IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received October 2, 2019, accepted November 20, 2019, date of publication November 25, 2019,
date of current version December 10, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2955863

A Novel Learning Model Based on Trust Diffusion and Global Item for Recommender Systems

YAKUN LI[1,2], JIAOMIN LIU[1], AND JIADONG REN[1,2]
[1]College of Information Science and Engineering, Yanshan University, Qinhuangdao 066004, China
[2]The Key Laboratory for Computer Virtual Technology and System Integration of Hebei Province, Yanshan University, Qinhuangdao 066004, China
Corresponding author: Yakun Li (15033564136@163.com)

ABSTRACT Recommender systems can provide users with an ordered list of various items, which greatly assists users to purchase products that they are satisfied with. However, item recommendation has been confronted with some inherent problems, such as sparse ratings and long-tail distribution, resulting in low accuracy of recommendations and insignificant marketing. In this paper, we propose a novel learning model based on trust diffusion and global item (TDGIL) to improve the accuracy of item rating prediction for recommender systems. Specifically, first, the rating information on items is mined and aggregated to the greatest extent based on trust diffusion characteristics among users. The benchmark prediction of item recommendation is updated by a user trust neighbor set and its item ratings, which are obtained by a trust diffusion algorithm. Then, the difference weights and compensation coefficients for all items are defined to learn users' potential preferences in the proposed global item model. Finally, the TDGIL learning algorithm is presented to train and learn the target networks by random gradient descent. The extensive experiments and results on two real-world datasets demonstrated that our proposed model can achieve significant improvements in the accuracy of rating prediction compared with some state-of-the-art methods.

INDEX TERMS Recommender system, sparse data, trust diffusion, global item, item rating prediction.

I. INTRODUCTION

With the explosive growth of Internet information and the emergence of new e-commerce services, users often suffer from information overload [1]. Instead of producing economic benefits, the diversity of choice reduces user satisfaction. At this time, recommender systems (RS) [2] have come into being and have proven to be effective in helping users select suitable items by modeling their real preferences, which has been widely used in information retrieval [3], e-commerce, advertising and other fields. In these RS, item rating prediction plays a key role in improving the accuracy of the recommendation model. In other words, the main task of recommendation systems is to predict a user's rating on unrated items. It has become a popular topic in the field of recommendation research today and has attracted the attention of a large number of researchers.

In general, the existing recommendation models for completing the rating prediction task are mainly divided into three categories: neighborhood-based models [4], content-based models [5] and matrix factorization (MF)-based models [6]. The main principle of neighborhood-based models is to predict users' ratings on new items by using the historical rating records of items. It includes two well-known recommendation methods: user-oriented and item-oriented collaborative filtering (CF) recommendations [7]–[9]. Content-based recommendation models analyze the common textual information of items that have been rated by a target user and then recommend new items containing the abovementioned textual characteristics to the target user. In addition, MF-based models are considered more popular and advanced methods in recommender systems. They mainly decompose the rating matrix of users and items to represent the interaction relationship among them and analyze the latent factor vector to locate user preferences and predict unobserved ratings.

Furthermore, some deep neural network models, such as the restricted Boltzmann machine (RBM) and various variant encoders [10], [11], also improve the performance of recommendation systems to some extent. However, these existing recommendation algorithms all have some defects or shortcomings in terms of item rating prediction. For example, CF-based methods can only capture a single type of relation

The associate editor coordinating the review of this manuscript and approving it for publication was Dominik Strzalka.

(user-user or item-item) and are more prone to suffer from cold start and data sparsity problems [12] due to the lack of rating information. Although the RBM method predicts item ratings by constructing corresponding independent models from the perspective of the user or item side information, they are not deep enough to capture complex features of a rating matrix. Moreover, content-based methods have difficulty in extracting all the characteristics of items from the user or item attribute profiles, thus failing to accurately mine the user's potential preferences and interests.

In this article, we argue that trust relationships among users and differences in all items can provide some or more relevance for recommendations. Based on this, we propose a novel recommendation learning model, in which trust diffusion features among users are utilized to obtain more item ratings. Then, the difference weights and compensation coefficients among global items are defined to train the objective neural network to reach the minimum error value. Different from conventional similarity recommendation methods, the TDGIL model utilizes the deep learning method to mine more rating information among users to improve the accuracy of item rating prediction. In particular, for the problem of data sparsity, the improved recommendation predictor can better learn and model users' real requirements and preferences for recommendations.

In detail, compared with other methods, the main differences of our proposed learning model are that it can utilize the trust diffusion among users and deep neural network to fully explore available rating records to alleviate the impact of the data sparsity problem on item recommendations. Moreover, taking full account of the differences in global items, our approach can more accurately locate users' interest preferences to find the most appropriate recommended items, which is also one of the main innovations of our work. Finally, to evaluate the effectiveness of the proposed model, empirical experiments are performed on two real-world datasets from different domains. The experimental results also prove that the proposed TDGIL algorithm has better accuracy than some state-of-the-art recommendation approaches.

The main contributions of this paper are as follows.

1. We introduce a novel pipeline to improve the accuracy of item rating prediction in recommendation systems, which first learns the user trust diffusion feature and the difference of global items to jointly improve the benchmark predictor and establish their neural networks.

2. In the proposed model, more rating data are collected and provided through a user trust diffusion algorithm for better neural network training. Then, the defined difference weights and compensation coefficients are utilized to learn the correlation of all items. Finally, the optimal solution of recommendation results is obtained by means of stochastic gradient descent.

3. Extensive experiments are conducted on two real datasets to study the performance of the proposed model, and the comparison results also show that our TDGIL outperforms other state-of-the-art methods.

The remainder of this paper is organized as follows. Section 2 presents the relevant work on recommendation algorithms. Section 3 provides background information about our model. In section 4, we develop the proposed TDGIL approach in detail. In section 5, empirical experiments are conducted to verify the accuracy of item rating prediction for recommender systems. Finally, section 6 concludes this article with some future directions.

II. RELATED WORKS

In this section, several major recommendation models related to our topic are reviewed, including factorization-based models, neighborhood-based models and deep learning models.

A. FACTORIZATION-BASED MODEL

The recommendation model based on matrix factorization is an advanced method that has achieved great success, and many researchers are committed to analyzing the characteristics of rating matrixes in recommender systems. Liu *et al.* [13] presents a list wise probabilistic matrix factorization (LPMF) framework, where item prediction ratings are generated by taking the preference orders of the users indicated by observed ratings as a whole instance and maximizing the log-posterior over the predicted preference order. Their approach is computationally efficient and can be applied to big dataset environments. Different from LPMF, a novel low-rank matrix factorization model with adaptive graph regularizer [14] was proposed to seek graph weight matrixes and low-dimensional representations of rating data for item recommendations. Although these two methods can adequately handle the interaction ratings between users and items, the data sparsity problem is still a bottleneck.

Therefore, a hybrid approach based on a topic model and matrix factorization [15] was introduced to predict the probability that a user will rate an item and the corresponding rating values to improve the accuracy of recommendations. Furthermore, Xing *et al.* [16] devises a joint convolution matrix factorization model that considers various factors. In the model, geographical influences from a user's check-in behavior, user social relations and latent factors of users and items can be effectively utilized in the recommendation model. Experimental results show that the proposed method achieves significantly superior precision. In addition, some deep matrix decomposition methods explore how to better improve the accuracy of rating prediction [17]–[19].

The aforementioned methods can solve the problem of data sparsity to some extent. However, the linear decomposition of potential features of the rating matrix may result in the loss of some implicit rating information and poor prediction accuracy, especially when the rating datasets are very sparse.

B. NEIGHBORHOOD-BASED MODEL

The neighborhood-based recommendation, as the earliest and a widely used model, can capture the association among the rating information and recommend some items that are not in the same type or are unknown to the target user,

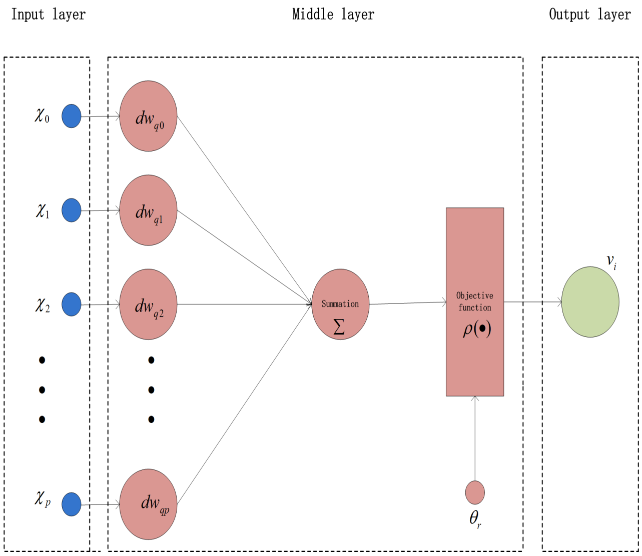

FIGURE 1. GNNL model for recommendation.

as long as his neighbors provide a very high rating. For example, Herlocker *et al.* [20] designed the neighborhood-based prediction systems that divide the whole framework into three components, namely, similarity computation, neighbor selection, and rating combination for recommendation improvements in accuracy. To enhance the performance of neighborhood-based collaborative filtering, time-aware information is integrated into a similarity measurement for high-quality Web service recommendation [21]. Additionally, a hybrid personalized random walk algorithm was presented to deduce user similarity and service similarity in the proposed scheme, and experiments on several real-world datasets were provided to verify the effectiveness of their approach.

In addition, some user-oriented [22], [23] and item-oriented [24], [25] neighborhood methods have been proposed to improve the accuracy of rating predictions. They are not only easy to implement and very efficient but can also produce relatively stable and accurate recommendation results. In addition to item-oriented and user-oriented CF algorithms, some researchers also incorporate these two algorithms into the similarity fusion framework [26], [27].

C. DEEP LEARNING MODEL

Deep learning recommendations have become increasingly important and indispensable in recommender systems due to their advanced performance. Deep learning has been proven to be a powerful approach for capturing complex relationships with more abstract hierarchical neural networks. A general neural network learning (GNNL) model is shown in Fig. 1 below.

In this review, the learning model is divided into three layers: the input layer, the middle layer, and the output layer. Specifically, the input layer responds to data entering the network. The middle layer receives weighted output from the input unit. The output layer responds to the weighted output in the middle layer and generates the final network output. In the recommendation application, the main advantage of the GNNL model is that it can handle nonlinear classification tasks compared to conventional algorithms, and due to the parallel attributes, it can efficiently train the network and generate the final recommendations even in the case of partial network damage.

Many researchers have explored the application of deep learning to recommender systems to improve the accuracy of item rating prediction. For example, Paradarami *et al.* [28] proposed a deep learning neural network framework that uses content-based features and user reviews to generate rating prediction. A stacked denoising autoencoder is presented to learn more useful and complex representations in a neural network with a local denoising criterion [29]. Liu *et al.* [30] extended the RBM model by making use of a naive Bayes classifier to approximate the missing entries in the user-item rating matrix, and then applied the improved RBM on a denser scoring matrix. Experimental results also showed that the proposed model performs better than pure RBM and other CF methods in terms of rating prediction.

Furthermore, to overcome the shortcomings of capturing only a single type of relation, a novel deep learning approach is proposed to learn the corresponding low-dimensional vector of user and item [31], which embeds semantic information to reflect the user-item correlation. Then, a feed-forward neural network is utilized to simulate interaction among users and items, where the pretrained representational vectors are adopted as inputs to neural networks.

In addition, there are some deep learning methods that have been proposed to improve the performance of the recommendation [32]–[35], which is closely relevant to our approach. Although these methods provide new ideas for recommendation systems from different perspectives, the disadvantages are that many parameters need to be learned, especially when the data are sparse, its ability to normalize is very weak.

Moreover, in these current deep learning recommendations, only a single type of interactive rating or semantic embedding is used to train the target network. This situation is likely to result in the recommender systems only capturing the user's explicit preferences, and the information behind the ratings or semantics are difficult to find and exploit. In contrast to the above deep learning methods, the main advantage of our proposed learning model is to comprehensively utilize user trust diffusion features and global item differences, to obtain more potential ratings and implicit preferences. Ultimately, the accuracy of item rating prediction is improved by training and minimizing the error function.

III. BACKGROUND

A. PROBLEM DEFINITION

In recommendation systems, rating prediction, which refers to predicting a user's rating on unrated items, is one of the most important tasks. In other words, given a triple (u, i, r) of

TABLE 1. Summary of notations and their meanings.

Symbol	Meaning
u, i, r	User, item, and rating respectively
U, I, R	The collections of user, item and rating respectively
$r_{v,i}$	The rating given by the user v on the item i
$\hat{r}_{ui}$	The prediction rating given by the user u on the item i
φ_{ui}	Benchmark prediction rating
μ	Mean of all the ratings
b_u	The bias of user ratings
b_i	The bias of item ratings
$TD_{u,v}$	Trust diffusion value between users u and v
$TL_{u,v}$	The minimum trust length between user u and user v
dw_{ij}	Difference weight of item j to item i
c_{ij}	Compensation coefficient
δ	Learning rate

the training set, which consists of a user $u \in U$, an item $i \in I$ and a rating $r \in R$, our goal is to build a learning function model Ψ to predict the rating of target user u for new item i (unrated by the target user u).

For convenience, we put some commonly used notations and their meanings in **Table 1**.

B. BENCHMARK PREDICTION RECOMMENDATION MODEL

Most CF methods attempt to capture the interaction between users and items to generate different predictive ratings. However, a majority of the observed ratings is related to either the user or the item, not the interaction between the user and item. Therefore, the benchmark predictor (BP), as a widely used basic model in recommendation systems, is defined by fully considering the bias of users and items.

$$\varphi_{ui} = \mu + b_u + b_i \tag{1}$$

where μ is the overall average rating, parameters b_u and b_i represent the deviation of user u and item i, respectively, from the average rating, and φ_{ui} is a benchmark prediction rating that considers both the user and item factors. For example, assume we want to build a user Curry's baseline prediction rating for the "Spider-Man: Far From Home" movie. Assume that the average rating μ for all movies is 2.9. In addition, since "Spider-Man: Far From Home" is better than general films, its rating is 0.8 higher than the average rating. Curry, on the other hand, as a discerning user, rated the film 0.6 lower than the average rating. As a result, Curry's benchmark prediction rating for "Spider-Man: Far From Home" is $2.9 - 0.6 + 0.8 = 3.1$.

IV. THE PROPOSED LEARNING MODEL

A. USER TRUST DIFFUSION

The recommender system is considered to be an information processing system that recommends suitable items to users by collecting numerous rating data. However, the lack of user or item rating information, known as data sparseness, has become an inherent bottleneck for recommendations. Hence, in view of the low accuracy of existing recommendation methods, the trust information among users is explored and mined to better model user preference in our paper.

For the sake of maximizing the use of the small amount of available data, this article takes advantage of the trust feature in networking sites that can be spread and diffused among different users [36], [37]. Then, both the user's ratings and those of his trusted neighbors can be gathered to make recommendations. In a user's trust network, any two users can be connected according to the small-world theory [38] and therefore the value of trust diffusion is inversely proportional to what we think of as their trust length in our work. Hereafter, the trust diffusion value between users u and v $TD_{u,v}$ is defined as follows.

$$TD_{u,v} = \frac{1}{TL_{u,v}} \tag{2}$$

In the formula, $TL_{u,v}$ is the minimum trust length between user u and user v computed by the breadth-first search method [39]. In practice, to avoid more cost and noise, we can set $TL_{u,v}$ less than or equal to 3 to achieve a better trust diffusion effect for recommendations.

Then, the set of diffused trust neighbors DTN_u for target user u is defined in our proposed recommendation model.

$$DTN_u = \{v | TD_{u,v} > \varepsilon, v \in U\} \tag{3}$$

where ε is the trust diffusion threshold, U represents the user set containing all users in a trust network, and v is any user in the user set U.

Since the $TL_{u,v}$ between two users is no more than 3, for simplicity, our method treats all diffused trust neighbors as available and sets $\varepsilon = 0$. In addition, it is important to note that each user trusts itself, so its DTN_u includes itself.

In general, most conventional recommendation methods directly utilize the original user set to compute the similarity between users to generate the recommendation list. Nevertheless, efficient recommendation results cannot be achieved due to data sparsity and other problems. Therefore, a trust diffusion algorithm (TDA) is designed to explore the target users' trust neighbors and mine more useful information in recommender systems for better modeling user preferences and improving the accuracy of item rating predictions. The pseudo-code of the TDA method is shown as below.

In line 1, since the target user u always trusts itself, DTN_u is initialized by user u. The next line of the algorithm traverses all users in a trusted network to search for trustworthy users. Line 3-4 determines whether v_i is on the trust path, the minimum trust length satisfies no more than 3, and the trust diffusion value is greater than the diffusion threshold.

Algorithm 1 Trust Diffusion Algorithm

Inputs: user set U and trust diffusion threshold ε;
Output: the diffused trust neighbors DTN_u;
1. **Initialize** DTN_u;
2. **for** v_i in U //v_i is any user in U
3. **if** v_i exists in a minimum directional path from u and $TL_{u,v} \leq 3$
4. **if** $TD_{u,v}$ is greater than the trust diffusion threshold ε
5. $v_i \rightarrow DTN_u$
6. Delete(v_i, U);
7. **end if**
8. **end for**

As shown in lines 5 and 6, eligible trusted users are filtered out and stored in DTN_u, and the user v_i is deleted from user set U. Finally, the DTN_u is output.

After identifying the diffused trust neighbors, a series of item ratings can be filtered out and expanded into a candidate rating set for next network training. The definition of the candidate rating set is as follows.

$$CRS_u = \{crs | r_{v,i} \in R, \exists v \in DTN_u, i \in I\} \tag{4}$$

In the formula, the sets of all original items and ratings are denoted I and R, respectively, and $r_{v,i}$ represents the rating given by the user v on item i, taking a certain value in a rating range defined by a recommender system, such as an integer from 1 to 5. In the previous algorithms, the item rating prediction process only considers related items of users similar to the target user, while more users after trust diffusion are mined and utilized to integrate into the prediction model in our proposed method. Therefore, in the benchmark predictor, we update the predicted value with the average of the item ratings of the trusted neighbors instead of the average of the overall item ratings.

$$\varphi'_{ui} = \mu' + b_u + b_i \tag{5}$$

where μ' is the average of CRS_u.

B. GLOBAL ITEM MODEL

Most item-based methods are local in nature because they focus only on a small subset of related item ratings, which results in low accuracy in many recommendation systems. Furthermore, these local methods also violate the principle that matrix decomposition techniques use all rating records to describe the characteristics of items and users. Considering the current research situation, we propose a new global item model, which is based on the relationship among items and improves the accuracy of recommendations by considering the potential differences among all items.

Specifically, the traditional models use the item set belonging to the target user's neighbors similar to item i to make recommendations. However, in many cold systems, the target user does not have many similar neighbors. In this case, we define the difference weights of all items that are not related to a target user to achieve global model optimization. Then, the difference weight of item j to item idw_{ij} is proposed and added to the benchmark predictor. The prediction formula for item rating is defined as follows.

$$\hat{r}_{ui} = \varphi'_{ui} + \sum_{j \in R_u} (r_{uj} - \varphi_{uj}) dw_{ij} \tag{6}$$

In the above definition, φ'_{ui} is the trusted but robust benchmark predictor. The predicted value $\hat{r}_{ui}$ is adjusted by summing all item ratings of target user u. In our work, the difference weight is regarded as a coefficient for adjusting and compensating the predicted value, and then the prediction accuracy of r_{ui} is improved based on the observed r_{uj} value.

By using the difference weights of global items rather than interpolation coefficients for a specific user, it emphasizes the impact of missing rating values. In other words, the user's preferences are reflected not only in the items the user rates but also in the items the user does not rate. For example, assume a book rating dataset shows that users who rate high on Harry Potter 1 will also rate higher on Harry Potter 2. The previous methods establish a high weight from Harry Potter 1 to Harry Potter 2. However, if a user does not rate Harry Potter 1 at all, the rating on Harry Potter 2 is penalized because some necessary weights are not added to the sum to participate in the predictions.

Therefore, similar to the difference weight, this article defines another compensation coefficient c_{ij} and adds it to the benchmark predictor. Accordingly, a more general item rating prediction model is optimized as follows.

$$\hat{r}_{ui} = \varphi'_{ui} + \sum_{j \in R_u} [(r_{uj} - \varphi_{uj}) dw_{ij} + c_{ij}] \tag{7}$$

For two items i and j, the implicit preference of user u for item j allows us to use c_{ij} to adjust the estimated value $\hat{r}_{ui}$.

Based on the aforementioned benchmark predictor, the prediction model is redefined as follows.

$$\hat{r}_{ui} = \mu' + b_u + b_i + \sum_{j \in R_u} [(r_{uj} - \varphi_{uj}) dw_{ij} + c_{ij}] \tag{8}$$

In addition, to normalize the prediction model, we further optimize the prediction formula as follows.

$$\hat{r}_{ui} = \mu' + b_u + b_i + |R_u|^{-\alpha} \sum_{j \in R_u} [(r_{uj} - \varphi_{uj}) dw_{ij} + c_{ij}] \tag{9}$$

The constant α controls the degree of normalization. Since the denormalized rule causes the predicted value of the user who provides many ratings to be more biased than the benchmark recommendation, and the fully normalized rule eliminates the bias effect of the number of ratings in the prediction, the constant α maintains the robustness of the prediction model. According to experience on the Netflix Prize [40], the general model achieves the best results when $\alpha = 0.5$, so the final prediction model is specifically improved as follows.

$$\hat{r}_{ui} = \mu' + b_u + b_i + |R_u|^{-0.5} \sum_{j \in R_u} [(r_{uj} - \varphi_{uj}) dw_{ij} + c_{ij}] \tag{10}$$

C. THE OBJECTIVE FUNCTION AND PARAMETER ESTIMATION

The accuracy error of item recommendations is defined as e_{ui}, where $e_{ui} = r_{ui} - \hat{r}_{ui}$. Therefore, the parameters of the proposed model can be learned by the regularized least squares method [41]. In other words, the objective function is defined to minimize the squared error between the predicted value and the real value.

$$\begin{aligned}\sum e_{ui}^2 = \min \sum (r_{ui} - \mu' - b_u - b_i \\ - |R_u|^{-0.5} \sum_{j \in R_u} ((r_{uj} - \varphi_{uj}) dw_{ij} + c_{ij}))^2 \\ + \lambda (b_u^2 + b_i^2 + \sum_{j \in R_u} dw_{ij}^2 + c_{ij}^2) \end{aligned} \tag{11}$$

Then, we can utilize the random gradient descent algorithm to iterate all known ratings, and adjust parameters dw_{ij} and c_{ij} by moving in the opposite direction to the gradient. The details are as follows.

$$\nabla dw_{ij} = \frac{\partial f_{ui}}{\partial dw_{ij}} = -|R_u|^{-0.5} \cdot \sum_{j \in R_u} (r_{uj} - \varphi_{uj}) \cdot e_{ui} + \lambda dw_{ij} \tag{12}$$

$$\nabla c_{ij} = \frac{\partial f_{ui}}{\partial c_{ij}} = -|R_u|^{-0.5} \cdot e_{ui} + \lambda c_{ij} \tag{13}$$

$$\nabla b_u = \frac{\partial f_{ui}}{\partial b_u} = -e_{ui} + \lambda b_u \tag{14}$$

$$\nabla b_i = \frac{\partial f_{ui}}{\partial b_i} = -e_{ui} + \lambda b_i \tag{15}$$

Finally, the parameters can be updated by

$$dw_{ij} = dw_{ij} - \delta \cdot \nabla dw_{ij} \tag{16}$$

$$c_{ij} = c_{ij} - \delta \cdot \nabla c_{ij} \tag{17}$$

$$b_u = b_u - \delta \cdot \nabla b_u \tag{18}$$

$$b_i = b_i - \delta \cdot \nabla b_i \tag{19}$$

where δ is the learning rate.

D. ALGORITHM LEARNING

The learning procedure of TDGIL has two steps. First, the TDGIL initializes the parameters of the proposed model and calls the TDA algorithm to obtain an improved benchmark in the pre-training phase. Then, in the iteration phase, we randomly select a small portion of items that has not yet been observed to update the parameters until the objective function converges. At this point, the network training is completed and the recommendation list is output.

In addition, we briefly analyze the time cost regarding the proposed TDGIL algorithm. If the rating matrix R is an n^*n matrix, the total number of variables we need to learn in TDGIL is $|\mathrm{R}|^*n^2$, which is expressed as V. Algorithm 2 shows the specific learning process of TDGIL. Its time consumption is mainly composed of three parts: initialization, regularization, and gradient updating. Specifically, since every variable needs to be initialized, the initialization time is $O(V)$. Furthermore, the time cost of regularization is $O(n^*|U| + n^*|I|) < O(V)$, and the time consumption of gradient updating is linear. Therefore, the final time cost of the proposed TDGIL algorithm is $O(V)$.

Algorithm 2 TDGIL Recommendation Algorithm

```
Inputs: User-item rating matrix R, iterations T, learning rate δ, user set U and trust diffusion threshold ε
Output: Recommendation items for each user
Initialize model parameters dw_ij and c_ij;
TDA(U, ε)        //Improved benchmark predictor φ'_ui is obtained
for count = 1, . . . , T do
  for u = 1, . . . , n do
    computePredictionValue(r̂_ui)
      e_ui ← r_ui − r̂_ui
      while not converged do
        Compute gradient ∇dw_ij_ij
        Compute gradient ∇c_ij
// Update parameters according to gradients
          Update parameter dw_ij
          Update parameter c_ij
      end while
  end for
end for
```

V. EXPERIMENT

A. EXPERIMENTAL DATASETS AND BASELINE METHODS

The performance of our proposed TDGIL on two real-world datasets is evaluated to verify the accuracy of item rating prediction. The two datasets used in our experiments are MovieLens and Epinions, and their original details are described below.

1. MovieLens is a movie community recommendation website created by the GroupLens team of the Department of Computer Science and Engineering at the University of Minnesota. It is a noncommercial, research-oriented online movie recommender. Specifically, the publicly available MovieLens 10M dataset[1] contains 10,000,054 ratings and 95,580 tags applied to 10,681 items by 71,567 users.

2. Epinions is an online review site where users can browse through various reviews of items to help them make the correct shopping choices, and they can also post any comments about the items. The open source dataset[2] our model adopted consists of 75,888 users, 681,213 items and 1,000,608 ratings, and it was collected by Richardson and Domingos [42] from Epinions.com.

In addition, we compare the proposed TDGIL model with some state-of-the-art baseline algorithms in our experiments. For each baseline method, the parameters are set to optimal to obtain the best comparison results.

[1] https://grouplens.org/datasets/movielens/
[2] http://alchemy.cs.washington.edu/data/epinions/

User-based collaborative filtering (UCF) [43] is one of the most used approaches to provide recommendation services for users. The basic principle of the neighborhood approach is to find similar users or items by considering both the local context features of user ratings and the global preference of user behavior.

Social group recommendation (SGR) [44] presents a novel framework that captures users' preferences by analyzing group members' interactions and social relationships. The group recommendation list is generated by averaging the group member prediction ratings.

Context-aware probabilistic matrix factorization (CPMF) [45] is an improved probabilistic matrix decomposition model, where textual, social, categorical and other information is used to integrate into the latent vector factors of users and items for point-of-interest recommendations.

The Bayesian personalized ranking method (BPR) [46] provides a generic learning algorithm with implicit feedback that is based on stochastic gradient descent with bootstrap sampling and the maximum posterior estimator derived from a Bayesian analysis for item recommendation.

The deep learning model (DLM) [31] is an advanced method based on a deep neural network that learns the low-dimensional vectors of users and items by embedding semantic information. In addition, a feed-forward neural network is exploited to express the interaction between user and item.

B. EXPERIMENTAL SETUP AND EVALUATION METRICS

To obtain objective and real experimental results, 80% of each dataset is selected as the training set, and the remaining 20% as the testing sets. The random gradient descent is adopted as the training method for our model, and L2- regularization is used to prevent overfitting of model training. In this way, the influence of noise and outliers is also minimized.

The learning rate [47], [48] is used to determine the coefficient of adjusting weights. If the learning rate is too large, it may be overcorrected, resulting in the error not being converged, and the neural network training is poor. Conversely, if the learning rate is too small, the convergence rate is very slow, resulting in the training time being too long. The learning rate in our model is set to 0.005 according to the experimental results in the next subsection.

In the experiments, a cross-validation method is adopted to predict the accuracy of item recommendation, and then to generate a small sample of random target data, we used the data generator to preprocess the datasets to meet the needs of experiments in the iterative process of the algorithm. In addition, the root mean square error (RMSE) and mean absolute error (MAE) are leveraged to evaluate the rating prediction accuracy of experimental results. These two evaluation metrics are currently the most popular in the rating prediction field. The RMSE between the predicted rating and the true rating is defined as follows.

$$RMSE = \sqrt{\frac{\sum_{u,i\in\psi} (\hat{r}_{ui} - r_{ui})^2}{|\psi|}} \tag{20}$$

where ψ is the validation dataset, $\hat{r}_{ui}$ denotes the predicted rating of the item, and r_{ui} represents the true rating of the item. The MAE is another commonly used evaluation metric, as shown below.

$$MAE = \frac{1}{|\psi|} \sum_{u,i\in\psi} |\hat{r}_{ui} - r_{ui}| \tag{21}$$

Compared to the MAE, the RMSE punishes large errors in different proportions, but they both measure the prediction accuracy of items in different ways; obviously, the lower the values are, the better the performance of the recommender system. Then, we can employ both the MAE and RMSE to evaluate the recommendation performance in our experiment.

In addition, to further verify the accuracy of recommendations, we also used three other evaluation metrics: precision, recall and F-measure to evaluate the Top-N recommendation task.

$$\mathrm{Pr}\,ecision = \frac{\sum |R \cap T|}{\sum |R|} \tag{22}$$

$$\mathrm{Rec}all = \frac{\sum |R \cap T|}{\sum |T|} \tag{23}$$

$$F - measure = \frac{2 \times \mathrm{Pr}\,ecision \times \mathrm{Rec}all}{\mathrm{Pr}\,ecision + Recall} \tag{24}$$

where R is the number of ratings in the recommendation list, and T is the number of ratings in the test set. Obviously, larger values of these three indicators reveal the recommendation results more accurately.

C. IMPACT OF PARAMETERS

In this section, the influences of the learning rate and regularization parameters on the accuracy of rating prediction are observed and analyzed. The Epinions dataset is utilized to train the proposed model in all parameter impact experiments. The final experimental results are presented by obtaining stable and optimal data.

1) IMPACT OF THE LEARNING RATE

In general, it is necessary to test the sensitivity of the proposed model to the learning rate, because it has a significant impact on the rating prediction accuracy of items. Therefore, we report the changing trend in the MAE and RMSE values with different learning rates.

As shown in Fig. 2, as the learning rate increases, the MAE and RMSE of the proposed algorithm gradually decrease. When the learning rate is 0.005, the MAE and RMSE reach the lowest value (0.764, 0.546) and then start to increase again. In other words, when the learning rate is 0.005, the model training achieves the optimal performance. As a result, in the next experiments, we set the learning rate to the optimal value of 0.005.

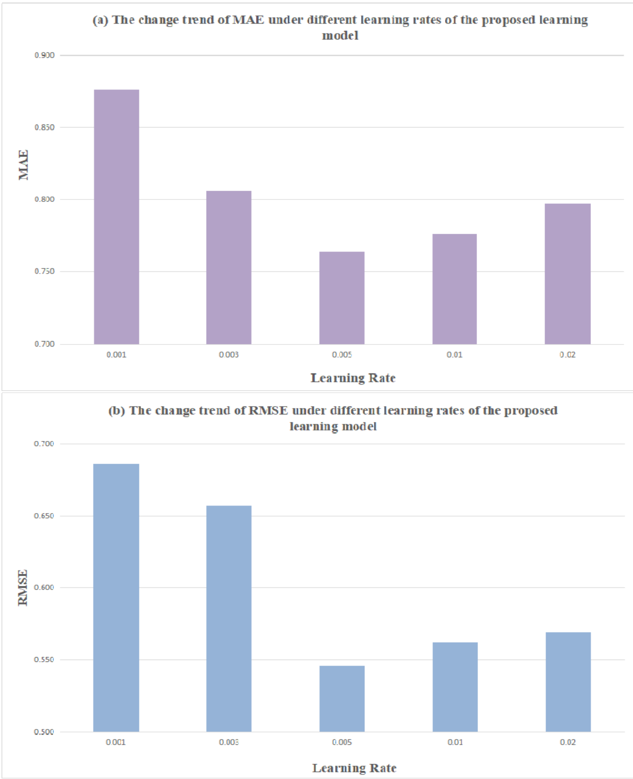

FIGURE 2. Impact of learning rate.

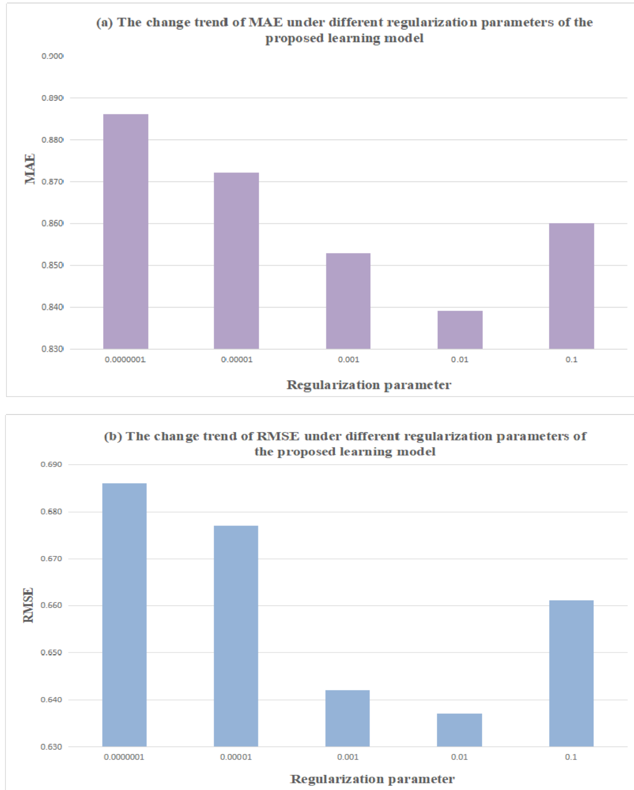

FIGURE 3. Impact of regularization parameter.

2) IMPACT OF REGULARIZATION PARAMETER

The regularization coefficient controls the regularization degree of the objective function, thus limiting the complexity of the model and avoiding the overfitting phenomenon, which has a great impact on the accuracy of rating prediction. Therefore, the repeated experiments are performed to verify the role of the regularization parameter in the model. The experimental results are presented in detail in Fig. 3. When the regularization parameter is too small, the accuracy of the predictions is relatively low, and the regularization term and network training are meaningless. However, when the parameter gradually increases to 0.01, the network performance reaches the optimal situation, where the MAE is 0.839, and the RMSE is 0.637. It can also be seen from Fig. 3 that the MAE and RMSE have similar impact tendencies.

D. PERFORMANCE COMPARISON ON RATING PREDICTION ACCURACY

It can be seen from the experimental results in the previous subsection that the proposed TDGIL model achieves the best performance when the learning rate and regularization parameter are 0.005 and 0.01, respectively. Therefore, these two optimal parameters are set for TDGIL in the next experiments. We observe the results of the comparative experiments in the training set of 15%, 30%, 45%, 60%, 75%, and 90%, as shown in Tables 2 and 3.

The similar trends of results are also obtained on the MovieLens 10M and Epinions datasets, as shown in Tables 2 and 3. More specifically, our proposed TDGIL model always achieves the best results in terms of MAE and RMSE among all the methods, while the accuracy of the UCF and SGR methods has been relatively poor. For example, when the Epinions dataset is set to 90%, the accuracy of TDGIL is improved by 17.37% in MAE and 22.18% in RMSE compared with UCF. These results show that the proposed approach can greatly improve the accuracy of item rating prediction.

At the beginning of dataset training, our method has greater improvement in accuracy. For example, when the MovieLens 10M dataset is set to 15%, the TDGIL algorithm improves the accuracy by 3.30% and 1.47%, respectively, in terms of MAE and RMSE compared with the second-best DLM method. In other words, due to inherent problems in recommender systems (cold start or data sparsity), the accuracy of many recommendation methods is relatively low, while our method is not obviously subject to the number of ratings. Therefore, from this perspective, the proposed model is more robust than other advanced methods to some extent. Consistently, we come to a conclusion that the TDGIL method outperforms the other approaches in accuracy, and significantly improves the performance of recommender systems.

E. VISUALIZATION OF PREDICTED RATINGS

To present the proposed model and experimental results more clearly, we visualize the predicted ratings of the proposed

TABLE 2. The predictive performance on the MovieLens 10M dataset.

	MAE					
	15%	30%	45%	60%	75%	90%
UCF	1.3361±0.0021	1.0006±0.0029	0.8327±0.0018	0.7611±0.0024	0.7533±0.0037	0.7419±0.0012
SGR	1.4634±0.0038	1.0255±0.0024	0.8422±0.0016	0.8027±0.0045	0.7672±0.0043	0.7544±0.0031
CPMF	1.3016±0.0064	0.9956±0.0043	0.8144±0.0025	0.7566±0.0032	0.7426±0.0011	0.7359±0.0028
BPR	1.2413±0.0025	0.9204±0.0020	0.7936±0.0037	0.7414±0.0016	0.7391±0.0068	0.7308±0.0009
DLM	1.2004±0.0053	0.8914±0.0027	0.8085±0.0019	0.7366±0.0027	0.7307±0.0033	0.7287±0.0018
TDGIL	**1.1608±0.0046**	**0.8492±0.0086**	**0.7673±0.0030**	**0.7283±0.0006**	**0.7162±0.0042**	**0.7039±0.0027**
	RMSE					
UCF	1.2996±0.0069	0.9013±0.0045	0.8638±0.0025	0.8301±0.0032	0.8063±0.0028	0.7508±0.0014
SGR	1.3191±0.0017	0.9444±0.0027	0.9094±0.0026	0.8655±0.0071	0.8276±0.0047	0.8004±0.0033
CPMF	1.1009±0.0046	0.8485±0.0018	0.8190±0.0008	0.7757±0.0012	0.7183±0.0036	0.6664±0.0027
BPR	1.2106±0.0028	0.8873±0.0034	0.8561±0.0017	0.8085±0.0010	0.7539±0.0024	0.6992±0.0056
DLM	0.9998±0.0031	0.8034±0.0023	0.7699±0.0022	0.7318±0.0037	0.6881±0.0016	0.6375±0.0084
TDGIL	**0.9851±0.0019**	**0.7726±0.0034**	**0.7468±0.0057**	**0.7102±0.0017**	**0.6588±0.0025**	**0.6126±0.0015**

TABLE 3. The predictive performance on the Epinions dataset.

	MAE					
	15%	30%	45%	60%	75%	90%
UCF	1.6038±0.0029	0.9789±0.0013	0.9127±0.0033	0.8667±0.0037	0.8218±0.0016	0.8014±0.0065
SGR	1.5246±0.0020	0.9427±0.0027	0.9015±0.0017	0.8429±0.0083	0.8107±0.0028	0.7857±0.0024
CPMF	1.3283±0.0049	0.9087±0.0008	0.8672±0.0019	0.8107±0.0024	0.7692±0.0067	0.7321±0.0007
BPR	1.4208±0.0015	0.9208±0.0031	0.8887±0.0021	0.8380±0.0058	0.7961±0.0004	0.7702±0.0018
DLM	1.3006±0.0014	0.8827±0.0035	0.8216±0.0037	0.7867±0.0025	0.7526±0.0054	0.7206±0.0040
TDGIL	**1.1862±0.0051**	**0.8418±0.0017**	**0.7762±0.0102**	**0.7325±0.0036**	**0.6991±0.0027**	**0.6622±0.0047**
	RMSE					
UCF	1.1129±0.0024	0.7694±0.0034	0.7208±0.0016	0.6815±0.0027	0.6672±0.0065	0.6533±0.0036
SGR	1.0005±0.0046	0.7437±0.0019	0.7183±0.0007	0.6518±0.0018	0.6322±0.0052	0.6109±0.0025
CPMF	0.9588±0.0024	0.6961±0.0020	0.6683±0.0017	0.6175±0.0035	0.5784±0.0023	0.5538±0.0094
BPR	0.9762±0.0011	0.7107±0.0015	0.6935±0.0026	0.6274±0.0037	0.6021±0.0072	0.5897±0.0050
DLM	0.9472±0.0017	0.6432±0.0022	0.6105±0.0019	0.5867±0.0026	0.5649±0.0010	0.5366±0.0048
TDGIL	**0.9177±0.0031**	**0.6219±0.0038**	**0.5972±0.0075**	**0.5533±0.0009**	**0.5344±0.0014**	**0.5084±0.0021**

model in a two-dimensional plane, where three categories of items (movies) randomly selected from the MovieLens 10M dataset are used for representation. Two visualizations of the conventional classic method (Benchmark predictor) and the proposed model are shown in Fig. 4. Different colors represent different categories.

As seen in Fig. 4, there are no obvious clusters and boundaries for positions of items in different categories of the conventional method, yet in our model, the items are clustered very well by different categories and the positions of items are regularly predicted. Therefore, from the perspective of visualization, the proposed

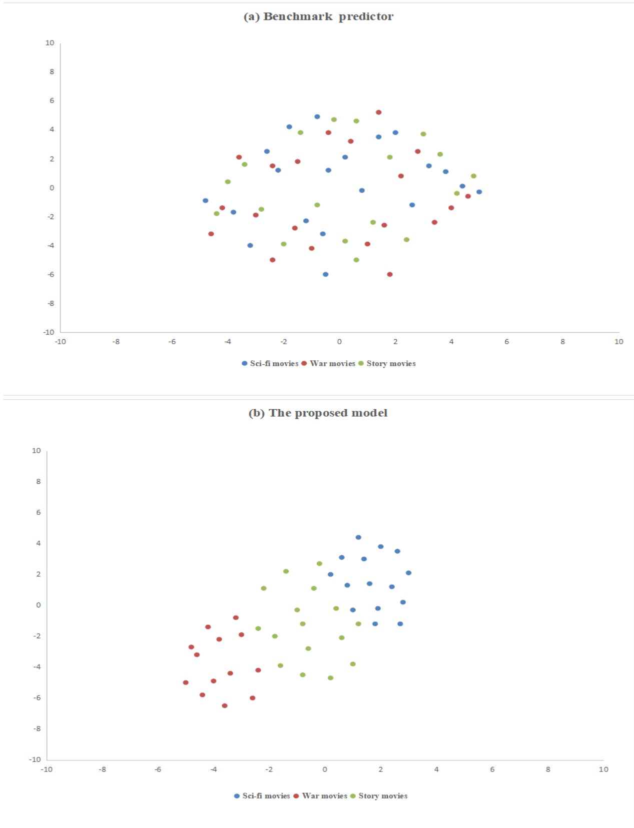

FIGURE 4. Clustering items in different categories.

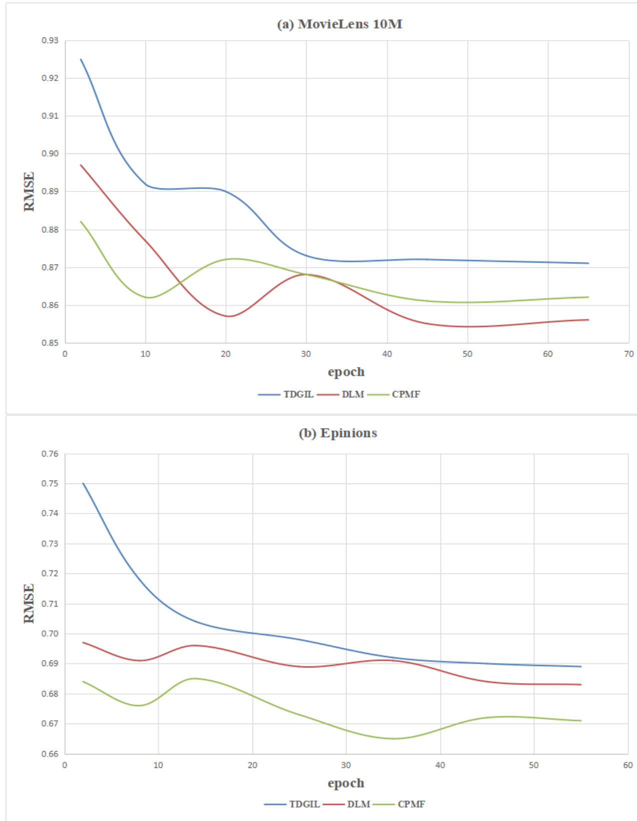

FIGURE 5. Convergence curves of our TDGIL model on MovieLens 10M and Epinions datasets.

TDGIL learning model is more accurate in item rating predictions.

F. CONVERGENCE AND COMPLEXITY ANALYSIS

The convergence curves of the proposed TDGIL and two other advanced methods on the MovieLens 10M and Epinions datasets are shown in Fig. 5. In the initial iteration, the RMSE of all methods is relatively high, and the model training is not stable. However, the RMSE of Fig. 5 becomes stable after four epochs. At this time, the experimental results tend to converge when the number of iterations reaches 35, and our TDGIL model always converges faster than other methods. Therefore, the proposed learning model has a better convergence rate and can be trained efficiently.

We trained the proposed learning model in a single GPU environment using Epinions and MovieLens 10M datasets. Table 4 displays the number of training parameters and the time cost of one epoch. (The parameters in Table 4 are in millions and the time is in minutes.) As seen in Table 4, although training time is relatively long in the proposed learning model, the time consumption is negligible throughout the rating prediction stage. This also proves that our learning model is very efficient.

G. ACCURACY ANALYSIS ON TOP-N TASK

In this subsection, we further evaluate the Top-N performance of our learning model and baseline methods on the Epinions dataset with precision, recall and F-measure metrics. Moreover, to obtain more accurate experimental results, the number of user neighbors is set from small to large, 5, 20, 35, 50, 65, 80 and 95. Fig. 6 shows the comparison performance, and the specific results and analysis are as follows.

TABLE 4. The complexity of our learning model.

Dataset	Parameters	Train Time	Test Time
Epinions	3.642	5	0.8
MovieLens 10M	10.289	130	7

As seen in Fig. 6 (a), it is clear that the precision performance of our learning model can significantly outperform the other five baseline methods. Additionally, when the user neighbor is equal to 35, our approach achieves the best 0.66. These results demonstrate that our model not only has impressive rating prediction accuracy but also has good Top-N accuracy.

Fig. 6 (b) and Fig. 6 (c) describe the comparison in terms of recall and F-measure, respectively. In contrast with conventional methods, our algorithm performs better as the number of user' neighbors increases. This is because the proposed trust diffusion and global items can capture more rating information to train the target neural network. Even compared to the popular advanced DLM model, our method has an obvious accuracy advantage in the rating prediction process.

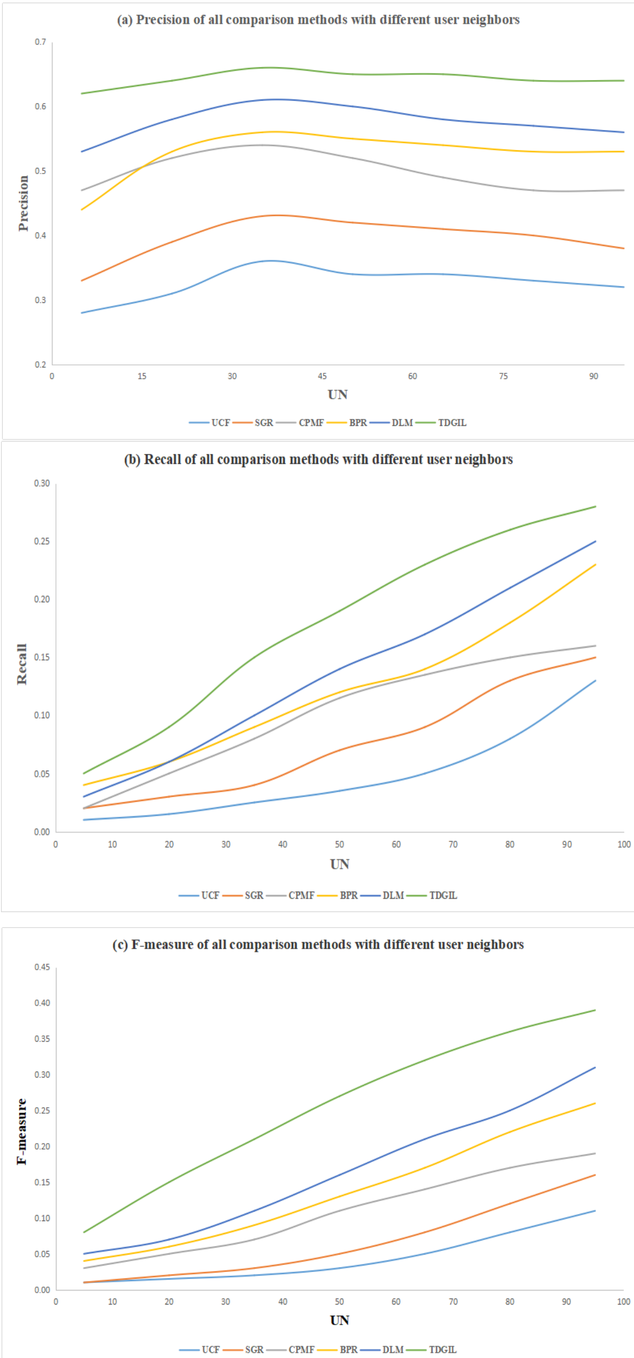

FIGURE 6. Accuracy comparison of Top-N recommendation under different methods.

Therefore, Fig. 6 (b) and Fig. 6 (c) above show that our learning model has impressive performance under both recall and F-measure indicators.

VI. DISCUSSION AND CONCLUSION

To conclude, in this paper, we proposed a novel learning model called TDGIL to improve the accuracy of item rating prediction in the recommendation systems. This framework considers both user trust diffusion and global items to generate final recommendations, which consists of two stages: 1) maximizing the use of item rating information in recommender systems through the trust diffusion feature among users and 2) generating predicted ratings by utilizing a global item model based on deep learning. The empirical analyses, over the MovieLens 10M and Epinions datasets, indicate that the proposed approach is a more accurate and efficient neural network training model than other advanced methods in terms of item rating prediction. Our aim for the future is to enrich the proposed learning model by incorporating temporal and contextual information into the neural network training.

ACKNOWLEDGMENT

The authors are grateful to the valuable comments and suggestions of the reviewers.

REFERENCES

[1] S. Bergamaschi, F. Guerra, and B. Leiba, "Guest editors' introduction: Information overload," *IEEE Internet Comput.*, vol. 14, no. 6, pp. 10–13, Nov./Dec. 2010.

[2] P. Resnick and H. R. Varian, "Recommender systems," *Commun. ACM*, vol. 40, no. 3, pp. 56–58, 1997.

[3] H. Zaragoza, "Information retrieval: Algorithms and heuristics," *Inf. Retr. J.*, vol. 5, nos. 2–3, pp. 271–274, Apr. 2002.

[4] A. P. Piotrowski, "Adaptive memetic differential evolution with global and local neighborhood-based mutation operators," *Inf. Sci.*, vol. 241, no. 12, pp. 164–194, Aug. 2013.

[5] L. Martínez, L. G. Pérez, and M. Barranco, "A multigranular linguistic content-based recommendation model," *Int. J. Intell. Syst.*, vol. 22, no. 5, pp. 419–434, May 2007.

[6] X. Luo, Y. Xia, and Q. Zhu, "Applying the learning rate adaptation to the matrix factorization based collaborative filtering," *Knowl.-Based Syst.*, vol. 37, pp. 154–164, Jan. 2013.

[7] Y. Shi, M. Larson, and A. Hanjalic, "Collaborative filtering beyond the user-item matrix: A survey of the state of the art and future challenges," *ACM Comput. Surv.*, vol. 47, no. 1, pp. 1–45, Jul. 2014.

[8] M. Ramezani, P. Moradi, and F. Akhlaghian, "A pattern mining approach to enhance the accuracy of collaborative filtering in sparse data domains," *Phys. A, Stat. Mech. Appl.*, vol. 408, pp. 72–84, Aug. 2014.

[9] H. Parvin, P. Moradi, and S. Esmaeili, "TCFACO: Trust-aware collaborative filtering method based on ant colony optimization," *Expert Syst. Appl.*, vol. 118, no. 15, pp. 152–168, 2019.

[10] C. L. P. Chen, C.-Y. Zhang, L. Chen, and M. Gan, "Fuzzy restricted Boltzmann machine for the enhancement of deep learning," *IEEE Trans. Fuzzy Syst.*, vol. 23, no. 6, pp. 2163–2173, Dec. 2015.

[11] T. Zhao, R. Zhao, and M. Eskenazi, "Learning discourse-level diversity for neural dialog models using conditional variational autoencoders," in *Proc. 55th Conf. Annu. Meeting Assoc. Comput. Linguistics*, 2017, pp. 654–664.

[12] B. K. Patra, R. Launonen, V. Ollikainen, and S. Nandi, "Exploiting bhattacharyya similarity measure to diminish user cold-start problem in sparse data," in *Proc. Int. Conf. Discovery Sci.* Cham, Switzerland: Springer, 2014, pp. 252–263.

[13] J. Liu, C. Wu, Y. Xiong, and W. Liu, "List-wise probabilistic matrix factorization for recommendation," *Inf. Sci.*, vol. 278, pp. 434–447, Sep. 2014.

[14] G. Lu, Y. Wang, and J. Zou, "Low-rank matrix factorization with adaptive graph regularizer," *IEEE Trans. Image Process.*, vol. 25, no. 5, pp. 2196–2205, May 2016.

[15] X. Zhao, Z. Niu, W. Chen, C. Shi, K. Niu, and D. Liu, "A hybrid approach of topic model and matrix factorization based on two-step recommendation framework," *J. Intell. Inf. Syst.*, vol. 44, no. 3, pp. 335–353, Jun. 2015.

[16] S. Xing, F. Liu, X. Zhao, and T. Li, "Points-of-interest recommendation based on convolution matrix factorization," *Appl. Intell.*, vol. 48, pp. 2458–2469, Aug. 2018.

[17] Y. Shi, M. Larson, and A. Hanjalic, "Mining contextual movie similarity with matrix factorization for context-aware recommendation," *ACM Trans. Intell. Syst. Technol.*, vol. 4, no. 1, Jan. 2013, Art. no. 16.

[18] J. Fan and J. Cheng, "Matrix completion by deep matrix factorization," *Neural Netw.*, vol. 98, pp. 34–41, Feb. 2018.

[19] S. Pero and T. Horváth, "Opinion-driven matrix factorization for rating prediction," in *Proc. Int. Conf. User Modeling, Adaptation, Personalization*, 2013, pp. 1–13.
[20] J. Herlocker, J. A. Konstan, and J. Riedl, "An empirical analysis of design choices in neighborhood-based collaborative filtering algorithms," *Inf. Retr.*, vol. 5, no. 4, pp. 287–310, Oct. 2002.
[21] Y. Hu, Q. Peng, X. Hu, and R. Yang, "Time aware and data sparsity tolerant Web service recommendation based on improved collaborative filtering," *IEEE Trans. Services Comput.*, vol. 8, no. 5, pp. 782–794, Sep. 2015.
[22] X. Yu, F. Jiang, J. Du, and D. Gong, "A user-based cross domain collaborative filtering algorithm based on a linear decomposition model," *IEEE Access*, vol. 5, pp. 27582–27589, Nov. 2017.
[23] J. Zhang, Y. Lin, M. Lin, and J. Liu, "An effective collaborative filtering algorithm based on user preference clustering," *Appl. Intell.*, vol. 45, no. 2, pp. 230–240, Sep. 2016.
[24] M. Gao, Z. Wu, and F. Jiang, "Userrank for item-based collaborative filtering recommendation," *Inf. Process. Lett.*, vol. 111, no. 9, pp. 440–446, Apr. 2011.
[25] M. Polato and F. Aiolli, "Exploiting sparsity to build efficient kernel based collaborative filtering for top-N item recommendation," *Neurocomputing*, vol. 268, pp. 17–26, Dec. 2017.
[26] J. Wang, A. P. de Vries, and M. J. T. Reinders, "Unifying user-based and item-based collaborative filtering approaches by similarity fusion," in *Proc. 29th Int. Conf. Res. Develop. Inf. Retr.*, Seattle, WA, USA, Aug. 2006, pp. 501–508.
[27] Y. Liu, L. Han, Z. Gou, and Y. Yang, "Personalized recommendation via trust-based diffusion," *IEEE Access*, vol. 7, pp. 94195–94204, 2019.
[28] T. K. Paradarami, N. D. Bastian, and J. L. Wightman, "A hybrid recommender system using artificial neural networks," *Expert Syst. Appl.*, vol. 83, pp. 300–313, Oct. 2017.
[29] J. Barbieri, L. G. M. Alvim, F. Braida, and G. Zimbrão, "Autoencoders and recommender systems: COFILS approach," *Expert Syst. Appl.*, vol. 89, pp. 81–90, Dec. 2017.
[30] Y. Liu, Q. Tong, Z. Du, and L. Hu, "Content-boosted restricted Boltzmann machine for recommendation," in *Proc. Int. Conf. Artif. Neural Netw.* Cham, Switzerland: Springer, 2014, pp. 773–780.
[31] M. Fu, H. Qu, Z. Yi, L. Lu, and Y. Liu, "A novel deep learning-based collaborative filtering model for recommendation system," *IEEE Trans. Cybern.*, vol. 49, no. 3, pp. 1084–1096, Mar. 2019.
[32] L. Zhang, T. Luo, F. Zhang, and Y. Wu, "A recommendation model based on deep neural network," *IEEE Access*, vol. 6, pp. 9454–9463, 2018.
[33] T. M. Phuong, T. C. Thanh, and N. X. Bach, "Neural session-aware recommendation," *IEEE Access*, vol. 7, pp. 86884–86896, 2019.
[34] Y. Li, J. Liu, and J. Ren, "Social recommendation model based on user interaction in complex social networks," *PLoS One*, vol. 14, no. 7, 2019, Art. no. e0218957.
[35] Q. Hu, Z. Han, X. Lin, Q. Huang, and X. Zhang, "Learning peer recommendation using attention-driven CNN with interaction tripartite graph," *Inf. Sci.*, vol. 479, pp. 231–249, Apr. 2019.
[36] N. Choudhary and K. K. Bharadwaj, "Leveraging trust behaviour of users for group recommender systems in social networks," in *Integrated Intelligent Computing, Communication and Security*, vol. 771. Singapore: Springer, 2019, pp. 41–47.
[37] T. Yu, J. Guo, W. Li, H. J. Wang, and L. Fan, "Recommendation with diversity: An adaptive trust-aware model," *Decis. Support Syst.*, vol. 123, Aug. 2019, Art. no. 113073.
[38] A. Karbasi, S. Ioannidis, and L. Massoulié, "From small-world networks to comparison-based search," *IEEE Trans. Inf. Theory*, vol. 61, no. 6, pp. 3056–3074, Jun. 2015.
[39] R. Berghammer and T. Hoffmann, "Relational depth-first-search with applications," *Inf. Sci.*, vol. 139, nos. 3–4, pp. 167–186, 2001.
[40] G. Linden, M. Conover, and J. Robertson, "The Netflix prize, computer science outreach, and Japanese mobile phones," *Commun. ACM*, vol. 52, no. 10, pp. 8–9, 2009.
[41] T. Pahikkala, A. Airola, M. Stock, B. D. Baets, and W. Waegeman, "Efficient regularized least-squares algorithms for conditional ranking on relational data," *Mach. Learn.*, vol. 93, nos. 2–3, pp. 321–356, Nov. 2013.
[42] M. Richardson and P. Domingos, "Mining knowledge-sharing sites for viral marketing," in *Proc. SIGKDD*, 2002, pp. 61–70.
[43] H. Liu, Z. Hu, A. Mian, H. Tian, and X. Zhu, "A new user similarity model to improve the accuracy of collaborative filtering," *Knowl.-Based Syst.*, vol. 56, pp. 156–166, Jan. 2014.
[44] I. Christensen, S. Schiaffino, and M. Armentano, "Social group recommendation in the tourism domain," *J. Intell. Inf. Syst.*, vol. 47, no. 2, pp. 209–231, 2016.
[45] X. Ren, M. Song, E. Haihong, and J. Song, "Context-aware probabilistic matrix factorization modeling for point-of-interest recommendation," *Neurocomputing*, vol. 241, pp. 38–55, Jun. 2017.
[46] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, "BPR: Bayesian personalized ranking from implicit feedback," in *Proc. 25th Conf. Uncertainty Artif. Intell.*, Montreal, QC, Canada: AUAI Press, Jun. 2009, pp. 452–461.
[47] K. J. Gurubel, A. Y. Alanis, E. N. Sanchez, and S. Carlos-Hernandez, "A neural observer with time-varying learning rate: Analysis and applications," *Int. J. Neural Syst.*, vol. 24, no. 1, 2014, Art. no. 1450011.
[48] L. Qiu, S. Gao, Q. Lyu, J. Guo, and P. Gallinari, "A novel non-Gaussian embedding based model for recommender systems," *Neurocomputing*, vol. 278, pp. 144–152, Feb. 2018.

YAKUN LI was born in 1988. He received the B.S. degree from the College of Information Science and Engineering, Yanshan University, China, in 2012, where he is currently pursuing the Ph.D. degree. He is currently focusing on a project on recommender systems. He is also proficient in Java and C++. His research interests include data mining and deep learning.

JIAOMIN LIU received the Ph.D. degree in electrical machinery and electrical appliances from the Hebei University of Technology. He is currently a Professor with the School of Information Science and Engineering, Yanshan University, China. He is also the Director of the Hebei Electrical Product Reliability Collaborative Innovation Center and the Chinese Association of Artificial Intelligence. His research interests include intelligent control, artificial intelligence, and data mining.

JIADONG REN is currently a Professor with the School of Information Science and Engineering, Yanshan University, China. His research interests include data mining, temporal data modeling, and software security. His research has been supported by the National Natural Science Foundation of China and the Science Foundation of Hebei Province. He is also a Senior Member of the Chinese Computer Society and a member of the IEEE SMC Society and ACM.