

Received 13 November 2023, accepted 4 December 2023, date of publication 7 December 2023, date of current version 15 December 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3340209

RESEARCH ARTICLE

Combining Social Relations and Interaction Data in Recommender System With Graph Convolution Collaborative Filtering

TIN T. TRAN¹, VACLAV SNASEL¹, (Senior Member, IEEE), AND LOC TAN NGUYEN²

¹Faculty of Electrical Engineering and Computer Science, VSB—Technical University of Ostrava, 708 00 Ostrava, Czech Republic

²Faculty of Information Technology, Ton Duc Thang University, Ho Chi Minh City 700000, Vietnam

Corresponding author: Tin T. Tran (trung.tin.tran.st@vsb.cz)

ABSTRACT A recommender system is an important subject in the field of data mining, where the item rating information from users is exploited and processed to make suitable recommendations with all other users. The recommender system creates convenience for e-commerce users and stimulates the consumption of items that are suitable for users. In addition to e-commerce, a recommender system is also used to provide recommendations on books to read, movies to watch, courses to take or websites to visit. Similarity between users is an important impact for recommendation, which could be calculated from the data of past user ratings of the item by methods of collaborative filtering, matrix factorization or singular vector decomposition. In the development of graph data mining techniques, the relationships between users and items can be represented by matrices from which collaborative filtering could be done with the larger database, more accurate and faster in calculation. All these data can be represented graphically and mined by today's highly developed graph neural network models. On the other hand, users' social friendship data also influence consumption habits because recommendations from friends will be considered more carefully than information sources. However, combining a user's friend influence and the similarity between users whose similar shopping habits is challenging. Because the information is noisy and it affects each particular data set in different ways. In this study, we present the input data processing method to remove outliers which are single reviews or users with little interaction with the items; the next proposed model will combine the social relationship data and the similarity in the rating history of users to improve the accuracy and recall of the recommender system. We perform a comparative assessment of the influence of each data set and calculation method on the final recommendation. We also propose and implement a model and compared it with base line models which include NGCF, LightGCN, WiGCN, SocialLGN and SEPT.

INDEX TERMS Recommendation system, social recommender system, collaborative filtering, graph convolution network.

I. INTRODUCTION

E-commerce develops strongly and gives users a good experience not only with complete information about products but also conveniences for customers such as providing items by catalog or comparing among products. Items selected according to certain criteria will be recommended to customers with relevant characteristics, such as age, gender,

The associate editor coordinating the review of this manuscript and approving it for publication was Fabrizio Messina¹.

interests, or place of residence. Storing customer and product information is a challenging task because the number of customers and products is very large, increasing rapidly and diversifying sources of information collected. With a huge size database of users and items, the speed of access will limit the effectiveness of e-commerce. The collaborative filtering (CF) algorithm calculates the similarity between users [1], [2], without retrieving their personal information. This algorithm evaluates the similarity between customers based on the large number of items they have interacted

with such as purchasing, rating, viewing, commenting. And then, the system will recommend to a target customer items in the interactive list of other customers who have a highly similarity with them. In these models, the correlation between the user and the item is recorded by an interaction matrix, which can be implicit or explicit. If the data are implicit, the matrix element will store whether the user is interested in or shopping for the item. If the data are explicit, the matrix element value represents the user's rating of the item. The similarity between two users is the distance between two vectors representing them, each of which contains information about that user's interactions with all items.

The explosion in the number of users and items leads to a huge matrix size, which will take up a large amount of memory for storage as well as computing resources for operation on matrices. The matrix factorization (MF) technique can decompose the interaction matrix into two or more matrices of smaller size without losing the original useful information. Singular value decomposition (SVD) algorithm generalizes the eigen decomposition of a square matrix which has an ortho normal eigen basis to any size matrix, and finally, the feature matrix is found with smaller size and minimum loss. However, most of the elements in these matrices have the value of zero, because almost users interested only a certain number of items and has no interaction with the rest of the item list. It made matrix sparse, they need to be compressed to both reduce the amount of memory needed as well as facilitate the computation. By matrix transformations, information is collected into embedding matrices and can be efficiently processed using TensorFlow and Keras libraries [3]. As the final step of the computation, the embedding matrices can be converted back to the original interaction matrix size for the front-end layers to make recommendations to the user.

Besides exploiting user and item interaction information, users' social relationships are also essential because they show the user's influence in real life. A shopping recommendation from a friend will have a greater impact on a user's decision than other advertising sources. Today's social networking platforms such as Foursquare, Facebook, Gowalla also provide a database of users and their friendship relationships. Mining these facts will enrich the information and support not only the recommendation process but also specific user groups [4], [5].

Both the interactions between users and items and the social relationships between users can be graphically represented and mined using graph neural network (GNN) [6]. In a graph, neighboring nodes in some hops show that they are related. Implicit information could be discovered by travel through the graph and look for nearby users who have the most common items reference. These high-order had been found in Neural graph collaborative filtering model (NGCF) [7]. Each matrix will also be transformed into embedded matrices that are significantly reduced in dimensions compared to the original matrix but still contain all the original information. The embedding matrices are decomposed to capture the

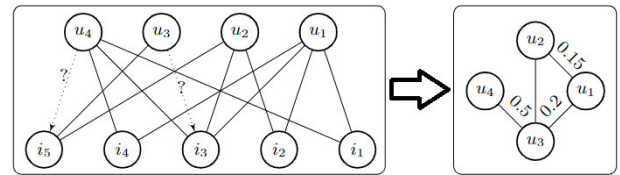


FIGURE 1. Collaborative Filtering algorithms calculates the correlation between users.

influence signals and then combine them into a more informative embedding matrix [8]. The learning process can be repeated many times to reach the target accuracy, in the iterations propagated embedding matrices are used to capture high-order connectivity in the item and user interaction graph.

In this publication, we propose a GCN model that could analyst and synthesize from three sources of information: the product and user interaction matrix, the influence weighted matrix between users and the matrix of social friendships derived from social network platforms. We also implement the model and conduct experiments on well-known data sets and compare the results with the base line models. This article will be shown the next sections. Chapter 2 will summarize the research work that are the foundation of the topic and the latest techniques. Chapter 3 propose model in which we will show how to preprocess data sets, the organization of neural layers, and the signal aggregator and decomposer. Chapter 4 presents the overall results and compares them with previous state-of-the-art models. Chapter 5 is our conclusions and some ideas for future research.

II. LITERATURE REVIEW

A. COLLABORATIVE FILTERING

Recommender system in the first stage exploited features of all items as well as the preferences of users to find suitable items and give reasonable recommendations to users [9]. For example, a book can be categorized into comics, detective, historical, archaeology, fine arts, and others. With users, information such as age, gender, address of residence or education could also used to enrich the input data [10]. However, with a very large number of items and frequent additions, it is difficult to find out and give items attributes, the systems need additive filtering methods. Collaborative filtering approach algorithms were used to find similarity between users or items without attribute information from them [11].

Collaborative filtering can be implemented by memory-based models [12] or model-based models [13]. With memory-based model, the history of every user's rating on items will be recorded in a rating matrix. There are many ways to define a rating scale, which can be an integer value from 1 to 5, or an implicit rating. In a space made up of user set, item set, and rating values, each user is represented by a feature vector e_u , also as each item is represented by feature vector e_i . The algorithms will find out the distance between each pair of users (or pair of items for item-based algorithms) to find the neighbors and form the recommended results.

There are many methods to calculate distance such as Cosine function, Jaccard similarity or Mean Squared Differences [8].

In the model-based collaborative filtering systems, the algorithms will search for patterns in the learning data to develop a model for future prediction [14]. The matrix that contains rating of users on items can be reduced by using MF techniques. In order to ensure accuracy of distance between users and items the rating matrix can be divided into user feature matrix and item feature matrix with smaller sizes [15]. The higher accuracy of user similarity calculating, the better recommendation prediction results.

The Graph convolutional matrix completion (GCMC) built a graph-based auto-encoder framework for matrix completion [16]. This model moved from matrix to graph by defining the main problem as prediction task on a bipartite graph. Graph construction also benefits from having additional sources of information such as friendships in social networks, which are also represented by graphs. The encoders traverse the graph and record signals from vertices, representing users and items, producing latent features of these nodes. The of collaborative filtering problems between rows (or columns) has now become about measuring the predicted edge weights between two objects, which present users or items.

B. GRAPH CONVOLUTION NETWORKS

A graph is a type of data structure that uses a set of vertices V to represent an object and a set of edges E to represent a relationship to depict the link between entities, people, data, and qualities. An edge on a graph may be directed or undirected, specifying the relationship's weight or just the relationship itself. In order to extract input data and identify properties of them, neural networks have had remarkable success setting up hidden layers [17], [18]. In the publication [19], neural networks were used with graph input data, and when vertex features were propagated and aggregated into one another during the learning process [20], favorable results were obtained. Graph neuron network (GNN) methods, which inherit from neural network algorithms, employ several propagation layers as well as various aggregation and update techniques. By means of pooling or attention techniques, the neighbor vertex's characteristics as well as those of the target vertex are better modified [21].

Graph Convolutional Network (GCN) is an iterative method of collecting information [22]. For example, when there is a similar item of interest to many of the target user's friends, it should be recommended with a higher degree than other items. GraphSAGE [23] embedded inductive for each vertex of the graph and learned the topological structure of the graph as well as the effect of vertices on neighboring vertices. This method not only focus on feature-rich graphs but also make use of structural features of all vertices.

NGCF [7] generates hop-by-hop propagation classes in the input graph under the assumption that the effect of the users varies depending on the distance between the graph's vertices. Attention signals from a user's neighbors are received by

the k^{th} layer of propagation, also known as k - order propagation, which also receives messages from items to that user. The input parameter k can be viewed as the number of propagation layers, and the value $k = 3$ is concluded to be the best one. The difference between the test set's actual and expected assessment value serves as the foundation for the loss function. Depending on the learning rate, the size of the embedding matrices, and the quantity of the learning data, the learning process to minimize the loss function occurs after a few iterations. LightGCN [24] deleted the feature weight transformation matrices during propagation and a non-linear activation function to remove adverse impacts on objects in the NGCF propagation process in a later version of the algorithm.

C. SOCIAL RECOMMENDER SYSTEM

The social counseling system has been developed since the emergence of online social platforms. In addition to user behavior, restrictions such as liking, sharing and commenting; Additional data from social networks is used to provide personalized headlines to users considering the influence of their friends [5], [25], [26]. Loads of information on social networks have influenced and made users of the same interests as their friends [27], [28]. ContextMF [29] recorded the social relational information in the recommendation data in a frame design that was collected from matrix coefficients and with regularization terms. This model has improved the accuracy of the recommendation results when the model only performs MF with the user and item relation matrix. The TrustSVD model [25] developed from SVD++ [30] incorporates the influence of friends on social relations as an implicit feedback add-on for an observing user. GCN usage recommendation models are also extended to include more user social relation information, such as GraphRec [31] which viewed a user in both importance matrix aspects. social system and in the matrix of interactions with items. The data influence between users was aggregated and conditioned by the Social Aggregation module during machine learning.

SocialLGN [32] designed deep models to capture the most distinctive characteristics of social and embedded networks and users of mixed embeddings, instead learning directly from their interaction data only users and items. Moreover, their model still uses GCN to study the reference of users whose information influence is affected through the process of social variables pervasive in social networks. User influence propagation also propagates through multiple logging layers, techniques used in NGCF [7] and Light GCN [24].

Technically, the process of enhancement between social relational data and user influence in shopping is a challenge because they change according to the time and nature of each kind of items. In the Socially-Aware Self-Supervised Tri-training (SEPT) model [33], three histogram encoders were created to get the times signal from the social relationship between users, the user relationship when interested in

TABLE 1. Ratio between number of users to number of items in surveyed datasets.

Model	Dataset	#Users	#Items	Ratio
NGCF	Gowalla	29,858	40,981	1.37
	Amazon-book	52,643	91,599	1.74
	Yelp2018	31,668	38,048	1.20
SocialLGN	Yelp	17,237	38,342	2.22
	Flickr	8,358	82120	9.83
SEPT	Last.fm	1,892	17,632	9.32
	Douban-Book	13,024	22,347	1.72
	Yelp	19,539	21,266	1.01
Our study	Gowalla	28,511	40,968	1.44
	Librarything	9,988	15,985	1.60
	Ciao	5,785	108,651	18.78
	Epinions	1,497	17,898	11.95

similar items and information shared information - augment the data between social information and shopping interaction information. The self-supervised learning process in the above model has both improved the effectiveness of recommendations.

III. OUR PROPOSED MODEL

In this section, we present a proposed model to exploit the rating matrix and social relationship between users to make recommendations for any user in the future. The model will be presented with data processing module, GCN module and prediction layer with loss calculator. An overview on our model show as Figure 2.

A. DATA PRE-PROCESSING

The data sets are always recorded continuously and redundantly and included noise data that are users and items with very little interaction. Eliminating these signals will limit recommendations to items that are no longer available. Furthermore, the reduced data sets, smaller size but without losing valuable information, will decrease the memory size required, as well as cut down the computation time of the models. In our experiments, we apply a 10-core setting, which means that users with less than 10 interactions will be dropped at this processing step.

With the selected set of items, we also keep only the number of users with the highest interactions that are measured by the Jaccard distance between the set of items that a user interacts with and the set of all filtered items in the database. The ratio of number of users to number of items can be selected equal to the ratio of the original data set. The ratio of all survey data sets in the publication [7], [24]. We summarize this ratio in Table 1. Users with a low Jaccard measure were removed as outliers, and the social friendship matrix is being rebuilt accordingly.

After pre-processing the data set, we've got 3 matrices as input to the model, where matrix A presents interaction between users and items, matrix C contains the similarity degree between users and matrix S represents social friendship of users. The matrix S is an optional matrix, which could be skipped if a data set does not provide a real social

Algorithm 1 Filter Out Items With Less Than 10 Interactions

Input: $\mathcal{U} \times \mathcal{I}$ {Interaction between users and items in original dataset}

Output: $R \subseteq \mathcal{U} \times \mathcal{I}$

```

1: for all item  $i \in \mathcal{I}$  do
2:   if item  $i$  has at least 10 interaction then
3:     {10-core setting}
4:      $I \leftarrow$  item  $i$ 
5:   end if
6: end for
7:  $p \leftarrow$  cardinality of set  $I$ 
8:  $q \leftarrow p \div \text{selected\_ratio}$ 
9: for all user  $u \in \mathcal{U}$  do
10:   $set_u \leftarrow$  list of items interacted by user  $u$ 
11:   $sim_u =$  Jaccard distance between set  $I$  and  $set_u$ 
12: end for
13:  $U \leftarrow q$  users have highest  $sim_u$ 
14: return  $R = U \times I$ 

```

relationship; or when the noise of social relation data is higher than the actual effect among users.

1) THE INTERACTION MATRIX A

The user-item interaction data set can be provided as an implicit or an explicit. Implicit data records interactions such as item view, item purchase, mouse click, or like item as a binary value of 0 or 1, while explicit data specifies a user rating an item with a rating value in a certain range, for example an integer between 0 and 5. In this article, we consider the interactions between the user and the item are implicit recorded and expressed by a bipartite graph $G = \{V, E\}$, where vertices set $V = \{U, I\}$ is union from set of users U and items set I ; the E contains edge $e_{i,j} = (u_i, i_j)$ if user u had interaction on item i . Thus, the matrix $R \subseteq \mathcal{U} \times \mathcal{I}$, represents the graph G , could be defined by (1).

$$R_{i,j} = \begin{cases} 1 & \text{if user } u_i \text{ refers to item } i_j \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

Interactive filtering by GCN will capture users' characteristics into user embedding while features of items will be learned into item embedding. In order for these two embedding matrices to be updated at the same time to facilitate propagation, we design the input matrix to be a Laplacian matrix showed in (2). If the matrix R represents user interactions on the items, the transpose matrix R^T will represent the items that the users interacted with. 0 is a zero matrix of suitable size.

$$A = \begin{bmatrix} 0 & R \\ R^T & 0 \end{bmatrix} \quad (2)$$

Because of convenience in calculation the embeddings in next layer in our graph convolution operations [7], the symmetrically normalized matrix \tilde{A} should be calculated once by (3) and accessed every time the embeddings are

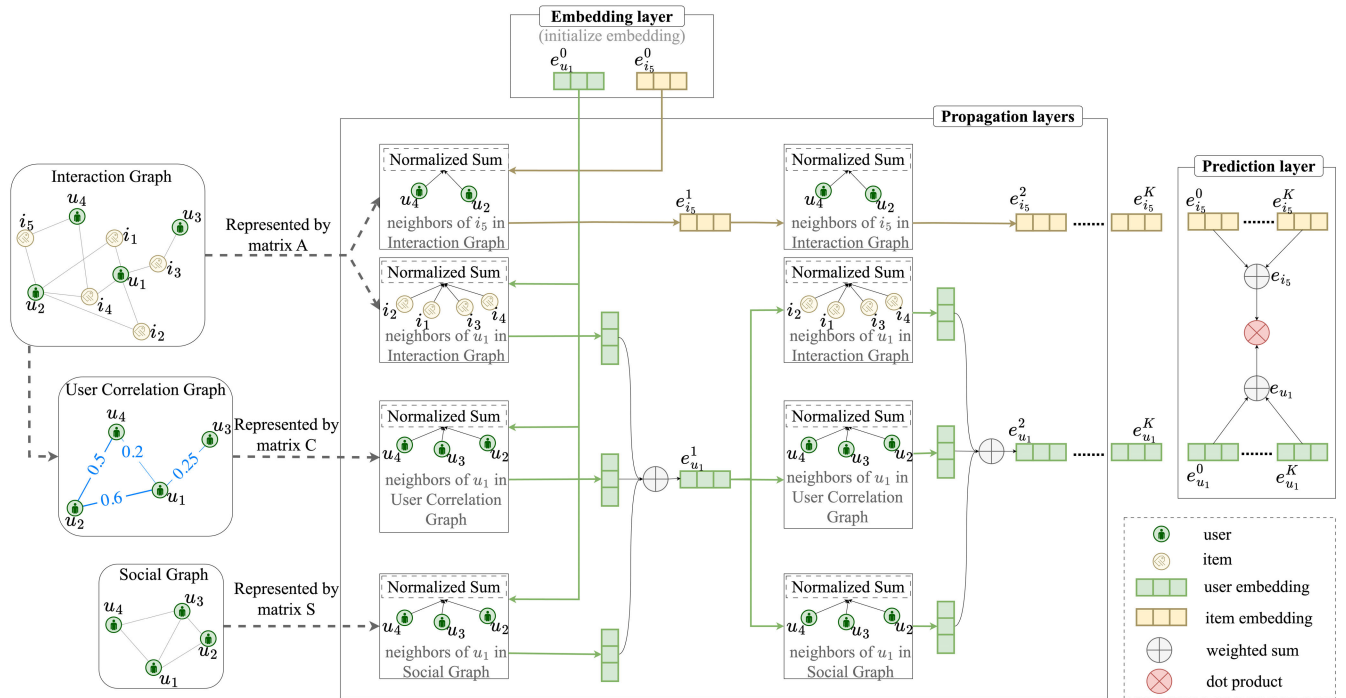


FIGURE 2. Overview on our proposed model.

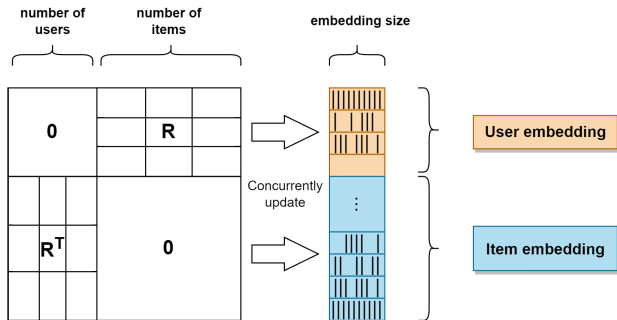


FIGURE 3. The form of matrix M helps embeddings updated concurrency.

convoluted.

$$\tilde{A} = D^{-\frac{1}{2}} A D^{-\frac{1}{2}} \quad (3)$$

where D is the diagonal degree matrix with each entry $D_{i,i}$ has value of number of nonzero entries in the $i - th$ row vector of degree matrix A and $D_{i,j} = 0$ with $i \neq j$. Finally, \tilde{A} is used as one of input sources of our GCN model.

2) THE USERS CORRELATION MATRIX C

The similarity between each pair of customers in the database depends on how many common items they interact with. Because the greater the number of items of common interest, the higher the influence between them. So that, in the publication [34], a weighted user influence matrix W_I was inputted as an additional source of information. This matrix can be calculated by $W_I = R \times R^T$ and it shows how many common items that user u_i and u_j have interaction by value of $W_{I_{i,j}}$. On the other hand, W_I also could be calculated by (5).

However, the matrix W_I only shows the number of intersections between the two sets of items I_i and I_j , but has not recorded the influence of couple of users $\{i, j\}$ to all interaction data. The higher the total number of items both of customers has interacted with, the more similarity should be counted, since it is clear that they interact with the items more often than other customers. We propose the matrix W_U has the same size as the matrix W_I and has the following initialization value in (4) and (5) with where: I_i and I_j are the sets of items interacted by user u_i and u_j , respectively.

$$W_U = |I_i| + |I_j| - |I_i \cap I_j| = |I_i \cup I_j| \quad (4)$$

$$W_I = |I_i \cap I_j| \quad (5)$$

Therefore, element-wise product $W_U \odot W_I^{-1}$ will return a normalized matrix whose elements get a value in range from 0.0 to 1.0, based on the Jaccard similarity between each pair of users.

Furthermore, GCN-based models collect the collaborative signals from the high-order connectivity graph. The matrix $W_U \odot W_I^{-1}$ inadvertently adds weight to the associations already extracted from high-order connectivity graph i.e. it makes the model more focused on the interaction points extracted from the high-order connectivity graph. This leads to a model that lacks extensible and is prone to over fitting. To solve this problem we use a function to convert value of elements of $W_U \odot W_I^{-1}$ into the typical values which showed in Table 2.

The conversion process of the resulting matrix $W_U \odot W_I^{-1}$ also demonstrates the clustering of all users into multiple

TABLE 2. Represents the function f classified elements in the $W_U \odot W_I^{-1}$ result matrix into denotation values.

Jaccard index	[0;0.1)	[0.1;0.4)	[0.4;0.6)	[0.6;0.9)	[0.9;1.0]
Typical value	0.0	0.005	0.05	0.5	1.0
Denote	No interaction	Very low effective	Median effective	High effective	Strongly effective

groups through their influence level, which is represented by the value of the matrix C .

Summarized all above ideas, the matrix C , which represents the degree of similarity between the behavior of the users, is implemented as the (6) with f is the value allocation conversion function, shown in Table 2. W_U is weighted user references matrix and W_I is the matrix that represents the union of two lists of items two users.

$$C = f(W_U \odot W_I^{-1}) \tag{6}$$

3) THE SOCIAL FRIENDSHIP MATRIX S

Because the data of social relation is not always provided in a data set, it is an optional part of our recommendation system. In the next part of the experiment, we will present the comparative results with social relations and without it. Some data sets and publications review the user i and user j relationships are represented as a directed edge in a graph if the user i follow user j (j might not know about i). In this article, we treat the relationship of any couple of users as equal, or in other words, the edge of the relation between them in the graph is an undirected edge. The social relation matrix $S \subseteq U \times U$ denotes the friendship among users in the real social life. If user i and user j are friends, $S_{ij} = 1$, otherwise $S_{ij} = 0$.

In recent studies, the matrix S is used to enhance users-items relationships in input data. SEPT model has created a sharing view A_s [33], calculated as (7). A_s represents the relationship weight between users not only by interesting in common items but also depending on whether they have a true relationship or not.

$$A_s = (RR^T) \odot S \tag{7}$$

We keep our matrix S separate from interaction data A and users correlation values C in the input to evaluate the influence of friendship relations on overall recommendation accuracy. We leave research on integrating social signals into other matrices for future research.

B. GRAPH CONVOLUTION NETWORK LAYERS

1) TRANSFORMATION WEIGHT MATRICES AND ACTIVATION FUNCTION

The NGCF model represents the most advanced models that use GCN for mining and recommendation. NGCF is designed with feature transformation matrices as well as a nonlinear activation function like other GCN standards [35]. Then

the non-linear activation will incorporate embedding in the output and smoothing the values so as not to introduce a biased value in overall result. The LeakyReLU function is commonly used in GCN models [7], [34].

However, developing the NGCF model on data with little description, where both user entries and each other are stored with only the valid ID, does not bring any benefit to the feature learning process. On the contrary, computer resources are spent a lot on storage and operations on digital matrices. The use of weighted transformation matrices and activation function has been discussed in [24]. In this publication, the LightGCN model eliminates all the weighted transformation W_i along with the activation function and achieves better overall results than the NGCF model. We again test this method on the SocialLGN model [32] with the variants eliminating either the transformation matrices, the function activations, or both.

- **SocialLGN-f** removes all feature transformation matrices W_1, W_2 and W_3 .
- **SocialLGN-n** eliminates the non-linear activation function $\sigma(\cdot)$.
- **SocialLGN-fn** eliminates both the feature transformation matrix W_i and the nonlinear activation function $\sigma(\cdot)$.

In all experiments, we apply the same hyperparameters with publications NGCF and LightGCN, including learning rate, regularization, dropout rate, Gowalla and Yelp2018 data. The results show in Figure 4 with Gowalla data set and Figure 5 with Yelp data set.

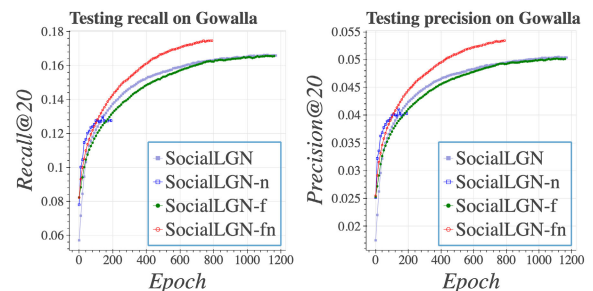


FIGURE 4. Compare variants of SocialLGN on data set Gowalla.

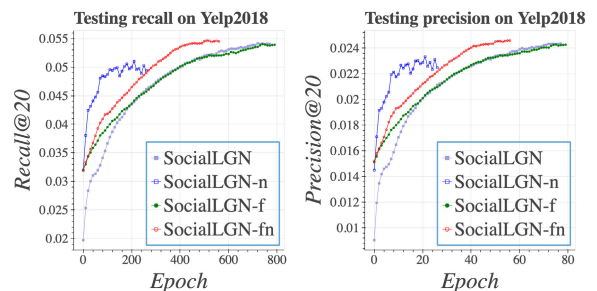


FIGURE 5. Compare variants of SocialLGN on data set Yelp2018.

When removing either feature transformation matrices W_1, W_2, W_3 or non-linear activation functions $\sigma(\cdot)$, the model performance is unstable and worse than the original

SocialLGN, but removing both feature transformation matrices and activation function the model improves significantly. Through this experiment we proved our inference about the negative impact of graph fusion operation to embedding users. To replace the graph fusion operator, at SocialLGN-fn we simply sum the embedding users social and the embedding users interaction together to generate embedding users for each propagation.

From the above presentation, we determine that eliminating both the feature transformation matrices and the nonlinear activation function is appropriate for the data sets considered in this article.

2) EMBEDDING LAYERS

Follow the model in publication [36] used two vectors for user's latent and item's latent, to define as a result are users embeddings and items embeddings, we also define users embeddings and items embeddings as $e_u \in \mathbb{R}^d$ and $e_i \in \mathbb{R}^d$ with d is the embeddings vector size. In the first round of propagation, embedding layer $e^{(0)} = e_u^{(0)} \parallel e_i^{(0)}$ will be initialized with normal weight initialization method. Based on the architecture of LGC [24], the initialized embeddings $e_u^{(0)}$ and $e_i^{(0)}$ were the only trainable parameters of our model. In matrix form we denote $E^{(0)} \in \mathbb{R}^{(n+m) \times d}$ is the set of all embeddings during propagation, i.e. $E^{(0)}$ contains the set of n user embeddings and m item embeddings.

$$E^{(0)} = E_U^{(0)} \parallel E_I^{(0)} = [e_{u_1}^{(0)}, \dots, e_{u_n}^{(0)}, e_{i_1}^{(0)}, \dots, e_{i_m}^{(0)}] \quad (8)$$

3) PROPAGATION PROCESS

In order to capture signals from three input matrices, every k propagation layers we propagate the k^{th} user and item embeddings e_u^k and e_i^k on the LGC architecture [24] and obtain four embeddings containing three signals are $e_{ua}^{(k+1)}$, $e_c^{(k+1)}$, $e_s^{(k+1)}$, $e_i^{(k+1)}$. Where $e_{ua}^{(k+1)}$ and $e_i^{(k+1)}$ hold user-item interaction signals in the matrix A , $e_s^{(k+1)}$ contain social signals between users in matrix S , and $e_c^{(k+1)}$ capture correlation signals between users in matrix C .

$$\begin{aligned} e_{ua}^{(k+1)} &= \sum_{i \in \mathcal{N}_u^A} \frac{1}{\sqrt{|\mathcal{N}_u^A| |\mathcal{N}_i^A|}} e_i^{(k)} \\ e_s^{(k+1)} &= \sum_{s \in \mathcal{N}_u^S} \frac{1}{\sqrt{|\mathcal{N}_u^S| |\mathcal{N}_s^S|}} e_u^{(k)} \\ e_c^{(k+1)} &= \sum_{c \in \mathcal{N}_u^C} \frac{1}{\sqrt{|\mathcal{N}_u^C| |\mathcal{N}_c^C|}} e_u^{(k)} \\ e_i^{(k+1)} &= \sum_{u \in \mathcal{N}_i^A} \frac{1}{\sqrt{|\mathcal{N}_i^A| |\mathcal{N}_u^A|}} e_u^{(k)} \end{aligned} \quad (9)$$

where $|\mathcal{N}_q^X|$ denote the number of neighboring users (or items) of item(or user) q in the matrix X , $X = [A, S, C]$. The essence of equation (9) is based on the symmetric normalization element was used in most GCN model [7], [24], [34] and was calculated by (3).

Then the $(k + 1)^{\text{th}}$ user embeddings are aggregated according to the equation (10), specifically, we combine three embedding by using the weighted sum of the embeddings.

$$e_u^{(k+1)} = \text{AGGREGATION}_k \left(e_{ua}^{(k+1)}, e_c^{(k+1)}, e_s^{(k+1)} \right) \quad (10)$$

After some loops of propagation process, we've got $(K + 1)$ embeddings corresponding to the number of propagations through K layers and including the initial embedding. The final embedding of users (and items) will be obtained by (11).

$$e_u = \frac{1}{K} \sum_{k=0}^K e_u^{(k)}; \quad e_i = \frac{1}{K} \sum_{k=0}^K e_i^{(k)} \quad (11)$$

Using equation (11), we takes the mean value of all embeddings at all layers, and that equation can be replaced with some other functions like maximum, median, weighted average. In order not to complicate the model and still ensure good performance in general, we use the mean function.

4) PREDICTION AND OPTIMIZATION

Based on the final embedding e_u and e_i calculated by (11), we calculate the prediction score of item i for user u by equation (12).

$$\hat{y}_{ui} = e_u^\top e_i \quad (12)$$

We build the loss function with Bayesian personalize ranking (BPR) for model to learn the parameters Φ which only include the user and item embedding. BPR is the best suitable method for implicit feedback data sets [37], it assumes observed interactions Ω_{ui}^+ have higher preferences than an unobserved interactions Ω_{ij}^- . To optimize the prediction model we use mini-batch Adam [38] and minimize the BPR loss in (13).

$$\text{Loss}_{BPR} = \sum_{\Omega_{ui}^+} \sum_{\Omega_{ij}^-} -\ln \sigma(\hat{y}_{ui} - \hat{y}_{ij}) + \lambda \|\Phi\|_2^2 \quad (13)$$

5) MATRIX FORM

To facilitate the implementation of the model, we expressed the propagation process in the form of matrix in (14).

$$\begin{aligned} E^{(k+1)} &= E_U^{(k+1)} \parallel E_I^{(k+1)} \\ &= (E_{UA}^{(k+1)} + E_S^{(k+1)} + E_C^{(k+1)}) \parallel E_I^{(k+1)} \\ &= (\tilde{R} E_I^{(k-1)} + \tilde{S} E_U^{(k-1)} + \tilde{C} E_U^{(k-1)}) \parallel \tilde{R}^\top E_U^{(k-1)} \end{aligned} \quad (14)$$

where \tilde{C} and \tilde{S} is a symmetrically normalized matrix of C and S as in (15).

$$\begin{aligned} \tilde{S} &= D_S^{-\frac{1}{2}} S D_S^{-\frac{1}{2}} \\ \tilde{C} &= D_C^{-\frac{1}{2}} C D_C^{-\frac{1}{2}} \end{aligned} \quad (15)$$

\tilde{R} and \tilde{R}^\top appeared as components in the input \tilde{A} in (3).

TABLE 3. Statistic of the experiment data sets.

Data set	#Users	#Items	#Edges	#Social
Gowalla	28,551	40,968	992,860	265,704
Librarything	9,988	15,985	372,967	28,736
Ciao	5,785	108,651	283,034	101,245
Epinions	1,497	17,898	25,191	9,493

IV. RESULTS

A. EXPERIMENTAL SETTING

We use setting 10-core with all data sets to ensure that each user has at least ten interactions. For each data set, we selected 20% of interactions which have the latest timestamp for testing set and the remaining (80% interactions) are for process of training. Summary of all data sets showed in Table 3. We configure our model with embedding fixed size to 64 for all models and the embedding parameters are initialized with the normalized weight method. The optimization process was done with Adam algorithm [38].

1) DATA SETS

- **Gowalla:** First published in [39], Gowalla is a web-based application that provides location-based social networking. Users can check-in and share public places. Gowalla also collects the friendship network of users and provides these data as an undirected graph.
- **Librarything:** is a data from a book review website called Library Thing [40]. It is an online service to help people organize their books easily and people can communicate with each other.
- **Ciao:** is an online shopping site that records users' ratings of items with timestamps. Customers can rate an item with a score from 1 to 5. Users can also add others to their friend lists and build a social network [41].
- **Epinions:** is a well-known consumer review site where users can rate items and add social friends to their trust lists [42]. The Ciao and Epinions data sets have been widely selected as benchmark data sets for social recommendations.

2) EVALUATION CRITERIA

The evaluative measurement should be selected appropriately for each algorithm [43]. A commonly evaluating method is dividing a data set into a test set and a train set. The algorithm is applied on the train set to make predictions and evaluated these results on the test set. The difference between the learning result and the actual data value shows the accuracy of the algorithm of proposed model. This difference can be represented in mean absolute error (MAE) and root mean square error (RMSE). Besides accuracy, recall, scalability, learning time, memory consumption or interpretability are also an important criterion in evaluating the recommended system.

For implicit data, interactions between the user and the item are recorded binary, rather than rated as a specific

value. Then, the algorithms will use the accuracy measure for the classification. The commonly used metrics are Precision and Recall [44], [45]. Precision is the ratio between correct predictions on the test set, and Recall is the sensitivity of the algorithm, or the proportion of relational assertions that have been retrieved from the test set.

$$\begin{aligned} \text{Precision} &= \frac{TP}{TP + FP} \\ \text{Recall} &= \frac{TP}{TP + FN} \end{aligned} \quad (16)$$

where TP is true positive set of correctly predicting interaction exists between users and items. FP is false positive set of missing prediction of interaction while FN (false negative) presents the number of predicted interactions not being exist on the test set.

Furthermore, Discounted Cumulative Gain score (DCG) [35] assumes that judges have assigned labels to each result and accumulates across the result vector a gain function G applied to the label of each result, scaled by a discount function D of the rank of the result, and it's normalized by the dividing DCG of an ideal result vector I . We get Normalized Discounted Cumulative Gain (NDGC) as in (17).

$$\text{NDGC}_u = \frac{\text{DCG}_u}{\text{CDG}_{max}} \quad (17)$$

In this publication, we use Precision and Recall (16) and NDCG (17) that considered at 20 items (NDGC@20).

3) BASE LINE MODELS

To evaluate our proposed model, we compare it with the following state-of-the-art models as base lines.

- **MF** [15] is the matrix factorization model demonstrated in the Netflix pricing competition. The model outperforms classic nearest neighbor techniques for generating product recommendations and allows the inclusion of additional information such as implicit feedback, temporal effects, and confidence. Even though this model is outdated, we still put it in comparison to show the evolution of recommendation models.
- **Social MF** [46] incorporates the trust information and its propagation into an MF model. The signals from direct friends are embedded in the final matrix.
- **Trust SVD** [25] is a trust-based MF technique, is built on the SVD++ algorithm. This model handles the explicit and implicit influence of rated items, by incorporating reviews from trusted friends, to make prediction of items for an active user.
- **GCMC** [16] assumes that making a future recommendation is a prediction of the association between the user and the item in the graph. Interaction data is represented by one bipartite graph with labeled edges denotes the observed ratings. The model then uses a graph auto encoding framework based on message passing on interactive graphs.
- **NGCF** [7] is one of the most cutting-edge GCN models. It performs propagation operations on embeddings using

a few iterations. High-order connectivity in the interactions graph is contained in the stacked embeddings on the output. The latent vectors contain the collaborative signal, which strengthens the model.

- **WiGCN** [34] based on NGCF model, the WiGCN added a weighted matrix as an additional input. That matrix contains the influence of users on each other. It leads to more data-gathering propagation and increased recommendation performance.
- **LightGCN** [24] in order to concentrate on the neighborhood aggregation element for CF, this model eliminates the weight matrices and the activation function. The users and items embeddings for the interaction graph are learned using linear propagation in this model. The weighted total of all learnt embeddings becomes the final embedding.
- **SocialLGN** [32] concentrate on the CF's neighborhood aggregation component. The users and items embeddings for the interaction graph are learned using linear propagation in this model. The weighted total of all learnt embeddings becomes the final embedding.
- **SEPT** [33] from the user social information, this model augments the user data views with the user social information. And then the framework builds three graph encoders upon the augmented views and iteratively improves each encoder with self-supervision signals from other two encoders.

B. OVERALL RESULT COMPARISON

Summary of all results from all models on the processed data are displayed in the Table 4 with three values in each data set: precision, recall and ndcg measured in 20 items. In order from the top row down, the chemical process of recommender system models can be clearly identified, that is, the group of methods using GCN has better results than the group of classical methods such as MF or SVD. In the group of GCN methods, models that do not use the transformation matrices, such as LightGCN, give better results than the remaining models. When compared in the group of models that do not consider social relationships, that is MF, GCMC, NGCF, LightGCN, WiGCN, our model with interaction embedding (**Our w/ interaction**) still gives the highest results.

In each pair of models such as MF and Social MF, LightGCN and SocialLGN, LightGCN and SEPT, almost models that use social relationship give better results; This confirms that mining actual friend data are as important as mining the correlation between users through items of common interest. Our proposed model also efficiently accounts for social relationship data when embedding them in the propagation process.

C. DETAILED MODEL ANALYSIS

1) EFFECT OF SOCIAL RELATION ON OVERALL RESULT

We compare each pair of models that are architecturally similar to each other in Table 5. In each row of comparison,

one model exploits the users and items relationship data while the other model adds users and users relationship data. In a pair of models, the signal propagation and reception formulas all have a high degree of similarity, using (or not using) the weighted transformation matrices and non-linear activation functions. The parameters setting in the experiment are the same.

- 1) From MF to SocialMF: in the traditional MF model, the characteristics of each user have been obtained from the decomposition of the users - items relationship matrix. Meanwhile, the SocialMF model also extracts characteristics of a user's direct friends to enrich information for that user. However, the SocialMF model did not deepen the indirect relationships between the user community, so it missed many valuable signals.
- 2) From LightGCN to SocialLGN: this is the only case where the results are worse across all measures. The process of combining signals from mining similarities between users with the social relationships of users is not in harmony, resulting in friend information not only reducing the accuracy of the recommendation results but also scattering the CF results.
- 3) From LightGCN to SEPT: although both SocialGCN and SEPT are models that extend from LightGCN with friendship information, it is clear that the SEPT model has optimized the embedding of the friendship matrix between users into the model and brings the benefits big improvement.
- 4) We also tested the influence of friendship data on the overall results by removing the S matrix in the model input and comparing the results between the model without the S matrix (Our model with only interaction data) and full model (Our model-all). We calculated the influence weights between users and clustering gathered those weights into each relationship level as shown in the Table 2. By doing that way, our model avoided the damage of SocialLGN that still captures valuable friend signals like in the SocialMF model and explores indirect relationships like other GCN models.

Although friendship data has been shown to influence the final recommendation results, it should be noted that relationships in real world change over time and can introduce bias in recommendations. An example is that friends in the same type of hobby will find it difficult to give good advice about another type of hobby.

2) TIME PERFORMANCE ANALYSIS

The time cost of our model lies in the interaction and social relation graph aggregation. Therefore, the total time complexity is acceptable in practice. We try to make the environment run the experiments as similar as possible each time, the average time results of each epoch compared to each data set on the LGC-based models are almost similar, the figures statistics are given in Table 6.

TABLE 4. Overall performance comparisons.

Data set	Gowalla			Librarything			Ciao			Epinion		
	recall	precision	ndcg*	recall	precision	ndcg*	recall	precision	ndcg*	recall	precision	ndcg*
MF	0.1091	0.0308	0.0920	0.0472	0.0114	0.0348	0.0318	0.0105	0.0222	0.0049	0.0006	0.0026
SocialMF	0.1167	0.0327	0.0980	0.0493	0.0121	0.0367	0.0348	0.0103	0.0241	0.0074	0.0009	0.0036
TrustSVD	0.1376	0.0380	0.1120	0.0539	0.0137	0.0374	0.0366	0.0107	0.0248	0.0067	0.0008	0.0035
GCMC	0.1357	0.0385	0.1097	0.0571	0.0151	0.0366	0.0466	0.0139	0.0334	0.0075	0.0009	0.0036
NGCF	0.1622	0.0445	0.1299	0.0680	0.0166	0.0442	0.0506	0.0149	0.0353	0.0118	0.0015	0.0065
WiGCN	0.1632	0.0445	0.1321	0.0734	0.0173	0.0455	0.0522	0.0153	0.0369	0.0122	0.0015	0.0058
LightGCN	0.1808	0.0496	0.1488	0.0768	0.0185	0.0507	0.0571	0.0168	0.0431	0.0150	0.0022	0.0085
SocialLGN	0.1705	0.0467	0.1411	0.0741	0.0180	0.0509	0.0521	0.0158	0.0384	0.0102	0.0015	0.0060
SEPT	0.1872	0.0525	0.1519	0.0763	0.0184	0.0499	0.0586	0.0176	0.0439	0.0138	0.0018	0.0071
Our w/ interact	0.1846	0.0506	0.1509	0.0770	0.0188	0.0509	0.0592	0.0173	0.0433	0.0139	0.0019	0.0073
Our w/ social	0.1924	0.0530	0.1564	0.0774	0.0185	0.0498	0.0578	0.0170	0.0439	0.0100	0.0019	0.0055
Our model-all	0.1940	0.0537	0.1572	0.0804	0.0191	0.0533	0.0594	0.0176	0.0444	0.0175	0.0023	0.0096

* ndcg@20

TABLE 5. Improvement with social friendship data on Gowalla data set.

Users-Items models	Social models	recall	precision	ndcg@20
MF	SocialMF	↑ 7.0%	↑ 6.2%	↑ 6.5%
LightGCN	SocialLGN	↓ 5.7%	↓ 5.8%	↓ 5.2%
LightGCN	SEPT	↑ 3.5%	↑ 5.8%	↑ 2.1%
Our w/ interaction	Our model-all	↑ 5.1%	↑ 6.1%	↑ 4.2%

TABLE 6. The number of epochs the models learned and the average time (in seconds) per epoch.

	WiGCN	LightGCN	SEPT	Our w/ interact	Our w/ social	Our model-all
Gowalla	350 (58.2s)	1140 (20.4s)	1090 (22.8s)	490 (22.5s)	1040 (22.8s)	440 (23.7s)
LibraryThing	380 (32.3s)	980 (12.3s)	970 (13.1s)	350 (13.3s)	990 (13.3s)	570 (13.5s)
Ciao	70 (25.8s)	240 (9.1s)	220 (10.2s)	180 (9.7s)	260 (10.1s)	160 (10.2s)
Epinions	50 (12.5s)	170 (4.2s)	160 (4.8s)	90 (5.3s)	170 (4.3s)	140 (4.9s)

An extremely outstanding feature of our proposed model is its convergence speed. Not only does it have higher recommendation predictions than other models, but its time efficiency is also significantly shorter than LGC architecture-based models such as LightGCN, SocialLGN, and SEPT. Specifically, in the Gowalla data set, our model with only interaction data takes 490 epochs and Our model only takes 440 epochs, while the two models SEPT and LightGCN take 1090 and 1140 epochs, respectively. For the remaining two data sets, Ciao and Epinions, there is also a similar trend, which demonstrates very good performance of the model training process. We conclude that the user correlation matrix helps the model focus on important data, thereby accelerating the learning process of our model.

3) POSITIVE IMPACT OF USER INTERACTION MATRIX

Our model converges faster than the baseline models, shown in the Figure 6 and 7. In our previous work at WiGCN model, adding a weight matrix as input and generating strong attention signals for nodes in the graph during propagation.

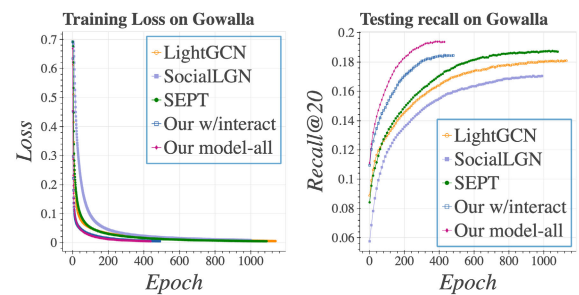


FIGURE 6. Training curves of LGC-based models, which are evaluated by training loss and testing recall@20 on Gowalla data set.

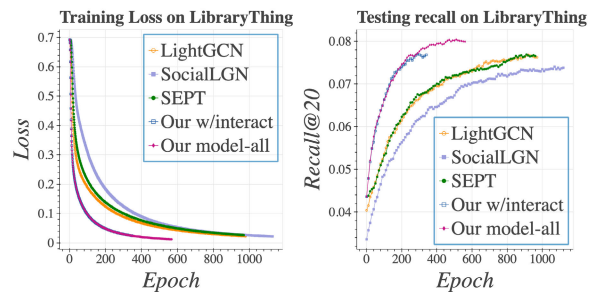


FIGURE 7. Training curves of LGC-based models, which are evaluated by training loss and testing recall@20 on LibraryThing data set.

The user correlation weight matrix based on the set of shared items only needs to be calculated once using basic matrix operations and used for all embedding layers. Therefore, in this article, the user's correlation matrix with carefully preprocessed $U \times I$ data has created a very high convergence speed for the propagation process.

D. EXPLAIN THE INFLUENCE OF INTERACTION DATA AND SOCIAL NETWORK FRIENDSHIP DATA

One of our main research questions is the influence of interaction data and social friendship data when considered separately and when aggregated. We implemented the model in a modular form, which can enable or disable interactive embedding and social friend embedding, to make our model explainable. In Table 7 and 8, we compare 4 models:

TABLE 7. Influence of interaction data and social network friendship data on Gowalla data set.

Models	recall	precision	ndcg@20
LightGCN	0.1808	0.0496	0.1488
Our w/ interaction	0.1846 (+2.1%)	0.0506 (+2.0%)	0.1509 (+1.4%)
Our w/ social	0.1924 (+6.4%)	0.0530 (+6.9%)	0.1564 (+5.1%)
Our full model	0.1940 (+7.3%)	0.0537 (+8.3%)	0.1572 (+5.6%)

TABLE 8. Influence of interaction data and social network friendship data on Librarything data set.

Models	recall	precision	ndcg@20
LightGCN	0.0768	0.0185	0.0507
Our w/ interaction	0.0770 (+0.3%)	0.0188 (+1.6%)	0.0509 (+0.4%)
Our w/ social	0.0774 (+0.8%)	0.0185 (+0%)	0.0498 (-1.8%)
Our full model	0.0804 (+4.7%)	0.0191 (+3.2%)	0.0533 (+5.1%)

LightGCN (can be considered a baseline model without interaction matrix and social network friendship data); our model only with interaction data or social network friendship data; and our full model.

We conclude that for data sets related to geographical locations on social networking platforms, social friendship data create the main impact. It contributes largely to the increase in precision and recall in our model. For platforms where friendship network is just a secondary function, such as the Librarything data set in Table 8, which have social network just for private messaging, the interaction data plays a crucial role. Finally, when combining both sources of data, the best results are achieved.

V. CONCLUSION AND FUTURE WORKS

A. CONCLUSION

In this work, we have presented an effective model that includes data filtering to remove noisy interactions between users and items, evaluating the influence of the weight matrix and activation function on the result, and a method of consolidating data when the similarity of users comes from multiple information sources, as well as comparing the effective recommender system models presented recently.

Designing the input with multi-embedding makes them quickly capture interaction signals and significantly increase the efficiency of the model. Our model has inherited the most effective features from the previously introduced algorithms, which are using the high-order connectivity learning structure, removing the weight matrix, and eliminating the nonlinear activation function. We also evaluated the influence of social friendships on the recommendation process and pointed out some of the difficulties of using this data.

B. FUTURE WORK

During the process of conducting research, we proposed several issues that need to be further explored. That is group

recommendation, which gives the results not only for a single user but for a whole group of friends. That recommendation model is useful in contexts where a group of friends go to the movies together, go to the same attraction, or a group of students take a course. Another problem is that items have been treated as individuals; they are not clustered or grouped into catalogs. This may have omitted a lot of information from the set of items in the final recommendation result. In fact, items are closely related when they belong to the same commodity group, when items are in geographically distant locations, or when the item is in movies in the same genre.

REFERENCES

- [1] J. Zhang, Z. Lin, B. Xiao, and C. Zhang, "An optimized item-based collaborative filtering recommendation algorithm," in *Proc. IEEE Int. Conf. Netw. Infrastruct. Digit. Content*, Nov. 2009, pp. 414–418.
- [2] A. S. Tewari, A. Kumar, and A. G. Barman, "Book recommendation system based on combine features of content based filtering, collaborative filtering and association rule mining," in *Proc. IEEE Int. Advance Comput. Conf. (IACC)*, Feb. 2014, pp. 500–503.
- [3] D. Grattarola and C. Alippi, "Graph neural networks in TensorFlow and keras with spektral [application notes]," *IEEE Comput. Intell. Mag.*, vol. 16, no. 1, pp. 99–106, Feb. 2021.
- [4] H. Ma, H. Yang, M. R. Lyu, and I. King, "SoRec: Social recommendation using probabilistic matrix factorization," in *Proc. 17th ACM Conf. Inf. Knowl. Manage.*, Oct. 2008, pp. 931–940, doi: 10.1145/1458082.1458205.
- [5] H. Ma, D. Zhou, C. Liu, M. Lyu, and I. King, "Recommender systems with social regularization," in *Proc. 4th ACM Int. Conf. Web Search Data Mining*, Jan. 2011, pp. 287–296.
- [6] M. Dossena, C. Irwin, and L. Portinale, "Graph-based recommendation using graph neural networks," in *Proc. 21st IEEE Int. Conf. Mach. Learn. Appl. (ICMLA)*, Dec. 2022, pp. 1769–1774.
- [7] X. Wang, X. He, M. Wang, F. Feng, and T.-S. Chua, "Neural graph collaborative filtering," 2019, *arXiv:1905.08108*.
- [8] L. Candillier, F. Meyer, and M. Boullé, "Comparing state-of-the-art collaborative filtering systems," in *Proc. Int. Workshop Mach. Learn. Data Mining Pattern Recognit.*, 2007, pp. 548–562.
- [9] M. Jalili, S. Ahmadian, M. Izadi, P. Moradi, and M. Salehi, "Evaluating collaborative filtering recommender algorithms: A survey," *IEEE Access*, vol. 6, pp. 74003–74024, 2018.
- [10] M. Pazzani, "A framework for collaborative, content-based and demographic filtering," *Artif. Intell. Rev.*, vol. 13, pp. 393–408, Dec. 1999.
- [11] G. Adomavicius, R. Sankaranarayanan, S. Sen, and A. Tuzhilin, "Incorporating contextual information in recommender systems using a multidimensional approach," *ACM Trans. Inf. Syst.*, vol. 23, no. 1, pp. 103–145, Jan. 2005.
- [12] E. Bojnordi and P. Moradi, "A novel collaborative filtering model based on combination of correlation method with matrix completion technique," in *Proc. 16th CSI Int. Symp. Artif. Intell. Signal Process. (AISP)*, May 2012, pp. 191–194.
- [13] D. Goldberg, D. Nichols, B. M. Oki, and D. Terry, "Using collaborative filtering to weave an information tapestry," *Commun. ACM*, vol. 35, no. 12, pp. 61–70, Dec. 1992, doi: 10.1145/138859.138867.
- [14] K. Yu, X. Xu, J. Tao, M. Ester, and H.-P. Kriegel, "Instance selection techniques for memory-based collaborative filtering," in *Proc. SIAM Int. Conf. Data Mining*, Apr. 2002, pp. 59–74, doi: 10.1137/1.9781611972726.4.
- [15] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, vol. 42, no. 8, pp. 30–37, Aug. 2009.
- [16] R. van den Berg, T. N. Kipf, and M. Welling, "Graph convolutional matrix completion," 2017, *arXiv:1706.02263*.
- [17] Q. Guo, F. Zhuang, C. Qin, H. Zhu, X. Xie, H. Xiong, and Q. He, "A survey on knowledge graph-based recommender systems," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 8, pp. 3549–3568, Aug. 2022.
- [18] Y. Gao, Y.-F. Li, Y. Lin, H. Gao, and L. Khan, "Deep learning on knowledge graph for recommender system: A survey," 2020, *arXiv:2004.00387*.
- [19] S. Wu, F. Sun, W. Zhang, X. Xie, and B. Cui, "Graph neural networks in recommender systems: A survey," 2020, *arXiv:2011.02260*.

- [20] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu, "A comprehensive survey on graph neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 1, pp. 4–24, Jan. 2021, doi: [10.1109/TNNLS.2020.2978386](https://doi.org/10.1109/TNNLS.2020.2978386).
- [21] X. Wang, H. Ji, C. Shi, B. Wang, P. Cui, P. Yu, and Y. Ye, "Heterogeneous graph attention network," 2019, *arXiv:1903.07293*.
- [22] R. Ying, R. He, K. Chen, P. Eksombatchai, W. L. Hamilton, and J. Leskovec, "Graph convolutional neural networks for web-scale recommender systems," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2018, pp. 974–983, doi: [10.1145/3219819.3219890](https://doi.org/10.1145/3219819.3219890).
- [23] W. L. Hamilton, R. Ying, and J. Leskovec, "Inductive representation learning on large graphs," 2017, *arXiv:1706.02216*.
- [24] X. He, K. Deng, X. Wang, Y. Li, Y. Zhang, and M. Wang, "LightGCN: Simplifying and powering graph convolution network for recommendation," 2020, *arXiv:2002.02126*.
- [25] G. Guo, J. Zhang, and N. Yorke-Smith, "A novel recommendation model regularized with user trust and item ratings," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 7, pp. 1607–1620, Jul. 2016.
- [26] M. Jiang, P. Cui, F. Wang, W. Zhu, and S. Yang, "Scalable recommendation with social contextual information," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 11, pp. 2789–2802, Nov. 2014.
- [27] R. M. Bond, C. J. Fariss, J. J. Jones, A. D. I. Kramer, C. Marlow, J. E. Settle, and J. H. Fowler, "A 61-million-person experiment in social influence and political mobilization," *Nature*, vol. 489, no. 7415, pp. 295–298, Sep. 2012, doi: [10.1038/nature11421](https://doi.org/10.1038/nature11421).
- [28] J. Qiu, J. Tang, H. Ma, Y. Dong, K. Wang, and J. Tang. (2018). *DeepInf: Modeling Influence Locality in Large Social Networks*. [Online]. Available: <https://api.semanticscholar.org/CorpusID:49239721>
- [29] J. Wang, D. Bagul, and S. Srihari. (2018). *ContextMF: A Fast and Context-Aware Embedding Learning Method for Recommendation Systems*. [Online]. Available: <https://api.semanticscholar.org/CorpusID:198186939>
- [30] Y. Koren, "Factorization meets the neighborhood: A multifaceted collaborative filtering model," in *Proc. 14th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2008, pp. 426–434, doi: [10.1145/1401890.1401944](https://doi.org/10.1145/1401890.1401944).
- [31] W. Fan, Y. Ma, Q. Li, Y. He, E. Zhao, J. Tang, and D. Yin, "Graph neural networks for social recommendation," 2019, *arXiv:1902.07243*.
- [32] J. Liao, W. Zhou, F. Luo, J. Wen, M. Gao, X. Li, and J. Zeng, "SocialLGN: Light graph convolution network for social recommendation," *Inf. Sci.*, vol. 589, pp. 595–607, Apr. 2022. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0020025522000019>
- [33] J. Yu, H. Yin, M. Gao, X. Xia, X. Zhang, and N. Q. V. Hung, "Socially-aware self-supervised tri-training for recommendation," 2021, *arXiv:2106.03569*.
- [34] T. T. Tran and V. Snašel, "Improvement graph convolution collaborative filtering with weighted addition input," in *Intelligent Information and Database Systems*. Cham, Switzerland: Springer, 2022, pp. 635–647, doi: [10.1007/978-3-031-21743-2_51](https://doi.org/10.1007/978-3-031-21743-2_51).
- [35] M. Najork and F. McSherry, "Computing information retrieval performance measures efficiently in the presence of tied scores," in *Proc. 30th Eur. Conf. IR Res. (ECIR)*, Apr. 2008, pp. 635–647. [Online]. Available: <https://www.microsoft.com/en-us/research/publication/computing-information-retrieval-performance-measures-efficiently-in-the-presence-of-tied-scores/>
- [36] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua, "Neural collaborative filtering," 2017, *arXiv:1708.05031*.
- [37] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, "BPR: Bayesian personalized ranking from implicit feedback," 2012, *arXiv:1205.2618*.
- [38] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.
- [39] E. Cho, S. A. Myers, and J. Leskovec, "Friendship and mobility: User movement in location-based social networks," in *Proc. 17th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2011, pp. 1082–1090, doi: [10.1145/2020408.2020579](https://doi.org/10.1145/2020408.2020579).
- [40] T. Zhao, J. McAuley, and I. King, "Improving latent factor models via personalized feature projection for one class recommendation," in *Proc. 24th ACM Int. Conf. Inf. Knowl. Manage.*, Oct. 2015, pp. 821–830, doi: [10.1145/2806416.2806511](https://doi.org/10.1145/2806416.2806511).
- [41] J. Tang, H. Gao, and H. Liu, "MTrust: Discerning multi-faceted trust in a connected world," in *Proc. 5th ACM Int. Conf. Web Search Data Mining*, Feb. 2012, pp. 93–102, doi: [10.1145/2124295.2124309](https://doi.org/10.1145/2124295.2124309).
- [42] M. R. Hamedani, I. Ali, J. Hong, and S. Kim, "TrustRec: An effective approach to exploit implicit trust and distrust relationships along with explicit ones for accurate recommendations," *Comput. Sci. Inf. Syst.*, vol. 18, pp. 93–114, Jan. 2021.
- [43] J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl, "Evaluating collaborative filtering recommender systems," *ACM Trans. Inf. Syst.*, vol. 22, no. 1, pp. 5–53, Jan. 2004, doi: [10.1145/963770.963772](https://doi.org/10.1145/963770.963772).
- [44] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Application of dimensionality reduction in recommender system a case study," Defense Tech. Inf. Center, Virginia, Tech. Rep. ADA439541, Aug. 2000. Accessed: Sep. 15, 2023. [Online]. Available: <https://apps.dtic.mil/sti/pdfs/ADA439541.pdf>
- [45] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Analysis of recommendation algorithms for e-commerce," in *Proc. 2nd ACM Conf. Electron. commerce*, Oct. 2000, pp. 158–167, doi: [10.1145/352871.352887](https://doi.org/10.1145/352871.352887).
- [46] M. Jamali and M. Ester, "A matrix factorization technique with trust propagation for recommendation in social networks," in *Proc. 4th ACM Conf. Recommender Syst.*, Sep. 2010, pp. 135–142, doi: [10.1145/1864708.1864736](https://doi.org/10.1145/1864708.1864736).



TIN T. TRAN received the M.S. degree in computer science from the HCMUT University of Technology, Ho Chi Minh City, Vietnam, in 2012. He is currently pursuing the Ph.D. degree with the VSB—Technical University of Ostrava, Czech Republic. He is a Lecturer with the Faculty of Information Technology, Ton Duc Thang University, Vietnam. His research interests include artificial intelligence, social networks, graph neural networks, and location-based application.



VACLAV SNAŠEL (Senior Member, IEEE) research and development experience includes over 25 years in the industry and academia. He works in a multi-disciplinary environment involving artificial intelligence, multidimensional data indexing, conceptual lattice, information retrieval, semantic web, knowledge management, data compression, machine intelligence, neural networks, web intelligence, data mining, and applied to various real-world problems. He has given more than ten plenary lectures and conference tutorials in these areas. He has authored/coauthored several refereed journal/conference papers and book chapters. He has published more than 400 papers (147 are recorded at Web of Science). He has supervised many Ph.D. students from Czech Republic, Jordan, Yemen, Slovakia, Ukraine, and Vietnam.



LOC TAN NGUYEN received the B.S. degree in computer science from Ton Duc Thang University, Ho Chi Minh City, Vietnam, in 2023. His research interests include data science, data processing, social networks, and graph neural networks.

• • •