



EAF-SR: an enhanced autoencoder framework for social recommendation

Taiheng Liu^{1,2} · Zhaoshui He^{3,4,5,6}

Received: 6 August 2021 / Revised: 23 June 2022 / Accepted: 12 September 2022 /
Published online: 8 October 2022

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2022

Abstract

In the recommendation systems, the user feedback data (e.g., user-item rating and the social information) is usually sparse, discrete and full of noise. However, most of existing methods use it to make recommendations directly, which leads to the reduction of their recommendation accuracy and quality ultimately. To address this problem, this paper proposes an Enhanced Autoencoder Framework with knowledge distillation (EAF-SR) for learning robust information from user feedback data, where it consists of three parts: Pre-training, knowledge distillation layers and Re-training. Specifically, Pre-training is proposed to generate the soft targets by using the stacked denoising autoencoder (SDAE) from user feedbacks and trust information, which aims to reduce data noise. Then, knowledge distillation layers are developed to learn the robust information from the generated soft targets. Finally, Re-training network combined with Pre-training is designed to make recommendations for each user. Extensive experiments done on three real-world datasets (e.g., Flixster, Epinions and Douban) show that the proposed EAF-SR is superior to the state-of-the-art approaches.

Keywords Deep learning · Recommender system · Stacked denoising autoencoder · Knowledge distillation · Social recommendation

✉ Zhaoshui He
zhshhe@gdut.edu.cn

Taiheng Liu
2111714019@mail2.gdut.edu.cn

- ¹ Guangdong Key Laboratory of Electromagnetic Control and Intelligent Robots, College of Mechatronics and Control Engineering, Shenzhen University, Shenzhen 518060, China
- ² College of Physics and Optoelectronic Engineering, Shenzhen University, Shenzhen 518060, China
- ³ School of Automation, Guangdong University of Technology, Guangzhou 510006, China
- ⁴ Key Laboratory for IoT Intelligent Information Processing and System Integration of Ministry of Education, Guangzhou 510006, China
- ⁵ Guangdong Key Laboratory of IoT Information Technology, Guangzhou 510006, China
- ⁶ Guangdong-HongKong-Macao Joint Laboratory for Smart Discrete Manufacturing, Guangzhou 510006, China

1 Introduction

With the rapid development of Internet technology, the amount of information on the Internet is growing exponentially, resulting in a serious problem of information overload. To overcome this problem, the recommendation systems (RS) emerge as the times required, which can help users to find the information that feeds their requirements [22]. RS can not only filter irrelevant information according to users' preferences, but also provide personalized recommendations for users. Among existing techniques of personalized recommendation [3, 16], collaborative filtering (CF) [18] is one of the most widely used ones. It has been widely used in many commercial websites, such as Amazon, YouTube, Netflix, E-commerce, Taobao, Douban and Last.fm. CF-based methods attempt to exploit the available user-item rating data to make predictions about users' preferences, in which they can be divided into two major categories [41]: memory-based and model-based. Specifically, the former [14] makes predictions based on the similarities between users and items, while the latter [25] tries to create a prediction model by utilizing machine learning.

Although CF has achieved huge success over the past two decades, some important issues in RS still remain unsolved, as follows:

- **Sparsity Problem:** In real-world, most users typically rate or experience only a small fraction of the available products. As a result, the density of the available ratings in the recommendation systems is always less than 1% [17]. Owing to this data sparsity, CF methods have a number of difficulties in trying to classify related users in the system. Therefore, the predictive quality of the recommendation system may be significantly restricted.
- **Cold-Start Problem:** So far, this is a universal challenge in the recommendation system. Here, the cold-start refers to the users who have expressed no or a few ratings, or items which have been rated by no or a few users. Owing to the lack of sufficient rating data, the similarity-based approaches fail to find out the nearest neighbor users or items and in turn, have degraded the consistency of recommendations by conventional recommendation algorithms.

Furthermore, the traditional CF approaches purely mines the user-item rating matrix for recommendations, which cannot sufficiently provide accurate and reliable predictions. However, in daily life, people usually like to refer to their trust friends' preferences to make decisions rather than the mass population. To this end, Qian et al. [31] present a novel social recommendation using global rating reputation and local rating similarity (SoRS) framework to make user social recommendations. Li et al. [21] propose a social matrix factorization method to optimize the prediction solution in both users' latent feature space and user-item rating space using the individual trust among users. Wang et al. [44] propose a social-enhanced content-aware recommendation method by fusing the social network, and item's and user's reviews. Although these CF approaches, incorporating social trust relationships, are very effective for the industrial recommendation, their performance may be limited by these linear models based on matrix factorization.

Recently, deep learning has made breakthroughs in a lot of domains, such as image processing [28, 42], natural language understanding [27] and speech recognition [10], which can bring new opportunities for the research of recommendation systems. Compared with traditional machine learning methods [34, 35], deep learning not only has a strong ability to learn the essential characteristics of data sets from samples, but also can obtain deep-level feature representations of users and items. Moreover, deep learning uses automatic feature

learning from multi-source heterogeneous data to map the different structures of data to the same hidden space, which can obtain a unified structure of the data. Based on these benefits of deep learning, it has attracted much attention on how to utilize deep learning to improve the recommendation performance of the recommendation system. For example, Wei et al. [45] propose an integrated recommendation models with CF and deep learning to solve the complete cold-start problem (IRCD-CCS). Fu et al. [11] propose a novel deep learning method that imitates an effective intelligent recommendation by understanding the users and items beforehand to overcome these setbacks that CF-based methods only grasp the single type of relation. Da'u et al. [6] propose a recommendation system that utilizes aspect-based opinion mining (ABOM) based on the deep learning technique to improve the accuracy of the recommendation process. Shamshoddin et al. [38] propose a deep learning with collaborative filtering technique for the recommendation system to predict user preferences from the Internet of Things (IoT) devices and Social Networks.

In general, deep learning needs massive data to train a robust and accurate model, which contains a lot of parameters to fit training data. It means that they need heavy computing resources. Moreover, for the recommendation systems, the data of user feedbacks are usually sparse and discrete, because of that user usually rates or experiences only a small fraction of the available items. If these sparse and discrete data are used directly to train the model, these problems may severely restrict the performance of deep neural networks for recommendation systems. Therefore, it is necessary to develop a method that can not only reduce computing resources, but also mine these sparse and discrete data in recommendation systems.

To this end, this paper proposes an Enhanced Autoencoder Framework to learn robust knowledge from the sparse and discrete data of use feedbacks, named as EAF-SR. In summary, the main contributions of this paper are summarized as follows:

- An Enhanced Autoencoder Framework (EAF-SR) is proposed for social recommendation by using the technique of knowledge distillation.
- For better learning robust information from the data of user feedbacks, an autoencoder is employed to map the different structures of data into the same hidden space, which can obtain a unified structure of the data.
- In order to alleviate the problem of data sparseness, a stacked denoising auto-encoder is proposed to generate soft targets, thereby learning the hidden information in users' social information.
- To reduce computing resources, a knowledge distillation-based recommendation method is proposed, thereby making it run in a low-cost system.
- To provide robust recommendations to users, the pre-training and re-training networks is proposed to make predictions.

The remaining of this paper is organized as follows. Section 2 reviews the previous works regarding recommendation systems. Section 3 gives the details of the proposed EAF-SR. In Section 4, a series of experiments are conducted to illustrate the performance of the proposed EAF-SR. Section 5 draws conclusions.

2 Related works

In this section, we review the related works briefly, including social recommendation, deep learning for recommendation systems and knowledge distillation.

2.1 Social recommendation

Social scientists have long believed that the users' preferences are similar to her/his social connections with the social correlation theories of homogeneity and social influence [48]. With the increasing popularity of online social networking platforms, the social network information has become an effective data source to alleviate data sparsity problem and enhance recommendation performance [40]. Since the latent factor-based models perform better than the neighborhood-based methods in CF, a popular direction is how to design more complex latent factor-based social recommendation models. For example, Ma et al. [26] propose a latent factor-based framework with social regularization for recommendation. SocialMF is proposed to incorporate the social influence theory into classical latent factor-based models [15]. By treating each users' social connections' preferences of an item as the auxiliary feedback of this user, researchers proposed a trust-based latent factor-based model that leveraged the auxiliary feedback [12].

With the observation that users tend to assign higher rankings to items that their friends prefer, a personalized ranking-based social recommendation is proposed that extends the classical BPR model with the observation [54]. Researchers also argued that both positive and negative links in social networks provide valuable clues for recommendation performance [9]. These social recommendation algorithms have also been extended to incorporate rich context information, such as social circle [30] and item content [55]. Since the performance of these latent factor-based models for social recommendation relied on the initialization of user and item latent factors, researchers proposed to apply autoencoder, an unsupervised deep learning technique in initialization [7]. These models showed improved performance over classical recommendation models. Nevertheless, few have explored the possibility of designing deep learning-based social recommendation models. Recently, neural social collaborative ranking is proposed to solve the problem of bridging a few overlapping users in the two domains of the social network domain and information domain [43]. Researchers also proposed to use deep learning models to model the social influence strength for temporal social recommendation [52]. This paper on social recommendation differs from these works as it focuses on learning robust information from user social network to generate soft labels.

2.2 Deep learning for recommendation systems

Recently, deep learning has achieved unexpected and great success in many fields due to its great learning ability, including Object Detection [49], Speech Recognition [46] and Natural Language Processing [24]. Motivated by it, there are many works proposed to use it to improve the performance of recommendation systems. Especially, Salakhutdinov et al. [36] propose the Restricted Boltzmann Machine (RBM) model by mapping the original input data from the visualization layer to the hidden layer and remapping the obtained hidden layer vector to the visible layer to obtain the rating of items. Sedhain et al. [37] develop an Autoencoders Meet Collaborative Filtering (AutoRec) model by constructing the missing part of the matrix and applying the constructed data to recommend products. Anil et al. [1] propose a LSTM-GRU-Hybrid method by combining deep learning neural architectures and collaborative filtering to provide an effective recommendation.

In a word, deep learning needs massive data to train deep neural networks. However, the data used in recommendation systems always faces serious sparse problems, which make it difficult to take full advantage of the deep neural networks in the recommendation system.

In most cases, the observed and unobserved user-item pairs are treated as ones and zeros in the recommendation system. However, these methods cannot fully reflect the real life. In fact, it is difficult to accurately describe the users’ preferences through simply using ones and zeros, which makes it not easy to model users with these noisy data. Based on this fact, in order to improve the accuracy of recommendation, a pre-training network is proposed to reduce the noise.

2.3 Knowledge distillation

Recently, Hinton et al. [13] propose the soft targets of a generation network containing abundant of information, which can be used to train another network and achieve competitive performance. Motivated by this idea, Cui et al. [5] propose a Multi-View Recurrent Neural Network (MV-RNN) model to use soft targets of separate structures to handle multi-view features. Dighe et al. [8] utilize the technique of soft targets to improve the mapping of far-field acoustics to close-talk senone classes. Zhao et al. [56] propose a new collaborative teaching knowledge distillation (CTKD) strategy to use the valuable information among the training process associated with training results to improve the performance of the student network. However, in the actual situation, these soft targets in the pre-training network also have a lot of noise, which is not perfect to fit the input data. Therefore, the phases of pre-training and re-training network are incorporated into a unified framework. In this way, the proposed model can propagate the training errors of the distillation process to tune the soft targets and reduce their noise to improve performance. Moreover, a novel distillation layer is proposed to adjust each unit of generated vectors to balance the effects of knowledge and noise based on the corresponding reliability. Finally, different from the previous works that make predictions solely based on the results of distillation network, this paper proposes to make final recommendations by integrating both results of the generation and distillation subnetworks.

3 An enhanced autoencoder framework for social recommendation

In this section, an enhanced autoencoder framework is proposed for social recommendation, named as EAF-SR, which consists of three parts (see Fig. 1): Pre-training, distilled

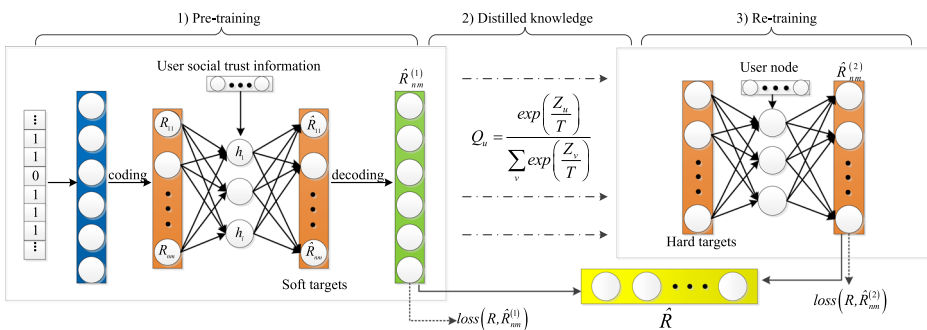


Fig. 1 The entire structure of the proposed EAF-SR, which consists of three modules: 1) Pre-training for generating soft targets with SDAE; 2) Distilled knowledge; 3) Re-training for learning implicit information from soft targets

knowledge layer, and Re-training. Specially, Pre-training is designed to generate soft targets by utilizing SDAE [2]. Next, the distillation layers are proposed to balance the knowledge and noise of outputs of Pre-training. After that, Re-training is developed to learn implicit knowledge from soft targets based on autoencoder structure. Finally, both Pre-training and Re-training network are combined by using distilled knowledge to make robust recommendations for users.

3.1 Notation and motivations

3.1.1 Notation

Notations used in this paper is first introduced, as shown in Table 1.

3.1.2 Motivations

For top-N recommendation, the observed and unobserved user-item pairs are regarded as ones and zeros, respectively. However, utilizing these hard targets are difficult to exactly reflect users' true interests on items. In fact, users may prefer some of the potential unobserved items than observed items. As a consequence, it is not easy to exactly model users by directly utilizing these discrete hard targets of user feedbacks.

For example, as demonstrated in Fig. 2, there are two figures to represents user feedbacks for items in views of hard and soft targets, respectively. Suppose there are three users $\{u_1, u_2, u_3\}$ and four items $\{v_1, v_2, v_3, v_4\}$ in recommender system, Fig. 2a demonstrates the observed user-item matrix with hard targets and Fig. 2b shows a possible user-item matrix to show latent user preferences with soft targets for items. Specially, in Fig. 2a, the preference

Table 1 List of Key Symbols

Symbol	Description
U	The user set
V	The item set
R	The user-item rating matrix
$r_u \in R$	The set of rows of the rating matrix R
D	The side information matrix in user social network
D_u	The set of rows of the side information matrix D
h_l	The corrupted output of ratings in the l layer
$g(x)$	The sigmoid function
W	The weight matrices about rating
S	The weight matrices about social network
b	The bias vector
p_i	The probability of category i
t	The temperature of knowledge distillation network
Q_u	The output of the distillation layer
θ_e	The values of trained parameters at iteration e
η	The learning rate for each iteration of training process
g_e	The gradients values at iteration e times

of user u_2 is closer to user u_1 than u_3 . However, in Fig. 2b, user u_2 is closer to user u_3 than u_1 . This example shows that directly learning user preferences from hard targets may lead to incorrect results.

Towards this problem, we propose to transfer the hard targets with discrete values to soft targets with continuous values by a pretrained model, and then utilize another network to learn user preferences from the generated soft targets for recommender systems. However, this technique of knowledge distillation is not a free lunch. Since the soft targets are generated by an imperfect model, there exist a lot of noise among the generated data. These noises have an adverse impact on the task of knowledge distillation. Therefore, how to develop a robust model to address this problem is remaining challenge.

In this paper, we propose a novel Enhanced Autoencoder Framework (EAF-SR) for Social Recommendation with knowledge distillation to learn robust information from soft targets for users. As demonstrated in Fig. 1, the overall structure of EAF-SR is consisted of three components: a Pre-Training network, a distillation layer and a Re-training network. Specially, the soft targets are generated by a SDAE network, which is proposed by injecting a user node into the hidden layer to exactly model users' interests. Followed by this network, we propose a novel distillation layer to adjust the generated targets to remain useful knowledge and reduce noise for retraining based on the corresponding reliability of each unit. Finally, we use an autoencoder network to learn implicit information from these soft targets for each user.

To learn the soft targets, we incorporate the generation and retraining stages into a unified framework for EAF-SR. As demonstrate in Fig. 1, the soft targets in EAF-SR model are constrained with hard targets in three perspectives: they are generated from hard targets, they are close to hard targets, and they can be used to reconstruct hard targets. So that the soft targets contain much less noise than hard targets, which make it easier to learn useful information from soft targets.

In particular, there exists a critical question about this idea. From the perspective of information theory, the useful information in generated soft targets is not more than that in original data. So that it raises a question of how can the EAF-SR model mine more information from soft targets than hard targets? Actually, this approach does not increase any information of users. As demonstrated in Fig. 1, this method turns the perspective to understand users by soft targets, which contains much less noise than hard targets. Therefore, by understanding users in different perspectives, this method is easier to learn robust knowledge from user feedbacks.

3.2 Pre-training for generating soft targets with SDAE

Recently, great achievements have been made in applying deep learning to recommendation systems. For example, Li et al. [20] propose a general deep architecture for collaborative filtering by integrating matrix factorization with deep feature learning. Wu et al. [47] present a novel method, called Collaborative Denoising Auto-Encoder (CDAE), for the top-N recommendation that utilizes the idea of Denoising Auto-Encoders. Zhang et al. [51] propose a new hybrid model by generalizing the contractive auto-encoder paradigm into a matrix factorization framework with good scalability and computational efficiency. Li et al. [19] propose a Bayesian generative model called collaborative variational autoencoder (CVAE) that considers both rating and content for the recommendation in a multimedia scenario. Liang et al. [23] extend variational autoencoders (VAEs) to collaborative filtering for implicit feedback. From these studies, it can be found that most of the approaches are based

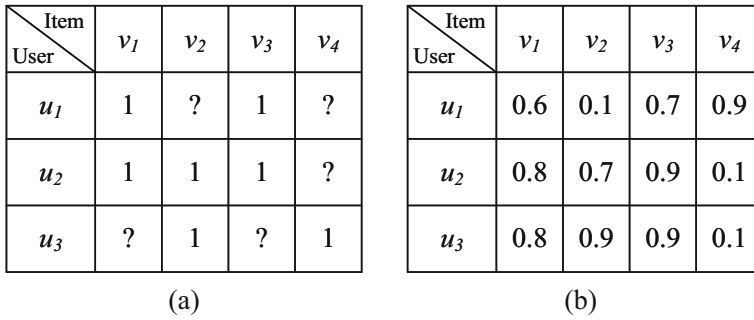


Fig. 2 An example of two user-item matrices in recommender systems: (a) A user-item matrix with hard targets for recommendation systems, where “1” for observed data and “?” for others; (b) A user-item matrix with soft targets to show latent user preferences for recommendation systems

on denoising autoencoder (DAE) technology, which has proved that it can improve the performance of recommendation systems. Therefore, in this subsection, a stacked denoising autoencoders (SDAE) based pre-training method (see Fig. 3) is proposed to generate soft target.

Suppose the user set $U = \{u_1, u_2, u_3, \dots, u_n\}$, the item set $V = \{v_1, v_2, v_3, \dots, v_m\}$ and the rating matrix $R \in \mathbb{R}^{n \times m}$. Herein, R_{nm} is the rating given the item m by user n that if the user m has rated the item n , its value is $R_{nm} \neq 0$, otherwise, it is $R_{nm} = 0$. Moreover, the rating set $r_u = \{r_{u_1}, r_{u_2}, \dots, r_{u_n}\} \in \mathbb{R}^{n \times m}$ and r_{u_1} is the rating vector of user u_1 . Next, suppose the user u side information D , where $D_u = \{d_{u_1}, d_{u_2}, \dots, d_{u_n}\} \in D$, and d_{u_1} is the trust vector of user u_1 . The steps of generating soft target with SDAE are as follows:

- 1) Each user $u \in U$ is first made to \tilde{u} by adding noise. Then, for each hidden layer $l = \{1, 2, 3, \dots, L - 1\}$ (where its optimal value is 30, as shown in Section 4.4.5), it can obtain

$$h_l = g(W_l^T h_{l-1} + S_l \tilde{D}_u + b_l), \tag{1}$$

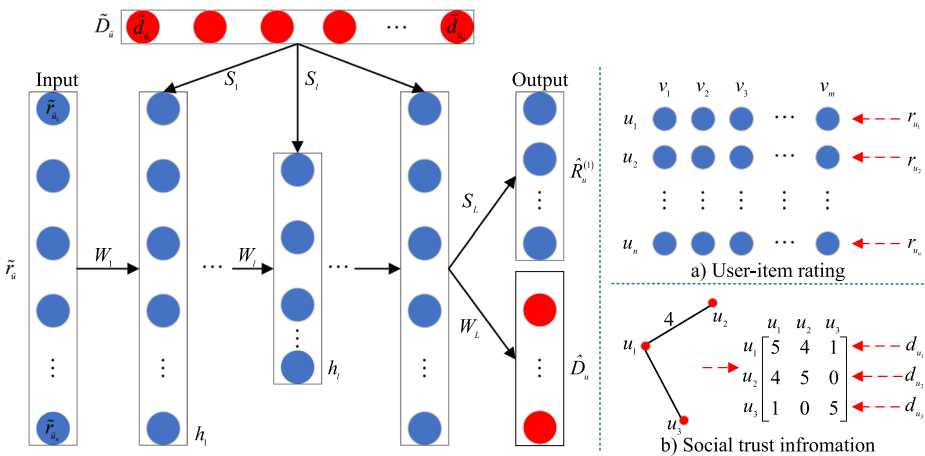


Fig. 3 Pre-training based on SDAE for recommendation with user side information, which includes two parts: (a) User-item rating; (b) Social trust information

where $g(x)$ is an activation function that can be described by the sigmoid function $g(x) = \frac{1}{1+e^{-x}}$, $W_l \in \mathbb{R}^{n \times k}$ and $S_l \in \mathbb{R}^{n \times k}$ indicate the weight matrices of the l layer, b_l represents the bias vector of the l layer, \tilde{D}_u denotes the side information of user u who has been added noise and $h_0 = \tilde{r}_u$ is one of corrupted inputs of ratings.

- 2) For output layer L , the calculation method for reconstructing the output layer is shown as

$$\begin{cases} \hat{R}_u^{(1)} = g(W_L h_L + b_{D_u}) \\ \hat{D}_u^{(1)} = g(S_L h_L + b_{D_u}) \end{cases} \tag{2}$$

where $W_L \in \mathbb{R}^{n \times k}$ and $S_L \in \mathbb{R}^{n \times k}$ indicate the weight matrices of the L layer, and b_{D_u} represents the bias vector.

- 3) Use the deep network integrated into the social information to reconstruct the input and minimize the square loss between the input and the reconstruction. The loss function is defined as

$$\begin{aligned} loss(R, D, \hat{R}_u^{(1)}, \hat{D}_u^{(1)}) &= \sum_u [(R - \hat{R}_u^{(1)})^2 + (D - \hat{D}_u^{(1)})^2] \\ &+ \lambda (\|W_l\|_F^2 + \|S_l\|_F^2 + \|b_{D_u}\|_F^2) \end{aligned} \tag{3}$$

where λ denotes a hyper-parameter to avoid the overfitting and $\|\cdot\|_F^2$ is the Frobenius norm.

3.3 Distilled knowledge layer

Knowledge distillation (KD) is proposed by Hinton [13]. It is to distill the knowledge learned in the complex teacher network into a simple student network. The “softmax” output layer is to normalize each type of value z_i generated by the neural network, and finally, the probability of correct classification p_i is obtained, where the greater the probability, the greater the probability of being classified into that category. The probability of each category is expressed as

$$p_i = \frac{\exp(z_i)}{\sum_j \exp(z_j)} \tag{4}$$

The core idea of knowledge distillation is to use soft targets to assist hard targets for training. First, the probability distribution of teacher network “softening” is calculated, and then it is used as the part of the total loss to induce student network training. Among them, the “softening” probability distribution calculation method is based on the “softmax”, and a temperature parameter t is introduced to obtain the probability distribution of “softening” as

$$q_i = \frac{\exp(\frac{z_i}{t})}{\sum_j \exp(\frac{z_j}{t})} \tag{5}$$

where z_i is the output logit before the softmax of the neural network. t denotes the temperature. When $t = 1$, it is the common softmax output probability. If t is larger, the distribution of classification probability will be relaxed. If t is smaller, the probability of misclassification will be enlarged, which is easy to incorporate unnecessary noise.

In recommendation systems, each user has multiple tags (such as Time, Age and Company), as shown in Fig. 4. Therefore, benefits from knowledge distillation, it can be used

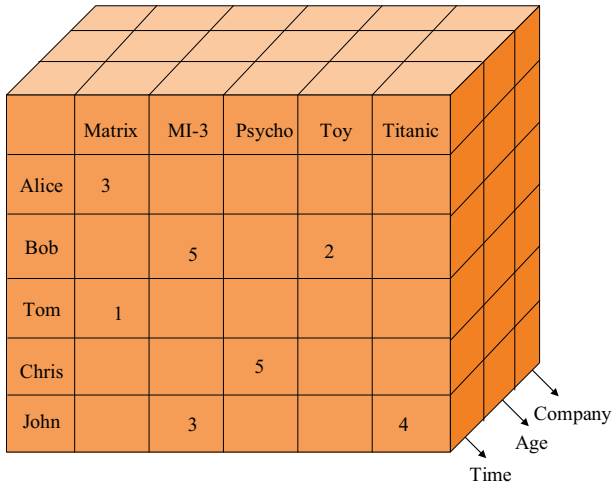


Fig. 4 The context_aware of the rating matrix

to assist hard labels to train by obtaining the information from users’ tags (soft labels). The output of the distillation layer is described as

$$\begin{cases} Q_u = \frac{\exp(Z_u)}{\sum_u \exp(Z_u)} \\ Z_u = W_L h_L + b_{D_u} \end{cases} \tag{6}$$

where Z_u indicates the output logit before “softmax” of the neural network. Herein, the distillation layer is developed to adjust each output unit of pre-training network.

3.4 Re-training for learning implicit information from soft targets

After distillation, there is some implicit information in the soft targets that aren’t learned. Thus, the re-training network is used to learn them, which its details are as follows:

- 1) The soft targets are first mapped to the low-dimensional hidden layer by

$$H_u = g(W_1^T Q_u + b_1), \tag{7}$$

where $W_1 \in \mathbb{R}^{k \times m}$ and $b_1 \in \mathbb{R}^k$ indicate training parameters for mapping the input vector of Q_u to a dimension of the low-dimensional space of k .

- 2) The implicit information is then learned from the low-dimensional hidden layer by using knowledge distillation. The output of the distillation process for user u can be expressed as

$$\begin{cases} \hat{R}_u^{(2)} = g(W_2^T H_u + b_2) \\ \hat{D}_u^{(2)} = g(W_3^T Q_u + b_3) \end{cases} \tag{8}$$

where $W_2 \in \mathbb{R}^{k \times m}$, $b_2 \in \mathbb{R}^k$ and $b_3 \in \mathbb{R}^k$ denote the training parameter to make predictions for each user.

- 3) The output of re-training is obtained by using deep network to integrate hard targets and users' node. The loss function of re-training is defined as

$$\begin{aligned} loss(R, D, \hat{R}_u^{(2)}, \hat{D}_u^{(2)}) &= \sum_u \left[(R - \hat{R}_u^{(2)})^2 + (D - \hat{D}_u^{(2)})^2 \right] \\ &+ \lambda (\|b_1\|_F^2 + \|b_2\|_F^2 + \|b_3\|_F^2) \end{aligned} \quad (9)$$

Next, the loss function from the two parts of pre-training and re-training networks is obtained. The loss function is described as

$$\begin{aligned} &loss(R, D, \hat{R}_u^{(1)}, \hat{D}_u^{(1)}, \hat{R}_u^{(2)}, \hat{D}_u^{(2)}) \\ &= \alpha \sum_u loss \left(R, D, \hat{R}_u^{(1)}, \hat{D}_u^{(1)} \right) + (1 - \alpha) loss \left(R, D, \hat{R}_u^{(2)}, \hat{D}_u^{(2)} \right) \\ &+ \lambda \left(\|W_l\|_F^2 + \|Z_l\|_F^2 + \|b_{D_u}\|_F^2 + \|b_1\|_F^2 + \|b_2\|_F^2 + \|b_3\|_F^2 \right) \end{aligned} \quad (10)$$

where α represents the balance parameter to be used to adjust the proportion of the generated network and the re-training network. λ means adjustment parameter to prevent overfitting.

Finally, four predict functions of generation and distillation subnetworks for each user is obtained, i.e., $\hat{R}_u^{(1)}$, $\hat{R}_u^{(2)}$, $\hat{D}_u^{(1)}$ and $\hat{D}_u^{(2)}$. Specially, these four kinds of results are training to make predictions from different perspectives. The Pre-training network is focused on the known user-item pairs while the Pre-training one is more focused on the unknown user-item pairs. Therefore, they are combined to make final recommendations for users by:

$$\hat{R}_u = \beta (\hat{R}_u^{(1)} + \hat{D}_u^{(1)}) + (1 - \beta) (\hat{R}_u^{(2)} + \hat{D}_u^{(2)}) \quad (11)$$

where β is employed to control the contribution of Pre-training and Re-training for predict final results. In this way, a more robust prediction function without additional training costs is achieved.

3.5 Model training

In order to get the optimized solution, the Stochastic Gradient Descent approach (SGD) [4] is used to train our EAF-SR model. In regards to each iteration, these training parameters of our model can be updated by gradients. The update formula is defined as follows

$$\theta_{e+1} = \theta_e - \eta g_e, \quad (12)$$

where θ_e is the values of trained parameters θ in EAF-SR model at iteration e . η denotes the learning rate for each iteration of training process. g_e indicates the gradients values at iteration e times. Herein, the details of this training process are summarized in Algorithm 1.

4 Experiments

In this section, a series of experiments are conducted to demonstrate the performance of the proposed EAF-SR, in which the batch size, learning rate and training epoch are set to 128, 0.0005 and 1000, respectively. Meanwhile, these experiments are implemented on PyCharm and carried out on a workstation with three NVIDIA GeForce RTX 2080Ti GPU and 10GB memory.

Input: Given the user-item rating matrix r_u , user-user social relationship matrix D_u , the number of hidden layer L , and the temperature t .

Output: Final recommendations for users \hat{R}_u .

- 1: **–Step1:** Pre-training.
- 2: **while** not converge **do**
- 3: **for** l **do**
- 4: **for** u **do**
- 5: **if** $l < L$ **then**
- 6: Each user $u \in U$ is first made to \tilde{u} by adding noise.
- 7: For each hidden layer l , its output can be obtained by $h_l = g(W_l^T h_{l-1} + S_l \tilde{D}_{\tilde{u}} + b_l)$.
- 8: **else** $[l == L]$
- 9: For output layer L , the output of reconstructing is obtained by $\hat{R}_u^{(1)} = g(W_L h_L + b_{D_u})$, and $\hat{D}_u^{(1)} = g(S_L h_L + b_{D_u})$
- 10: **end if**
- 11: **end for**
- 12: **end for**
- 13: **end while**
- 14: Obtaining the output of Pre-training $loss(R, D, \hat{R}_u^{(1)}, \hat{D}_u^{(1)})$, and generating soft targets $Z_u = W_L h_L + b_{D_u}$.
- 15: **–Step2:** Knowledge distillation layer
- 16: Assisting hard label to train by obtaining information from user labels (soft targets), i.e., $Q_u = \frac{\exp(\frac{Z_u}{t})}{\sum_u \exp(\frac{Z_u}{t})}$
- 17: **–Step3:** Re-training
- 18: **while** not converge **do**
- 19: **for** u **do**
- 20: Mapping soft targets to the low-dimensional hidden layer by $H_u = g(W_1^T Q_u + b_1)$.
- 21: Learning implicit information from low-dimensional hidden layer, i.e., $\hat{R}_u^{(2)} = g(W_2^T H_u + b_2)$ and $\hat{D}_u^{(2)} = g(W_3^T Q_u + b_3)$
- 22: **end for**
- 23: **end while**
- 24: Obtaining the output of Re-training $loss(R, D, \hat{R}_u^{(2)}, \hat{D}_u^{(2)})$.
- 25: **–Step4:** Predicting final results by combining Pre-training and Re-training.
- 26: Combining Pre-training and Re-training to obtain the total loss function $loss(R, D, \hat{R}_u^{(1)}, \hat{D}_u^{(1)}, \hat{R}_u^{(2)}, \hat{D}_u^{(2)})$.
- 27: Making final recommendations for users by \hat{R}_u

Algorithm 1 EAF-SR.

4.1 Description of datasets

To validate the performance of the proposed approach, three real-world datasets related to social CF are adopted and they are taken from popular social networking websites, including

Table 2 The statistics of three real-world datasets

Data sets	Flixster	Epinions	Douban
Number of users	147,612	49,290	129,490
Number of items	48,794	139,738	58,541
Number of ratings	8,196,077	664,823	16,830,839
Scale of ratings	[1.0, 5.0]	[1.0, 5.0]	[1.0, 5.0]
Rating sparsity	0.0117%	0.011846%	0.222%
Trust relations	2,538,746	487,181	1,692,952

Flixster,¹ Epinions² and Douban,³ which permit users to present their interests by posting feedback and ratings for all items. Furthermore, these datasets have different rating sparsity and social information, in which their statistics are presented in Table 2 and their details are as follows:

Flixster is an American social-networking movie website for discovering new movies, learning about movies, and meeting others with similar tastes in movies. The platform helps users to watch movie trailers as well as read about new and upcoming movies at the box office. Its dataset contains 1,049,511 users, 66,726 items, 8,196,077 ratings and 11,794,648 social relations. However, most of users have not rated any items. Therefore, those users or items with no ratings that are meaningless to experimental evaluation are removed. Finally, this dataset is collected, including 147,612 users, 48,794 items, 8,196,077 ratings and 2,538,746 social relations.

Epinions is a well-known website for product review established in 1999 [29]. Users can rate products from 1 to 5 and submit their personal reviews. These ratings and feedback will affect other consumers as they determine whether to purchase same product. In addition, users are also able to specify who to trust and create a social trust network. In social trust network, user feedback and ratings can be reliably found to be important by their trustees. Finally, this dataset contains 49,289 users, 139,738 items, 664,823 ratings and 487,183 trust relations.

Douban is a Chinese social networking service website that allows registered users to record information and create content related to film, books, music, recent events, and activities. For registered users, the website recommends potentially interesting books, movies, and music to them in addition to serving as a social network website such as WeChat. Users can assign 5-scale integer ratings (from 1 to 5) to movies, books and music. Finally, this dataset contains 129,490 users, 58,541 items, 16,830,839 ratings and 1,692,952 friend relations.

In the experiments, these datasets are divided into two disjoint user sets: a training set and a test set. Among them, the test set is constructed by randomly selecting 1000 users with at least 500 ratings and 20 social relationships, and the remaining users and their ratings are kept in the training set.

¹www.flixster.com

²www.epinions.com

³www.douban.com

4.2 Evaluation metrics

In recommendation system, a lot of previous studies usually use the Mean Absolute Error (MAE) and the Root Mean Squared Error (RMSE) to evaluate the performance of the model. However, many users only pay attention to the list of some items of their interests rather than all items in real life, which means that MAE and RMSE cannot accurately reflect the performance of the recommended model. Thence, in this paper, the Mean Average Precision (MAP) and Normalized Discounted Cumulative Gain@N (NDCG@N) are adopted to evaluate the performance of the proposed method, where the larger the values of MAP and NDCG@N, the more the recommendation accuracy and the higher the recommendation quality. The metrics of MAP and NDCG@N are respectively defined as

$$\begin{cases} AP_u = \frac{1}{I^u} \sum_{i \in I^u} \frac{\delta(rank_{uj} < rank_{ui}) + 1}{rank_{ui}} \\ MAP = \frac{\sum_{u=1}^{|U^{te}|} AP_u}{|U^{te}|} \end{cases} \tag{13}$$

and

$$\begin{cases} DCG_u = \sum_{i=1}^N \frac{2^{rel_i} - 1}{\log_2(i + 1)} \\ IDCG_u = \sum_{i=1}^N \frac{1}{\log_2(i + 1)} \\ NDCG_u@N = \frac{DCG_u@N}{IDCG_u} \\ NDCG@N = \frac{\sum_{u \in U^{te}} NDCG_u@N}{|U^{te}|} \end{cases} \tag{14}$$

where I^u denotes that the correct set is recommended in the recommendation list of user u , $rank_{ui}$ indicates the ranking position of item i in the recommendation list of user u , and U^{te} represents all users in the test set. Moreover, rel_i indicates the relevance of the recommendation result at position i , which means that if the item at top N is adopted, its value is 1, otherwise, it is 0.

4.3 Comparison

To evaluate the performance of the proposed method, the following ones are selected as competitors:

- **BPR** [33]: This is a sorting recommendation algorithm that sorts items according to the users’ interest in items and then selects the items with the highest priority to recommend to users, where $k = 128$ and $\lambda = 0.01$.
- **PRFM** [32]: This is a Ranking Factorization Machine (Ranking FM) model, which applies the Factorization Machine model to microblog ranking on basis of pairwise classification, where $k = 3$, $\lambda_w = 10^{-6}$ and $\lambda_v = 10^{-4}$.
- **trustMF** [50]: This is a matrix factorization method based on either rating data or trust data, where $\lambda = 0.001$ and $\beta_1 = \beta_2 = 0.5$.

- **aSDAE** [2]: This is a Top-N recommendation model by using the side information of the user based on the reconstruction function of the stacked denoising autoencoder, where $k = 50$.
- **FunkR-pDAE** [53]: This is a funk singular value decomposition recommendation method by using pearson correlation coefficient and Deep Auto-Encoders, where $\alpha = 0.5$, $\beta = 0.5$ and $\gamma = 1$.
- **DVMF** [39]: This is a deep learning based fully Bayesian treatment recommendation framework by integrating users' information, where $\varphi = 0.5$ and $\lambda = 0.001$.

4.3.1 Comparison results

Table 3 shows the results of experiments comparing the proposed EAF-SR with other latest methods. It can be seen that the proposed EAF-SR can obtain 7.59%, 6.56%, 5.61%, 1.95%, 1.20% and 0.99% relative best improvement in the term of MAP@10 in the Flixster dataset compared with BPR, PRFM, trustMF, aSDAE, FunkR-pDAE and DVMF. And it also obtains 5.50%, 3.89%, 1.61%, 1.18%, 0.79% and 0.54% relative best improvement in the term of MAP@10 in the Epinions dataset compared with those approaches. Last but not least, EAF-SR obtains 8.42%, 7.60%, 4.61%, 3.21%, 2.76% and 1.94% relative best improvement in the term of MAP@10 in the Douban dataset compared with those approaches. Moreover, EAF-SR obtains 5.86%, 5.44%, 3.17%, 2.63%, 2.17% and 1.99% relative best improvement in the term of NDCG@10 in the Flixster dataset compared with those approaches. And EAF-SR obtains 6.99%, 5.84%, 0.58%, 0.42%, 0.28% and 0.18%

Table 3 The comparison results of seven approaches in three different datasets

Data sets	Evaluation index	BPR	PRFM	trustMF	aSDAE	FunkR-pDAE	DVMF	EAF-SR
Flixster	MAP@1	0.7752	0.7836	0.7934	0.8203	0.8459	0.8603	0.8772
	MAP@5	0.8061	0.8255	0.8327	0.8398	0.8669	0.8701	0.8878
	MAP@10	0.8366	0.8447	0.8523	0.8829	0.8894	0.8913	0.9001
	Improve	7.59%	6.56%	5.61%	1.95%	1.20%	0.99%	
	NDCG@10	0.8768	0.8803	0.8997	0.9044	0.9085	0.9101	0.9282
	Improve	5.86%	5.44%	3.17%	2.63%	2.17%	1.99%	
Epinions	MAP@1	0.7983	0.8299	0.8417	0.8769	0.8803	0.8891	0.8924
	MAP@5	0.8341	0.8458	0.8803	0.8990	0.8996	0.9002	0.9054
	MAP@10	0.8633	0.8767	0.8964	0.9002	0.9037	0.9059	0.9108
	Improve	5.50%	3.89%	1.61%	1.18%	0.79%	0.54%	
	NDCG@10	0.8809	0.8905	0.9371	0.9386	0.9399	0.9408	0.9425
	Improve	6.99%	5.84%	0.58%	0.42%	0.28%	0.18%	
Douban	MAP@1	0.8004	0.8165	0.8279	0.8479	0.8538	0.8683	0.8710
	MAP@5	0.8368	0.8399	0.8425	0.8655	0.8794	0.8802	0.8957
	MAP@10	0.8542	0.8607	0.8853	0.8973	0.9012	0.9085	0.9261
	Improve	8.42%	7.60%	4.61%	3.21%	2.76%	1.94%	
	NDCG@10	0.8835	0.9097	0.9388	0.9417	0.9492	0.9532	0.9655
	Improve	9.28%	6.13%	2.84%	2.53%	1.71%	1.29%	

The bold denotes the best result

relative best improvement in the term of NDCG@10 in the Epinions dataset compared with those approaches. In addition, EAF-SR obtains 9.28%, 6.13%, 2.84%, 2.53%, 1.71% and 1.29% relative best improvement in the term of NDCG@10 in the Douban dataset compared with those approaches. Furthermore, it can also be observed that as N in top- N increases, the performance of the methods also increases (see MAP@1, MAP@5 and MAP@10).

In a word, as the increase of MAP@10 and NDCG@10 represents the improvement of recommendation accuracy and quality, experimental results show that EAF-SR is more powerful than the BPR, PRFM, trustMF, aSDAE, FunkR-pDAE and DVMF.

4.3.2 Comparisons of running time

In order to assess the computational efficiency of the model, many experiments are conducted on a workstation with three NVIDIA GeForce RTX 2080Ti GPU and 10GB memory to compare the running time of the proposed EAF-SR with other methods (e.g., BPR, PRFM, trustMF, aSDAE, FunkR-pDAE, and DVMF). For fair comparisons, all the experiments of these methods are performed in 100 iterations. In the actual situation, it is known to all that the complexity of using CPU to run the EAF-SR model is lower than using the GPU. However, parallel computing on the CPU to train deep neural networks is not efficient. Therefore, all experiments are performed in parallel on the GPU.

As shown in Table 4, with the GPU capacity, the running time of EAF-SR is less than BPR, PRFM, trustMF, aSDAE, FunkR-pDAE, and DVMF on all datasets, which shows that the proposed EAF-SR is better in terms of computational efficiency in the recommendation system.

4.4 Parameters analysis

From the objective function (10), it can be seen that the values of five parameters need to be determined, i.e., t , k , α , λ , and L . Thus, this section will discuss their effects on the performance of the proposed EAF-SR. Herein, to reduce the complexity of the experiment, the terms of MAP@10 and NDCG@10 are used to evaluate its performance.

4.4.1 The impact of parameter t

In the proposed EAF-SR, the parameter t denotes a temperature to regulate the output values of the distillation layer. To study its effects, it is set to {1, 1.5, 2, 2.5, 3, 3.5, 4, 4.5 and 5}. In other words, the experiments are conducted to test the influence of difference t on the proposed EAF-SR. It can be observed from Fig. 5 that the values of MAP@10 and NDCG@10 rise fastest with the increase of the temperature t at the beginning of the experiment. When t is 3, their values are the biggest and the recommended result is the best. The results indicate that a larger or smaller value of temperature t can affect the performance of knowledge distillation.

Table 4 The comparisons of running time (s) of seven approaches with $k = 80$ in three different datasets

Datasets	BPR	PRFM	trustMF	aSDAE	FunkR-pDAE	DVMF	EAF-SR
Flixster	4,238	4,247	4,261	4,233	4,208	4,195	4,122
Epinions	2,242	2,276	2,281	2,235	2,226	2,187	2,158
Douban	4,156	4,189	4,194	4,106	4,098	4,069	4,027

The bold denotes the best result

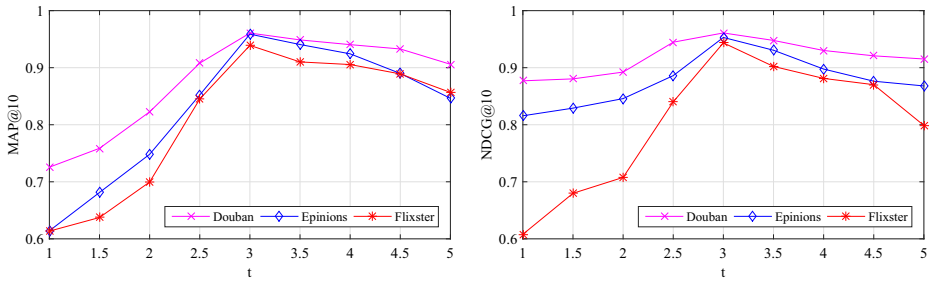


Fig. 5 The performance of the proposed EAF-SR with different t values, where the best result is achieved at $t = 3$

4.4.2 The impact of parameter k

In order to discuss the effect of dimensionality k , numerous experiments are carried out on the EAF-SR model with different values of dimensionality k in $\{10, 20, 30, 40, 50, 60, 70, 80, 90$ and $100\}$. And the grid-search approach is used to find the best combination values. Figure 6 illustrates that with the increase of dimensionality k , the values of MAP@10 and NDCG@10 gradually increase at first. However, when the dimensionality k exceeds a threshold (about 80 for Flixster, Epinions, and Douban datasets), the values of MAP@10 and NDCG@10 are decreasing. For the results of the experiments, there are two explanations for this observation:

- 1) A relatively larger dimension is contributing to better performance.
- 2) When the dimensionality reaches a certain threshold, it can trigger the problem of over-fitting, which turns out to degrade the accuracy of the prediction.

4.4.3 The impact of parameter α

The parameter α is adopted to balance the importance of pre-training and re-training networks in Formula (10). A larger value of α indicates that the retraining network has more impact. An extremely small values of α makes the proposed EAF-SR model focus on the soft target generation network. To study the influence of parameter α , it is set to $\{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9\}$. Figure 7 shows the result of experiment of the proposed

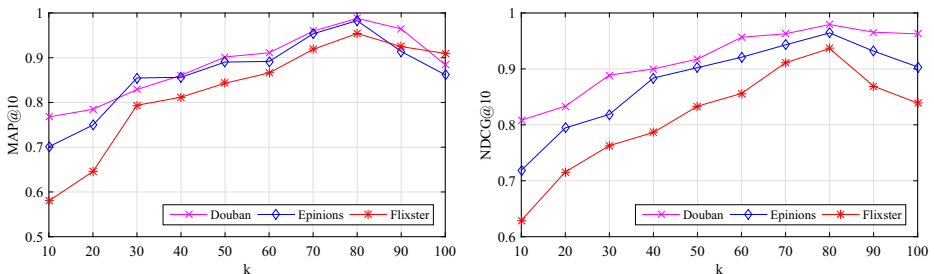


Fig. 6 The performance of the proposed EAF-SR with different k values, where the best result is achieved at $k = 80$

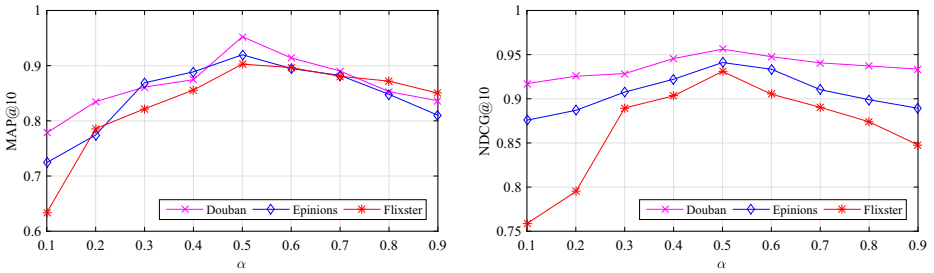


Fig. 7 The performance of the proposed EAF-SR with different α values, where the best result is achieved at $\alpha = 0.5$

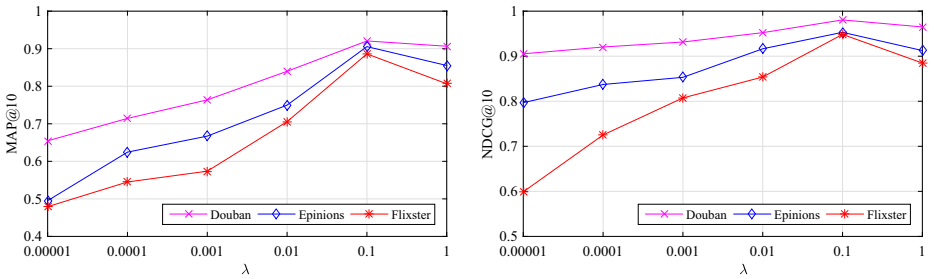


Fig. 8 The performance of the proposed EAF-SR with different λ values, where the best result is achieved at $\lambda = 0.1$

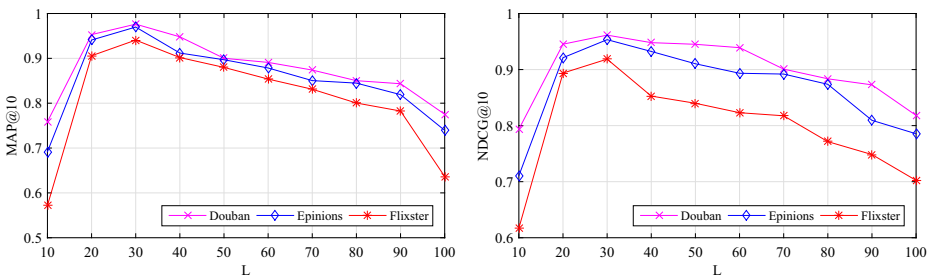


Fig. 9 The performance of the proposed EAF-SR with different layers L , where the best result is achieved at $L = 30$

Table 5 The best values of parameters t , k , α and λ in three data sets

Parameters	t	k	α	λ	L
Values	3	80	0.5	0.1	30

method with diverse values of α . It can be observed that with the increase of α , the values of MAP@10 and NDCG@10 dramatically increase. In other words, with the values of α obtaining larger, the performances of EAF-SR go better to a certain point (around 0.5 for Flixster, Epinions, and Douban datasets). However, when α surpasses 0.5, the values of MAP@10 and NDCG@10 are decreasing. Therefore, the experimental results indicate that the recommended result of the proposed EAF-SR is the best when $\alpha = 0.5$.

4.4.4 The impact of parameter λ

The parameter λ is a hyper-parameter to prevent the over-fitting. To study its impact on the the performances of the proposed EAF-SR, it is set to {0.00001, 0.0001, 0.001, 0.01, 0.1 and 1}. By using the grid-search approach, the values of λ are obtained. From Fig. 8, it can be seen that when $\lambda = 0.1$, the values of MAP@10 and NDCG@10 are the largest. In other word, the optimal value of λ is 0.1.

4.4.5 The impact of parameter L

The parameter L denotes the number of layers of the proposed EAF-SR. In order to discuss its effect, a series of experiments are conducted on the EAF-SR model with different values of L in {10, 20, 30, 40, 50, 60, 70, 80, 90 and 100}. Figure 9 illustrates that with the number of layers L increases, the values of MAP@10 and NDCG@10 gradually increase at first. However, when it exceeds a threshold (about 30 for Flixster, Epinions, and Douban datasets), the values of MAP@10 and NDCG@10 are decreasing, i.e., the optimal value of the number of layers L is 30.

In summary, the optimal values of parameters t , k , α , λ and L are listed as Table 5.

4.5 Ablation studies on epinions dataset

For a better understanding of the proposed EAF-SR, the effects of Pre-training and Re-training are first investigated on Epinions dataset. As shown in Table 6, when only Pre-training is plugged into the network, the MAP@10 and NDCG@10 are 0.7584 and 0.8359, respectively. Likewise, Re-training is simply added to the model, and the MAP@10 and

Table 6 Ablation results of the proposed EAF-SR

Number	Re-training	Knowledge distillation	Re-training	Evaluation metrics	
				MAP@10	NDCG@10
No.1	✓			0.7584	0.8359
No.2			✓	0.7003	0.8231
No.3	✓	✓	✓	0.9108	0.9425

The bold denotes the best result

NDCG@10 are 0.7003 and 0.8231. Finally, both Pre-training and Re-training are plugged into the network, EAF-SR further improves the MAP@10 and NDCG@10 to 0.9108 and 0.9425.

5 Conclusions

In this paper, an Enhanced Social Recommendation Algorithm (EAF-SR) is proposed to address the issues, such as cold-start, data noise and data sparsity. In the EAF-SR framework, the issue of learning hidden features from the soft target which is produced by the pre-training network is also investigated. Specially, a tightly coupled system is proposed to integrate the pre-training and re-training phases into a unified framework. By this way, the model can dynamically update the soft target by adjusting the training errors of the pre-training and retraining networks in real time, which is to reduce the influence of noise and preserve knowledge. Furthermore, a new knowledge distillation layer is designed to regulate the outputs of pre-training model for dynamic training based on the reliability of each output unit. Then, a new measurement approach is proposed to calculate the reliability of the output unit based on the number of corresponding positive markers. Finally, the recommendation is made by combining the predictions of pre-training and re-training networks.

Acknowledgements This work was supported in part by Guangdong Key Laboratory of Electromagnetic Control and Intelligent Robots; in part by the National Natural Science Foundation of China under Grants U1813212, 62273105, U1911401, and 61727810; in part by the Ten Thousand Talent Program approved in 2018; in part by the Guangdong Province Foundation under Grants 2019B1515120036, and 501200069; in part by Science and Technology Planning Project of Guangdong Province under Grant 2020B121201012, and in part by Shenzhen Science and Technology Planning Project under Grant JSGG20210802153155018.

Data Availability All data in the manuscript come from the public websites, which can be found at: <https://www.flixster.com>, <https://www.epinions.com>, and <https://www.douban.com>. Furthermore, all other data (i.e., processed data) are available from the authors upon reasonable request.

Declarations

Conflict of Interests All authors certify that they have no affiliations with or involvement in any organization or entity with any financial interest or non-financial interest in the subject matter or materials discussed in this manuscript.

References

1. Anil D, Vembar A, Hiriyanaiyah S, Siddesh GM, Srinivasa KG (2018) Performance analysis of deep learning architectures for recommendation systems. In: Proceedings of the 25th IEEE International Conference on High Performance Computing Workshops (HiPCW), pp 129–136
2. Bao R, Sun Y (2019) Top-N recommendation model based on SDAE. *J Phys: Conf Ser* 1168(5):52036–52045
3. Bobadilla J, Ortega F, Hernando A, Gutiérrez A (2013) Recommender systems survey. *Knowl-Based Syst* 46:109–132
4. Bottou L (2012) Stochastic gradient descent tricks. In: Proceedings of the neural networks: tricks of the trade, pp 421–436
5. Cui Q, Wu S, Liu Q, Zhong W, Wang L (2020) MV-RNN: a multi-view recurrent neural network for sequential recommendation. *IEEE Trans Knowl Data Eng* 32(2):317–331
6. Da' u A, Salim N, Rabi' u I, Osman A (2020) Recommendation system exploiting aspect-based opinion mining with deep learning method. *Inf Sci* 512:1279–1292
7. Deng S, Huang L, Xu G, Wu X, Wu Z (2016) On deep learning for trust-aware recommendations in social networks. *IEEE Trans Neur Netw Learn Syst* 28(5):1164–1177

8. Dighe P, Asaai A, Bourlard H (2018) Far-field ASR using low-rank and sparse soft targets from parallel data. In: Proceedings of the IEEE Spoken Language Technology Workshop (SLT), pp 581–587
9. Eirinaki M, Louta MD, Varlamis I (2013) A trust-aware system for personalized user recommendations in social networks. *IEEE Trans Syst Man Cybern: Syst* 44(4):409–421
10. Fayek HM, Lech M, Cavedon L (2017) Evaluating deep learning architectures for speech emotion recognition. *Neural Netw* 92:60–68
11. Fu M, Qu H, Yi Z, Lu L, Liu Y (2018) A novel deep learning-based collaborative filtering model for recommendation system. *IEEE Trans Cybern* 49(3):1084–1096
12. 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. In: Proceedings of the AAAI conference on artificial intelligence, vol 29, pp 123–129
13. Hinton G, Vinyals O, Dean J (2015) Distilling the knowledge in a neural network. *Comput Sci* 14(7):38–39
14. Jain A, Nagar S, Singh PK, Dhar J (2020) EMUCF: enhanced multistage user-based collaborative filtering through non-linear similarity for recommendation systems. *Expert Syst Appl* 161:113724
15. Jamali M, Ester M (2010) A matrix factorization technique with trust propagation for recommendation in social networks. In: Proceedings of the 4th ACM conference on recommender systems, pp 135–142
16. Ji Z, Pi H, Wei W, Xiong B, Woźniak M, Damasevicius R (2019) Recommendation based on review texts and social communities: a hybrid model. *IEEE Access* 7:40416–40427
17. Koren Y (2010) Factor in the neighbors: scalable and accurate collaborative filtering. *ACM Trans Knowl Discov Data (TKDD)* 4(1):1–24
18. Koren Y, Rendle S, Bell R (2022) Advances in collaborative filtering. *Recommender Systems Handbook*, 91–142
19. 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
20. 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
21. Li J, Chen C, Chen H, Tong C (2017) Towards context-aware social recommendation via individual trust. *Knowl-Based Syst* 127:58–66
22. Li H, Li K, An J, Zheng W, Li K (2019) An efficient manifold regularized sparse non-negative matrix factorization model for large-scale recommender systems on GPUs. *Inf Sci* 496:464–484
23. Liang D, Krishnan RG, Hoffman MD, Jebara T (2018) Variational autoencoders for collaborative filtering. In: Proceedings of the 27th world wide web conference, pp 689–698
24. Liu Z, Lin Y, Sun M (2020) Representation learning for natural language processing. Springer Nature, 1–334
25. Luo X, Zhou M, Xia Y, Zhu Q (2014) An efficient non-negative matrix-factorization-based approach to collaborative filtering for recommender systems. *IEEE Trans Industr Inf* 10(2):1273–1284
26. Ma H, Zhou D, Liu C, Lyu MR, King I (2011) Recommender systems with social regularization. In: Proceedings of the 4th ACM international conference on web search and data mining, pp 287–296
27. Magassouba A, Sugiura K, Quoc AT, Kawai H (2019) Understanding natural language instructions for fetching daily objects using GAN-based multimodal target–source classification. *IEEE Robot Autom Lett* 4(4):3884–3891
28. Maier A, Syben C, Lasser T, Riess C (2019) A gentle introduction to deep learning in medical image processing. *Z Med Phys* 29(2):86–101
29. Massa P, Avesani P (2004) Trust-aware collaborative filtering for recommender systems. In: Proceedings of the move to meaningful internet systems: CoopIS, DOA, and ODBASE, OTM confederated international conferences, pp 492–508
30. Qian X, Feng H, Zhao G, Mei T (2013) Personalized recommendation combining user interest and social circle. *IEEE Trans Knowl Data Eng* 26(7):1763–1777
31. Qian F, Zhao S, Tang J, Zhang Y (2016) SoRS: social recommendation using global rating reputation and local rating similarity. *Physica A* 461:61–72
32. Qiang R, Liang F, Yang J (2013) Exploiting ranking factorization machines for microblog retrieval. In: Proceedings of the 22nd ACM international conference on information & knowledge management, pp 1783–1788
33. Rendle S, Freudenthaler C, Gantner Z, Schmidt-Thieme L (2009) BPR: bayesian personalized ranking from implicit feedback. In: Proceedings of the 25th conference on uncertainty in artificial intelligence, Montreal, pp 452–461
34. Roy PK, Chahar S (2020) Fake profile detection on social networking websites: a comprehensive review. *IEEE Trans Artif Intell* 1(3):271–285

35. Roy PK, Bhawal S, Subalalitha CN (2022) Hate speech and offensive language detection in dravidian languages using deep ensemble framework. *Comput Speech Lang* 75:101386
36. Salakhutdinov R, Mnih A, Hinton G (2007) Restricted boltzmann machines for collaborative filtering. In: *Proceedings of the 24th international conference on machine learning*, pp 791–798
37. 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
38. Shamshoddin S, Khader J, Gani S (2020) Predicting consumer preferences in electronic market based on IoT and social networks using deep learning based collaborative filtering techniques. *Electron Commer Res* 20(2):241–258
39. Shen X, Yi B, Liu H, Zhang W, Zhang Z, Liu S, Xiong N (2021) Deep variational matrix factorization with knowledge embedding for recommendation system. *IEEE Trans Knowl Data Eng* 33(5):1906–1918
40. Tang J, Hu X, Liu H (2013) Social recommendation: a review. *Soc Netw Anal Min* 3(4):1113–1133
41. Tuzhilin A (2010) Towards the next generation of recommender systems. In: *Proceedings of the 1st International Conference on E-Business Intelligence (ICEBI2010)*, pp 553–557
42. Udendhran R, Balamurugan M, Suresh A, Varatharajan R (2020) Enhancing image processing architecture using deep learning for embedded vision systems. *Microprocess Microsyst*, 103094–103101
43. Wang X, He X, Nie L, Chua T (2017) Item silk road: recommending items from information domains to social users. In: *Proceedings of the 40th international ACM SIGIR conference on research and development in information retrieval*, pp 185–194
44. Wang X, Yang X, Guo L, Han Y, Liu F, Gao B (2019) Exploiting social review-enhanced convolutional matrix factorization for social recommendation. *IEEE Access* 7:82826–82837
45. Wei J, He J, Chen K, Zhou Y, Tang Z (2017) Collaborative filtering and deep learning based recommendation system for cold start items. *Expert Syst Appl* 69:29–39
46. Wilson BS, Finley CC, Lawson DT, Wolford RD, Eddington DK, Rabinowitz WM (1991) Better speech recognition with cochlear implants. *Nature* 352(6332):236–238
47. Wu Y, DuBois C, Zheng AX, Ester M (2016) Collaborative denoising auto-encoders for top-N recommender systems. In: *Proceedings of the 9th ACM international conference on web search and data mining*, pp 153–162
48. Wu L, Sun P, Hong R, Ge Y, Wang M (2021) Collaborative neural social recommendation. *IEEE Trans Syst Man Cybern: Syst* 51(1):464–476
49. Wu X, Sahoo D, Hoi StevenCH (2020) Recent advances in deep learning for object detection. *Neurocomputing* 396:39–64
50. Yang B, Lei Y, Liu J, Li W (2016) Social collaborative filtering by trust. *IEEE Trans Pattern Anal Mach Intell* 39(8):1633–1647
51. Zhang S, Yao L, Xu X (2017) AutoSVD++ an efficient hybrid collaborative filtering model via contractive auto-encoders. In: *Proceedings of the 40th international ACM SIGIR conference on research and development in information retrieval*, pp 957–960
52. Zhang C, Wang Y, Zhu L, Song J, Yin H (2021) Multi-graph heterogeneous interaction fusion for social recommendation. *ACM Trans Inform Syst* 40(2):1–26
53. Zhang P, Xiong F, Leung HaretonKN, Song W (2021) FunkR-pDAE: personalized project recommendation using deep learning. *IEEE Trans Emerg Top Comput* 9(2):886–900
54. Zhao T, McAuley J, King I (2014) Leveraging social connections to improve personalized ranking for collaborative filtering. In: *Proceedings of the 23rd ACM international conference on conference on information and knowledge management*, pp 261–270
55. Zhao Z, Lu H, Cai D, He X, Zhuang Y (2016) User preference learning for online social recommendation. *IEEE Trans Knowl Data Eng* 28(9):2522–2534
56. Zhao H, Sun X, Dong J, Chen C, Dong Z (2020) Highlight every step: knowledge distillation via collaborative teaching. *IEEE Transactions on Cybernetics*, <https://doi.org/10.1109/TCYB.2020.3007506>

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

Springer Nature or its licensor holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.