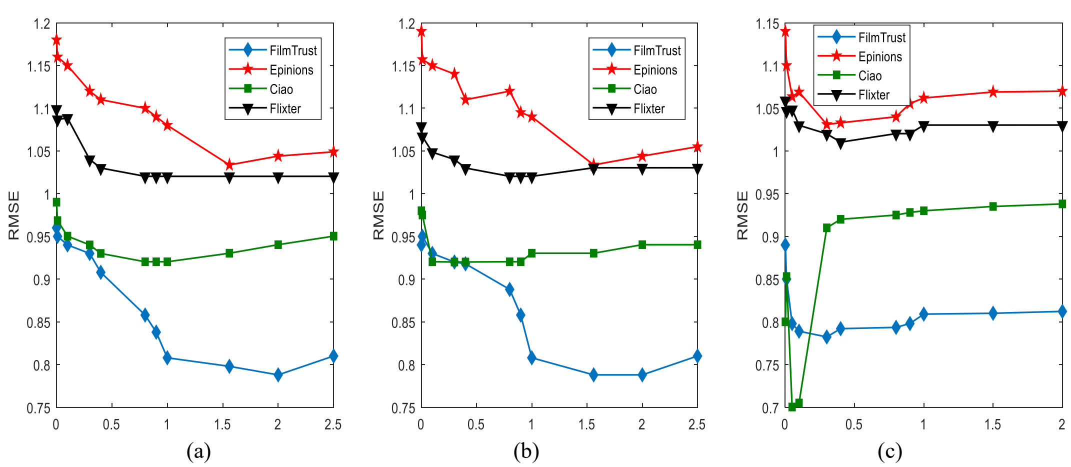

Fig. 2. (a) The effect of parameter α, (b) the effect of parameter β and (c) the effect of parameter γ.

evolutionary algorithms and TOPSIS fuzzy model. At first, TOPSIS determines a positive ideal solution as a matrix containing four columns to save factors investigated in this study. Then, the desired indicators and their criteria are evaluated using the TOPSIS model and decision matrix, and the ideal points are identified through the conversion of qualitative values into quantitative ones to resolve the problem and the distances of other points or the same tourist places in each city to the ideal points are then computed (Forouzandeh, Rostami, & Berahmand, 2022).

CFMT. This approach provides a trust-based matrix factorization technique known as CFMT, which models user roles as trusters and trustees given the structural information of the trust network and employs the social network data in the recommendation process. By incorporating information sources like ratings and trust statements into the recommendation model, the suggested solution aims to mitigate the cold start and data sparsity challenges (Khaledian & Mardukhi, 2022).

In the approaches that were compared, parameters were set, as indicated in Table 4.

4.4. Comparison to trust-based models

For assessment of the performance of our proposed method, it was compared to that of a number of well-reputed state-of-the-art related

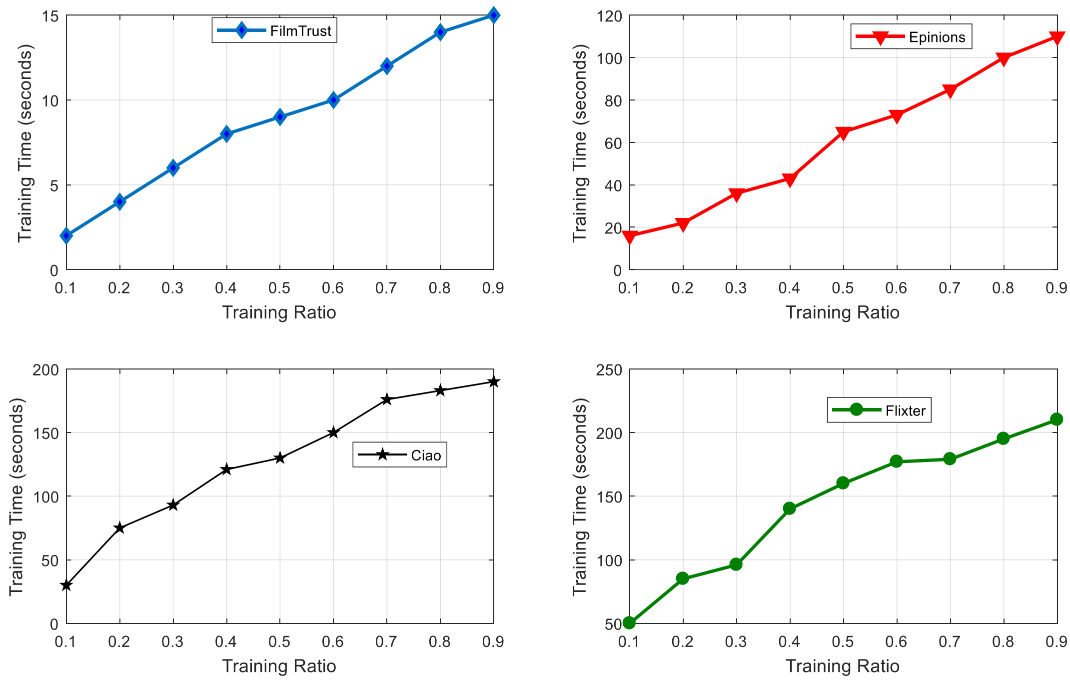

Fig. 3. The scalability of *ourmethod* across all the datasets.

methods, including Emr (Kartoglu & Spratling, 2018), TrustMF (Yang et al., 2016), SocialMF (Jamali & Ester, 2010), TrustSVD (Guo et al., 2015), SLAM (Feng et al., 2018), TrustANLF (Parvin et al., 2019b), Ayub (Ayub et al., 2019), GA (Ar & Bostanci, 2016), UTV (Forouzandeh et al., 2021a), CCI-TrustWalker (Forouzandeh et al., 2021b), and ABC-T (Forouzandeh et al., 2022). The MAE and RMSE findings obtained using these approaches are displayed in Table 5. It should be noted that some of the methods, such as TrustANLF, Ayub, and GA, did not use the Flixter dataset in their evaluation, so the values in the table are null for them.

Table 5 shows the best results for each dataset in bold. TrustDL outperforms the compared methods in most cases. It achieves the best performance on FilmTrust in the MAE criterion and exhibits the lowest RMSE value on the Flixter dataset. TrustDL significantly reduces MAE with respect to the values obtained over EMR, TrustMF, SocialMF, TrustSVD, SLAM, TrustANLF, Ayub, GA, ABC-T, UTV, CCI, and CFMT by 10.31%, 20.38%, 17.76%, 5.43%, 10.03%, 1.71%, 14.07%, 14.58%, 4.49%, 3.52%, 2.38%, and 1.71%, respectively, on FilmTrust. In addition, TrustDL improves RMSE over EMR, TrustMF, SocialMF, TrustSVD, SLAM, ABC-T, UTV, CCI, and CFMT by 30.63%, 1.12%, 22.43%, 26.49%, 31.51%, 35%, 33.77%, 31.84%, and 3.83%, respectively, on Flixter.

The results indicate that our proposed strategy is consistently superior to the best of the approaches in the majority of cases, as detailed below. Our approach and TrustMF perform better than trust-based models. The experiments demonstrate the importance of the trust inference of relationships in the proposed method, which obtains excellent results. It exhibits higher accuracy than the others due to its

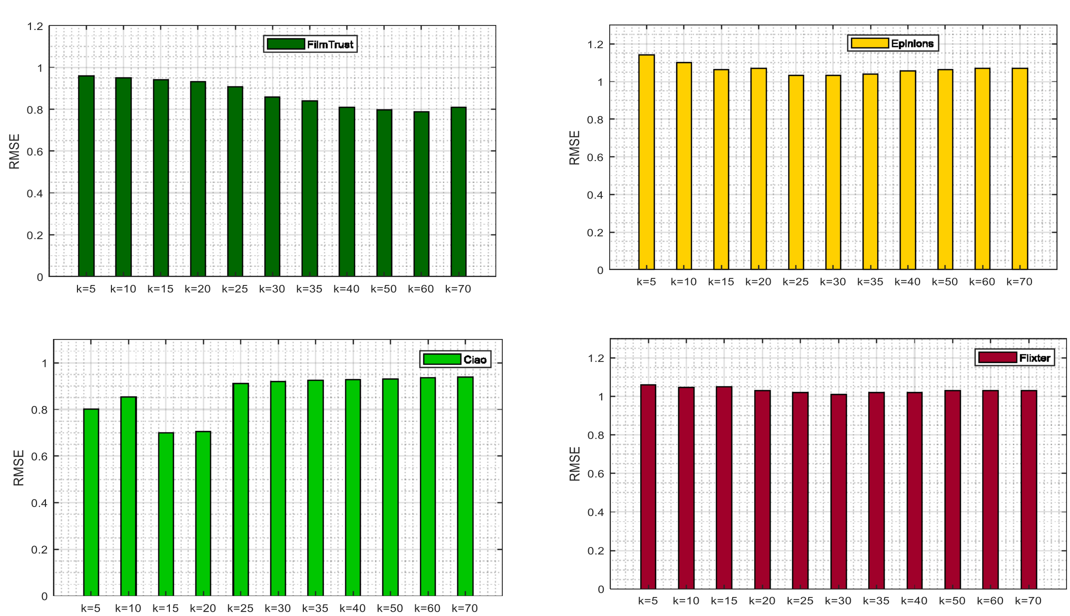

Fig. 4. The impacts of latent dimensionality on four different datasets.

application of the users' trust data. The proposed method employs the RSDL strategy to update the hidden factors along with subsidiary information, so it exhibits higher accuracy than the other methods in most cases. Furthermore, the proposed method maintains the local data structure using manifold regularization. However, the high dimensions in the recommender system data make up a major issue. To deal with the curse of dimensionality, an effective solution is to apply dimension reduction methods. Fortunately, DL represents data using a smaller set of basic functions through dimension reduction; it can reduce the problem dimensionality to improve the recommendations.

4.5. Performance in the face of cold-start users

The efficiency of the proposed strategy was tested on cold-start users and reported in this subsection. Handling new users or ones with low rating counts is a critical concern for the success of an RS, as this type of user frequently occurs in real-world scenarios and applications, making it necessary to manage cold-start users as a substantial obstacle in present systems. Table 6 compares the efficacy of the techniques mentioned above and reveals that using trust statements significantly improves recommendation performance quality. This conclusion is interesting, as it exposes that it is feasible to ease the scarcity of user ratings for new and cold-start users by adding trust ties among them. In most cases, the proposed strategy outperformed the trust-based ones in the tests.

According to the table, TrustDL properly resolves the cold start problem. Its best performance pertains to the FilmTrust dataset. On average, the MAE and RMSE criteria improved by more than 8.86% and 8.9%, respectively. TrustDL reduced MAE over EMR, TrustMF, SocialMF, TrustSVD, SLAM, TrustANLF, Ayub, GA, ABC-T, UTV, CCI, and CFMT by 15.44%, 7.10%, 2.37%, 11.53%, 11.26%, 5.27%, 9.16%, 11.80%, 10.43%, 7.25%, 5.89%, and 9.16%, respectively. It also improved RMSE over EMR, TrustMF, SocialMF, TrustSVD, SLAM, TrustANLF, Ayub, GA, ABC-T, UTV, CCI, and CFMT by 12.44%, 12.24%, 5.37%, 8.40%, 8.72%, 1.27%, 6.06%, 10.82%, 12.73%, 10.82%, 9.47%, and 9.46%, respectively.

It is notable that in various genuine apps, most users express their interests mainly in small numbers of objects. With up to five stated ratings, these individuals are referred to as cold-start users, who constitute shares of more than 50% of the users in the four datasets. This discovery accords tremendous significance to the success of any recommendation mechanism in the event of cold users. The outcomes demonstrate that the proposed method has superior performance compared to the previous ones along with Ayub because we include both in-link and out-link impacts of user rating and user trust to tackle the cold start and data sparsity issues. The comparison suggests that the strategy provided in this research exhibits greater performance in terms of prediction accuracy than the trust-based recommendation strategies. The superiority and advantages of our proposed method to the others result from the proper use of subsidiary information to contribute to the ratings to gain resistance against cold start and data sparsity and thus exhibit better performance in most cases. TrustDL makes it possible to embed trust in the new data representation along with the rating matrix. The use of trust data has an effective role in the generation of a valid, accurate model, which makes it possible to deal with a cold start since DL can improve accuracy by providing more efficient data representations.

4.6. Comparisons with other models

As an additional particular issue, the performance conditions of well-reputed rating-based techniques were explored and compared to those in the proposed method. For that purpose, the efficiency of our proposed method was compared to that of the benchmark and rating-based algorithms, namely UAvg, IAvg, PMF (Mnih & Salakhutdinov, 2008), SVD++ (Koren, 2008), RSDL (Luo et al., 2014), LLORMA (Lee et al., 2013), TrustANLF (Parvin, et al., 2019b) and CFMT (Khaledian & Mardukhi, 2022). Table 7 presents the outcomes of the experiments conducted with the above seven algorithms on the datasets. Before comparing the proposed algorithm to the others that simply use rating information, we made examinations with and without trust relationships. Compared to rating-based models, the technique proposed in this research performed very well in terms of prediction accuracy. Rating-based methods do not use subsidiary information, so they have very low accuracy on large datasets. However, the proposed method exhibits higher accuracy compared to the others due to the proper use of subsidiary information.

Although the proposed method performs well in most cases, it is not the best on some datasets. This is due to the nature of learning methods and the influence of hyper-parameters on them. These methods need to be adjusted according to the type, distribution, and noise in the data. In our study, we considered all the experiments under the same conditions for better comparison. This issue can have a negative effect on learning. Therefore, we examined the impacts of meta-parameters on learning results.

4.7. Parameter-sensitive tests

The consequences of the parameter values considered in the proposed method are examined in this section. α, β, and γ are some of these factors. Several tests were conducted to specify how modifications in these factors affect recommendation accuracy. The findings for parameters α, β, and γ are shown in Fig. 2. For the first experiment, λ_Q and β were given a fixed value of 1; then, the proposed approach was tested as the value of α enhanced from 0.001 to 3. The findings in Fig. 2 show that the proposed approach failed to converge satisfactorily as the value of α declined. In other words, the performance of our model was found to be connected to the value of α. The appropriate α value for obtaining the required outcome on various datasets was $\alpha = 0.3$. The value of $\beta = 0.08$ also provided a proper setting for a variety of data types. The β parameter specified the amount of social network data that the proposed method would utilize to construct the observed rating matrix. It can be observed that the algorithm ignored the information on user trust for small values of β and simply used the perceived user rating for factorization. Where γ was given a large value, however, the trust information controlled the learning process, led to a poor performance. As a result, in order to prevent damage to recommendation performance, a rational value needs to be specified for the parameter of social regularization, accomplished by an examination of how the combination of these factors affects recommendation performance. Because the ideal β value to obtain great performance is likely to vary from one dataset to the next, it is acceptable to apply a value of 0.05 to this parameter.

4.8. The scalability of the proposed method

The capacity of recommender systems to handle huge amounts of data and users effectively without sacrificing their predicted accuracy is known as scalability. As the amount of the dataset increases, the recommender system's scalability becomes increasingly important. A system that performs well with a small dataset might not be effective with a larger dataset. Memory-based algorithms provide recommendations based on the complete dataset, while model-based algorithms model the data to generate predictions. Although model-based techniques require more computational resources, they are often more scalable. Since TrustDL is a model-based method, hence, we have evaluated the time required to train the model based on the increase in the volume of the dataset. For scenarios when our technique was applied to change dataset percentages, the scalability of our approach in terms of training time was investigated, notably for the range from 0.1 percent to 1 percent with intervals of 0.1. The findings in Fig. 3 demonstrate that training time increased linearly as the amount of training data grew. As a result, this strategy is capable of resolving large-scale CF issues and is susceptible to the applicability to big datasets. Thus, it may be employed in big data applications with limited numbers of tuning processes.

4.9. Impact of the k dimension

In matrix factorization, the dimension of the hidden space refers to the number of hidden factors or features employed to indicate the original data matrix. The number of hidden dimensions is a meta-parameter that should be selected based on the complexity of the data and the desired accuracy in its display. This section discusses the impact of latent space dimensions on the function of TrustDL. Fig. 4 shows the impacts of different dimensions on the FilmTrust, Epinions, Ciao, and Flixster datasets. Given the huge amount of available data, the greater the values of the characteristics, the better the efficiency. However, it is worth noting that raising the latent space dimensions increased the computational complexity of the algorithm, so we needed to obtain the optimal number to achieve the best results. Fig. 4 indicates that TrustDL technique performed well on the four datasets for $k = 10$ (solution). It means the value of $k = 10$ yielded the best results on all four datasets.

5. Conclusion

Statements of utility trust were used in the recommendation process in this research. Experimental studies have demonstrated the relevance of trust in traditional RSs as a useful strategy for the resolution of CF issues. To improve feature representation learning, we applied the dictionary learning approach to the rating feature space and generated the trust consistency regularization term. On the other hand, most CF methods consider only the global Euclidean structure, disregarding another structure that is often significant for many real applications, *i.e.*, the local manifold structure. To exploit the local manifold data structure in this research, the objective function was equipped with a manifold regularization that featured a Laplacian graph. Compared to the popular relevant methods, the proposed method exhibited low computational cost, high speed, and sufficient effectiveness when confronted with data sparsity and cold-start users. We employed social trust relationships as an extra source of information to generate more accurate predictions at a reduced computational cost and still with excellent accuracy. The overall findings indicated that our strategy was far better than well-known state-of-the-art related works in terms of prediction recommendation accuracy. We plan to develop the proposed method into richer ones for future work by utilizing users' comments and other subsidiary information on items. Another possible direction is to employ more sophisticated technologies in artificial intelligence, such as deep learning, to simultaneously explore more helpful information from the explicit and implicit impacts of item rating and user trust.

CRediT authorship contribution statement

Navid Khaledian: Conceptualization, Methodology, Software, Writing – original draft. **Amin Nazari:** Visualization, Data curation, Writing – original draft. **Keyhan Khamforoosh:** Supervision, Validation. **Laith Abualigah:** Validation, Writing – review & editing. **Danial Javaheri:** Investigation, Formal analysis, Supervision, Writing – review & editing.

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.

Data availability

The authors do not have permission to share data.

Acknowledgement

This research was supported by Basic Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science and ICT (RS-2022-00166712), Research fund from Chosun University (2022).

References

Ahmadian, S., Joorabloo, N., Jalili, M., & Ahmadian, M. (2022). Alleviating data sparsity problem in time-aware recommender systems using a reliable rating profile enrichment approach. *Expert Systems with Applications, 187*, Article 115849.
Alhijawi, B., & Kilani, Y. (2020). A collaborative filtering recommender system using genetic algorithm. *Information Processing & Management, 57*(6), Article 102310.
Amestoy, P., Buttari, A., Higham, N., l'Excellent, J.-Y., Mary, T., & Vieuble, B. (2022). Combining sparse approximate factorizations with mixed precision iterative refinement, 49(1), 1–29.
Ar, Y., & Bostanci, E. (2016). A genetic algorithm solution to the collaborative filtering problem. *Expert Systems with Applications, 61*, 122–128.
Ayub, M., Ghazanfar, M. A., Mehmood, Z., Saba, T., Alharbey, R., Munshi, A. M., & Alrige, M. A. (2019). Modeling user rating preference behavior to improve the performance of the collaborative filtering based recommender systems. *PloS One, 14*(8), Article e0220129.
Azadjalal, M. M., Moradi, P., Abdollahpouri, A., & Jalili, M. (2017). A trust-aware recommendation method based on Pareto dominance and confidence concepts. *Knowledge-Based Systems, 116*, 130–143.
Behera, G., & Nain, N. (2022a). Handling data sparsity via item metadata embedding into deep collaborative recommender system. *Journal of King Saud University-Computer and Information Sciences, 34*(10), 9953–9963.
Behera, G., & Nain, N. (2022b). Trade-off between memory and model-based collaborative filtering recommender system. *Proceedings of the international conference on paradigms of communication, computing and data sciences*, 137-146.
Chen, G., Xu, C., Wang, J., Feng, J., & Feng, J. (2020). Nonnegative matrix factorization for link prediction in directed complex networks using PageRank and asymmetric link clustering information. *Expert Systems with Applications, 148*, Article 113290.
Chen, L.-J., & Gao, J. (2018). A trust-based recommendation method using network diffusion processes. *Physica A: Statistical Mechanics and its Applications, 506*, 679–691.
Du, H., Zhang, Y., Ma, L., & Zhang, F. (2021). Structured discriminant analysis dictionary learning for pattern classification. *Knowledge-Based Systems, 216*, Article 106794.
Feng, X., Wu, S., Tang, Z., & Li, Z. (2018). Sparse latent model with dual graph regularization for collaborative filtering. *Neurocomputing, 284*, 128–137.
Forouzandeh, S., Berahmand, K., & Rostami, M. (2021a). Presentation of a recommender system with ensemble learning and graph embedding: A case on MovieLens. *Multimedia Tools and Applications, 80*(5), 7805–7832.
Forouzandeh, S., Rostami, M., & Berahmand, K. (2021b). Presentation a Trust Walker for rating prediction in recommender system with Biased Random Walk: Effects of H-index centrality, similarity in items and friends. *Engineering Applications of Artificial Intelligence, 104*, Article 104325.
Forouzandeh, S., Rostami, M., & Berahmand, K. (2022). A hybrid method for recommendation systems based on tourism with an evolutionary algorithm and Topsis model. *Fuzzy Information and Engineering*, 1–25.
Geluvaraj, B., & Sundaram, M. (2022). A hybrid approach to resolve data sparsity and cold start hassle in recommender systems. In *Pervasive Computing and Social Networking* (pp. 499–510). Springer.
Guo, G., Zhang, J., & Yorke-Smith, N. (2015). Trustsvd: Collaborative filtering with both the explicit and implicit influence of user trust and of item ratings. *Proceedings of the AAAI conference on artificial intelligence*, 29(1), 123-129.
Guo, J., Zhou, Y., Zhang, P., Song, B., & Chen, C. (2021). Trust-aware recommendation based on heterogeneous multi-relational graphs fusion. *Information Fusion, 74*, 87–95.
Jakomin, M., Bosnić, Z., & Curk, T. (2020). Simultaneous incremental matrix factorization for streaming recommender systems. *Expert Systems with Applications, 160*, Article 113685.
Jamali, M., & Ester, M. (2010). A matrix factorization technique with trust propagation for recommendation in social networks. *Proceedings of the fourth ACM conference on Recommender systems*, 135-142.
Javaheri, D., Gorgin, S., Lee, J.-A., & Masdari, M. (2022). An improved discrete harris hawk optimization algorithm for efficient workflow scheduling in multi-fog computing. *Sustainable Computing: Informatics and Systems, 36*, Article 100787. https://doi.org/10.1016/j.suscom.2022.100787
Jing, X.-Y., Wu, F., Li, Z., Hu, R., & Zhang, D. (2016). Multi-label dictionary learning for image annotation. *IEEE Transactions on Image Processing, 25*(6), 2712–2725.
Kartoglu, I. E., & Spratling, M. W. (2018). Two collaborative filtering recommender systems based on sparse dictionary coding. *Knowledge and Information Systems, 57*(3), 709–720.
Khaledian, N., & Mardukhi, F. (2022). CFMT: A collaborative filtering approach based on the nonnegative matrix factorization technique and trust relationships. *Journal of Ambient Intelligence and Humanized Computing*, 1–17.
Kiran, R., Kumar, P., & Bhasker, B. (2020). DNNRec: A novel deep learning based hybrid recommender system. *Expert Systems with Applications, 144*, Article 113054.

Kordabadi, M., Nazari, A., & Mansoorizadeh, M. (2022). A movie recommender system based on topic modeling using machine learning methods. *International Journal of Web Research, 5*(2), 19–28. https://doi.org/10.22133/ijwr.2022.370251.1139

Koren, Y. (2008). Factorization meets the neighborhood: a multifaceted collaborative filtering model. *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, 426-434.

Kuo, R., Chen, C.-K., & Keng, S.-H. (2021). Application of hybrid metaheuristic with perturbation-based K-nearest neighbors algorithm and densest imputation to collaborative filtering in recommender systems. *Information Sciences, 575*, 90–115.

Lee, J., Kim, S., Lebanon, G., & Singer, Y. (2013). Local low-rank matrix approximation. *International Conference on Machine Learning*, 82–90.

Li, H., Li, K., An, J., Zheng, W., & Li, K. (2019a). An efficient manifold regularized sparse non-negative matrix factorization model for large-scale recommender systems on GPUs. *Information Sciences, 496*, 464–484.

Li, J., Tao, J., Ding, W., Zhang, J., & Meng, Z. (2022). Period-assisted adaptive parameterized wavelet dictionary and its sparse representation for periodic transient features of rolling bearing faults. *Mechanical Systems and Signal Processing, 169*, Article 108796.

Li, W., Mo, J., Xin, M., & Jin, Q. (2018). An Optimized trust model integrated with linear features for cyber-enabled recommendation services. *Journal of Parallel and Distributed Computing, 118*, 81–88.

Li, W., Ye, Z., Xin, M., & Jin, Q. (2017). Social recommendation based on trust and influence in SNS environments. *Multimedia Tools and Applications, 76*(9), 11585–11602.

Li, W., Zhou, X., Shimizu, S., Xin, M., Jiang, J., Gao, H., & Jin, Q. (2019). Personalization recommendation algorithm based on trust correlation degree and matrix factorization. *IEEE Access, 7*, 45451–45459.

Livne, A., Tov, E. S., Solomon, A., Elyasaf, A., Shapira, B., & Rokach, L. (2022). Evolving context-aware recommender systems with users in mind. *Expert Systems with Applications, 189*, Article 116042.

Luo, X., Zhou, M., Xia, Y., & Zhu, Q. (2014). An efficient non-negative matrix-factorization-based approach to collaborative filtering for recommender systems. *IEEE Transactions on Industrial Informatics, 10*(2), 1273–1284.

Mnih, A., & Salakhutdinov, R. R. (2008). Probabilistic matrix factorization. *Advances in Neural Information Processing Systems, 20*, 1257–1264.

Nazari, A., Kordabadi, M., & Mansoorizadeh, M. (2023). Scalable and data-independent multi-agent recommender system using social networks analysis. *International Journal of Information Technology & Decision Making, 22*(4), 1–22. https://doi.org/10.1142/s021962202350030x

Papadakis, H., Papagrigoriou, A., Panagiotakis, C., Kosmas, E., & Fragopoulou, P. (2022). Collaborative filtering recommender systems taxonomy. *Knowledge and Information Systems, 64*(1), 35–74.

Parvin, H., Moradi, P., & Esmaeili, S. (2019). TCFACO: Trust-aware collaborative filtering method based on ant colony optimization. *Expert Systems with Applications, 118*, 152–168.

Parvin, H., Moradi, P., Esmaeili, S., & Qader, N. N. (2019b). A scalable and robust trust-based nonnegative matrix factorization recommender using the alternating direction method. *Knowledge-Based Systems, 166*, 92–107.

Permiakova, O., & Burger, T. (2022). Sketched Stochastic Dictionary Learning for large-scale data and application to high-throughput mass spectrometry. *Statistical Analysis and Data Mining: The ASA Data Science Journal, 15*(1), 43–56.

Qi, Z., Yue, K., Duan, L., Wang, J., Qiao, S., & Fu, X. (2021). Matrix factorization based Bayesian network embedding for efficient probabilistic inferences. *Expert Systems with Applications, 169*, Article 114294.

Rashidi, R., Khamforoosh, K., & Sheikhahmadi, A. (2021). Proposing improved meta-heuristic algorithms for clustering and separating users in the recommender systems. *Electronic Commerce Research*, 1–26.

Rodpysh, K. V., Mirabedini, S. J., & Banirostam, T. (2021). Employing singular value decomposition and similarity criteria for alleviating cold start and sparse data in context-aware recommender systems. *Electronic Commerce Research*, 1–27.

Tahmasbi, H., Jalali, M., & Shakeri, H. (2021). TSCMF: Temporal and social collective matrix factorization model for recommender systems. *Journal of Intelligent Information Systems, 56*(1), 169–187.

Talmon, R., Mallat, S., Zaveri, H., & Coifman, R. R. (2015). Manifold learning for latent variable inference in dynamical systems. *IEEE Transactions on Signal Processing, 63*(15), 3843–3856.

Wang, J., Zhu, L., Dai, T., Xu, Q., & Gao, T. (2021). Low-rank and sparse matrix factorization with prior relations for recommender systems. *Applied Intelligence, 51*, 3435–3449.

Wang, W., Yan, Y., Nie, F., Yan, S., & Sebe, N. (2018). Flexible manifold learning with optimal graph for image and video representation. *IEEE Transactions on Image Processing, 27*(6), 2664–2675.

Yang, B., Lei, Y., Liu, J., & Li, W. (2016). Social collaborative filtering by trust. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 39*(8), 1633–1647.

Zhang, F., Qi, S., Liu, Q., Mao, M., & Zeng, A. (2020). Alleviating the data sparsity problem of recommender systems by clustering nodes in bipartite networks. *Expert Systems with Applications, 149*, Article 113346.

Zhou, L., Du, G., Lü, K., & Wang, L. (2021). A network-based sparse and multi-manifold regularized multiple non-negative matrix factorization for multi-view clustering. *Expert Systems with Applications, 174*, Article 114783.

Zhu, B., Liu, J. Z., Cauley, S. F., Rosen, B. R., & Rosen, M. S. (2018). Image reconstruction by domain-transform manifold learning. *Nature, 555*(7697), 487–492.