ExpoEv: Enhancing Social Recommendation Service with Social Exposure and Feature Evolution

Li Ma[1], Zuowu Zheng[1], Xiuqi Huang[1], Zhaoxiang Zhang[1], Xiaofeng Gao[1,*], Jianxiong Guo[2], and Guihai Chen[1]
[1] MoE Key Lab of Artificial Intelligence, Department of Computer Science and Engineering,
Shanghai Jiao Tong University, Shanghai, China
[2] Advanced Institute of Natural Sciences, Beijing Normal University, Zhuhai, China
{mali-cs, waydrow, huangxiuqi, zzx-sjtu}@sjtu.edu.cn, gao-xf@cs.sjtu.edu.cn, jianxiongguo@bnu.edu.cn, gchen@cs.sjtu.edu.cn

***Abstract*—Social networks are widely recognized as highly effective information sources for social recommendation services. However, previous social recommendation methods assumed that a user's preference factor and social trust factor shared a common latent feature space. Additionally, few studies have explored the incorporation of social information into the item domain for recommendations. To address these gaps, we propose ExpoEv, a deep collaborative filtering recommendation model that integrates social exposure based on feature evolution for social recommendation services. Specifically, we propose a social exposure module for both user and item domains that considers the number of items that a user's social friends interact with. Furthermore, we introduce a feature evolution component that enables the incorporation of social exposure information with social trust and attribute factors in the context of social recommendation services. Experiments demonstrate the effectiveness of our model in the quality of recommendation service.**

***Index Terms*—Social Recommendation Service, Collaborative Filtering, Social Exposure, Feature Evolution, Quality of Service**

I. Introduction

The recommendation system filters information for users by predicting their preferences and generates personalized recommendations. Traditional recommendation systems employ Collaborative Filtering (CF) methods to predict what current users are most interested in based on previous user behaviors. With the rise of social service platforms such as Facebook and Twitter, people can communicate with each other more conveniently, which brings a great deal of social information. Driven by the knowledge that human beings usually acquire and disseminate information through their acquaintances, researchers try to combine conventional recommendation methods with social networks to improve the quality of recommendation services [20], [26].

However, there are some limitations to the existing methods. We summarize them as three points. First, existing methods assume that user preferences and social trust share a common latent space and use the same latent vector to represent both of them. While user preference latent features are strongly related to ratings and social networks, social trust latent features are more strongly associated with social networks and have a relatively weaker relationship with ratings. As a result, user preference and social trust latent features should be represented in separate domains. Second, these methods assume that a user and their friends share similar preferences [8], [9] but overlook the specific relationships between them. Friends may have entirely different opinions on a particular item due to factors such as gender, age, or location. Nevertheless, even when a user holds different views from their friends, they can still interact with them. For example, a user may keep track of the items consumed by their friends. Sometimes, mere exposure can also draw a user's attention. Therefore, an social exposure domain should be considered. Third, item attributes are treated as static and independent latent factors in current models. However, social networks are dynamic in reality, and the influence of friends varies depending on the item and situation. Previously, there are merely appropriate methods to directly combine the dynamic interaction latent factors of an item with social networks. These limitations prevent researchers from accurately capturing the latent features of social networks and the rating matrix, thereby reducing the quality of recommendation services.

To effectively address the aforementioned limitations, we propose ExpoEv, a deep collaborative filtering model, that incorporates social Exposure based on feature Evolution for social recommendation service. In ExpoEv, we define an exposure parameter for each interacted item of each user, and take into account the ratings of the user's friends on social networks. This parameter is modeled using user and item exposure latent factors. The exposure vector collaboratively works with traditional user embeddings to capture the diversity of user interests from the perspective of social networks. Moreover, we employ two distinct factors to describe the user's preferences and trust separately. To integrate preference, exposure, and trust information, we utilize a feature evolution unit for both the user and item domains. Vectors pass through a series of evolution units, merging each other's information based on a probability distribution. The advantages of the feature evolution unit include: (i) the ability to fuse more

This work was supported by the National Key R&D Program of China [2020YFB1707900], the National Natural Science Foundation of China [62272302, 62202055, 62172276], Shanghai Municipal Science and Technology Major Project [2021SHZDZX0102], and DiDi GAIA Research Collaboration Plan [202204]. *Xiaofeng Gao is the corresponding author.

979-8-3503-0485-5/23/$31.00 ©2023 IEEE
DOI 10.1109/ICWS60048.2023.00027

than two vectors, such as three vectors in the user domain in our model; (ii) the integration of preference, exposure, and trust information without compromising their individual features; and (iii) the use of a more general probability distribution, effectively avoiding local optima. Furthermore, using the aforementioned methods, we can directly combine item attributes with social information, enhancing the model's performance and applicability.

The contribution of the paper can be summarized as follows: (i) we introduce the concept of social exposure, which collaborates with user preference and trust information for social recommendation services; (ii) we propose an enhanced feature evolution method that fuses multiple factors, including latent factors in the user domains as well as those in the item domains, which can directly combine item attributes with social information; and (iii) we conduct comprehensive experiments on three public datasets, and the experimental results demonstrate that our model outperforms both trust-based and high-performing ratings-only baseline methods.

II. Related Works

Collaborative filtering is one of the most popular recommendation systems techniques. Early literature makes predictions for users by calculating the behavior similarity of users or items [1], [15], [16]. Later, matrix factorization based methods [7], [9], [10] are proposed to encode users and items into a joint space and allow incorporation of additional information. In recent years, some recent methods utilize deep learning [3], [5], [17] methods to learn high-level representations because of its effectiveness in feature extraction, and introduce the idea of graph neural network [13], [18], [21], [28] to model user-item interaction to learn user or item representations.

In light of the fact that humans typically acquire and disseminate information through their social networks, researchers are attempting to combine conventional recommendation methods and social networks to enhance the quality of recommendation services [26]. Guo et al. [4] employ matrix factorization and incorporate both implicit and explicit influence of trusted users based on [9]. Jamali et al. [8] focus on user's preference learning and assume that a user's preference is similar to his/her social friends. Li et al. [11] integrate social information and user interest in the process of searching for the nearest neighbor. The advantages of social information are leveraged in [14], [22] to learn accurate representations for users and items. Yu et al. [27] propose a multi-channel hyper-graph convolutional network to improve social recommendation by exploiting high-order user relations. Chen et al. [2] propose a method based on the user's periodic pairs of interest and graph structure to obtain as much effective information as possible to recommend items.

However, these methods fail to explicitly represent the social trust factor and user preference factor in separate domains, as well as to integrate social information in the item domain.

III. Proposed Model: ExpoEv

In this section, we will introduce the proposed ExpoEv to effectively tackle the limitations mentioned above. Before that, we first review the preliminaries of social recommendation services. Let $U = \{u_1, u_2, \cdots, u_m\}$ and $V = \{v_1, v_2, \cdots, v_n\}$ denote the set of users and items respectively. r_{ij} is the rating of u_i on v_j, ranging from 1 to 5. All the ratings are recorded in a rating matrix $R \in \mathbb{R}^{m\times n}$. $S \in \{0,1\}^{m\times m}$ is the social network between users, where $S_{ik} = 1$ means u_i and u_k are friends. Given the rating matrix and social network matrix, our goal is to predict the unobserved ratings of a user on an item. Fig. 1 shows the overall framework of ExpoEv model.

A. Embedding Layer

In the user domain, we encode each user as a preference factor, a trust factor, and an exposure factor. We first denote u_i as a one-hot vector $\mathbf{x}_i \in \mathbb{R}^{1\times m}$, where m indicates the number of users, and $\mathbf{x}_{ik} = 1$ and other elements are zero if u_i is the k-th user. The one-hot vector is fed into the embedding layer to learn a low-dimensional representation. The preference embedding of users is obtained by $\mathbf{p}_i = \mathbf{P}\mathbf{x}_i^T$, where $\mathbf{P} \in \mathbb{R}^{D\times m}$ is the preference embedding matrix and D is the dimension size. Similarly, the trust embedding can be obtained by $\mathbf{t}_i = \mathbf{T}\mathbf{x}_i^T$, where $\mathbf{T} \in \mathbb{R}^{D\times m}$ are the trust embedding matrix. In the item domain, we encode each item as an attribute factor and an exposure factor in the same way. Denote $\mathbf{y}_j \in \mathbb{R}^{1\times n}$ as the one-hot vector for item v_j, where n is the number of items. The attribute embedding is obtained by $\mathbf{q}_j = \mathbf{Q}\mathbf{y}_j^T$, where $\mathbf{Q} \in \mathbb{R}^{D\times n}$. For the representations of user exposure factor and item exposure factor, we give details in the following *Social Exposure Model* part.

B. Social Trust Model

Social network based recommendation methods assume that users have similar representations if they are friends. The social trust model is designed to capture the social trust factor between a user and his/her friends. To achieve this goal, we expect to learn a vector to measure the similarity between a user and his/her friends. We first concatenate the trust embeddings of two friends, and then feed them into a two-layer feed-forward network to obtain a predicted similarity score $\hat{S}_{ik}$. The formulation is as follows:

$$\hat{S}_{ik} = \mathbf{W}_2^n \tanh\left(\mathbf{W}_1^n \left[\mathbf{t}_i, \mathbf{t}_k\right] + \mathbf{b}_1^n\right) + \mathbf{b}_2^n, \tag{1}$$

where $\mathbf{W}_1^n, \mathbf{W}_2^n$ are weight matrices, and $\mathbf{b}_1^n, \mathbf{b}_2^n$ are biases.

C. Social Exposure Model

Even if a user has different preferences than his friends, he may still see from moments and blogs which items his friends interact with, which triggers exposure for users and contribute to attracting user's interests. [12] introduces a probability model to illustrate the exposure property. However, it ignores the utilization of the preference property.

The challenge is to measure the social exposure, denoted as μ_{ij}, which indicates the exposure probability of u_i on v_j. Intuitively, the greater the percentage of friends with whom

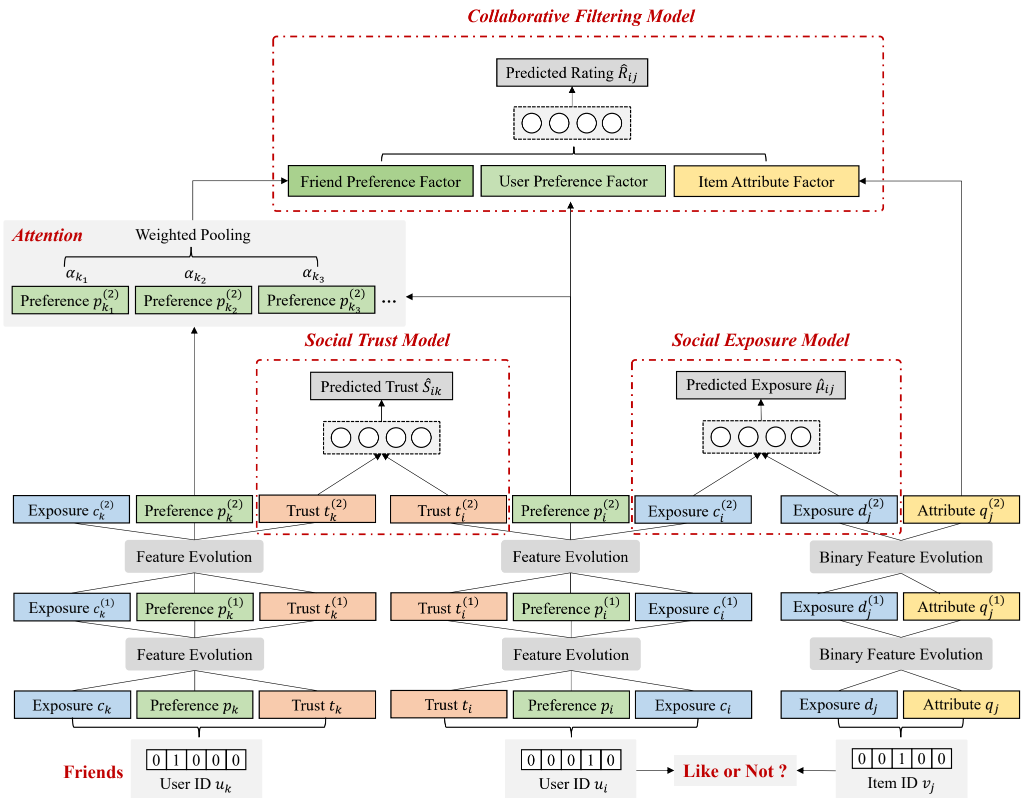


Fig. 1: The overall framework of ExpoEv.

a user has interacted with an item, the greater the likelihood that the user will see it. For example, u_i have 100 friends and 50 of them interact with v_j, so u_i may have the probability of $f_{ij} = \frac{1}{2}$ to see v_j. u_k have 30 friends and 20 of them interact with v_j, so $f_{kj} = \frac{2}{3}$. We can infer that v_j is more likely to be seen by u_k. However, we cannot define this likelihood in terms of quantity only, as the number of friends varies greatly from user to user. A user may have many friends who have not interacted with this item, and these friends can interfere with the user's judgment. The probability of the item being exposed to the user should increase exponentially with the percentage of friends interacting with the item. We propose the function to indicate the social exposure probability: $f_{ij}^2 + (\mu_{ij} - 1)^2 = 1$, where $f_{ij}, \mu_{ij} \in$ [0, 1]. We can conclude when $f_{ij} = 0$, $\mu_{ij} = 0$, when $f_{ij} = 1$, $\mu_{ij} = 1$. That is to say, if no friend of u_i interacts with v_j, then the social exposure probability of u_i on v_j is 0. On the contrary, if all of the friends of u_i interact with v_j, v_j definitely is exposed to u_i.

Based on the design above, we can obtain the social exposure probability of each user on the interacted item. This definition has two advantages: (i) the exposure probability is normalized, ranging from 0 to 1, which can prevent slow convergence or poor results due to excessive differences; (ii) the slope of the function continues to increase. When users see an item for the first time, they may not be impressed, but the impression will be deep after several exposures.

Consequently, the user exposure factor $\mathbf{c}_i$ and the item exposure factor $\mathbf{d}_j$ are defined as, $\mathbf{c}_i = \sum_{a_t \in N_{u_i}} a_t \cdot \mathbf{q_t}$ with $a_t = \mathbf{p_i} \cdot \mathbf{q_t}$, $\mathbf{d}_j = \sum_{b_t \in N_{v_j}} b_t \cdot \mathbf{p_t}$ with $b_t = \mathbf{q_j} \cdot \mathbf{p_t}$, where N_{u_i} indicates the items set that u_i have interacted, N_{v_j} represents the set of users who have interacted with v_j, and a_t, b_t means the aggregation weight for u_i and v_j, respectively.

Similar to the social trust model, we concatenate $\mathbf{c}_i$ and $\mathbf{d}_j$ and feed them into a two-layer feed-forward model to capture the social exposure probability between users and items. The formulation is defined as:

$$\hat{\mu}_{ij} = \mathbf{W}_2^e \tanh\left(\mathbf{W}_1^e \left[\mathbf{c}_i, \mathbf{d}_j\right] + \mathbf{b}_1^e\right) + \mathbf{b}_2^e, \tag{2}$$

where $\mathbf{W}_1^e, \mathbf{W}_2^e$ are weight matrices, and $\mathbf{b}_1^e, \mathbf{b}_2^e$ are biases.

D. Collaborative Filtering Model

CF model utilizes user preference factor and item attribute factor to predict the probability of user-item interaction. To incorporate more social information, we use attention mechanisms to aggregate the preference factors of a user's friends. For u_i, the friend's attentive preference factor is defined as $\mathbf{f}_i = \sigma\left(\sum_{u_k \in F_i} \alpha_{ik} \mathbf{W}_f \mathbf{p}_k + \mathbf{b}_f\right)$, where $\alpha_{ik} = \frac{\mathrm{attn}(\mathbf{W}_p \mathbf{p}_i, \mathbf{W}_p \mathbf{p}_k)}{\sum_{u_j \in F_i} \mathrm{attn}(\mathbf{W}_p \mathbf{p}_i, \mathbf{W}_p \mathbf{p}_j)}$ and $\mathrm{attn}(\mathbf{x}, \mathbf{y}) = LeakyReLU(\mathbf{w}_a(\mathbf{x} \| \mathbf{y}))$, F_i is the set of friends of u_i, σ is the activation function, $\mathbf{W}_f, \mathbf{W}_p$ are weight matrices, $\mathbf{b}_f$ is bias vector, and $\mathbf{w}_a$ is a weight vector.

Then we concatenate the user preference vector, item attribute vector, and friend's attentive preference vector, and feed them into a two-layer feed-forward network to obtain the predicted rating $\hat{R}_{ij}$ of u_i on item v_j, the calculation process is defined as follows:

$$\hat{R}_{ij} = \mathbf{W}_2^c \tanh\left(\mathbf{W}_1^c \left[\mathbf{f}_i, \mathbf{p}_i, \mathbf{q}_j\right] + \mathbf{b}_1^c\right) + \mathbf{b}_2^c, \tag{3}$$

where $\mathbf{W}_1^c, \mathbf{W}_2^c$ are weight matrices and $\mathbf{b}_1^c, \mathbf{b}_2^c$ are biases.

E. Loss Function

As we design above, we will obtain the prediction results for three tasks, and now we combine the three losses to calculate the final loss, the loss function is defined as follows:

$$L = \sum_{u_i \in U} \sum_{v_j \in V} \left(\hat{R}_{ij} - R_{ij}\right)^2 + \sum_{u_i \in U} \sum_{u_k \in F_i} \left(\hat{S}_{ik} - S_{ik}\right)^2 + \sum_{u_i \in U} \sum_{v_j \in V} \left(\hat{\mu}_{ij} - \mu_{ij}\right)^2 + L_{reg}, \tag{4}$$

$$L_{reg} = \lambda_C \left(\sum_{u_i \in U} \|\mathbf{p}_i\|_F + \sum_{v_j \in V} \|\mathbf{q}_j\|_F\right) + \lambda_S \sum_{u_i \in U} \|\mathbf{t}_i\|_F + \lambda_E \left(\sum_{u_i \in U} \|\mathbf{c}_i\|_F + \sum_{v_j \in V} \|\mathbf{d}_j\|_F\right), \tag{5}$$

where $\|\cdot\|_F$ denotes the Frobenius norm and λ_C, λ_S and λ_E are regularization parameters. We use stochastic gradient descendant algorithm to optimize the objective function.

F. Feature Evolution Unit

To fuse preference and trust information in the user domain, [23] propose an element-wise exchange method between vectors. However, this method could only fuse two vectors. In this paper, we generalize this method to make it feasible for three vectors and more.

In the user domain, our goal is to fuse the preference factor $\mathbf{p}_i$, the trust factor $\mathbf{t}_i$ and the exposure factor $\mathbf{c}_i$, and output the fused vectors as $\mathbf{p}_i^{(1)}$, $\mathbf{t}_i^{(1)}$ and $\mathbf{c}_i^{(1)}$ by the first feature evolution, thus each fused vector contains features from others. First, we generate a vector $\mathbf{e} \in \mathbb{R}^{D\times 1}$, each element $\mathbf{e}_k$ of $\mathbf{e}$ follows a multi-nomial distribution, and in a single test, there are 3 possible results, which are $-1, 0, 1$, and the probability of each result is $\frac{1}{3}$. Then, we construct three vectors $\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3 \in \mathbb{R}^{D\times 1}$. Specifically, we have the following definitions for $\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3$,

$$\mathbf{e}_{ik} = \begin{cases} 1, & \mathbf{e}_k = i - 2 \\ 0, & \text{otherwise.} \end{cases} \tag{6}$$

where $\mathbf{e}_{ik}$ represents the k-th element of $\mathbf{e}_i$.

Fig. 2 (b) displays the framework of feature evolution unit. Without loss of generality, let $\mathbf{x}_1$, $\mathbf{x}_2$, $\mathbf{x}_3$ denote $\mathbf{p}_i$, $\mathbf{t}_i$, and $\mathbf{c}_i$, as input vectors of the feature evolution unit, and $\mathbf{y}_1$ denotes the output vectors of the feature evolution unit for three input vectors. A new vector $\mathbf{w}_1$ is generated based on $\mathbf{e}_1$, $\mathbf{e}_2$ and $\mathbf{e}_3$, the calculation process is defined as $\mathbf{w}_1 = \mathbf{e}_1 \otimes \mathbf{x}_1 \oplus \mathbf{e}_2 \otimes \mathbf{x}_2 \oplus \mathbf{e}_3 \otimes \mathbf{x}_3$, where $\otimes$ denotes the element-wise product and $\oplus$ denotes the element-wise plus. Then, a fully connected layer is applied to obtain more condenser representation: $\mathbf{w}_1' = \tanh\left(\mathbf{W}_1^f \mathbf{w}_1 + \mathbf{b}_1^f\right)$, where $\mathbf{W}_1^f$ is the weight matrix and $\mathbf{b}_1^f$ is the bias vector. Finally, we use the linear combination of the original vector $\mathbf{x}_1$ and the feature evolution vector $\mathbf{w}_1'$ to obtain the output vector $\mathbf{y}_1$: $\mathbf{y}_1 = \alpha \cdot \mathbf{x}_1 + (1-\alpha)\cdot \mathbf{w}_1'$. Similarly, we can obtain $\mathbf{y}_2$ and $\mathbf{y}_3$ based on the operations mentioned above, as shown in Fig. 2 (a).

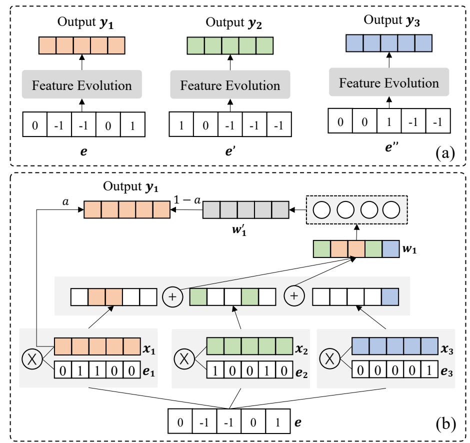


Fig. 2: Framework of feature evolution unit for user domain.

In the item domain, we use the same method as the feature evolution unit in [23], which is designed to fuse two input vectors and could be regarded as a simplification of feature evolution unit. Specifically, each element $\mathbf{e}_k$ of $\mathbf{e}$ follows a Bernoulli distribution. The aggregation vector is calculated by $\mathbf{w}_2 = (\mathbf{1} - \mathbf{e}_1) \otimes \mathbf{x}_1 \oplus \mathbf{e}_1 \otimes \mathbf{x}_2$.

The design of feature evolution unit is inspired by independent assortment in biogenetics, where offspring will inherit a subset of their parents' genetic feature traits. With this approach, vectors can exchange some of their elements, carry information about each other from different vectors, and combine various latent vectors to describe users or items. The social trust model and social exposure model help to integrate trust and exposure information into recommendation services.

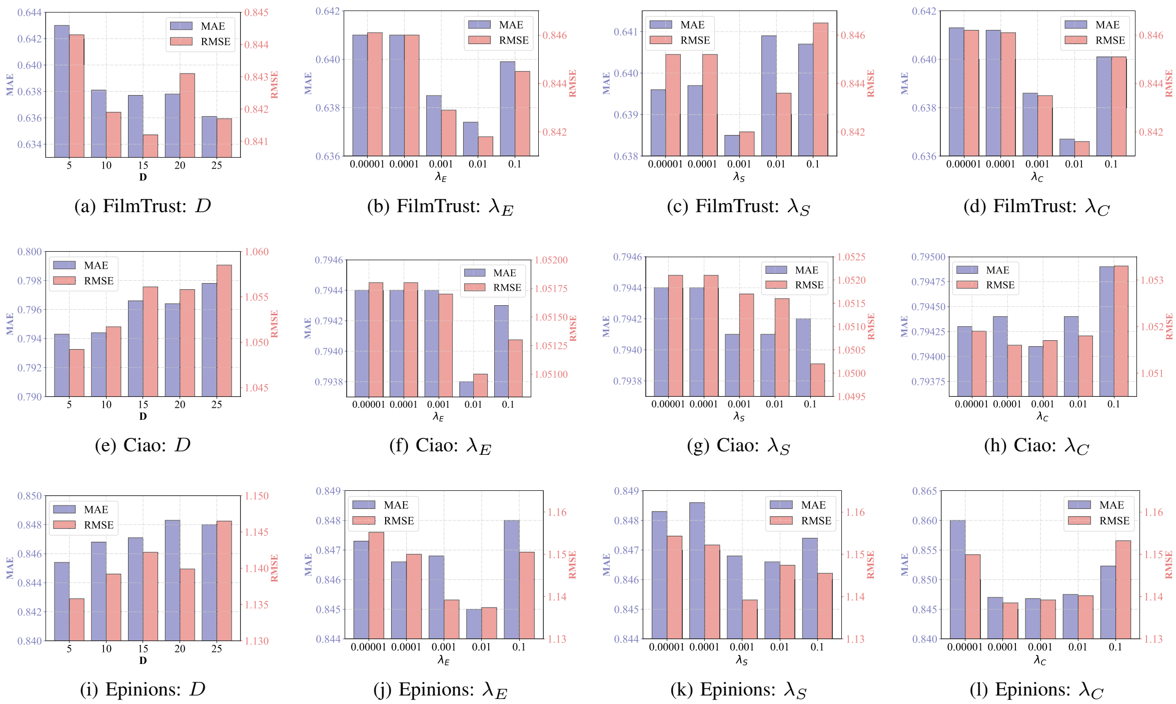
IV. Performance Evaluation

A. Datasets

We choose three recommendation benchmarks to carry out the experiments: (1) *FilmTrust:* It is crawled from the FilmTrust website, which is widely used in research on social recommendation services; (2) *Ciao:* It is crawled from a product review site Ciao, which contains rating matrix, social information and side information like item categories; and (3) *Epinions:* It is crawled from a product review site Epinions. The statistic details are shown in Table I.

B. Experiment Settings

We implement our model by TensorFlow. The embedding size is set as $D = 10$. We set $\lambda_C = \lambda_S = \lambda_E = 0.001$. Mini-



(a) FilmTrust: D (b) FilmTrust: λ_E (c) FilmTrust: λ_S (d) FilmTrust: λ_C

(e) Ciao: D (f) Ciao: λ_E (g) Ciao: λ_S (h) Ciao: λ_C

(i) Epinions: D (j) Epinions: λ_E (k) Epinions: λ_S (l) Epinions: λ_C

Fig. 3: Performance comparisons with different hyper-parameters on three datasets.

TABLE I: Details of the datasets

Datasets	#Users	#Items	#Ratings	#Links	Density
FilmTrust	1508	2071	35497	1853	1.1400%
Ciao	7375	99746	278483	111781	0.0379%
Epinions	40163	139738	664824	487138	0.0118%

batch gradient descent method is adopted with batch size 64 to optimize the parameters. The learning rate is decaying from 0.01 to 0.001 during the training process.

The dataset is randomly divided into three parts: 70% for training, 20% for validation, and 10% for testing. Mean absolute error (MAE) and root mean square error (RMSE) are used to evaluate the performance of our model. The smaller the values are, the better the model performs.

C. Baselines

We choose 8 models as baselines in the experiments: (1) *SVD++ [9]:* A typical recommendation system model based on SVD and matrix factorization; (2) *TrustMF [25]:* It is a trust-based method, which introduces latent factors to retrieve the social network matrix; (3) *TrustSVD [4]:* It is a trust-based method based on SVD++, which incorporates both implicit and explicit influence of trusted users; (4) *NCF [6]:* A deep learning-based recommendation model uses only the rating matrix without social information; (5) *NSCR [19]:* A regularization-based deep learning recommendation method; (6) *SREPS [12]:* It takes the preference space into account to model the differences between user preferences in recommendation systems and social networks; (7) *DANSER [24]:* It is a dual graph attention network, which captures the deep representations of preference and multifaceted social effects of users; and (8) *TrustEV [23]:* A powerful social recommendation mode, which proposes the feature evolution method without considering the social exposure information.

D. Overall Performance

We present all the comparative results in Table II. We have several findings as follows. First, our model outperforms all the competitors. MAE and RMSE are both improved in all three datasets. Overall, the improvements suggest that social exposure information and the feature network have a favorable impact on the recommendation service. Second, the degrees of improvement varies between datasets. In Ciao, the improvement degree on two metrics are the least and the performance on FilmTrust is better than that on Ciao. This may be due to the data density is highest in FilmTrust. For Epinions, the improvements are the best, which implies that a large dataset may provide more information on user preference and social information. Third, TrustEV beats other baselines on most of the results, which indicates the importance of feature evolution. The out-performance of our model compared to TrustEV indicates the importance of social exposure.

E. Parameter Sensitivity

Dimension D. We try 5 embedding sizes ranging from 5 to 25. The results are shown in Fig. 3a, Fig. 3e, and Fig. 3i. For

TABLE II: Performance comparisons of ExpoEv. The bold value marks the best one in each column. The improvements are compared with the best competitors marked by underlines. All experiments are repeated 5 times and taken the average.

Method	FilmTrust		Ciao		Epinions	
	MAE (mean±std)	RMSE (mean±std)	MAE (mean±std)	RMSE (mean±std)	MAE (mean±std)	RMSE (mean±std)
SVD++	0.9490±0.0004	1.1321±0.0019	1.0944±0.0004	1.3212±0.0012	0.9995±0.0001	1.2270±0.0009
TrustMF	0.6675±0.0014	0.8627±0.0002	1.2159±0.0004	1.4159±0.0002	1.1076±0.0004	1.3124±0.0001
TrustSVD	0.7244±0.0004	0.9289±0.0007	1.1439±0.0014	1.3355±0.0015	0.9712±0.0001	1.1730±0.0012
NCF	0.6584±0.0020	0.8589±0.0020	0.8269±0.0018	<u>1.0628</u>±0.0011	0.8753±0.0005	1.1657±0.0005
NSCR	0.6660±0.0004	0.8586±0.0007	0.8162±0.0003	1.0888±0.0003	0.8827±0.0012	1.1712±0.0013
SREPS	0.6634±0.0015	0.8638±0.0009	0.8181±0.0012	1.0787±0.0009	0.8825±0.0003	1.1632±0.0011
DANSER	0.6983±0.0012	0.9038±0.0008	<u>0.8034</u>±0.0004	1.0723±0.0014	0.8698±0.0006	1.1644±0.0009
TrustEV	<u>0.6543</u>±0.0002	<u>0.8550</u>±0.0019	0.8065±0.0020	1.0731±0.0006	<u>0.8690</u>±0.0007	<u>1.1576</u>±0.0007
ExpoEv	**0.6385±0.0013**	**0.8419±0.0008**	**0.7941±0.0006**	**1.0517±0.0001**	**0.8468±0.0008**	**1.1392±0.0009**
Impv.	2.41%	1.53%	1.17%	1.04%	2.55%	1.59%

FilmTrust, we can find that MAE shows a declining trend when D ranges from 5 to 25, while the RMSE flicks up and down. While for Ciao and Epionions, when $D = 5$, the performance on both two metrics is the best since these two datasets are much larger than FilmTrust. In conclusion, the performance is closely connected to both the embedding size D and the scale of datasets. When the dataset is small, a larger embedding size may catch more useful features. However, when the dataset is huge, the large embedding size may make the representation sparse and unsuitable for optimization.

Regularization Weight. λ_E, λ_S and λ_C. For these three regularization weight, we try values from $1e-5$ to $1e-1$. The results are shown in Fig. 3. We can see that regularization weights have a great influence on both MAE and RMSE, and the trade-off between optimization and regularization should be balanced appropriately. For λ_E, the proper value lies in the range of [0.001, 0.01] for three datasets and we finally choose 0.01. For λ_S, results from three datasets show distinct trends. But $\lambda_S = 0.01$ is still the most proper one. As for λ_C, the performance is degrading with the increasing weight on Ciao. We still choose $\lambda_S = 0.01$ considering all the datasets.

F. Ablation Study

In this section, we remove the feature evolution unit from the model and evaluate its performance. In total, we test three versions. In ExpoEv-I, we only use the feature evolution unit in the item domain, and let the three latent factors in the user domain share a common feature space. Therefore, there are no cross-feature operations in the user domain. In ExpoEv-U, similarly, we only use the feature evolution unit in the user domain, and let the two latent factors in the item domain share a common feature space. ExpoEv-N is a combination of ExpoEv-I and ExpoEv-U, where latent factors share a common feature space in both the user domain and item domain.

The results are shown in Table III. For the three datasets, we can find that the performance of ExpoEv is the best, which proves that fusing social trust and exposure information in both user domain and item domain is beneficial for performance. Moreover, for FilmTrust, the MAE of ExpoEv-U is smaller than ExpoEv-I, while the RMSE of ExpoEv-U is larger than

TABLE III: Performance of ExpoEv variants on three datasets.

Variants	FilmTrust		Ciao		Epinions	
	MAE	RMSE	MAE	RMSE	MAE	RMSE
ExpoEv-I	0.6543	0.8513	0.8013	1.0710	0.8501	1.1546
ExpoEv-U	0.6535	0.8562	0.8060	1.0719	0.8512	1.1553
ExpoEv-N	0.6536	0.8526	0.7979	1.0670	0.8475	1.1501
ExpoEv	**0.6385**	**0.8419**	**0.7941**	**1.0517**	**0.8468**	**1.1392**

ExpoEv-I. For Ciao and Epinions, the performance of ExpoEv-I is better than ExpoEv-U. This demonstrates a larger dataset may be more general. Therefore, the evolution method in the item domain can be even more important than that in the user domain. Note that in Ciao, the performance of ExpoEv-N is much better than ExpoEv-I and ExpoEv-U.This is because the evolution method should be applied to both domains simultaneously to avoid information unbalancing in the two domains. We are supposed to either use it in both domains or not use it. If we use it in just one domain and don't apply it to the other, the data information in the two domains may be unbalanced. Consequently, the performance may be even worse. In conclusion, the feature evolution units are useful for both user and item domains to obtain a better performance.

V. Conclusion

In this paper, we propose a deep collaborative filtering model ExpoEv based on feature evolution for social recommendation services. We introduce a novel concept of social exposure, which augments social information by considering the interaction information of a user's friends with respect to items, thus enriching user and item representations. Then we propose a feature evolution unit to refine the representation embeddings of users, which plays a positive role in prediction performance. It is applied to fuse the user preference factor, user social trust factor, and user exposure factor and generate an evolved representation. Experimental results on three datasets show that our model outperforms eight baseline models in the quality of recommendation services.

References

[1] Bai, T., Wen, J., Zhang, J., Zhao, W.X.: A neural collaborative filtering model with interaction-based neighborhood. In: ACM International Conference on Information and Knowledge Management (CIKM). pp. 1979–1982 (2017)

[2] Chen, H., Feng, Z., Chen, S., Xue, X., Wu, H., Sun, Y., Xu, Y., Han, G.: Capturing users' fresh interests via evolving session-based social recommendation. In: IEEE International Conference on Web Services (ICWS). pp. 182–187 (2022)

[3] Chen, J., Zhang, H., He, X., Nie, L., Liu, W., Chua, T.: Attentive collaborative filtering: Multimedia recommendation with item- and component-level attention. In: International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR). pp. 335–344 (2017)

[4] Guo, G., Zhang, J., Yorke-Smith, N.: Trustsvd: Collaborative filtering with both the explicit and implicit influence of user trust and of item ratings. In: AAAI Conference on Artificial Intelligence (AAAI). vol. 29, pp. 123–129 (2015)

[5] He, X., Liao, L., Zhang, H., Nie, L., Hu, X., Chua, T.: Neural collaborative filtering. In: International World Wide Web Conference (WWW). pp. 173–182 (2017)

[6] He, X., Liao, L., Zhang, H., Nie, L., Hu, X., Chua, T.S.: Neural collaborative filtering. In: International Conference on World Wide Web (WWW). pp. 173–182 (2017)

[7] He, X., Zhang, H., Kan, M., Chua, T.: Fast matrix factorization for online recommendation with implicit feedback. In: International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR). pp. 549–558 (2016)

[8] Jamali, M., Ester, M.: Trustwalker: a random walk model for combining trust-based and item-based recommendation. In: ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD). pp. 397–406 (2009)

[9] Koren, Y.: Factorization meets the neighborhood: a multifaceted collaborative filtering model. In: ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD). pp. 426–434 (2008)

[10] Koren, Y., Bell, R.M., Volinsky, C.: Matrix factorization techniques for recommender systems. Computer **42**(8), 30–37 (2009)

[11] Li, D., Wang, C., Li, L., Zheng, Z.: Collaborative filtering algorithm with social information and dynamic time windows. Applied Intelligence **52**(5), 5261–5272 (2022)

[12] Liu, C.Y., Zhou, C., Wu, J., Hu, Y., Guo, L.: Social recommendation with an essential preference space. In: AAAI Conference on Artificial Intelligence (AAAI). vol. 32, pp. 346–353 (2018)

[13] Liu, K., Xue, F., He, X., Guo, D., Hong, R.: Joint multi-grained popularity-aware graph convolution collaborative filtering for recommendation. IEEE Transactions on Computational Social Systems **10**(1), 72–83 (2023)

[14] Luo, F., Wu, J., Wang, H.: Semi-discrete social recommendation (student abstract). In: AAAI Conference on Artificial Intelligence (AAAI). vol. 35, pp. 15835–15836 (2021)

[15] Miyahara, K., Pazzani, M.J.: Improvement of collaborative filtering with the simple bayesian classifier. Information Processing Society of Japan **43**(11) (2002)

[16] Sarwar, B.M., Karypis, G., Konstan, J.A., Riedl, J.: Item-based collaborative filtering recommendation algorithms. In: International World Wide Web Conference (WWW). pp. 285–295 (2001)

[17] Song, W., Xiao, Z., Wang, Y., Charlin, L., Zhang, M., Tang, J.: Session-based social recommendation via dynamic graph attention networks. In: ACM International Conference on Web Search and Data Mining (WSDM). pp. 555–563 (2019)

[18] Wang, H., Wang, N., Yeung, D.: Collaborative deep learning for recommender systems. In: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Sydney, NSW, Australia, August 10-13, 2015. pp. 1235–1244. ACM (2015)

[19] Wang, X., He, X., Nie, L., Chua, T.S.: Item silk road: Recommending items from information domains to social users. In: International ACM SIGIR conference on Research and Development in Information Retrieval (SIGIR). pp. 185–194 (2017)

[20] Wang, X., Zhu, W., Liu, C.: Social recommendation with optimal limited attention. In: ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD). pp. 1518–1527 (2019)

[21] Wang, Y., Li, C., Liu, Z., Li, M., Tang, J., Xie, X., Chen, L., Yu, P.S.: An adaptive graph pre-training framework for localized collaborative filtering. ACM Transactions on Information Systems **41**(2), 1–27 (2022)

[22] Wu, J., Fan, W., Chen, J., Liu, S., Li, Q., Tang, K.: Disentangled contrastive learning for social recommendation. In: ACM International on Conference on Information and Knowledge Management (CIKM). pp. 4570–4574 (2022)

[23] Wu, Q., Jiang, L., Gao, X., Yang, X., Chen, G.: Feature evolution based multi-task learning for collaborative filtering with social trust. In: International Joint Conference on Artificial Intelligence (IJCAI). pp. 3877–3883 (2019)

[24] Wu, Q., Zhang, H., Gao, X., He, P., Weng, P., Gao, H., Chen, G.: Dual graph attention networks for deep latent representation of multifaceted social effects in recommender systems. In: The Web Conference (WWW). pp. 2091–2102 (2019)

[25] Yang, B., Lei, Y., Liu, D., Liu, J.: Social collaborative filtering by trust. In: International Joint Conference on Artificial Intelligence (IJCAI). pp. 2747–2753 (2013)

[26] Yang, X., Guo, Y., Liu, Y., Steck, H.: A survey of collaborative filtering based social recommender systems. Computer Communications **41**, 1–10 (2014)

[27] Yu, J., Yin, H., Li, J., Wang, Q., Hung, N.Q.V., Zhang, X.: Self-supervised multi-channel hypergraph convolutional network for social recommendation. In: The Web Conference (WWW). pp. 413–424 (2021)

[28] Zhang, B., Yu, J.: An adaptive preference retention collaborative filtering algorithm based on graph convolutional method. Electronic Research Archive **31**(2), 793–811 (2023)