



Deep matrix factorization via feature subspace transfer for recommendation system

Weichen Wang¹ · Jing Wang¹

Received: 25 October 2023 / Accepted: 9 March 2024
© The Author(s) 2024

Abstract

The sparsity problem remains a significant bottleneck for recommendation systems. In recent years, deep matrix factorization has shown promising results in mitigating this issue. Furthermore, many works have improved the prediction accuracy of deep matrix factorization by incorporating the user's and/or items' auxiliary information. However, there are still two remaining drawbacks that need to be addressed. First, the initialization of latent feature representations has a substantial impact on the performance of deep matrix factorization, and most current models utilize a uniform approach to this initialization, constraining the model's optimization potential. Secondly, many existing recommendation models lack versatility and efficiency in transferring auxiliary information from users or items to expand the feature space. This paper proposes a novel model to address the issues mentioned above. By using a semi-autoencoder, the pre-trained initialization of the latent feature representation is realized in this paper. Simultaneously, this model assimilates auxiliary information, like item attributes or rating matrices from diverse domains, to generate their latent feature representations. These representations are then transferred to the target task through subspace projection distance. With this, this model can utilize auxiliary information from various sources more efficiently and this model has better versatility. This is called deep matrix factorization via feature subspace transfer. Numerical experiments on several real-world data show the improvement of this method compared with state-of-the-art methods of introducing auxiliary information about items. Compared with the deep matrix factorization model, the proposed model can achieve 6.5% improvement at most in the mean absolute error and root mean square error.

Keywords Deep matrix factorization · Feature subspace transfer · Semi-autoencoder · Cross-domain recommendation

Introduction

The exponential growth in online information has ushered in the development and application of recommendation systems (RS). Such systems are instrumental in mitigating the challenge of information overload and finding applications across various platforms like Amazon, Taobao, Jingdong, Facebook, Microblog, etc [1]. RS proactively suggests information or products aligned with user preferences by analyzing their past interaction records. Broadly, the traditional

techniques employed in RS can be grouped into two main categories: Content-based methods and Collaborative Filtering (CF) methods [2]. The former recommends items by drawing parallels between them and the user's historical preferences. In contrast, CF techniques primarily rely on user or item similarity metrics to formulate recommendations. They all relate to machine learning [3].

Collaborative Filtering (CF) methods enjoy widespread adoption due to their straightforward implementation and robust efficiency. A significant proportion of these CF approaches anchor on matrix factorization (MF). However, most existing MF models have defects in nonlinear feature learning. Additionally, the sparsity problem is a major bottleneck for MF. Recently, deep learning methods have shown the power of learning latent feature representation, and it has a wide range of applications in various fields [4–7]. Inspired by deep learning, many works combine deep learning with MF and propose deep matrix factorization models to address the aforementioned shortcomings and have demonstrated com-

Weichen Wang and Jing Wang contributed equally to this work.

✉ Jing Wang
wroaring@hqu.edu.cn
Weichen Wang
21013083025@hqu.stu.edu.cn

¹ School of Computer Science and Technology, Huaqiao University, No. 668 Jimei Avenue, Xiamen 361021, Fujian Province, China

mendable outcomes. Autoencoders stand out as a favored approach in this domain [8–10]. For instance, Sedhain et al. employed an autoencoder to learn feature representations of either items or users to Implement collaborative filtering [8]. Another popular method is the Multi-Layer Perceptron (MLP) [11, 12]. Fan et al. innovated by leveraging multi-layer neural networks to learn the latent representations of users or items to approximate the nonlinear latent variable model [11]. This model is more versatile and applicable to various tasks of matrix completion and has achieved satisfactory results.

Most methods that introduce auxiliary information from users and/or items have demonstrated significant prowess in addressing the sparsity challenge [13]. Zhang and Zou et al. introduced additional item/user attributes on the collaborative filtering model to improve the accuracy of rating prediction [14, 15]. However, due to the high cost of obtaining label knowledge and the increment awareness of user privacy protection, it is difficult to capture the auxiliary information of items/users attributes. He and Man et al. take advantage of rating information from closely correlated domains using transfer learning methodologies for recommendations [16, 17]. Yet, transfer learning often falls short when there are large differences in the structure of scoring data across different domains. Recently, autoencoder-based methods have been widely used in recommendation systems for the superiority of no-label requirement and fast convergence speed. Some works employ semi-autoencoder to co-embed the item's attribute information and the graph features of the items for rating prediction to improve prediction results [9, 10, 18–20]. Nonetheless, these models ignore the exploration of the latent feature representation, which curtails their optimization efficiency. Moreover, many existing recommendation models lack versatility and efficiency in expanding the feature space with user or item auxiliary information.

To address the challenges of the various models mentioned above, this paper proposes a novel model, called deep matrix factorization via feature subspace transfer (FSTDMF). Specifically, this paper introduces the semi-autoencoder [14] model to initialize the features of the item from the user-item rating matrix and integrate it into the FSTDMF model to facilitate advanced latent feature representation learning. Simultaneously, this work incorporates a subspace projection distance regular term to transfer the latent feature representations of the item's auxiliary information to the target task. Remarkably, FSTDMF is adept at harnessing auxiliary information from diverse sources, encompassing attribute data and rating information from other domains, etc. The key contributions to this paper are summarized as follows:

- To solve the low generality and low efficiency in transferring the auxiliary information of users or items to expand the feature space, this paper takes the subspace projection distance between the item auxiliary information latent feature and the item latent feature as the penalty term to realize the feature transfer.
- To solve the limitation of simple feature initialization method on model capability, the proposed model uses semi-autoencoder to initialize latent feature representation, and integrates it into FSTDMF model to obtain better latent feature representation and improve recommendation performance.
- The experimental results conducted on several public real-world datasets demonstrate the effectiveness of the proposed method for capturing more powerful feature representations compared with the start-of-the-art methods.

The remainder of this paper is organized as follows. In “Related work” section, the related work is briefly reviewed, and then the model of FSTDMF is proposed in “FSTDMF model” section. In “Experiments” section, The experimental setup and the experimental results of five real-world datasets are provided in detail. In “Conclusion” section, The proposed model is summarized, and future work is introduced.

Related work

Recommendation systems aim to gauge users' preferences for items and suggest those they might appreciate [21]. In the initial stages of recommendation system research, collaborative filtering emerged as the most popular and broadly utilized method [22]. Previous collaborative filtering methods can be divided into two categories, memory-based and model-based methods [23]. Memory-based methods are further subdivided into user-based and item-based methods. While both methods leverage rating data to determine the similarity between users or items, their principles are similar. The other category is model-based methods, and the representative technique is matrix factorization (MF) [24]. MF decomposes the rating matrix into a user's and an item's matrix and then makes predictions based on the two decomposed matrices. Singular value decomposition (SVD) stands out as a renowned matrix factorization model [24]. It decomposes the rating matrix into the inner product of two low-dimensional feature matrices and then makes predictions based on the reduced-dimensional matrix. An enhancement of SVD, known as SVD++, integrates implicit feedback [24]. While collaborative filtering offers effective recommendations in data-rich scenarios, its efficacy wanes in data-sparse situations.

The current methods to solve the problem of data sparsity include deep matrix factorization (DMF) methods [25]

and methods that integrate auxiliary information pertaining to users and items [26]. At the core of DMF, methodologies lies the adoption of deep learning to construct a nonlinear model, transcending the confines of traditional linear MF paradigms. Coinciding with the rapid advancements in deep learning, a lot of deep architecture-based models have found their application in recommender systems. This includes but isn't limited to multi-layer perceptrons (MLP) [27, 28], convolutional neural networks (CNN) [29, 30], recurrent neural networks (RNN) [31, 32], adversarial networks (AN) [33], autoencoders (AE) [34–37] among other evolving architectures.

Another way to mitigate data sparsity involves integrating auxiliary information related to users and/or items. Strub et al. proposed a collaborative filtering network model that combines additional information into an autoencoder for rating matrix completion [38]. To utilize the additional information flexibly, Zhang et al. innovated an autoencoder structure, subsequently unveiling a semi-autoencoder model tailored for both rating prediction and personalized top-n recommendations [14].

In addition, cross-domain recommendation presents another avenue for the inclusion of auxiliary user and item information [39]. Pan et al. utilized the user embedding learned in the source domain to initialize the user embedding in the target domain and restrict them to being closed [40]. Man et al. proposed an Embedding and Mapping model (EMCDR) for cross-domain recommendation to use auxiliary information [17]. The introduction of external additional information is also one of the important means to solve the data sparsity.

Despite the contributions of the aforementioned methods, they exhibit shortcomings in their initialization processes, subsequently curbing their model performance. Furthermore, these techniques, when introducing auxiliary information, often lack versatility and efficiency. For instance, works [14, 38] are tailored specifically for item-side information, while works [17, 39, 40] emphasize item rating data from other domains, suggesting a lack of versatility in their approaches. In contrast, this approach diverges significantly. This work employs a semi-autoencoder model to initialize the latent feature representation of items derived from the user-item rating matrix, amalgamating this into this model to facilitate advanced latent feature learning. Additionally, inspired by literature [41], this study discerned that by utilizing subspace projection distances, this work can seamlessly incorporate diverse auxiliary information, achieving enhanced outcomes through merely elementary learning from this auxiliary information. Specifically, this study transforms item attribute data or denser item rating data from other fields into a low-rank latent feature representation. This is then approximated using subspace projection distance against the target task's latent feature representation, enabling the target task to effectively learn from the auxiliary information.

FSTDMF model

Problem setting

Suppose there is a partial observed rating matrix $R \in R^{m \times n}$ representing ratings from m users for n items. Additionally, there is a binary matrix $\psi \in R^{m \times n}$, where the entry ψ_{ij} is equal to 1 if R_{ij} is observed and 0 otherwise and auxiliary information about the items is available. This auxiliary information might derive from item attributes, item ratings in other relevant domains, and so forth. Given this item auxiliary information, the aim is to ascertain its latent feature representation $V_A \in R^{n \times d}$, which behaves as a partial orthonormal matrix, i.e., $V_A^T V_A = E$. This paper presents two methodologies to derive V_A , the specific procedures are detailed in "Learning latent representations from item auxiliary information" section. Then according to R , V_A , the objective is to secure an effective latent feature representation $V \in R^{n \times r}$ that allows us to reconstruct the missing entries in R subsequently enabling the prediction of all user ratings for every item.

FSTDMF framework

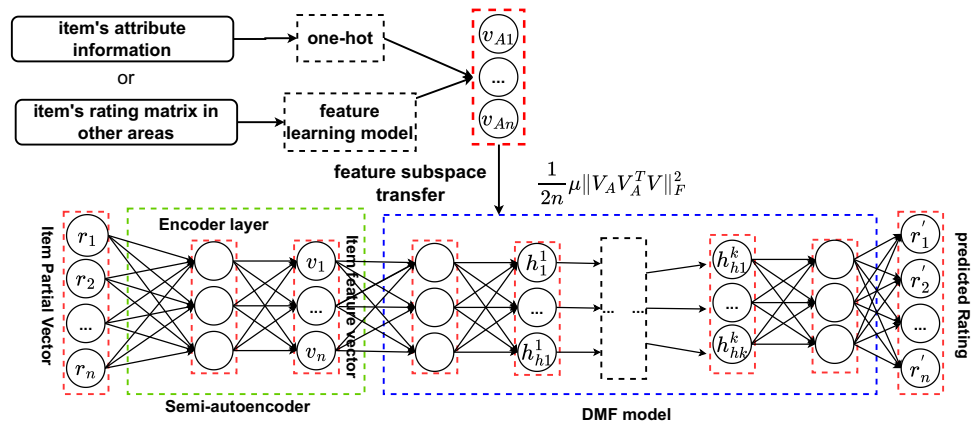
The proposed FSTDMF framework is depicted in Fig. 1. First, this model takes the scoring matrix as input of a semi-autoencoder model to initialize the latent feature representation V of the item. In addition, this study employs two distinct methods (onehot coding, and a semi-autoencoder model) to extract the latent feature representation V_A from the item's auxiliary information A (comprising both item attribute details and item rating matrices from other domains). Subsequently, the initial feature matrix V of the item is obtained by semi-autoencoder as the input of the DMF model. and the regular term (10) is added to the objective function of the model to transfers the features from V_A to V in pursuit of a superior V which in turn facilitates improved user-item rating predictions. In the following subsections, each component will be discussed in detail.

DMF model combined semi-autoencoder

Matrix completion is one matrix factorization method, the low-rank assumption of conventional matrix completion methods indicates that R is from a linear latent variable model, i.e.,

$$R = UV^T \quad (1)$$

Fig. 1 The framework of FSTD MF model



where $U \in R^{m \times r}$ denotes the latent feature representation of users. The target function can be expressed as:

$$\min_{U, V} \left\| \psi \odot (R - UV^T) \right\|_F^2 + \lambda \|V\|_F^2 + \beta \|U\|_F^2 \quad (2)$$

In Eq. (2), λ and β are regularization parameter, \odot denotes the Hadamard product. By solving Eq. (2), the missing entries can be recovered.

Fan et al. [11] proposed to use multi-layer neural networks to learn the latent feature representations of users or items to approximate the nonlinear latent variable model instead of Eq. (2):

$$\min_{V, W, b} \frac{1}{2n} \sum_{i=1}^n \left\| \psi_i \odot (r_i - g(W\hat{v}_i + b)) \right\|^2 + \frac{1}{2n} \lambda \|V\|_F^2 + \frac{1}{2} \beta \|W\|_F^2 \quad (3)$$

where $W \in R^{m \times r}$ denotes the weight matrix, $b \in R^m$ denotes the bias vector, ψ_i is the i th column of ψ , \hat{v}_i is i th column of V^T , r_i is the i th column of R , and $g(\cdot)$ denotes the activation function such as sigmoid function or hyperbolic tangent function.

It is well known that multi-layer neural networks are often more effective than single-layer neural networks in approximating nonlinear functions. Therefore, the final target function (3) can be further approximated as:

$$\min_{V, W, b} \frac{1}{2n} \sum_{i=1}^n \left\| \psi_i \odot (r_i - g^{(K+1)}(g^{(K)}(\dots g^{(1)}(\hat{v}_i, \Theta^{(1)}), \dots, \Theta^{(K)}), \Theta^{(K+1)})) \right\|^2 + \frac{1}{2n} \lambda \|V\|_F^2 + \frac{1}{2} \beta \sum_{j=1}^{K+1} \|W^{(j)}\|_F^2 \quad (4)$$

where $g^{(j)} = (W^{(j)}t + b^{(j)})$, $\Theta^{(j)} = \{W^{(j)}, b^{(j)}\}$. $j = 1, 2, K + 1$, and K is the number of hidden layers. By solving Eq. (4), V can be obtained and then the missing entries of R can be recovered.

While the initialization of the latent feature representation V is constructed using a Gaussian distribution, this can restrict the model's learning capability, making it challenging to achieve a better latent feature representation. To solve this problem, this work combines the semi-autoencoder with DMF to improve the model learning capability.

Semi-autoencoder is an unsupervised model that attempts to learn a compact representation of the input in the output layer [14]. The semi-autoencoder is generally composed of a three-layer network, it does not require the dimensions of the input layer and the output layer to be equal, which is shown in Fig. 2. Given the input $\{x_1, x_2, x_3, \dots, x_n\}$, where, $x_i \in R^m$, the semi-autoencoder tries to learn a function $f_{Q,p}(x) \approx x$. The encoder and decoder of a semi-autoencoder are expressed as follows:

$$\xi_i = f(Qx_i + p) \quad (5)$$

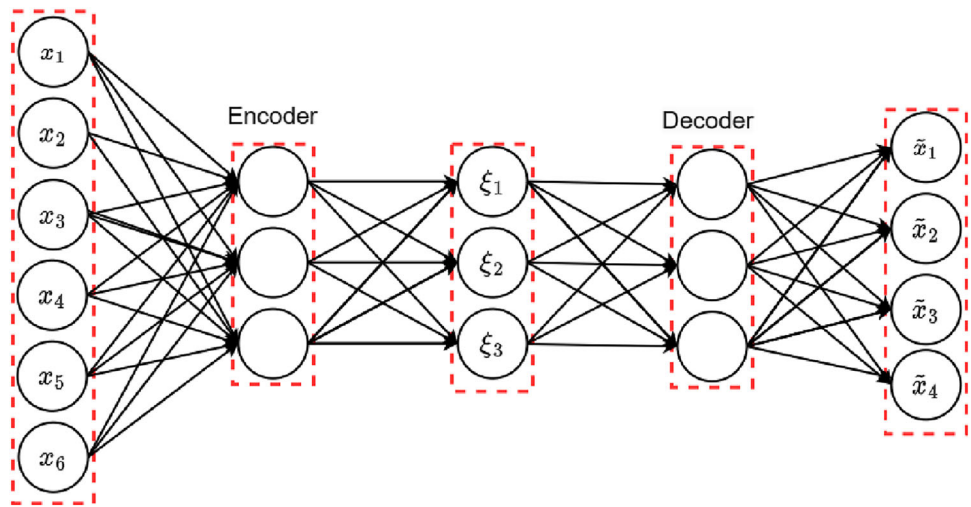
$$\tilde{x}_i = g(\tilde{Q}\xi_i + \tilde{p}) \quad (6)$$

where $Q \in R^{r \times m}$ and $\tilde{Q} \in R^{m \times r}$ are the weight matrixes, $p \in R^r$ and $\tilde{p} \in R^m$ are the bias vectors, $\xi_i \in R^r$ is the output of encoder, \tilde{x}_i is the result of x_i refactoring, f and g are nonlinear activation functions. The aim of the semi-autoencoder model is to minimize the reconstruction error by learning the parameters Q , p , \tilde{Q} , and \tilde{p} as Eq. (7):

$$\min_{Q, \tilde{Q}, p, \tilde{p}} \sum_{i=1}^n \|x_i - \tilde{x}_i\|^2 = \min_{Q, \tilde{Q}, p, \tilde{p}} \sum_{i=1}^n \left\| x_i - g(\tilde{Q}(f(Qx_i + p)) + \tilde{p}) \right\|^2 \quad (7)$$

Semi-autoencoders have found extensive use in recommendation systems with promising results [25]. For instance,

Fig. 2 The framework of semi-autoencoder model



Dong et al. [9] employs a semi-autoencoder to co-embed the attributes and the graph features of the items for rating prediction. Geng et al. [18] extract auxiliary information from DBpedia and leverage the LSI model to learn hidden relations on top of item features, and subsequently combine them with the original rating matrix and side information, fed into a semi-autoencoder for recommendation prediction.

In this paper, the semi-autoencoder is solely employed to derive the latent feature representation V and not for output prediction. The sigmoid function is applied as the encoding activation function. This process can be expressed as:

$$\hat{v}_i = f(Qr_i + p) \tag{8}$$

By integrating it into the DMF model, a better initialization of latent feature representation V is achieved. Finally, the prediction results are obtained by solving Eq. (4).

However, solely relying on rating data limits the model’s predictive efficacy, particularly when the rating matrix is sparse. To mitigate this issue, the introduction of a subspace projection distance has been proposed to incorporate auxiliary information, thereby enhancing predictive accuracy. The details will be described in “DMF model via feature subspace transfer” section.

DMF model via feature subspace transfer

Assuming the latent feature representation V_A is obtained from the auxiliary information, the latent feature can be transferred to the target domain. In general, a regularization term $Sim(V, V_A)$ can be added to the optimization model to transfer the auxiliary features V_A . The regularization term $Sim(V, V_A)$ is a measurement to represent the similarity between V and V_A .

This paper uses the subspace projection distance to measure the similarity $Sim(V, V_A)$. Denote the i th column of

V by v_i . The projection distance from v_i to the subspace spanned by V_A can be defined by [41]:

$$dist(v_i, V_A) = \|v_i - V_A V_A^T v_i\|_F \tag{9}$$

where V_A is a partial orthonormal matrix. Hence, the similarity $Sim(V, V_A)$ can be defined by:

$$\begin{aligned} Sim(V, V_A) &= \sum_{i=1}^n dist^2(v_i, V_A) = \sum_{i=1}^n \|v_i - V_A V_A^T v_i\|_F^2 \\ &= \|V - V_A V_A^T V\|_F^2 = \|V\|_F^2 - \|V_A V_A^T V\|_F^2 \end{aligned} \tag{10}$$

Adding the similarity $Sim(V, V_A)$ as a penalty term to the model (4), the target function can be formed as follows:

$$\begin{aligned} \min_{V, W, b} \frac{1}{2n} \sum_{i=1}^n \|\psi_i \odot (r_i - g^{(k+1)}(g^k(\dots g^{(1)}(\hat{v}_i \\ p), \Theta^{(1)}), \dots, \Theta^{(k)}), \Theta^{(k+1)}))\|^2 + \frac{1}{2n} \lambda \|V\|_F^2 \\ + \frac{1}{2} \beta \sum_{j=1}^{k+1} \|W^{(j)}\|_F^2 + \frac{1}{2n} \mu \|V\|_F^2 - \frac{1}{2n} \mu \|V_A V_A^T V\|_F^2 \end{aligned} \tag{11}$$

Here, μ is regularization parameter, the model (11) can be further simplified as:

$$\begin{aligned} \min_{V, W, b} \frac{1}{2n} \sum_{i=1}^n \|\psi_i \odot (r_i - g^{(k+1)}(g^k(\dots g^{(1)}(\hat{v}_i \\ + p), \Theta^{(1)}), \dots, \Theta^{(k)}), \Theta^{(k+1)}))\|^2 + \frac{1}{2n} \alpha \|V\|_F^2 \\ + \frac{1}{2} \beta \sum_{j=1}^{k+1} \|W^{(j)}\|_F^2 - \frac{1}{2n} \alpha \eta \|V_A V_A^T V\|_F^2 \end{aligned} \tag{12}$$

with $\alpha = \lambda + \mu$, $\eta = \mu/\alpha$. The confidence parameter $\eta \in [0, 1]$ represents the relatedness between the auxiliary information and target data. A large η means that the auxiliary and target domain share a similar feature structure. If $\eta = 0$, the model reduces to Eq. (4), which means that the auxiliary data can not improve the prediction performance of the target domain.

The optimization problem of FSTDMF in Eq. (12) is non-convex. Fan et al. [11] propose to find the local minima by nonlinear optimization techniques such as BFGS [42] and improved resilient backpropagation (iRprop+) algorithm [43]. In [44], it was shown that iRprop+ often outperformed other methods in optimizing neural networks and iRprop+ is more efficient than BFGS for large-scale problems. This paper follows Fan et al. [11] to use BFGS to solve DMF when the size of R is relatively small (e.g., $m, n < 1000$) and use iRprop+ otherwise.

Learning latent representations from item auxiliary information

As previously mentioned, the proposed model processes the provided item auxiliary information A to obtain its latent feature representation V_A . For this purpose, two distinct methodologies are employed.

Items' auxiliary information typically falls into two categories: item side information and the rating matrices that grade items in various domains. While the item side information encompasses a wide range, including attributes, images, comments, videos, etc., obtaining certain types like images, videos, and comments can be challenging. However, attribute information and rating matrices from different domains are more readily accessible. This work leverages both the item attribute information and the domain-specific rating matrices to extract the latent feature representations.

More precisely, each item comes with its attribute data, such as the year of production and genres. Given that there are multiple categories within each attribute, this work adopts one-hot encoding to handle the attribute data of items. For instance, considering the MovieLens dataset, each span of 7 years is treated as a distinct code for the attribute 'year', while every genre is a separate code for the attribute 'genres'. After encoding, the data is represented as $C \in R^{n \times d}$. Since it is not orthogonal, the next step is to orthogonalize it. Subsequently, the latent feature representation V_A of the items' auxiliary information is obtained. The process is illustrated in Fig. 3.

In contrast to deriving latent features from item attributes, when it comes to the rating matrices of how items are graded in other domains, a feature learning model is used. This approach is reminiscent of cross-domain recommendation methods [39]. Dong et al. [9] employs a semi-autoencoder to co-embed item attributes with their graph features for more

accurate rating predictions. It connects items rating R , items attributes A , and graph structures G of items, feeding them into the semi-autoencoder to learn the hidden representations of the item:

$$\epsilon = f(Q\text{con}(R, A, G) + p) \quad (13)$$

Here, $\epsilon \in R^{n \times d}$ represents the latent feature representations to be learned. In this context, since A and G are redundant, Eq. (13) simplifies to Eq. (14):

$$\epsilon = f(QR + p) \quad (14)$$

This bears similarity to a standard semi-autoencoder, so, by solving Eq. (7), the latent features are obtained from the rating matrices that indicate how items are graded in external domains. Following the regularization process, the latent feature representation V_A is derived. The process is depicted in Fig. 4.

Experiments

Datasets

This work conducts experiments on four real-world datasets to assess the effectiveness of the proposed FSTDMF model.

MovieLens100K¹: Recognized as a standard benchmark dataset, MovieLens100K is frequently utilized to gauge the accuracy of rating prediction algorithms. This dataset comprises 100,000 ratings (ranging from 1 to 5) provided by 943 users for 1682 items, with both users and items having at least 20 interactions. Additionally, it includes attributes for users and items, with item attributes covering genre and release year.

MovieLens1M²: Serving as an expanded version of the MovieLens100K, the MovieLens1M dataset is also a popular choice for assessing recommendation systems. It contains 1,000,209 ratings (from 1 to 5) given by 6040 users across 3952 items.

DoubanMovie [45]: This dataset revolves around well-structured movie ratings on Douban. It comprises 1,287,869 ratings (from 1 to 5) offered by 2712 users for 34,893 items, where both users and items have a minimum of 5 interactions.

MovieLens20M [46]: Part of the MovieLens family, the MovieLens20M dataset has 1,462,905 ratings (from 0 to 5) provided by 10,000 users for 9395 items. Both users and items in this dataset have at least 5 interactions.

In the case of only containing item's attribute information, this study conduct experiments on the MovieLens100k and

¹ <http://files.grouplens.org/datasets/movielens/ml-100k.zip>.

² <http://files.grouplens.org/datasets/movielens/ml-1m.zip>.

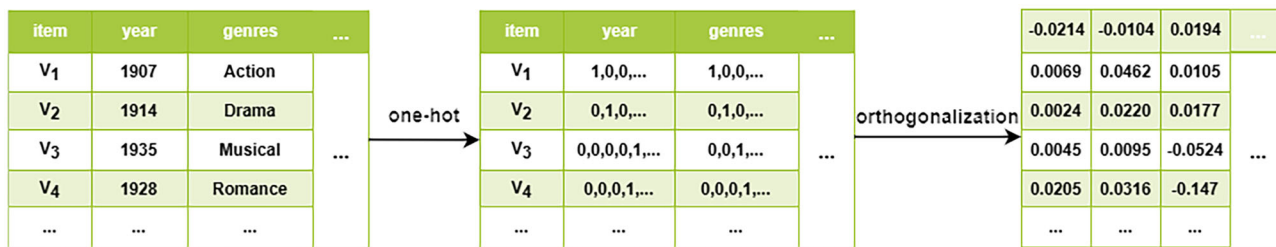


Fig. 3 The process of learning latent feature representations form item attribute information

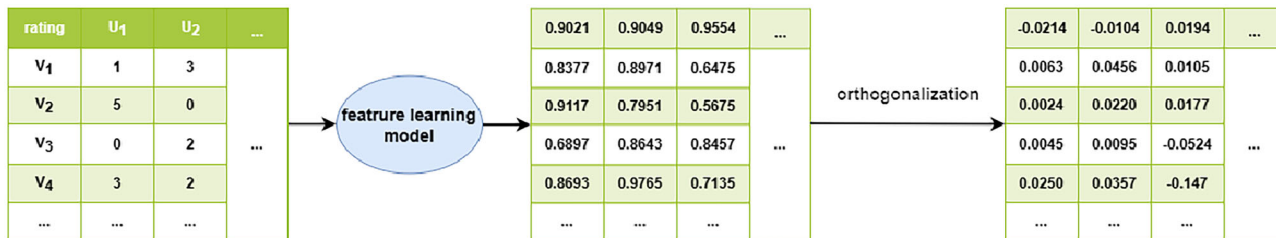


Fig. 4 The process of learning latent feature representations form item attribute information

Movielens1M data set to evaluate the performance of the proposed FSTD MF.

Conversely, in scenarios exclusively involving the item rating matrix from other domains, this work conducts experiments on the Movielens100k, Movielens1M, DoubanMovie, and Movielens20M data set. This paper designated two distinct tasks for these datasets:

Task 1: Utilizing MovieLens1M as the source domain and MovieLens100K as the target domain.

Task 2: Employing DoubanMovie as the source and MovieLens20M as the target domain.

For both tasks, this work only performs latent representation learning of auxiliary information for items that exist in both domains simultaneously. The details of these two data sets are summarized in Table 1.

Compared methods

This paper compares proposed FSTD MF with the following baseline methods:

- PMF [47]: PMF models the latent attributes of users and items using a probabilistic linear model with Gaussian observation noise. It is derived from a probabilistic version of SVD factorization.
- DMF [11]: DMF is on the basis of a nonlinear latent variable model. It is formulated as a deep-structure neural network, in which the inputs are the low-dimensional unknown latent variables and the outputs are the partially observed variables.

- NCF [27]: NCF leverages only implicit feedback. It begins with randomly initialized user and item representations and subsequently employs a multi-layer perceptron to learn the user-item interaction function.
- I-AUTO [8]: The Item-based AutoRec (I-AUTO) model leverages an autoencoder to discern effective item feature representations for recommendation.
- GraphRec [10]: GraphRec co-embeds users and items in a unified latent space. It harnesses the user-item interaction graph’s Laplacian to glean graph-based attribute features. Remarkably, it relies solely on the rating matrix, and all attribute information is extracted from the structure of the graph.
- HCRSA [14]: HCRSA breaks the restriction that the dimensions of the autoencoder’s input and output layers be equal and introduces auxiliary information for representation learning.
- MFSAE [18]: MFSAE extracts auxiliary information from DBpedia and leverages the LSI model to learn hidden relations on top of item features. Finally, combined with the original rating matrix and side information, are fed into a semi-autoencoder for recommendation prediction.
- PRKG [19]. PRKG draws side information from DBpedia and encodes it into a low-dimensional representation using an autoencoder. It then introduces a semi-autoencoder to blend this side information into the recommendation process.
- Item-Agrec [9]: Item-Agrec uses a semi-autoencoder to jointly embed the attributes and graph features of the items for rating prediction.

Table 1 Statistics of two tasks in the experiment of transferring the rating matrix of items in other areas

Task	Domain	Dataset	User	Items	Interaction	Density (%)	Rating
Task1	Source	Movielens1M	6040	1594	481,829	5	1–5
	Target	Movielens100k	943	1682	100,000	6.30	1–5
Task2	Source	DoubanMovie	2439	3040	147,398	1.99	1–5
	Target	Movielens20M	9995	3040	543,550	1.79	0–5

- CMF [48]: CMF offers a joint matrix factorization technique, where entities' latent factors are cohesively shared across both source and target domains.
- EMCDR [17]: EMCDR adopts Matrix Factorization (MF) to learn embedding first and then utilize a network to bridge the user embeddings from the auxiliary domain to the target domain.
- PTUPCDR [49]: PTUPCDR learned a meta-network fed with users' characteristic embeddings to generate personalized bridge functions to achieve personalized transfer of preferences for each user.

Here, PMF belongs to the ordinary matrix factorization and only the rating matrix is used. DMF, NCF, and I-AUTO belong to the deep matrix factorization and only the rating matrix is used. GraphRec, HCRSA, MFSAE, PRKG, and Item-Agrec belong to the deep matrix factorization and the rating matrix and item attribute information are used, for GraphRec, this paper uses GraphRec1 to denote GraphRec with item attribute information, GraphRec2 denotes GraphRec with item attribute information and graph. CMF, EMCDR, and PTUPCDR belong to the cross-domain recommendation.

Parameter settings and evaluation metrics

In the experiments, this study sets the same parameter values of the compared methods as those in the cited paper if the data sets are the same as those used in the proposed method. If the compared methods are not used for the data sets used in this paper, the parameter values for optimal results are selected through repeated experiments. For the proposed model, the parameter values for optimal results are selected through repeated experiments. In addition, the autoencoder includes an input layer and hidden layer, DMF model includes an input layer, hidden layer, and output layer. Since we want to use the output of the autoencoder hidden layer as the input to the DMF model, the output dimension of the autoencoder hidden layer is the same size as the input dimension of the DMF model. The dimension size of the hidden layer of the DMF model is selected in the range that it is larger than the feature dimension of the project and smaller than the number of users. The parameter values for the compared methods and proposed method are presented in Table 2.

This paper uses Root Mean Square Error (RMSE) and Mean Absolute Error (MAE), which are commonly used in the recommendation, to evaluate the experimental effects of the model mentioned. Their expressions are as follows:

$$\text{MAE} = \frac{\sum_{(i,j) \in \omega_{\text{test}}} |R_{ij} - \tilde{R}_{ij}|}{|\omega_{\text{test}}|} \quad (15)$$

$$\text{RMSE} = \sqrt{\frac{\sum_{(i,j) \in \omega_{\text{test}}} (R_{ij} - \tilde{R}_{ij})^2}{|\omega_{\text{test}}|}} \quad (16)$$

Here ω_{test} denote the indices of test entries, \tilde{R}_{ij} denote the predicted result. Obviously, the more accurate the prediction, the lower value of MAE and RMSE.

Experimental results

This work conducted experiments on four datasets: Movielens100k, Movielens1M, Movielens1M-Movielens100K, and DoubanMovie-Movielens20M. The aim was to highlight the enhancements of the proposed model when introducing both item attribute information and the item's rating matrix from other domains. For the rating matrix in other domains, this work allocated 90% for training and the remaining for testing to obtain V_A . For the target data set, this work experimented with 70%, 50%, and 30% as training data, using the remainder for testing. The average results of all models with five repeated experiments on the data sets are reported in Tables 3, 4, 5, and 6.

From the results, it's evident that PMF consistently lags behind other methods across all datasets. This performance gap can be attributed to the datasets' non-linear structure, which contrasts with PMF's linear modeling approach. In contrast, deep matrix factorization, being non-linear, is aptly suited for handling such data.

As it can be seen in Table 3, GraphRec, HCRSA, MFSAE, PRKG, and FSTDMF outperform DMF, NCF, and I-AUTO on the Movielens100K data set. This demonstrates the usefulness of transferring knowledge from the item attribute information. In addition, GraphRec2 achieves the best result except for FSTDMF, which shows the effectiveness of using the Laplacian function of the user-item interaction graph to gather graph-based attribute features. Notice, when the train-

Table 2 Parameter settings for each algorithm

Model	λ : learning rate	α : regularization	r : feature dimension	β : regularization	batch_size
PMF	{0.001,0.005,0.01,0.05,0.1}	{0.001,0.005,0.01,0.05,0.1}	10	Not required	Not required
DMF	0.0123	{0.001,0.005,0.01,0.1}	{10,100,300}	{0.001,0.005,0.01,0.1}	Not required
NCF	{0.0001,0.0005,0.001,0.005}	Not required	{8,16,32,64}	Not required	[128,256,512,1024]
I-AUTO	{0.001,0.005,0.01,0.05,0.1}	{0.001,0.01,0.1,1,100,1000}	{10,20,40,80,100,200,300,400,500}	Not required	Not required
GraphRec1	0.0002	{0.08,0.05}	{50,40}	{0.06,0.05}	Not required
GraphRec2	{0.00003,0.000005}	0.05	{10,50}	{0.02,0.06}	Not required
HCRSA	0.001	0.1	500	Not required	Not required
MFSAE	0.001	0.005	200	Not required	Not required
PRKG	0.001	0.1	300	Not required	Not required
Item-Agrec	{0.000001,0.00001,0.0001,0.001,0.01}	{0.001,0.005,0.01,0.05,0.1,0.5,1,1,5,10}	{32,64,128,256,512,1024}	Not required	Not required
CMF	{0.001, 0.005, 0.01, 0.02, 0.1}	Not required	{5,10,20}	Not required	{32,64,128,256,512}
EMCDR	{0.001, 0.005, 0.01, 0.02, 0.1}	Not required	{5,10,20}	Not required	{32,64,128,256,512}
PTUPCDR	{0.001, 0.005, 0.01, 0.02, 0.1}	Not required	{5,10,20}	Not required	{32,64,128,256,512}
FSTDMF	{0.0123,0.00123}	{0.1,0.15,0.2,0.25,0.3,0.35,0.4}	{100,300,500}	{0.1,0.15,0.2,0.25,0.3,0.35,0.4}	Not required

ing set is 30%, I-AUTO falls behind DMF and NCF. This is because that I-AUTO is based on the autoencoder model, and the performance of the model is limited when dealing with a sparse training set. Compared to DMF, the MAE and RMSE results of FSTDMF were improved by 3.8% and 3.2%, 4.2% and 3.7%, 3% and 2.8% in different training set and test set partition cases, respectively. And FSTDMF achieved the best results in all different cases. Notably, FSTDMF can achieve better improvements than GraphRec2 when the training set is 50% and 30%. This is because that when the training set is more sparse, FSTDMF can still effectively utilize the item auxiliary information.

In Table 4, it can be seen that for the Movielens1M data set, the methods GraphRec, HCRSA, MFSAE, and FSTDMF outperform DMF, NCF, and I-AUTO. This indicates that incorporating attribute information can still enhance recommendation performance on a large and sparse data set. In addition, GraphRec still achieves the best results with the exception of FSTDMF. This demonstrates the effectiveness of leveraging the semi-autoencoder to combine item auxiliary information. However, it is worth noting that the performance of GraphRec2 is similar to that of GraphRec1. This suggests that utilizing the Laplacian function of the user term interaction graph to gather graph-based attribute features does not significantly improve the model’s performance in this data. Notice that FSTDMF can achieve the best result in all cases. Moreover, compared to DMF, the MAE and RMSE results of FSTDMF were improved by 3.3% and 3.2% with the training set is 30%. This indicates that FSTDMF can still effectively utilize the item auxiliary information when the data set is large and sparse.

In Table 5, it is evident that CMF, EMCDR, PTUPCDR, and FSTDMF clearly outperform DMF, PMF, and NCF on the Movielens1M-Movielens100K data set. These results demonstrate the usefulness of transferring knowledge from the rating matrix on how the item is graded in other areas. On the other hand, PTUPCDR outperformed CMF, which illustrates that the efficiency of transfer learning using a personalized bridge generated through a meta-network. Notably, CMF is based on a transfer learning version of the MF. However, a notable standout is that Item-Agrec clearly outperforms CMF, EMCDR, and PTUPCDR. This can be attributed to Item-Agrec’s advanced neural network model and the introduction of auxiliary item information. FSTDMF achieved the best results in all cases. In a comprehensive review of Table 5, the MAE and RMSE results of FSTDMF were increased by 3.7% and 3.3%, 4.4% and 3.9%, 3.9% and 3.6% compared with DMF respectively. This indicates that FSTDMF can effectively extract and transfer the features of items from the auxiliary domain to the target domain, regardless of the sparsity of the training set.

As it can be seen in Table 6, CMF, EMCDR, and PTUPCDR are clearly inferior to other methods. This may

Table 3 Performance of various algorithms on the Movielens100k data set

Training sets metrics	70%		50%		30%	
	MAE	RMSE	MAE	RMSE	MAE	RMSE
PMF	0.748	0.957	0.772	0.989	0.813	1.04
DMF	0.729	0.926	0.744	0.944	0.767	0.972
NCF	0.741	0.941	0.746	0.946	0.767	0.972
I-AUTO	0.718	0.907	0.726	0.934	0.770	1.069
GraphRec1	0.709	0.906	0.726	0.921	0.753	0.955
GraphRec2	0.703	0.899	0.721	0.918	0.746	0.948
HCRSA	0.713	0.907	0.727	0.927	0.753	0.951
MFSAE	0.706	0.899	0.723	0.917	0.765	0.965
PRKG	0.714	0.913	0.729	0.928	0.772	0.970
FSTDMF	0.701	0.893	0.713	0.909	0.744	0.945

The best results are highlighted in bold

Table 4 Performance of various algorithms on the Movielens1M data set

Training sets metrics	70%		50%		30%	
	MAE	RMSE	MAE	RMSE	MAE	RMSE
PMF	0.694	0.884	0.709	0.906	0.740	0.943
DMF	0.693	0.854	0.700	0.871	0.720	0.914
NCF	0.681	0.866	0.692	0.880	0.707	0.898
I-AUTO	0.714	0.896	0.722	0.912	0.735	0.928
GraphRec1	0.673	0.858	0.684	0.872	0.703	0.888
GraphRec2	0.672	0.857	0.683	0.871	0.704	0.889
HCRSA	0.674	0.859	0.687	0.877	0.705	0.897
MFSAE	0.672	0.857	0.684	0.872	0.705	0.890
PRKG	0.690	0.881	0.705	0.898	0.720	0.918
FSTDMF	0.667	0.850	0.678	0.863	0.696	0.885

The best results are highlighted in bold

be due to the different rating scales of the DoubanMovie and Movielens20M data set, which may lead to a bias between the features extracted from the auxiliary and target domain by these approaches. In this case, these methods can not achieve better results. In a comprehensive review of Table 6, FSTDMF achieves relatively good experimental results in most cases. It is worth noting that, when the training set is 30%, the MAE and RMSE results of FSTDMF were improved by 6.5% and 6.5% over DMF. This proves that FSTDMF can efficiently extract the item's features from the auxiliary rating matrix in different rating scales. The subspace projection distance measurement between the auxiliary and target domain can eliminate the influence of different rating scales.

In addition, 10 repeated experiments were conducted on the Movielens100k data sets to verify the validity of the FSTMF model by double-tail t test with a significance level was 5%. This paper presents two hypotheses:

H_0 : the performance difference between FSTDMF and the comparison method is not significant.

H_1 : the performance difference between FSTDMF and the comparison method is significant.

When H_0 is rejected and H_1 is accepted, the advantages and disadvantages of the method are judged by comparing the mean.

The experimental results are shown in Table 7, where '+' indicates that the MAE value obtained by FSTDMF is superior to the statistical significance distinction results of other methods in the double-tail t test. As can be seen from Table 7, FSTDMF were significantly different ($p < 0.05$) from other methods in three different training set test set partition cases. Therefore, the H_0 is rejected and H_1 is accepted, which shows that FSTDMF presents significant changes compared with other methods. It can be seen that FSTDMF achieves the best mean in all cases, which proves that FSTDMF is statistically significantly better than the contenders. Notice that the experimental phenomenon on Movielens1M and DoubanMovie-Movielens20M datasets are the same as those on Movielens100k. For brevity, we do not give the result of these two data sets.

Table 5 Performance of various algorithms on the MovieLens1M-MovieLens100K data set

Training sets metrics	70%		50%		30%	
	MAE	RMSE	MAE	RMSE	MAE	RMSE
PMF	0.748	0.957	0.772	0.989	0.813	1.04
DMF	0.729	0.926	0.744	0.944	0.767	0.972
NCF	0.741	0.941	0.746	0.946	0.761	0.964
Item-Agrec	0.706	0.893	0.722	0.915	0.753	0.950
CMF	0.728	0.935	0.738	0.944	0.751	0.963
EMCDR	0.740	0.934	0.750	0.950	0.765	0.965
PTUPCDR	0.723	0.925	0.732	0.938	0.746	0.956
FSTDMF	0.702	0.895	0.711	0.907	0.737	0.937

The best results are highlighted in bold

Table 6 Performance of various algorithms on the DoubanMovie-MovieLens20M data set

Training sets metrics	70%		50%		30%	
	MAE	RMSE	MAE	RMSE	MAE	RMSE
PMF	0.670	0.872	0.681	0.889	0.710	0.933
DMF	0.676	0.883	0.700	0.914	0.749	0.985
NCF	0.663	0.860	0.672	0.875	0.709	0.926
Item-Agrec	0.655	0.847	0.669	0.871	0.701	0.919
CMF	0.930	1.169	0.967	1.209	0.966	1.214
EMCDR	0.833	1.055	0.909	1.136	0.913	1.143
PTUPCDR	0.833	1.060	0.864	1.102	0.911	1.162
FSTDMF	0.650	0.846	0.667	0.871	0.700	0.921

The best results are highlighted in bold

Table 7 Statistical analysis results of various algorithms on the MovieLens100k dataset

Training sets	70%			50%			30%		
	Mean	<i>t</i> test		Mean	<i>t</i> test		Mean	<i>t</i> test	
FSTDMF	0.700			0.713			0.744		
PMF	0.754	5.13E−19	+	0.773	3.53E−24	+	0.813	2.79E−24	+
DMF	0.726	2.07E−09	+	0.744	1.54E−28	+	0.769	2.71E−33	+
NCF	0.730	2.54E−20	+	0.743	1.68E−13	+	0.760	1.19E−15	+
I-AUTO	0.715	1.87E−08	+	0.738	9.65E−17	+	0.775	9.61E−19	+
GraphRec1	0.715	2.08E−11	+	0.730	1.21E−14	+	0.756	1.5E−09	+
GraphRec2	0.703	0.0035	+	0.718	5.71E−08	+	0.748	2.73E−6	+
HCRSA	0.712	1E−8	+	0.727	5.03E−10	+	0.749	1.14E−08	+
MFSAE	0.706	1.3E−6	+	0.724	1.06E−09	+	0.764	2.43E−16	+
PRKG	0.710	1.29E−09	+	0.728	5.13E−12	+	0.772	6.68E−19	+

The best results are highlighted in bold

Parameter sensitivity

To show the effect of different parameters on the prediction precision, this study conducts experiments on several data sets. The specific experimental results will be presented in the next.

trade-off parameter η

The trade-off parameter η is important for the FSTDMF model since it determines how large the impact of the auxil-

iary data will be on the target data. To show the effect of η on the prediction precision, this work conducts experiments on the MovieLens100k data set and the MovieLens1M-MovieLens100K data set with different η . For both sets, this work chose 70%, 50%, and 30% as the training set, and the rest as the test set. The prediction performance is shown in Fig. 5.

As can be seen, The value of MAE and RMSE decreases monotonically as the value of η increases monotonically on both data sets. And the FSTDMF model achieves the best result with $\eta = 1$. This is because the target and auxiliary

domains are quite related in the experiments. The item knowledge learned from the auxiliary data can make a positive impact on the target data with large tradeoff parameters. Note that when η decreases to 0, the feature subspace approach model reduces to the DMF model with semi-autoencoder (4). Hence the prediction precision of FSTDMF decreases greatly when η is close to 0.

Number of item feature dimension

As different models have different sensitivities to the latent feature representations, it is curious to see how many latent feature representation dimensions are beneficial to the recommendation task. Towards this end, this work conducts experiments on the Movielens1M and DoubanMovie-Movielens20M data sets with different latent feature representation dimensions. The prediction performance is shown in Fig. 6.

As can be seen from Fig. 6a–c, on the Movielens1M data set, when the training set is 70%, the MAE of PMF also increases faster than that of other methods with the increase of dimension. This explains the high sensitivity of PMF to dimension and the low sensitivity of deep matrix factorization to dimension. Compared with NCF and I-AUTO, DMF shows more stable changes under the change of dimensions, and the effect of DMF is almost better, which shows the superiority of the DMF model. For GraphRec, HCRSA, MFSAE, PRKG, and FSTDMF, their change trends are relatively stable, which indicates that the introduction of auxiliary information can reduce the sensitivity of the model to dimensions. When the training set was 50% and 30%, MAE of PMF, DMF, NCF, and I-AUTO showed different changing trends with the increase of dimension, respectively. This shows that the size of the training set affects the sensitivity of the model to dimensionality when no auxiliary information is introduced. For GraphRec, HCRSA, MFSAE, PRKG, and FSTDMF, the variation trend of MAE is the same as when the training set is 70%, which indicates that the introduction of auxiliary information can reduce the influence of training set size on dimensional sensitivity. Overall, FSTDMF is almost always the best in different dimensions for the three different training set sizes, which illustrates its effectiveness. In addition, FSTDMF shows relatively stable results under different dimensions and different training set sizes, which indicates that FSTDMF is almost unaffected by training set size and dimension size, and has stronger stability.

In Fig. 6d–f, it can be observed that on the DoubanMovie-Movielens20M dataset, PMF once again exhibits higher sensitivity to dimensionality and sparsity compared to DMF. Similarly, NCF is highly sensitive to both dimensionality and training set size. On the other hand, Item-Agrec shows lower sensitivity to dimensionality and training set size due to the incorporation of autoencoders and auxiliary informa-

tion. It is important to note that the experimental results for CMF, EMCDR, and PTUPCDR are not provided here due to their poor MAE performance. For CMF, when the training set size is 70%, the MAE results are 0.942, 1.261, 1.928, 2.919, 3.471, and 4.2. For EMCDR, the MAE result is 0.862. For PTUPCDR, the MAE results are 0.929, 1.036, 1.136, 1.324, 1.576, and 5.307. These methods exhibit high sensitivity to dimensionality, which is consistent across training set sizes of 50% and 30%. Overall, FSTDMF continues to perform exceptionally well across different dimensions on the DoubanMovie-Movielens20M dataset. Additionally, FSTDMF demonstrates relatively stable results regardless of dimensionality or training set size, indicating greater efficiency and stability compared to the other methods.

Number of layers

The number of layers of neural networks is one of the key factors affecting the performance of neural networks, so the FSTDMF with different hidden layers is further studied in this paper. In this paper, extensive experiments are conducted on the Movielens100k, Movielens1M, and DoubanMovie-Movielens20M datasets to study the proposed model with different numbers of hidden layers. Pay attention, the target score of Movielens100k and Movielens1M-Movielens100k are the same, but the auxiliary information is different, so only Movielens100k is used in this experiment. For detailed comparison, Table 8 shows the performance of different layers at different training set ratios.

As can be seen from Table 8, the results indicate that the 2-layer model achieved the best performance among all the data set. This finding is highly encouraging as it demonstrates the effectiveness of utilizing deep models for collaborative recommendations. The improved performance can be attributed to the increased nonlinearity introduced by stacking multiple nonlinear layers. The relatively poor performance of the 1-layer model suggests that a single-layer neural network may not have sufficient depth to achieve optimal learning results. However, it is worth noting that adding more layers beyond two does not appear to yield further improvements. In fact, the 3-layer model even exhibits a decrease in performance. This phenomenon could be attributed to the higher layers of the network causing overfitting, where the model becomes too specialized to the training data and fails to generalize well to unseen data. In summary, the results indicate that a 2-layer model strikes a good balance between capturing nonlinearity and avoiding overfitting, resulting in improved performance for recommendations.

Ablation study and execution time analysis

In this section, the performance of feature subspace transfer (FST) and semi-autoencoder (SA) is compared using four

Fig. 5 Influence of parameter η size on FSTD MF prediction error

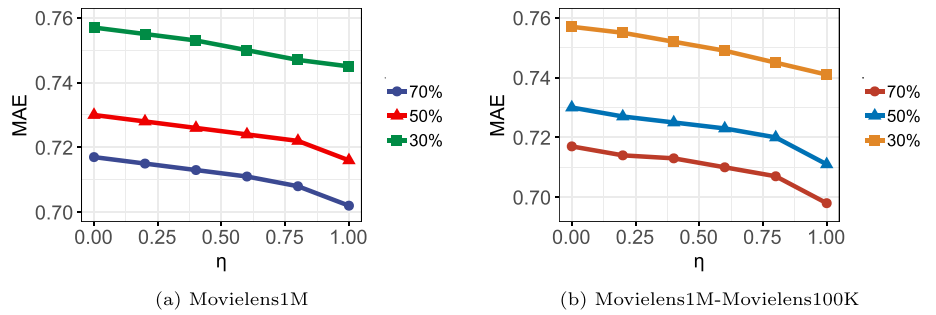
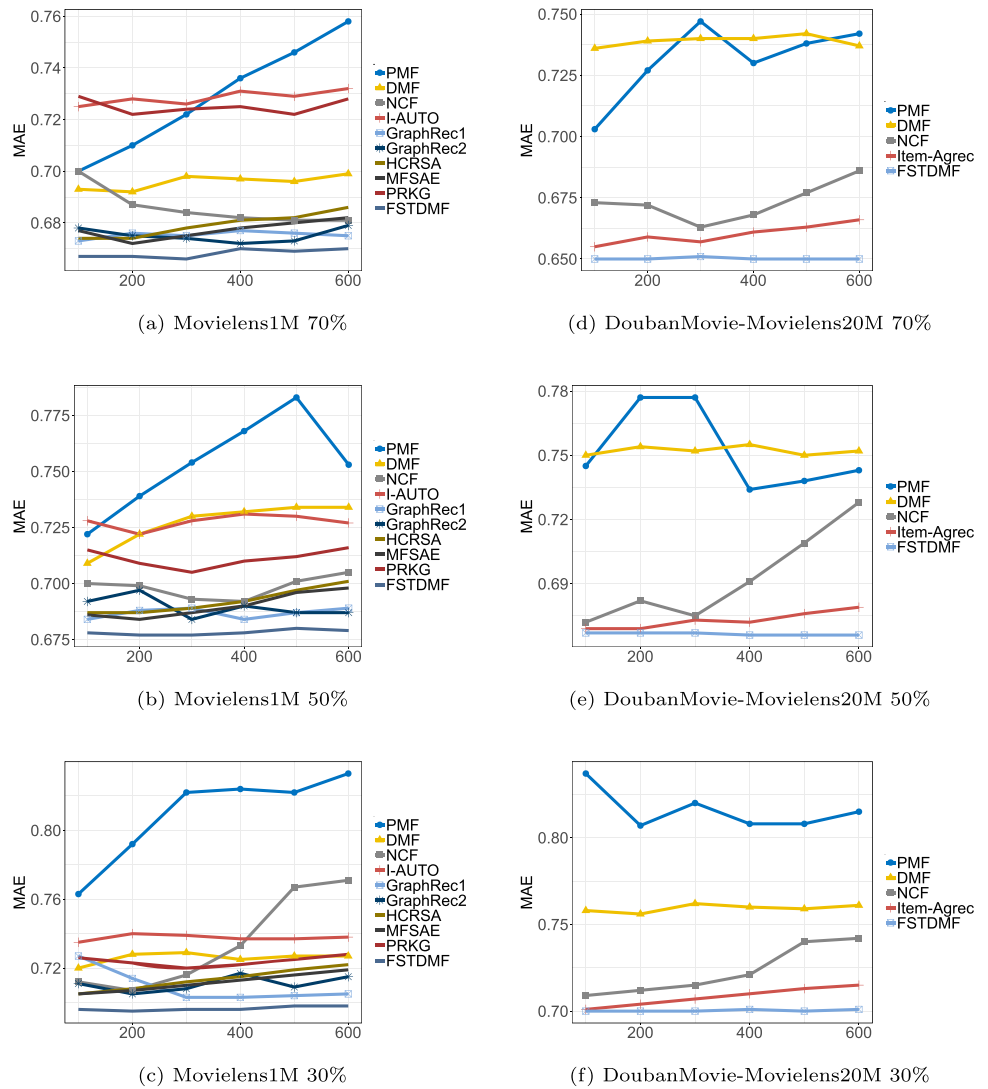


Fig. 6 The influence of different dimensions r on the prediction error of each algorithm in different data sets



datasets: Movielens100k, Movielens1M-Movielens100k, Movielens1M and DoubanMovie-Movielens20M. In addition, the optimized execution time of each method is analyzed, to ensure fairness, each method’s hyperparameters are consistent on each data set. Since there is little difference in execution time under different proportions, the execution time of each algorithm is the average of its execution time

under three different proportions. The experimental results for both methods are presented in Table 9.

As can be seen, the results demonstrate that both DMF (+SA) and DMF (+FST) consistently outperform the standalone DMF across all datasets, indicating their effectiveness. Specifically, on the Movielens100k and Movielens1M-Movielens100k datasets, DMF (+FST) performs better than DMF (+SA), suggesting that feature subspace transfer is

Table 8 The influence of different layers on FSTD MF prediction error in different data sets

Layer	Metrics	MAE			RMSE		
		1-layer	2-layer	3-layer	1-layer	2-layer	3-layer
Movielens100k	70%	0.718	0.701	0.714	0.910	0.893	0.912
	50%	0.734	0.713	0.722	0.931	0.909	0.923
	30%	0.773	0.744	0.746	0.977	0.945	0.951
Movielens1M	70%	0.683	0.667	0.693	0.865	0.850	0.888
	50%	0.699	0.678	0.689	0.885	0.863	0.883
	30%	0.726	0.696	0.705	0.918	0.885	0.900
DoubanMovie-Movielens20M	70%	0.662	0.650	0.657	0.859	0.846	0.854
	50%	0.679	0.667	0.674	0.884	0.871	0.878
	30%	0.713	0.700	0.704	0.935	0.921	0.924

The best results are highlighted in bold

Table 9 The influence of feature subspace transfer and semi-autoencoder on different data sets

Training sets metrics		70%		50%		30%		Execution time (s)
		MAE	RMSE	MAE	RMSE	MAE	RMSE	
Movielens100K	DMF	0.720	0.917	0.742	0.943	0.767	0.972	28
	DMF(+SA)	0.715	0.911	0.726	0.927	0.756	0.960	29
	DMF(+FST)	0.713	0.912	0.722	0.926	0.746	0.951	30
	FSTD MF	0.701	0.893	0.713	0.909	0.744	0.945	31
Movielens1M-Movielens100K	DMF	0.720	0.917	0.742	0.943	0.767	0.972	28
	DMF(+SA)	0.715	0.911	0.726	0.927	0.756	0.960	29
	DMF(+FST)	0.704	0.899	0.716	0.913	0.743	0.946	111
	FSTD MF	0.702	0.895	0.711	0.907	0.737	0.937	112
Movielens1M	DMF	0.693	0.884	0.700	0.906	0.720	0.914	146
	DMF(+SA)	0.671	0.855	0.684	0.872	0.704	0.896	147
	DMF(+FST)	0.668	0.857	0.680	0.871	0.703	0.897	148
	FSTD MF	0.667	0.850	0.678	0.863	0.696	0.885	149
DoubanMovie-Movielens20M	DMF	0.676	0.883	0.700	0.914	0.749	0.985	176
	DMF(+SA)	0.664	0.866	0.680	0.888	0.706	0.931	177
	DMF(+FST)	0.662	0.863	0.678	0.886	0.704	0.928	237
	FSTD MF	0.650	0.846	0.667	0.871	0.700	0.921	238

The best results are highlighted in bold

particularly advantageous for smaller datasets. On the other hand, for the Movielens1M and DoubanMovie-Movielens20M datasets, although DMF (+FST) improves the performance in terms of MAE and RMSE, the improvement is not significantly higher than that of DMF (+SA). This implies that when dealing with large sparse datasets, initializing latent feature representations using semi-automatic encoders can achieve almost the same level of improvement as feature subspace transfer. Taking an overview of these results, it can be concluded that FSTD MF consistently achieves the best results across all cases, highlighting the effectiveness of combining FST and SA.

It can be observed that DMF exhibits the lowest execution time across all data sets. FSTD MF has a higher time complexity compared to DMF due to the additional computing costs involved in incorporating SA and FST. The execution

time of DMF (+SA) is not significantly different from that of DMF. This is because SA is used for initializing the latent features of the item and is not involved in the model optimization process. Notably, the execution time of DMF (+FST) exhibits a slight increase on the Movielens100k and Movielens1M datasets. This is because, on the Movielens100k and Movielens1M datasets, the latent features of the item's auxiliary information are encoded by One-hot encoding, which takes less extra time. In addition, the execution time is significantly increased for the Movielens1M-Movielens100k and DoubanMovie-Movielens20M. This is because that the feature-extracting process on the auxiliary rating matrix Movielens1M and DoubanMovie require additional execution time. In general, the transfer learning methods are more time-consuming than the non-transfer methods due to the same reason.

Conclusion

This paper proposes a novel deep matrix factorization method with an item feature subspace transfer model for recommendation systems. This approach surpasses prior methods by innovatively integrating item auxiliary information, optimizing both the feature representations and the learning model's parameters. In the model framework, this work employed one-hot encoding in tandem with a semi-autoencoder model to extract the latent feature representation of item auxiliary information. Through the utilization of subspace projection distance, the latent feature representations are seamlessly transferred into the target task. Additionally, this paper leverages the semi-autoencoder to pre-initialize the latent feature representations of items. Extensive experiments conducted on five real-world datasets show the proposed framework outperforms competing methods in effectiveness.

Although the proposed model can make use of various auxiliary information of the project, its ability will be limited when the auxiliary information of the item is scarce. In addition, the model ignores the use of user auxiliary information, which also limits the ability of the model. In future research, the graphical features of the item will be considered and further combined with the auxiliary information of the user and the item, as well as the graphical features of the user and the item, to improve the predictive performance in the target task.

Acknowledgements We wish to express our appreciation to the reviewers for their helpful suggestions which greatly improved the presentation of this paper.

Author contributions The main idea of this paper is proposed by JW. The algorithm design and experimental environment construction are completed by WCW. The experimental verification was completed by WCW. The writing of the article is completed by WCW. And the writing guidance is completed by JW, English polish is jointly completed by JW and WCW.

Funding Not applicable.

Availability of data and materials The data included in this paper are available without any restriction.

Code availability We can provide these if needed.

Declarations

Conflict of interest The authors declare that they have no conflicts of interest to report regarding the present study.

Ethics approval All the authors have approved.

Consent to participate All the authors have approved.

Consent for publication All the authors have approved publication.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Zhao X, Ma Z, Zhang Z (2018) A novel recommendation system in location-based social networks using distributed ELM. *Memet Comput* 10:321–331
- Batmaz Z, Yurekli A, Bilge A, Kaleli C (2019) A review on deep learning for recommender systems: challenges and remedies. *Artif Intell Rev* 52:1–37
- Karimi-Mamaghan M, Mohammadi M, Meyer P, Karimi-Mamaghan AM, Talbi E-G (2022) Machine learning at the service of meta-heuristics for solving combinatorial optimization problems: a state-of-the-art. *Eur J Oper Res* 296(2):393–422
- Dobrojevic M, Zivkovic M, Chhabra A, Sani NS, Bacanin N, Amin MM (2023) Addressing internet of things security by enhanced sine cosine metaheuristics tuned hybrid machine learning model and results interpretation based on SHAP approach. *PeerJ Comput Sci* 9:1405
- Nematzadeh S, Kiani F, Torkamanian-Afshar M, Aydin N (2022) Tuning hyperparameters of machine learning algorithms and deep neural networks using metaheuristics: a bioinformatics study on biomedical and biological cases. *Comput Biol Chem* 97:107619
- Bacanin N, Stoean C, Zivkovic M, Rakic M, Strulak-Wójcikiewicz R, Stoean R (2023) On the benefits of using metaheuristics in the hyperparameter tuning of deep learning models for energy load forecasting. *Energies* 16(3):1434
- Zhao Y, Hu H, Song C, Wang Z (2022) Predicting compressive strength of manufactured-sand concrete using conventional and metaheuristic-tuned artificial neural network. *Measurement* 194:110993
- Sedhain S, Menon AK, Sanner S, Xie L (2015) Autorec: autoencoders meet collaborative filtering. In: *Proceedings of the 24th international conference on World Wide Web*, pp 111–112
- Dong B, Zhu Y, Li L, Wu X (2021) Hybrid collaborative recommendation of co-embedded item attributes and graph features. *Neurocomputing* 442:307–316
- Rashed A, Grabocka J, Schmidt-Thieme L (2019) Attribute-aware non-linear co-embeddings of graph features. In: *Proceedings of the 13th ACM conference on recommender systems*, pp 314–321
- Fan J, Cheng J (2018) Matrix completion by deep matrix factorization. *Neural Netw* 98:34–41
- Xue H-J, Dai X, Zhang J, Huang S, Chen J (2017) Deep matrix factorization models for recommender systems. In: *IJCAI*, vol 17. Melbourne, Australia, pp 3203–3209
- Chicaiza J, Valdiviezo-Diaz P (2021) A comprehensive survey of knowledge graph-based recommender systems: technologies, development, and contributions. *Information* 12(6):232
- Zhang S, Yao L, Xu X, Wang S, Zhu L (2017) Hybrid collaborative recommendation via semi-autoencoder. In: *Neural information processing: 24th international conference, ICONIP 2017*,

- Guangzhou, China, 14–18 Nov 2017, proceedings, part I 24. Springer, pp 185–193
15. Zou H, Chen C, Zhao C, Yang B, Kang Z (2019) Hybrid collaborative filtering with semi-stacked denoising autoencoders for recommendation. In: 2019 IEEE international conference on dependable, autonomic and secure computing, international conference on pervasive intelligence and computing, international conference on cloud and big data computing, international conference on cyber science and technology congress (DASC/PiCom/CBDCCom/CyberSciTech). IEEE, pp 87–93
 16. He J, Liu R, Zhuang F, Lin F, Niu C, He Q (2018) A general cross-domain recommendation framework via Bayesian neural network. In: 2018 IEEE international conference on data mining (ICDM). IEEE, pp 1001–1006
 17. Man T, Shen H, Jin X, Cheng X (2017) Cross-domain recommendation: an embedding and mapping approach. In: IJCAI, vol 17, pp 2464–2470
 18. Geng Y, Zhu Y, Li Y, Sun X, Li B (2022) Multi-feature extension via semi-autoencoder for personalized recommendation. *Appl Sci* 12(23):12408
 19. Yang Y, Zhu Y, Li Y (2022) Personalized recommendation with knowledge graph via dual-autoencoder. *Appl Intell* 52:6196–6207
 20. Yin F, Ji M, Wang Y, Yao Z, Feng X, Li S (2022) Enhanced graph recommendation with heterogeneous auxiliary information. *Complex Intell Syst* 8(3):2311–2324
 21. Adomavicius G, Tuzhilin A (2005) Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. *IEEE Trans Knowl Data Eng* 17(6):734–749
 22. Schafer JB, Frankowski D, Herlocker J, Sen S (2007) Collaborative filtering recommender systems. In: Brusilovsky P, Kobsa A, Nejdl W (eds) *The adaptive web: methods and strategies of web personalization*, vol 4321. Springer, Berlin, Heidelberg, p 291
 23. Su X, Khoshgoftaar TM (2009) A survey of collaborative filtering techniques. *Adv Artif Intell* 2009:421425
 24. Koren Y, Bell R, Volinsky C (2009) Matrix factorization techniques for recommender systems. *Computer* 42(8):30–37
 25. Zhang S, Yao L, Sun A, Tay Y (2019) Deep learning based recommender system: a survey and new perspectives. *ACM Comput Surv (CSUR)* 52(1):1–38
 26. Sun Z, Guo Q, Yang J, Fang H, Guo G, Zhang J, Burke R (2019) Research commentary on recommendations with side information: a survey and research directions. *Electron Commer Res Appl* 37:100879
 27. He X, Liao L, Zhang H, Nie L, Hu X, Chua T-S (2017) Neural collaborative filtering. In: *Proceedings of the 26th international conference on World Wide Web*, pp 173–182
 28. Bobadilla J, Ortega F, Gutiérrez A, González-Prieto Á (2023) Deep variational models for collaborative filtering-based recommender systems. *Neural Comput Appl* 35(10):7817–7831
 29. Kim D, Park C, Oh J, Lee S, Yu H (2016) Convolutional matrix factorization for document context-aware recommendation. In: *Proceedings of the 10th ACM conference on recommender systems*, pp 233–240
 30. Oord A, Dieleman S, Schrauwen B (2013) Deep content-based music recommendation. In: *Proceedings of the 26th international conference on neural information processing systems*, pp 2643–2651
 31. Tan YK, Xu X, Liu Y (2016) Improved recurrent neural networks for session-based recommendations. In: *Proceedings of the 1st workshop on deep learning for recommender systems*, pp 17–22
 32. Wu S, Ren W, Yu C, Chen G, Zhang D, Zhu J (2016) Personal recommendation using deep recurrent neural networks in netease. In: 2016 IEEE 32nd international conference on data engineering (ICDE). IEEE, pp 1218–1229
 33. Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A, Bengio Y (2020) Generative adversarial networks. *Commun ACM* 63(11):139–144
 34. Li S, Kawale J, Fu Y (2015) Deep collaborative filtering via marginalized denoising auto-encoder. In: *Proceedings of the 24th ACM international on conference on information and knowledge management*, pp 811–820
 35. Chen Y, Rijke M (2018) A collective variational autoencoder for top-n recommendation with side information. In: *Proceedings of the 3rd workshop on deep learning for recommender systems*, pp 3–9
 36. Li X, She J (2017) Collaborative variational autoencoder for recommender systems. In: *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, pp 305–314
 37. Liang D, Krishnan RG, Hoffman MD, Jebara T (2018) Variational autoencoders for collaborative filtering. In: *Proceedings of the 2018 World Wide Web conference*, pp 689–698
 38. Strub F, Mary J, Gaudel R (2016) Hybrid collaborative filtering with neural networks. [arXiv:1603.00806](https://arxiv.org/abs/1603.00806)
 39. Zhu F, Wang Y, Chen C, Zhou J, Li L, Liu G (2021) Cross-domain recommendation: challenges, progress, and prospects. In: 30th international joint conference on artificial intelligence, pp 4721–4728
 40. Pan W, Xiang E, Liu N, Yang Q (2010) Transfer learning in collaborative filtering for sparsity reduction. In: *Proceedings of the AAAI conference on artificial intelligence*, vol 24, pp 230–235
 41. Wang J, Ke L (2014) Feature subspace transfer for collaborative filtering. *Neurocomputing* 136:1–6
 42. Liu DC, Nocedal J (1989) On the limited memory BFGS method for large scale optimization. *Math Program* 45(1–3):503–528
 43. Igel C, Hüsken M (2000) Improving the RPROP learning algorithm. In: *Proceedings of the second international ICSC symposium on neural computation (NC 2000)*, vol 2000, pp 115–121
 44. Igel C, Hüsken M (2003) Empirical evaluation of the improved RPROP learning algorithms. *Neurocomputing* 50:105–123
 45. Zhu F, Wang Y, Chen C, Liu G, Zheng X (2020) A graphical and attentional framework for dual-target cross-domain recommendation. In: IJCAI, pp 3001–3008
 46. Harper FM, Konstan JA (2015) The movielens datasets: history and context. *ACM Trans Interact Intell Syst (TIIS)* 5(4):1–19
 47. Salakhutdinov R, Mnih A (2007) Probabilistic matrix factorization. In: *Proceedings of the 20th international conference on neural information processing systems*, pp 1257–1264
 48. Singh AP, Gordon GJ (2008) Relational learning via collective matrix factorization. In: *Proceedings of the 14th ACM SIGKDD international conference on knowledge discovery and data mining*, pp 650–658
 49. Zhu Y, Tang Z, Liu Y, Zhuang F, Xie R, Zhang X, Lin L, He Q (2022) Personalized transfer of user preferences for cross-domain recommendation. In: *Proceedings of the fifteenth ACM international conference on Web search and data mining*, pp 1507–1515

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.