Contents lists available at ScienceDirect

Electronic Commerce Research and Applications

journal homepage: www.elsevier.com/locate/elerap

ImprovedGCN: An efficient and accurate recommendation system employing lightweight graph convolutional networks in social media

Sanjeev Dhawan [a], Kulvinder Singh [a], Adrian Rabaea [b,*], Amit Batra [a]

[a] Department of Computer Science and Engineering, University Institute of Engineering and Technology, Kurukshetra University, Kurukshetra 136119, Haryana, India
[b] Department of Mathematics and Computer Science, North University Center of Baia Mare, Technical University of Cluj-Napoca, Victoriei 76, Baia Mare 430122, Romania

ARTICLE INFO

Keywords:
Recommendation system
Graph neural network
Embedding
Collaborative filtering

ABSTRACT

Graph Convolutional Networks (GCNs) have emerged as a hot topic of interest for collaborative filtering among researchers in the recent past. The research which exists in literature and is applied to recommendation does not analyze all the facets of GCN, as GCN is introduced for graph classification activities. It is observed that the two facets of GCNs namely, feature transformation and non-linear activation have a small influence on increasing the effectiveness of *collaborative filtering* (CF). Furthermore, the inclusion of these two facets increases the complexity of training and even decreases the recommendation performance. In this paper, a novel approach namely *Improved Graph Convolutional Network* (ImprovedGCN) has been proposed which only makes use of the important part of GCN termed neighborhood aggregation for CF. The aforesaid model can be implemented and trained which leads to significant improvements as compared to a similar approach termed *Neural Graph Collaborative Filtering* (NGCF).

1. Introduction

To circumvent the problem of information explosion on the internet, a *recommendation system* (RS) has been extensively employed to carry out filtering of the personalized data (Covington et al., 2016; Ying et al., 2018; Yuan et al., 2020). The significant objective of the RS is to create a forecast about the interaction of an individual (user) with a product (item) in the future.

CF approach works by learning latent characteristics also termed embeddings to denote a user and an item and these embedding vectors are utilized to carry out the prediction (Cheng et al., 2018; He et al., 2017). CF is the basic approach for making predictions and hence better-personalized recommendations by concentrating on the exploration of past user-item interactions (He et al., 2017; Ebesu et al., 2018; Liang et al., 2018; Wang et al., 2019b). One of such approaches is *matrix factorization* (MF), where the single ID of an individual is transformed into embedding (Koren et al., 2009).

Many studies have observed that enhancing individual ID with the historical interaction can be utilized as feedback to boost the feature of the embedding. As an instance, *Singular Value Decomposition* (SVD++) (Koren, 2008) has shown the advantages of utilizing individual interaction history in estimating the ratings, the significance of items in the interaction history has been distinguished by *Neural Attentive Item Similarity* (NAIS) (He et al., 2018) and exhibited enhancements in estimating item ranking.

NGCF approach has been proposed recently by Wang et al. (2019b) which makes utilization of the subgraph arrangement and several hop neighbors and hence acquired good performance for CF. NGCF has been built by making use of several properties of GCN (Hamilton et al., 2017; Kipf and Welling, 2016) using the same propagation mechanism to make good embeddings: feature transformation, neighborhood aggregation, and non-linear activation. The user-item interaction graph can utilize the subgraph structure of an individual, particularly, one hop adjacent neighbors to enhance the learning of the embeddings.

It is believed that when developing a framework for a recommendation, thorough ablation studies are required to understand the influence of each operation. Otherwise, introducing less relevant procedures will needlessly complicate the framework, make training more complex, and even reduce the framework's performance (Fu et al., 2022; Chen et al., 2022a; Wei et al., 2022; Wang et al., 2022; Wu et al., 2022; Chen et al., 2022b; Liu et al., 2022a; Lin et al., 2022; Yang et al., 2022).

Specifically, GCN has been designed for node classification tasks on a graph with attributes, where these attributes can be used as input characteristics, while in an item-user collaboration graph for CF, every

* Corresponding author.
E-mail address: adrian1803@gmail.com (A. Rabaea).

https://doi.org/10.1016/j.elerap.2022.101191
Received 10 November 2021; Received in revised form 14 June 2022; Accepted 10 August 2022
Available online 13 August 2022

node says item or user is defined by one hot ID, and it carries no real meaning except considering as an identifier.

Rigorous studies have been attempted on NGCF, to corroborate our opinions. With the aid of strong experiments carried out on the same dataset and using the same assessment procedure, the decision is made that two properties of GCN namely non-linear activation and feature change put no influence on the performance of NGCF. In this situation, utilizing the input of ID embedding, carrying out several levels of non-linear feature change that is significant to the achievement of state-of-the-art neural networks (He et al., 2016) will result in no gains and advantages, but increases the complexity of model training. Furthermore, it is a matter of surprise that by eliminating these two features of GCN, there is a significant improvement in the metric of accuracy. This mirrors the problems of augmenting these features that are not appropriate for the job in *graph neural network* (GNN), which not only conveys no advantages, however, also deteriorates the framework performance. Particularly, after linking every item/user to an ID embedding, these embeddings are propagated on the item-user collaboration graph to improve them.

Inspired by these experimental observations, a fresh framework namely ImprovedGCN is suggested with the inclusion of the utmost necessary part of GCN called neighborhood aggregation for CF. Afterward, the embeddings generated at several propagation layers are combined alongside a weighted total to acquire the last embedding to make a prediction. The aforesaid framework is straightforward as well as uncomplicated, and it not only is easy to be trained, moreover also acquires good experimental effectiveness as compared to NGCF and other modern approaches such as Mult-VAE (Variational Autoencoders) (Liang et al., 2018).

Our work incorporates the following suggestions: it is experimentally proved that two features in GCN namely feature transformation and non-linear activation have no significant contribution to the performance of CF. ImprovedGCN simplifies the model by including only the necessary parts for a recommendation. Experimentally a contrast is made between ImprovedGCN and NGCF by observing the same background and it exhibits great enhancements. Exhaustive investigations have been carried out to prove the rationality of ImprovedGCN from both practical and experimental viewpoints.

The difference between NGCF and ImprovedGCN lies in the fact that there is no feature transformation and nonlinear activation in ImprovedGCN, unlike NGCF.

ImprovedGCN simplifies the GCN by removing the non-linearity from the GCN and reducing the simple pre-processing of node features.

The paper is structured as follows: Section 2 includes a brief discussion of related work. Our proposed model is presented in Section 3. In Section 4 we provide the analysis of a model and in Section 5 experimental results. Finally, Section 6 concludes the paper.

2. Literature reviews

As recommendation systems have become increasingly important in a multitude of areas, many approaches (like CF, recommendation based on graph approaches and NGCF) have been applied to improve them.

2.1. Collaborative filtering

CF is an existing approach that is utilized for building robust recommendation systems (Covington et al., 2016; Ying et al., 2018). One usual example of the CF approach is to consider items and users as embeddings and know the different constraints of embedding by rebuilding past interactions among users and items.

As an example, previous CF methods e.g., MF (Koren et al., 2009; Rendle et al., 2009) convert the ID of an individual (or a product) to an embedding vector. Besides utilizing only ID information, another category of CF models takes into consideration the past items as the prevailing characteristics of a user to well represent the user.

The current neural RSs, for instance, *Latent Relational Metric Learning* (LRML) (Tay et al., 2018) and *Neural Collaborative Filtering* (NCF) (He et al., 2017) utilize the same embedding part and improve the interaction modeling using neural networks. For instance, SVD++ (Koren, 2008) and *factored item similarity models* (FISM) (Kabbur et al., 2013) employ the weighted average of the ID embeddings of past products as the objective individual's embedding.

Afterward, attention-based approaches are incorporated to catch different contributions, for instance, NAIS (He et al., 2018) and *Auto-correlation function* (ACF) (Chen et al., 2017), to learn the significance of every item.

Currently, investigators feel that historical items contribute differently to figuring personal interests. Taking into consideration past collaborations as an individual-product bipartite graph, the enhancement in the effectiveness may be due to local neighborhood i.e., one hop neighbors which enhances the embedding knowledge. Each matrix multiplication diffuses the current embeddings to their one-hop neighbors. The final embedding of a user/item is obtained by aggregating embeddings of its multi-hop neighboring nodes. The simplification leads to better empirical performance than NGCF (Cao et al., 2022; Chu et al., 2022).

2.2. Recommendation based on graph approaches

Another pertinent investigation is to exploit the user-item graph arrangement for a recommendation. Currently employed GNNs exploit the graph arrangement, particularly multi-hop neighbors to learn the embedding (Hamilton et al., 2017; Kipf and Welling, 2016).

Previous contributions, for instance, ItemRank (Gori and Pucci, 2007) utilize the approach of label propagation to spread user favourite ratings across the graph e.g., similar labels are assigned to associated nodes.

Previous investigations perform graph convolution on the spectral region, for instance, Chebyshev polynomials (Defferrard et al., 2016) and Laplacian eigendecomposition (Bruna et al., 2014), which are computation-wise costly.

Because of its efficacy of it, it has become an existing formulation of GNNs and is broadly employed (Feng et al., 2019; Qiu et al., 2018; Zhao et al., 2019). Afterward, GCN (Kipf and Welling, 2016) and GraphSage (Hamilton et al., 2017) re-illustrate graph convolution in the spatial region i.e., the target's node embedding can be achieved by aggregating the embeddings of neighbors.

The GCN has been broadly adopted to the user-item interaction graph by being encouraged by the power of graph convolution in contributions like *Graph Convolutional Matrix Completion* (GC-MC) (Berg et al., 2017), NGCF (Wang et al., 2019b), and PinSage (Ying et al., 2018), catching CF signals in multi-hop neighbors for a recommendation. Specifically, Wu et al. (2019a) exhibit the unneeded intricacy of GCN, thereby introducing a *simplified GCN* (SGCN) method by alleviating the non-linearity and utilizing a single matrix instead of multiple weights matrices.

Different current investigations deeply explore GNNs (Wu et al., 2019a; Klicpera et al., 2018; Li et al., 2018) which has motivated us to develop ImprovedGCN. The major variance is that ImprovedGCN and SGCN are built for diverse jobs, therefore, the reason for approach oversimplification is diverse. Particularly, SGCN has been designed for node classification, carrying out simplification for method efficacy.

In comparison, ImprovedGCN is on CF, when every node carries an ID characteristic only. If the accuracy of the node classification is taken into consideration, SGCN performs similarly to GCN and in some cases, it is less robust than GCN.

Therefore, the simplification is carried out for a robust motive: weight matrices and non-linearity are of no use for CF, and further increases the training difficulty. Whereas, if the accuracy of CF is considered, ImprovedGCN performs better than GCN by a big factor (16% enhancement over NGCF).

In the end, another study carried out in a similar duration (Chen et al., 2020), also observes that the non-linearity is not required in NGCF and introduces a linear GCN approach for CF. In comparison, our ImprovedGCN goes one footstep more i.e., all the extra parameters are eliminated and only ID embeddings are maintained, thereby, building the approach simpler than MF.

2.3. Neural graph collaborative filtering

In this section, NGCF (Wang et al., 2019b) is introduced, which is a GCN design for a recommendation, and is proved that both usual properties of GCN, namely feature change and non-linear activation, are not advantageous for CF.

The high-level concept behind NGCF is the organization of historical interactions as a customer-product bipartite graph. NGCF captures CF signals through high-order connectivity. It may be defined as the paths that reach u_ℓ from any node with the path length ℓ larger than 1. E.g., why u_1 may like i_4?

(1) $u_1 \leftarrow i_2 \leftarrow u_2 \leftarrow i_4$
(2) $u_1 \leftarrow i_3 \leftarrow u_3 \leftarrow i_4$

The aforesaid example has been shown in Fig. 1. Therefore, the contribution of NGCF is towards explicit modeling high order connectivity through GNNs. One of the demerits of NGCF is that its designs of NGCF are rather burdensome.

From the beginning, every item and user is linked to an ID embedding. Assume, $z_u^{(0)}$ describe the ID embedding of individual u and $z_i^{(0)}$ describe the ID embedding of a product i. The item-individual collaboration plot is utilized by NGCF for the propagation of embeddings as underneath:

$$\begin{aligned} z_u^{(\ell+1)} &= \sigma\left(W_\ell z_u^{(\ell)} + \sum_{i\in N_u} \frac{1}{\sqrt{|N_u||N_i|}}\left(W_\ell z_i^{(\ell)} + B_l\left(z_i^{(\ell)} \odot z_u^{(\ell)}\right)\right)\right) \\ z_i^{(\ell+1)} &= \sigma\left(W_\ell z_i^{(\ell)} + \sum_{u\in N_i} \frac{1}{\sqrt{|N_u||N_i|}}\left(W_\ell z_u^{(\ell)} + B_\ell\left(z_u^{(\ell)} \odot z_i^{(\ell)}\right)\right)\right). \end{aligned} \tag{1}$$

Here, $z_u^{(\ell)}$ and $z_i^{(\ell)}$ describe the improved embedding of user u and item i afterward propagating ℓ levels, σ describes the non-linear activation operation (e.g. ReLU), N_u describes the group of items which are collaborated by user u, N_i describes the group of users which collaborate with item i, $W_\ell \in \mathbb{R}^{d'\times d}$ and $B_\ell \in \mathbb{R}^{d'\times d}$ describe the *trainable weight matrices* to carry out feature transformation in every layer, d is the embedding size, d' is the transformation size and $\odot$ denotes the element-wise product.

NGCF utilizes the GCN (Kipf and Welling, 2016) incorporating the use of feature change matrices W_ℓ and B_ℓ and non-linear activation operation σ. NGCF achieves $L+1$ embedding to define an individual ($z_u^{(0)}$, $z_u^{(1)}$, …, $z_u^{(L)}$) and a product ($z_i^{(0)}$,$z_i^{(1)}$, …, $z_i^{(L)}$). The following three simple variations of NGCF are implemented:

NGCF-t, eliminates the feature transformation matrices W_ℓ and B_ℓ.

NGCF-a, eliminates the non-linear activation operation σ.

NGCF-ta, eliminates together non-linear activation function and feature transformation matrices.

There is twofold propagation in NGCF. In first-order propagation, the message is constructed by generating a message from one neighbor. It is depicted in the equation as underneath:

$$m_{u\leftarrow i} = \frac{1}{\sqrt{|N_u||N_i|}}\left(W_1 z_i + W_2(z_i \odot z_u)\right) \tag{2}$$

where, $W_1, W_2 \in \mathbb{R}^{d'\times d}$ are the trainable weight matrices to distill significant information for propagation, and d' is the transformation size. The message aggregation is carried out by updating the ego node's representation by an aggregating message from all neighbors. It is shown as described by the following equation:

$$z_u^{(1)} = LeakyReLU\left(m_{u\leftarrow u} + \sum_{i\in N_u} m_{u\leftarrow i}\right). \tag{3}$$

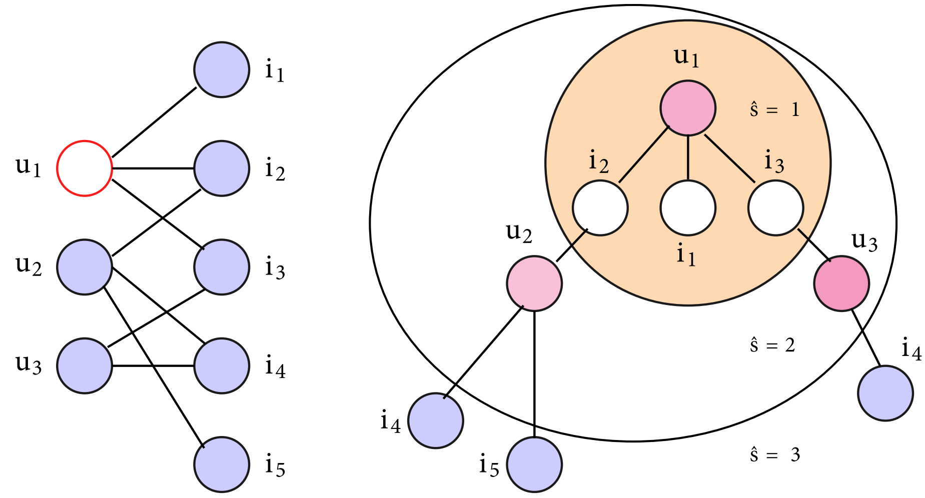
In higher-order propagation, more embedding propagation layers are stacked to explore the high-order connectivity information as shown in the equation underneath:

$$z_u^{(\ell)} = LeakyReLU\left(m_{u\leftarrow u}^{(\ell)} + \sum_{i\in N_u} m_{u\leftarrow i}^{(\ell)}\right). \tag{4}$$

Apart from these approaches, few other methods have been suggested in the literature as illustrated underneath.

Zhang et al. (2020) have proposed a new factorization framework that amalgamates multi-view visual information with the implicit feedback information for restaurant forecast and ranking. The visual characteristics of pictures are mined by employing a deep convolution network and are combined into a collaborative filtering model.

Chen and Xia (2021) have proposed a context-aware recommendation method centered on embedded characteristic selection. The context redundancy is eliminated by producing the least subset of all contextual data and assigning the weight to every context properly.



Customer-Pro duct interaction High-order connectivity for u1

Fig. 1. Customer-Product bipartite graph and high order connectivity.

Wu et al. (2020) have proposed the interval-valued triangular fuzzy rating framework. This framework substitutes interval-valued triangular fuzzy numbers in place of crunchy ratings centered on the individuals' rating statistics data, which can effectively compute the individuals' likings.

Wen et al. (2021) have proposed a neural attention framework for a recommendation, which deepens factorization machines by augmenting an attention system and fully connected layers. By employing the attention procedure, the aforesaid approach can know the diverse significance levels of low-order characteristic interactions.

Xing et al. (2019) have proposed a content-aware *point of interest* (POI) recommendation centered on a convolutional neural network. The researchers employ a convolutional neural network as the basis of a combined POI recommendation model and incorporate three categories of content data, comprising POI characteristics, individual likings, and sentiment signs.

Chang et al. (2021) have presented a framework called Multi Attention Network to know the contextual influence of both individuals and POIs on POI recommendation. This framework comprises: an individual-friend component and a POI neighborhood component. The individual-friend component utilizes an attention-centered memory part to produce particular relation vectors which can distinguish the effect from the viewpoint of interest, and utilizes a friend-level attention network to determine the likings of individuals.

Zhang et al. (2019) have suggested a collaborative filtering recommendation algorithm centered on probability matrix factorization. The suggested approach decomposes the rating matrix into two nonnegative matrices employing a predictive rating framework. The likings for products that individuals have not rated can be forecasted.

Xiao and Shen (2019) have proposed a deep generative framework called Neural Variational Matrix Factorization, which introduces side data (characteristics) of both individuals and products to learn the best latent interpretations of them for more efficient collaborative filtering recommendation.

Duan et al. (2022) have proposed a collaborative filtering recommendation approach (ETBRec) that not only takes into account the trust difference among individuals but also suggests the definition of experts and takes into account the direct impact of expert individuals on forecast ratings.

Feng et al. (2018) have proposed a new group recommendation framework that introduces individual social networks into the random walk with a restart framework which can assist us in better illustrating groups liking and enhance the performance of group recommender systems.

3. Proposed model

The above section illustrates that NGCF is a heavy-weight GCN design for CF. The benefits of keeping GCN simple are many folds as the model becomes easy to train and analyze. In this part, firstly our ImprovedGCN design is presented as depicted in Fig. 2.

With these observations, our objective is to design a lightweight and robust framework that increases the performance with the inclusion of only necessary components of GCN for an effective recommendation. Afterward, our implemented model is analyzed to prove its effectiveness. In the end, it is shown how to carry out the training for a recommendation.

3.1. ImprovedGCN

The core of the GCN is to learn the description for nodes by enhancing the characteristics of the graph (Kipf and Welling, 2016; Wu et al., 2019a). To acquire this, the graph convolution is carried out repeatedly i.e., the characteristics of the adjacent neighbors are aggregated as the new description of an objective node. This combination of neighbors may be represented as:

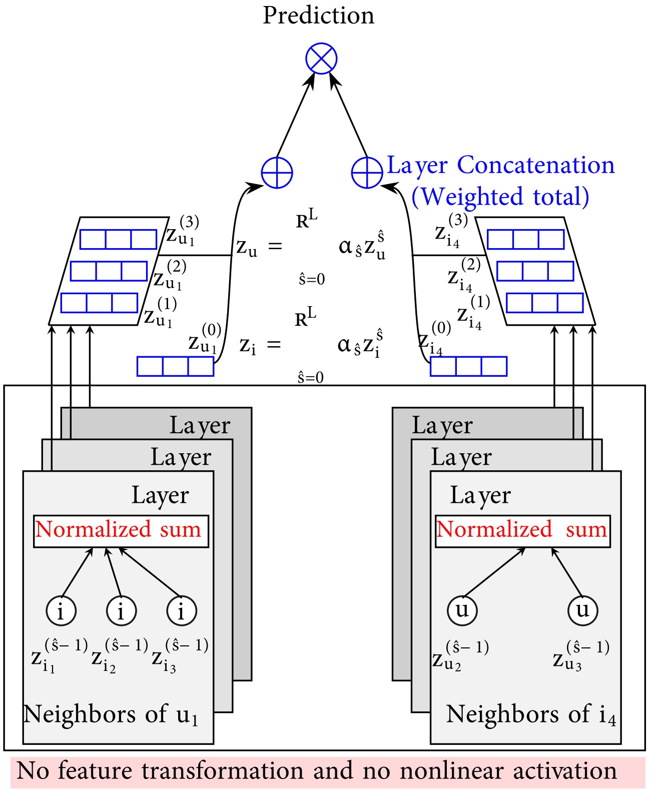


Fig. 2. ImprovedGCN framework architecture.

$$z_u^{(\ell+1)} = AGG\left(z_u^{(\ell)}, \left\{z_i^{(\ell)} : i \in N_u\right\}\right). \tag{5}$$

AGG function has been defined by several studies, for example, as *Long short-term memory* (LSTM) aggregator in GraphSAGE (Hamilton et al., 2017), weighted sum aggregator in GIN (Xu et al., 2018), etc. The major component of graph convolution is an aggregation function AGG which takes into consideration the $\ell-th$ level's description of the objective node and its adjacent neighbors. But these studies associate non-linear activation and feature transformation with AGG. These operations can be advantageous to node and graph classification jobs that have meaningful input characteristics, these can be troublesome for CF.

3.2. Improved graph convolution (IGC)

In ImprovedGCN, the straightforward weighted total aggregation is utilized and non-linear activation and feature transformation are eliminated. The *Graph Convolution* (GC) function which is also termed as propagation law (Wang et al., 2019b) in ImprovedGCN is described as:

$$\begin{aligned} z_u^{(\ell+1)} &= \sum_{i \in N_u} \frac{1}{\sqrt{|N_u|}\sqrt{|N_i|}} z_i^{(\ell)} \\ z_i^{(\ell+1)} &= \sum_{u \in N_i} \frac{1}{\sqrt{|N_i|}\sqrt{|N_u|}} z_u^{(\ell)}. \end{aligned} \tag{6}$$

The aforementioned scenario is in contrast to the other graph convolution approaches (Hamilton et al., 2017; Kipf and Welling, 2016; Veličković et al., 2017; Zhu et al., 2020; Wang et al., 2019b) that take into consideration self-connection, especially neighborhood connections. The symmetric normalization quantity $\frac{1}{\sqrt{|N_u|}\sqrt{|N_i|}}$ is as per accordance with GCN (Kipf and Welling, 2016), which may circumvent any increase in embeddings with graph convolutional tasks. Other terms may also be utilized like L_1 norm; however, it has been found experimentally that the aforementioned symmetric normalization exhibits better performance. Different layers are combined which achieves the same result as that of the self-connections as shown in the forthcoming section. There is no necessity for the inclusion of self-connections in IGC.

3.3. Layer concatenation and model prediction

In ImprovedGCN, the embeddings of the $0-th$ level, i.e., $z_u^{(0)}$ for all individuals and $z_i^{(0)}$ for all products can be considered as only trainable model parameters. When the values of these parameters are known, the embeddings at higher layers may be determined using IGC as described in expression (6). Afterward, L layers IGC, the embeddings determined at every layer are concatenated to achieve the final representation of an item (a user):

$$z_u = \sum_{\ell=0}^{L} \alpha_\ell z_u^{(l)}; \qquad z_i = \sum_{\ell=0}^{L} \alpha_\ell z_i^{(\ell)}; \tag{7}$$

where $\alpha_\ell \geqslant 0$ describes the significance of the $\ell-th$ layer embedding to form the final embedding.

In our empirical explorations, it has been observed that equating α_ℓ to $1/(L+1)$ exhibits decent performance. It can be taken into consideration as a hyper constraint that can be tweaked with hands or as a framework constraint (e.g., outcome of an attention network (Chen et al., 2017)) which can be fine-tuned instinctively. Therefore, there is no necessity of introducing any special element to improve α_ℓ, to circumvent making ImprovedGCN complex needlessly. This builds the model simple. There are three explanations for carrying out layer concatenation to obtain the final representation.

Firstly, when the number of layers is increased, the embeddings get over-smoothed (Li et al., 2018). Therefore, employing the last layer is troublesome.

Secondly, the embeddings at diverse layers catch different meanings. For example, the first level applies flatness to products and individuals who have connections. The individuals/products which have overlap with collaborated products/individuals are smoothed by the second layer and the upper layers catch upper-order closeness (Wang et al., 2019b). Therefore, the concatenation of all the layers creates the representation as a complete one.

Thirdly, the embeddings at diverse layers are concatenated alongside the weighted total to achieve the influence of GC with self-association, a significant aspect of GCN. The prediction for the model may be described as the inner product of the item and individual last embeddings:

$$\widehat{r_{ui}} = z_u^T z_i \tag{8}$$

which can be utilized as the ranking score for generating a recommendation. If the data is less intricate and carries low dimensions or features, one or two layers can be utilized, otherwise, three to four layers need to be used.

3.4. Matrix formulation

Assume the customer-product interaction matrix as $R \in \mathbb{R}^{m \times n}$, here m and n define the count of customers and products correspondingly, and every entry R_{ui} is equal to 1 if u has collaborated with i, or else zero.

ImprovedGCN can be expressed in matrix representation to ease its development and compare it with works that exist in literature. Therefore, the adjacency matrix of the item-user plot may be described as

$$A = \begin{pmatrix} 0 & R \\ R^T & 0 \end{pmatrix}. \tag{9}$$

Assume the $0-th$ layer embedding matrix as $Z^{(0)} \in \mathbb{R}^{(m+n) \times d}$ where d is the size of the embedding. Thereafter, the IGC in matrix form can be achieved as:

$$Z^{(\ell+1)} = \left(D^{-\frac{1}{2}} A D^{-\frac{1}{2}}\right) Z^{(\ell)} \tag{10}$$

where, D is an $(m+n) \times (m+n)$ diagonal matrix, here every entry D_{jj} describes the count of non-zero values in the $j-th$ row vector of the adjacency matrix A (as well termed as degree matrix). In the end, the last embedding matrix employed for framework forecast is obtained as underneath:

$$\begin{aligned} Z &= \alpha_0 Z^{(0)} + \alpha_1 Z^{(1)} + \ldots + \alpha_L Z^{(L)} \\ &= \alpha_0 Z^{(0)} + \alpha_1 \widetilde{A} Z^{(0)} + \ldots + \alpha_L \widetilde{A}^L Z^{(0)} \end{aligned} \tag{11}$$

where, $\widetilde{A} = D^{-\frac{1}{2}} A D^{-\frac{1}{2}}$ is called symmetric normalization matrix.

4. Analysis

4.1. Analysis of the model

The analysis of the framework has been carried out to show the rationality and usefulness of the ImprovedGCN.

Firstly, it is compared with SGCN (Wu et al., 2019a) which combines self-connections in graph convolution. This analysis demonstrates that by carrying out layer concatenation, ImprovedGCN implicitly comprises the impact of self-association, therefore, it is of no necessity for ImprovedGCN to augment self-association in the adjacency matrix. The analysis demonstrates the fundamental correspondence among ImprovedGCN and *Approximate Personalized Propagation of Neural Predictions* (APPNP), therefore, our ImprovedGCN exhibits the same advantages in propagating distance with manageable smoothness.

Secondly, the connection alongside APPNP (Klicpera et al., 2018) is observed that will be a new GCN variation which discusses over smoothness by motivating from Personalized PageRank (Haveliwala, 2002).

In the end, the second layer IGC is analyzed to demonstrate in what manner it smooths a user with second-order neighbors, and hence provides in-depth details of the working procedure of ImprovedGCN.

4.2. Association with SGCN

GCNs are an effective variation of CNNs on graphs. In this paper, it is observed that GCNs inherit significant complexity from their deep learning roots, which can be heavy, troublesome, and needless for applications that are less in demand.

Therefore, this extra complexity of GCNs is mitigated by recurrently alleviating the nonlinearities among GCN layers and folding the resultant function into a single linear transformation. This linear framework is referred to as *Simple Graph Convolution* (SGC) as shown in Fig. 3 which contrasts GCN and SGCN.

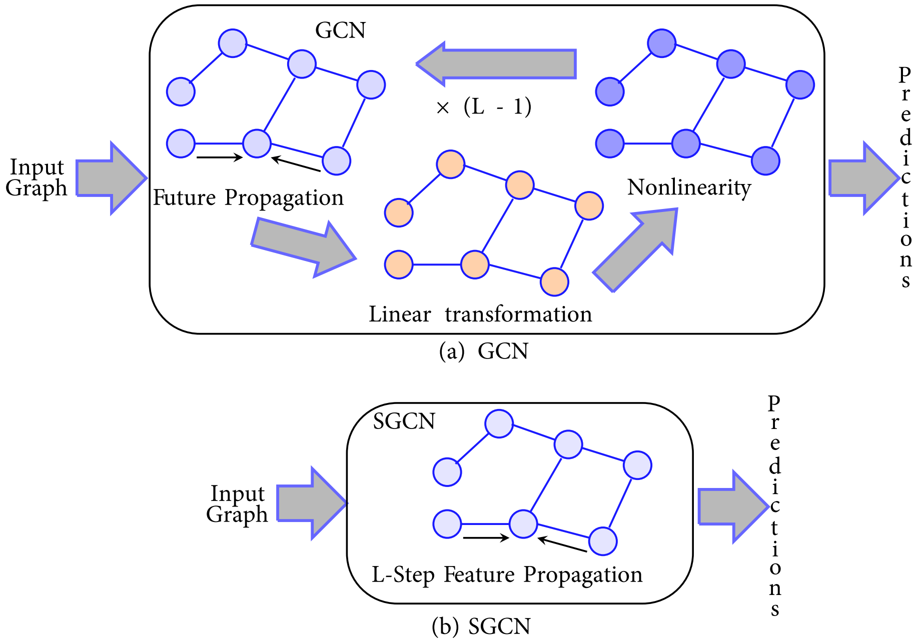
In GCN, the feature vectors are transformed recurrently throughout L layers and afterward linear classification is applied to the last interpretation.

In contrast, in SGCN, the whole process is reduced to a simple feature propagation phase. The authors in Wu et al. (2019a) claim that GCN is unessentially complex when used for node classification and suggest SGCN, where the nonlinearities are eliminated and the weight matrices are collapsed into a single weight matrix (Liu et al., 2022b; Ni et al., 2022).

The GC operation in SGCN is described underneath:

$$Z^{(\ell+1)} = (D+I)^{-\frac{1}{2}} \left(A + I\right) (D+I)^{-\frac{1}{2}} Z^{(\ell)} \tag{12}$$

where $I \in \mathbb{R}^{(m+n) \times (m+n)}$ is known as the identity matrix and self-connections are included by adding it on A. To keep the model simple, the $(D+I)^{-1/2}$ terms are omitted because these are only employed for re-scaling of embeddings. In SGCN, the last layer of final embeddings is



(a) GCN

(b) SGCN

Fig. 3. Comparison of GCN vs SGCN.

utilized for making a prediction, which is illustrated as:

$$
\begin{aligned}
Z^{(L)} &= \left(A+I\right)Z^{(L-1)} = (A+I)^{(L)}Z^{(0)} \\
&= \binom{L}{0}Z^{(0)} + \binom{L}{1}AZ^{(0)} + \ldots + \binom{L}{L}A^{L}Z^{(0)}.
\end{aligned}
\tag{13}
$$

The aforementioned equation demonstrates that adding self-connections into A and as embeddings are propagated on it, is necessarily the same as propagating the weighted sum of the embeddings at every IGC layer. The complexity of GCN and simplicity of SGCN are shown in Fig. 3.

4.3. Association with APPNP

Motivated by the design of Personalized PageRank, APPNP applies initial features (i.e., $0-th$ layer embeddings) to every propagation layer which can maintain the trade-off between the necessity of maintaining neighborhood (i.e., remaining nearer to the target node to circumvent over smoothing) and aggregating the data about a big neighborhood. The architecture design of APPNP is shown in Fig. 4.

Predictions are firstly produced from every node's feature set by a neural network and afterward propagated employing an adjustment of personalized PageRank. The framework is trained end to end. In a new study (Klicpera et al., 2018), the researchers associate GCN alongside Personalized PageRank (Haveliwala, 2002), and by taking motivation from which they suggest a variation of GCN termed APPNP that can propagate distant eliminating the danger of over smoothing.

In APPNP, the propagation layer can be described as underneath:

$$
Z^{(\ell+1)} = \gamma Z^{(0)} + \left(1-\gamma\right)\widetilde{A}Z^{(\ell)}
\tag{14}
$$

where γ is the likelihood to regulate the retention of initial characteristics in the propagation and $\widetilde{A}$ is the standardized adjacency matrix.

The final-place level is employed for making the concluding estimate in APPNP as underneath:

$$
\begin{aligned}
Z^{(L)} &= \gamma Z^{(0)} + \left(1-\gamma\right)\widetilde{A}Z^{(L-1)} \\
&= \gamma Z^{(0)} + \gamma\left(1-\gamma\right)\widetilde{A}Z^{(0)} + (1-\gamma)^{2}\widetilde{A}^{2}Z^{(L-2)} \\
&= \gamma Z^{(0)} + \gamma\left(1-\gamma\right)\widetilde{A}Z^{(0)} + \ldots + (1-\gamma)^{L}\widetilde{A}^{L}Z^{(0)}.
\end{aligned}
\tag{15}
$$

Moreover, ImprovedGCN utilizes the power of APPNP in controlling smoothness by giving appropriate value to α, employing a large L for

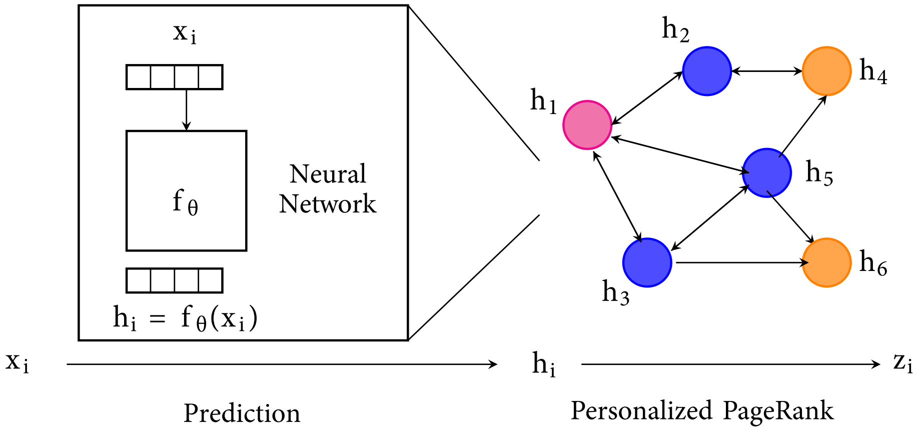


Fig. 4. Design of APPNP.

distant modeling is permitted.

Utilizing expression (11), it can be observed that by providing a value to α_ℓ appropriately, ImprovedGCN can generate the prediction embedding employed by APPNP.

Another slight dissimilarity is that APPNP augments self-association into the adjacency matrix. But since it has been demonstrated earlier, it gives rise to redundancy because of the weighted sum of different layers.

Note that the propagation scheme of this model does not require any additional parameters to train - as opposed to models like GCN, which typically require more parameters for each additional propagation layer. Therefore, it can be propagated very far with very few parameters. Predictions are first generated from each node's features by a neural network and then propagated using an adaptation of personalized PageRank. The model is trained end-to-end.

4.4. Smoothness of second-order embedding

A two-layer ImprovedGCN is analyzed to show its effectiveness. Our proposed model ImprovedGCN is linear and simple, so it can be determined in-depth how the embeddings are smoothed by it. By considering the user part as an instance, instinctively, the 2nd layer smooths users which are overlapped in the interacted items. Specifically, the following is achieved:

$$z_u^{(2)} = \sum_{i\in N_u}\frac{1}{\sqrt{|N_u|}\sqrt{|N_i|}}z_i^{(1)} = \sum_{i\in N_u}\frac{1}{|N_i|}\sum_{w\in N_i}\frac{1}{\sqrt{|N_u|}\sqrt{|N_w|}}z_w^{(0)}. \tag{16}$$

It can be observed that if another user w collaborated with the objective individual u, the power of the smoothness of w on u is determined by the following coefficient (or else zero):

$$c_{w\rightarrow u} = \frac{1}{\sqrt{|N_u|}\sqrt{|N_w|}}\sum_{i\in N_u\cap N_w}\frac{1}{|N_i|}. \tag{17}$$

The aforementioned coefficient can be interpreted: the effect of a 2*nd* order neighbor w on u is computed by

1) the count of items with which co-interacted, greater this count, greater the impact
2) the popularity of these items, lower is the popularity (a sign of user personalized liking), greater is the impact
3) the activity of w, if the activity is small, the greater is the impact.

Following the symmetric property of ImprovedGCN, the alike analysis can be carried out on the item part also. This shows how CF computes the user similarity (Chen et al., 2019b; Wang et al., 2006) and proves the rationality of ImprovedGCN. Our proposed ImprovedGCN framework generates smoother embeddings which are more suitable for recommendation and hence prove the usefulness and rationality of ImprovedGCN. If the number of layers is increased, it may enhance the performance, but the advantages get smaller. It has been observed that increasing the layer number from 0 to 1 yield the greatest performance gain and utilizing layer number of three yields satisfactory performance in most cases.

4.5. Training of the model

The parameters and constraints of ImprovedGCN which can be trained are the embeddings at the $0-th$ level i.e., $Z(0)$, so the intricacy of the model remains the same as in the case of MF. BPR loss (Rendle et al., 2009), a pairwise loss, is utilized that supports the fact that the prediction of a seen item is larger than its unobserved ones:

$$L_{BPR} = -\sum_{u=1}^{m}\sum_{i\in N_u}\sum_{j\notin N_u}\ln\sigma\left(\widehat{r_{ui}} - \widehat{r_{uj}}\right) + \lambda\left|\left|Z^{(0)}\right|\right|^2 \tag{18}$$

where λ regulates the strength of regularization L_2.

Observe that the dropout procedures are not incorporated, which are generally employed in NGCF and GCNs. Different optimizers can be utilized, in particular, Adam (Kingma and Ba, 2014) optimizer is employed and it is utilized in a mini-batch fashion. Other innovative negative sampling approaches can enhance the training of our ImprovedGCN, for example, hard negative sampling (Rendle and Freudenthaler, 2014) and adversarial one (Ding et al., 2019). This is because the feature transformation weight matrices are not utilized in ImprovedGCN, therefore, utilizing L_2 regularization on the embedding level is appropriate to eliminate overfitting. This exhibits benefits of ImprovedGCN including its simplicity and easiness for training and fine-tunes as compared to NGCF which requires an addition to tune two dropout ratios (message and node dropout) and the embedding of every level needs to be normalized to unit length.

But it is observed that no enhancement can be achieved by learning α. This is because appropriate signals are not contained in the training data to learn the best value of α through which generalization can be applied to unseen data. Furthermore, it is possible to learn the layer concatenation coefficients $\{\alpha_l\}_{l=0}^{L}$, or they can be parameterized using the attention network. An attempt has been made to know α from validation data as motivated by one of the studies (Chen et al., 2019a) which learns hyper-parameters on validation data. There is a slight improvement in the performance i.e., less than one percent.

5. Experimentations

Firstly, a setup for experiments is defined and then in-depth contrast with NGCF is performed (Wang et al., 2019b), the model which is pertinent to ImprovedGCN and is more complex. Also, a contrast with other models is made in the forthcoming sections. To prove the rationality and efficiency of ImprovedGCN, different investigations and the analysis of embeddings are carried out.

5.1. Experimental setup

The measurements of the experimental datasets which are employed for training and testing are illustrated in Table 1.

The embedding size of 64 and the learning rate of 0.001 are used for each of the three datasets. The batch size of 2048 is used for Gowalla and Yelp2018 dataset and the batch size of 8192 is used for the Amazon-book dataset. To mitigate the experimental burden and make the contrast reasonable, the same setup as used by NGCF (Wang et al., 2019b) is employed. The Amazon-Book and Gowalla datasets are the same as those employed for NGCF, the only dataset which has been revised is the Yelp2018 because, in an earlier version, cold start items have not been filtered in the training set. Recall@20 and ndcg@20 are the evaluation metrics that have been determined by the all-ranking procedure i.e., the items which are not collaborated by a user can be considered as contenders. Therefore, NGCF is executed again on the Yelp2018 dataset.

5.2. Contrasted models

Since the comparison has been carried out on the same datasets and utilizing the same evaluation guidelines and metrics, no further comparison has been done with these models. Furthermore, apart from NGCF, a comparison has been drawn with two more pertinent contending CF models.

Table 1
Measurements of the empirical information.

Dataset	User#	Item#	Interaction#	Density
Amazon-Book	52640	91590	2984100	0.00062
Gowalla	29850	40980	1027360	0.00084
Yelp2018	31660	38040	1561410	0.00130

NGCF is the major contending model, which has exhibited performance that is better than many other models like PinSage (Ying et al., 2018), CMN (Ebesu et al., 2018), factorization centered methods like MF (Rendle et al., 2009), *high-order proximity for implicit recommendation* (HOP-Rec) (Yang et al., 2018), GC-MC (Berg et al., 2017) and neural network centered methods NeuMF (He et al., 2017).

The outcomes of the two-layer arrangement on the Amazon-Book, Gowalla, and Yelp2018 datasets have been shown in Table 2.

This table shows the effectiveness of NGCF and its variants. In NGCF-t, feature transformation is removed, in NGCF-a, nonlinear activation operation is removed, in NGCF-ta, both feature transformation and nonlinear activation are removed. Moreover, apart from removing feature transformation and nonlinear activation, no self-connections are included in ImprovedGCN, however, self-connections are present in NGCF-ta. Therefore, the proposed ImprovedGCN is compared to NGCF-ta, and it outperforms NGCF. It has now been explained in the paper as well as in this response what each row means. The subsequent figures demonstrate the effectiveness of NFCF and its variants in terms of recall and training loss.

To learn why NGCF deteriorates with the above two operations, the curves for recall and training loss have been depicted in Fig. 5.

The architecture of NGCF differs from our proposed approach ImprovedGCN in several ways. As stated earlier, the designs of NGCF are rather burdensome. Several functions are straightforwardly derived from GCN lacking explanation. The input data to GNN comprise rich node features e.g., text, image data, attributes whereas NGCF considers only node ID.

When it comes to feature transformation, GNNs are capable of distilling valuable information whereas NGCF only generates ID embeddings. Furthermore, NGCF undesirably intensifies the trouble for model training. Our proposed approach namely ImprovedGCN outperforms NGCF since, in the end, only a simple weighted sum aggregator remains. Furthermore, there is no feature transformation and no nonlinear activation which in turn significantly decreases training loss and also improves the testing accuracy.

Normalized Discounted Cumulative Gain (NDCG) is an approach for determining the quality of a set of search outcomes. *Discounted Cumulative Gain* (DCG) is defined as underneath:

$$DCG = \frac{1}{m}\sum_{u=1}^{m}\sum_{j\in I_u}\frac{g_{uj}}{\log_2(v_j+1)} \tag{19}$$

where the discount factor of product *j* is set to $\log_2(v_j+1)$ and v_j is the rank of product *j* in the test set I_u.

Therefore, NDCG is specified as the ratio of DCG to its *ideal* value (IDCG).

$$NDCG = \frac{DCG}{IDCG}. \tag{20}$$

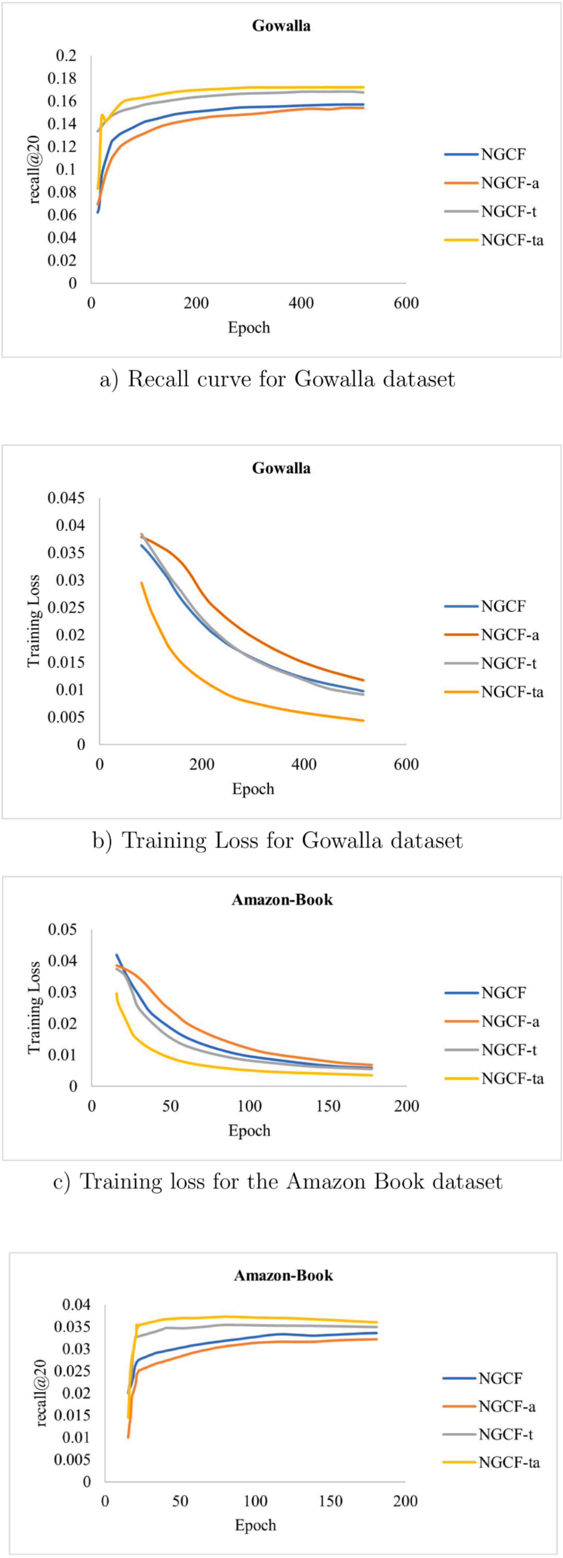
In this section, a comparison is made among state-of-the-art approaches cum algorithms and proposed ImprovedGCN. These state-of-the-art methods are described in brief as underneath.

BPR is a collaborative filtering matrix factorization algorithm used to predict the ranking of items through implicit feedback.

NCF is a deep learning algorithm with improved performance for implicit feedback.

Table 2
Effectiveness of NGCF and variations.

	Amazon-Book		Gowalla		Yelp2018	
	ndcg	recall	ndcg	recall	ndcg	recall
NGCF	0.0261	0.0341	0.1324	0.1571	0.0459	0.0564
NGCF-t	0.0282	0.0367	0.1438	0.1685	0.0464	0.0566
NGCF-a	0.0257	0.0335	0.1294	0.1535	0.0462	0.0565
NGCF-ta	0.0302	0.0398	0.1475	0.1741	0.0469	0.0571



a) Recall curve for Gowalla dataset

b) Training Loss for Gowalla dataset

c) Training loss for the Amazon Book dataset

d) Recall curve for Amazon-Book dataset

Fig. 5. Curves for Training of NGCF and its streamlined variations.

SVD is a collaborative filtering matrix factorization algorithm used to predict explicit rating feedback in not very large datasets.

ALS is a collaborative filtering matrix factorization algorithm used for explicit or implicit feedback in big datasets. It is optimized by Spark Machine Learning Library (MLLib) to achieve scalability and distributed computing capabilities.

SAR is a collaborative filtering similarity-centered algorithm for implicit feedback datasets.

FAST is a collaborative filtering general-purpose algorithm with embeddings and biases for customers and products.

BiVAE is a collaborative filtering generative framework for dyadic data (e.g., customer-product interactions).

Table 3 illustrates how our proposed framework ImprovedGCN outperforms the state-of-the-art approaches which exist in the literature. A recommendation framework is trained to employ each of the collaborative filtering algorithms as depicted in Table 3.

Although the proposed approach falls behind in a few numbers of precision and recall, however, it overall performs much better as compared to other state-of-the-art techniques. For example, as far as the recall is concerned only BPR and BiVAE have better recall than ImprovedGCN, and the proposed approach outperforms NCF, SVD, ALS, SAR, and FastAI. The same is the case for precision. It has been observed that as the number of products is increased, similar values are obtained and the approach still outperforms SOTA techniques.

For ranking measures, k = 10 (top 10 recommended products) is employed. It can be observed that our proposed ImprovedGCN model outperforms other state-of-the-art approaches.

To the best of our knowledge, the proposed approach namely ImprovedGCN outperforms when compared to the state-of-the-art approaches which have been proposed in recent times in terms of prediction accuracy, RMSE, MAE, Precision, Recall and other ranking measures like NDCG.

The performance of ImprovedGCN is compared to SAR and NCF on MovieLens datasets of 100 k and 1 m. The technique of loading the data and splitting is the same as described earlier in the article and the GPU employed was a GeForce GTX 1080Ti.

The configuration used across all three frameworks is as follows: seed = 42, epochs = 15. The configuration for ImprovedGCN is: batch_size = 1024, embed_size = 64, learning_rate = 0.015, decay = 0.0001, n_layers = 3; for NCF is: batch_size = 1024, learning_rate = 0.001, layer_sizes=[16, 8, 4], n_factors = 4; for SAR is: time_now = None, time_decay_coefficient = 30, timedecay_formula = True, similarity_type=”jaccard”.

A deep comparison of the proposed approach with these two states of the art approaches i.e., SAR and NCF have been depicted in Table 4.

The model, when compared to NCF and SAR, takes more training time as compared to SAR. However, the table depicts that the model performs far better in terms of recommending time, recall, NDCG, precision, and mean average precision.

The depicted results reveal that ImprovedGCN performs much better as compared to the other two frameworks. Furthermore, our proposed approach called ImprovedGCN performs better than other state-of-the-art algorithms when it comes to novelty, serendipity, and diversity.

The following investigation shows how our proposed approach performs better in terms of these metrics. Although in our experimentation, the recommendation accuracy of our framework has been evaluated in terms of ‘recall@k’ and ‘ndcg@k’. However, since accuracy itself doesn’t assure acceptable recommendations, a few evaluation metrics are also assessed like ‘serendipity@k’.

Serendipity means how surprising and relevant a recommendation is. Surprise is determined as a weighted average of the differences between the probability $P_i(u)$ that a product i is suggested for a particular customer u and the probability $P_i(U)$ that product i is suggested for any customer. It is determined as follows:

$$SRDP@k = \frac{1}{|U|}\sum_{u\in U}\frac{1}{|I_k(u)|}\sum_{i\in I_k(u)}\max(P_i(u) - P_i(U), 0). \tag{21}$$

Table 5 illustrates the values of serendipity for the state-of-the-art approaches and ImprovedGCN.

Serendipity gives unexpected and surprising recommendations and the table shows that the model gives good results when it comes to serendipity.

The Diversity (i.e., novelty) of recommendations is determined as follows:

$$Nov@k = \frac{1}{|U|}\sum_{u\in U}\sum_{i\in I_k(u)}\frac{-\log_2 D(i)/|U|}{|I_k(u)|}. \tag{22}$$

The performance of the proposed approach and NGCF has been shown in Fig. 6 when it comes to novelty.

5.3. Fast Loss

BPR is a broadly employed pairwise loss to optimize recommender frameworks. But BPR randomly samples customer-product interactions to constitute a mini-batch.

Therefore, it completely fails to utilize the parallel computing capability of the GPU. BPR samples the interactions to constitute a mini-batch, and the information cannot constitute a well-arranged matrix.

BPRMF is executed using Amazon-Book data set employing C++ on CPU and TensorFlow on GPU.

The aforesaid statistics for BPR have been illustrated in Fig. 7.

The fast loss on ImprovedGCN can be illustrated using the following expression:

$$\sum_{u\in U}\sum_{i\in V} d_{ui}(x_{ui} - \widehat{x}_{ui})^2 \tag{23}$$

where $d_{ui} = \beta, x_{ui} = 1$; there exists an interaction among u and i and $d_{ui} = 1, x_{ui} = 0$; otherwise. The above expression can be mathematically transformed into the following:

$$\sum_{u\in U}\sum_{i\in V_u^+}\left[(\beta - 1)\widehat{x}_{ui}^2 - 2\beta\widehat{x}_{ui}\right] + \sum_{r=1}^{c}\sum_{s=1}^{c}\left(\sum_{u\in U} p_{ur}p_{us}\right)\left(\sum_{i\in V} q_{ir}q_{is}\right). \tag{24}$$

Expression (23) takes into consideration all possible customer-product pairs whereas expression (24) takes historical and past customer-product pairs.

Therefore, the best features of fast loss on ImprovedGCN are as follows:

(1) It can favor any framework of inner product arrangement.
(2) The time complexity is given by $\mathscr{O}(|R|c + |N|c^2)$.
(3) It is linear to the number of viewed interactions.

Table 3
Comparison of the state of the art approaches and ImprovedGCN.

Algorithm	NDCG	Recall	RMSE	Precision	MAP	MAE
BPR	0.441995	0.212521	0.928765	0.388228	0.132476	0.74265
NCF	0.396117	0.180774	0.938054	0.347295	0.107719	0.74237
SVD	0.095929	0.032782	0.938680	0.091197	0.012872	0.74268
ALS	0.044238	0.017795	0.965037	0.048461	0.004731	0.75300
SAR	0.382460	0.176384	1.253804	0.330752	0.110590	1.04848
FastAI	0.147865	0.053823	0.943083	0.130328	0.025502	0.744336
BiVAE	0.475076	0.219144	0.927543	0.411770	0.146125	0.73214
ImprovedGCN	0.419845	0.182835	0.918643	0.379625	0.088525	0.72546

Table 4
Deep Comparison of the state-of-the-art approaches (NCF and SAR) and ImprovedGCN.

Dataset size	Framework	Recommend -ing time	Training time	Recall @10	nDCG @10	Precision @10	MAP @10
1 m	NCF	85.4566	1601.5845	0.180774	0.396117	0.347295	0.107719
1 m	SAR	2.8356	4.5592	0.176384	0.382460	0.330752	0.110590
1 m	ImprovedGCN	1.4342	396.7297	0.182835	0.419845	0.379625	0.088525
100 k	NCF	7.7659	116.3173	0.184579	0.397602	0.347502	0.108724
100 k	SAR	0.1143	0.4894	0.178384	0.385458	0.338752	0.115589
100 k	ImprovedGCN	0.6444	27.8864	0.205815	0.436296	0.381865	0.129235

Table 5
Comparison of serendipity for the state of the art approaches and ImprovedGCN.

Amazon-Book	SRDP@20	Gowalla	SRDP@20	Yelp2018	SRDP@20
ImprovedGCN	**0.0170**	ImprovedGCN	**0.0142**	ImprovedGCN	**0.0155**
NGCF	0.0128	NGCF	0.0111	NGCF	0.0150
BPRMF	0.0096	BPRMF	0.0102	BPRMF	0.0099
GC-MC	0.0092	GC-MC	0.0083	GC-MC	0.0076

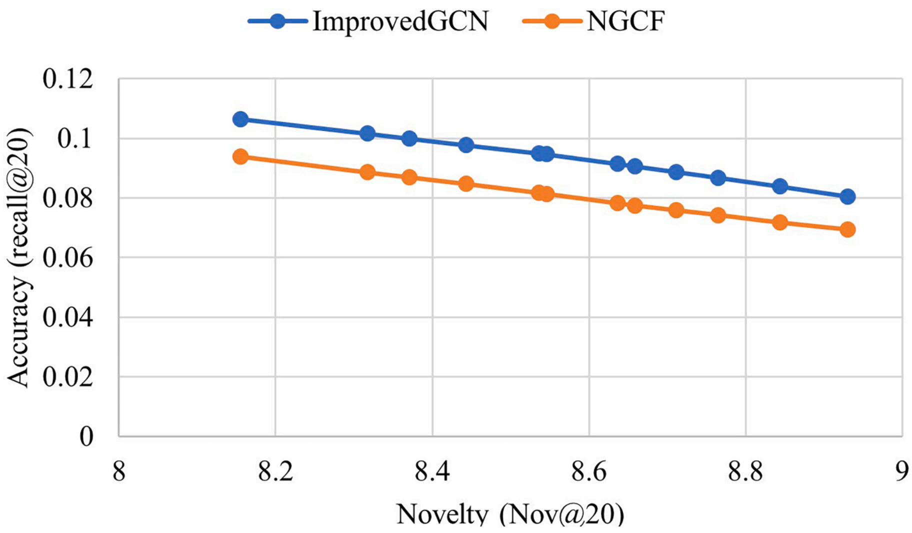
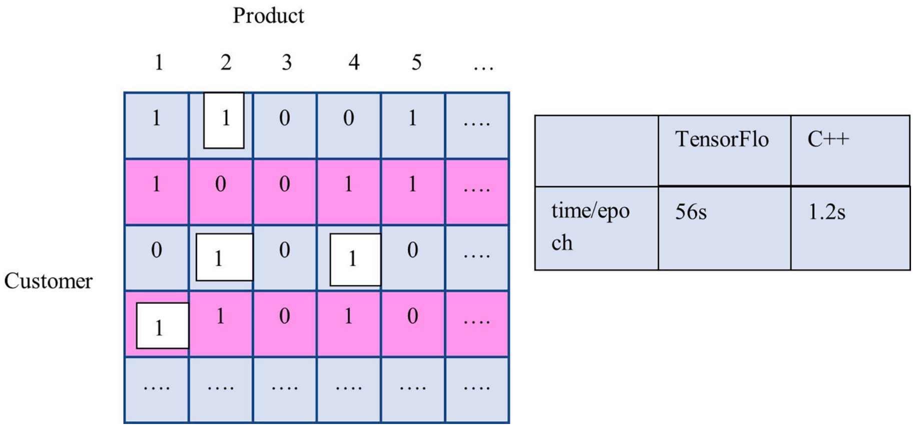


Fig. 6. Performance of proposed approach and NGCF.



1	2	3	4	5	…
1	1	0	0	1	….
1	0	0	1	1	….
0	1	0	1	0	….
1	1	0	1	0	….
….	….	….	….	….	….

	TensorFlo	C++
time/epo ch	56s	1.2s

Fig. 7. Statistics for BPR.

In contrast to BPR which samples interactions as a batch, Fast Loss samples customers (rows) as a batch. The information contained in the batch is well arranged. The advantage of this fact is that it permits the best utilization of the speedup of CPU/GPU. Fast Loss conveys two to three times speedup in contrast to BPR. Furthermore, ImprovedGCN augmented with Fast Loss may obtain better performance in contrast to BPR loss. Table 6 shows the speedup comparison of BPR and Fast loss on ImprovedGCN.

This table shows the fact that fast-loss brings comparable performance and great efficiency improvement compared with BPR.

Table 6
Speedup Comparison of BPR and Fast Loss on ImprovedGCN.

	Fast-loss			BPR		
	s	N	T	s	N	T
Amazon-Book	1.6 s	870	21.9 min	436 s	340	41.9 h
Gowalla	0.6 s	860	87 s	66 s	840	15.2 h
Yelp2018	0.9 s	1250	16.8 min	116 s	520	16.7 h

ImprovedGCN optimized with Fast Loss can achieve comparable performance to that with BPR loss. The best features of Fast Loss have been depicted in Fig. 8.

5.4. Mult-VAE

It is an item-centered CF approach to the *variational autoencoder* (VAE). It is based on the assumption that data is produced using multinomial distribution and utilizing variational interpretation for the evaluation of the parameters. The source code for the model has been run by fine-tuning the dropout fraction in [0, 0.2, 0.5] and the γ in [0.2, 0.4, 0.6, 0.8].

5.5. Graph regularized matrix factorization (GRMF)

To make the contrast reasonable on items recommended, the score prediction loss is changed to BPR loss. In this model, a graph Laplacian regularizer is utilized to smooth MF.

The *graph regularized matrix factorization* GRMF (Rao et al., 2015) can be described as:

$$L = -\sum_{u\in U}\sum_{i\in N_u}\left(\sum_{j\notin N_u}\ln\sigma\left(z_u^T z_i - z_u^T z_j\right) + \lambda_g\left|\left|z_u - z_i\right|\right|^2\right) + \lambda\left|\left|Z\right|\right|^2 \tag{25}$$

where λ_g lies in the values $\{10^{-5}, 10^{-4}, \ldots, 10^{-1}\}$. The setup of other hyperparameters is kept the same as ImprovedGCN. Furthermore, a comparison is made with a variation that augments normalization to graph Laplacian: $\lambda_g\left|\left|\frac{z_u}{\sqrt{|N_u|}} - \frac{z_i}{\sqrt{|N_i|}}\right|\right|^2$, which is known as GRMF-norm. The embeddings are smoothed by employing a Laplacian regularizer using two GRMF approaches whereas our ImprovedGCN acquires smoothness in embeddings in the prediction method.

Table 7 depicts the overall performance contrast among ImprovedGCN and contending models.

In this table the model is compared to a different set of approaches since the features of these frameworks are common and similar to ImprovedGCN. The results presented show that ImprovedGCN has better performance compared to the other frames except Mult-VAE for the Amazon-Book dataset but the difference is very small in this case as well.

Table 7
Overall performance contrast between ImprovedGCN and contending models.

Dataset	Amazon-Book		Gowalla		Yelp2018	
Model	ndcg	recall	ndcg	recall	ndcg	recall
Mult-VAE	0.0313	0.0405	0.1333	0.1640	0.0449	0.0582
GRMF	0.0269	0.0352	0.1203	0.1475	0.0461	0.0569
GRMF-norm	0.0267	0.0350	0.1260	0.1555	0.0452	0.0560
NGCF	0.0261	0.0341	0.1324	0.1571	0.0459	0.0564
ImprovedGCN	0.0311	0.0404	0.1548	0.1828	0.0529	0.0645

5.6. Setup of hyperparameters

Our ImprovedGCN is optimized with Adam (Kingma and Ba, 2014) and a learning rate of 0.001 and minimum batch dimensions of 1024 is employed (in Amazon-Book the minimum batch dimension is increased to 2048 to achieve pace). As in the case of NGCF, the size of the embeddings is set to 64 for all the approaches and the initialization of the embedding parameters is carried out using the Xavier approach (Glorot and Bengio, 2010). The value of L_2 regularization factor λ lies in the values $\{10^{-6}, 10^{-5}, \ldots, 10^{-2}]\}$ and in many instances, the optimized estimate of 10^{-4} is employed. A value of $\frac{1}{1+L}$ is uniformly assigned to layer concatenation coefficient, here L specifies the count of layers. Better performance may be obtained when a value of 3 is assigned to L. The value of 1 to 4 for L is usually used. Particularly, our ImprovedGCN can be converged if the count of epochs is assigned as 1000. Our suggested framework has been implemented in PyTorch.

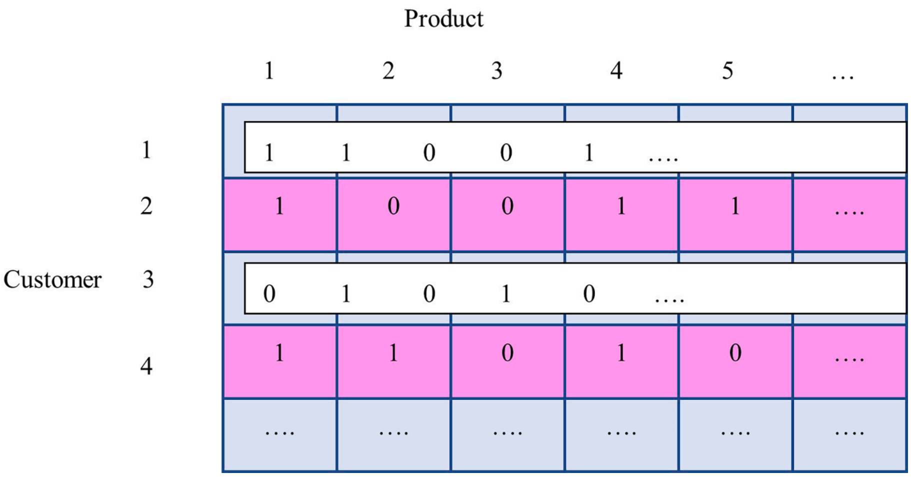
5.7. Comparison of performance with NGCF

An in-depth comparison has been carried out alongside NGCF, observing the effectiveness at several levels in Table 8 which also demonstrates relative improvement proportion on every measure.

When the number of layers is increased, the performance can improve, however, the advantages get reduced. It has been observed generally that when the layer number is increased from 0 to 1, it results in the largest performance gain, and utilizing a layer number of 3 results in acceptable performance in most scenarios.

Furthermore, the training plots of training loss and testing recall have been plotted in Fig. 9 to exhibit the benefits of ImprovedGCN.

The following can be observed: in every scenario, ImprovedGCN performs far better as compared to NGCF.


Product

Customer	1	2	3	4	5	…
1	1	1	0	0	1	….
2	1	0	0	1	1	….
3	0	1	0	1	0	….
4	1	1	0	1	0	….
	….	….	….	….	….	….

Fig. 8. Best features of Fast Loss.

Table 8
Comparison of performance among NGCF and ImprovedGCN at diverse layers.

Dataset		Amazon-Book		Gowalla		Yelp2018	
Layer	Model	ndcg	recall	ndcg	recall	ndcg	recall
Layer 1	ImprovedGCN	0.0297	0.0383	0.1491	0.1754	0.0514	0.0630
	NGCF	0.0240	0.0312	0.1314	0.1555	0.0441	0.0542
Layer 2	ImprovedGCN	0.0314	0.0410	0.1523	0.1776	0.0503	0.0621
	NGCF	0.0253	0.0329	0.1306	0.1546	0.0464	0.0565
Layer 3	ImprovedGCN	0.0317	0.0409	0.1554	0.1822	0.0524	0.0638
	NGCF	0.0260	0.0336	0.1323	0.1568	0.0476	0.0578
Layer 4	ImprovedGCN	0.0311	0.0404	0.1548	0.1828	0.0529	0.0645
	NGCF	0.0261	0.0341	0.1324	0.1571	0.0459	0.0564

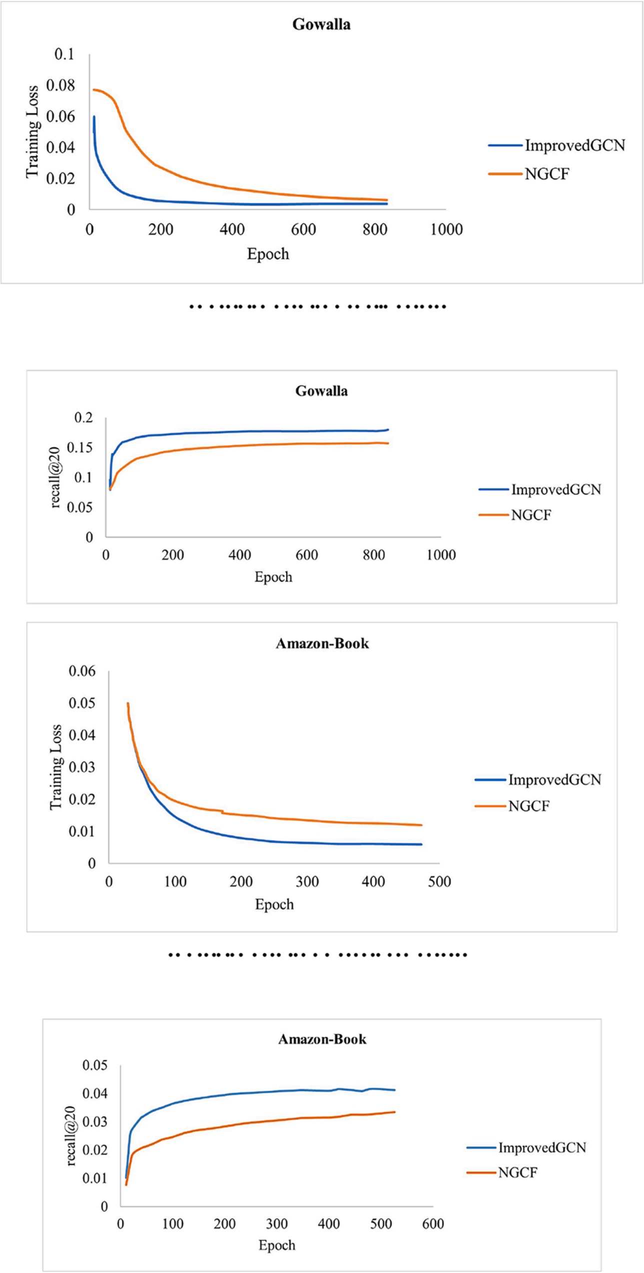


Fig. 9. Curves for training of ImprovedGCN and NGCF.

Observing Table 7 in alignment with Table 2, it can be noticed that the performance of ImprovedGCN is far best when compared to NGCF-ta, the variation of NGCF which eliminates non-linear activation and feature transformation.

NGCF-ta differs from ImprovedGCN on the lines that no self-connections are included in ImprovedGCN. As an instance, using the Gowalla dataset, the greatest recall recorded in NGCF is 0.1569, whereas our ImprovedGCN can achieve 0.1829 in a four-layer setup, which is marginally 16.55% greater. If the average recall improvement on three datasets is considered, it is 16.51% and the enhancement in NDCG is 16.86%.

Since NGCF-ta yet comprises more operations in contrast to ImprovedGCN i.e., self-connection, dropout, and interaction amid user and item embeddings in graph convolution. This tells us that these operations are of no use for NGCF-ta.

If the number of layers is increased, then the performance can be enhanced, but the benefits are decreased. During the training task, low training loss occurs in ImprovedGCN, which demonstrates that the training data is fitted in ImprovedGCN far better as compared to NGCF.

Furthermore, low training loss is effectively transferred to good accuracy, which demonstrates the power of ImprovedGCN. One can notice that a large performance gain is achieved when the number of layers is increased from 0 (i.e., MF method (Wang et al., 2019b)) to 1, and utilizing the number of layers as 3 exhibits acceptable performance in many scenarios. In comparison, the high training loss with low testing accuracy in NGCF results in trouble to train such a burdensome model.

Observe that in the figures, the training task for the optimal setup of hyperparameters for both approaches is demonstrated. However, enhancing the learning rate of NGCF may result in low training loss (yet smaller as compared to ImprovedGCN), but one cannot observe improvement in recall, since making the training loss low in this fashion only determines the insignificant and direct solution for NGCF.

5.8. Comparison of performance with other models

The better score which can be achieved for every model is demonstrated. The contrast of effectiveness with other contending approaches has been depicted in Table 7.

It can be noticed that ImprovedGCN achieves better than other models on all datasets depicting its efficiency with simplicity. The effectiveness of GRMF is higher than MF which acknowledges the usefulness of applying smoothness to embeddings utilizing the Laplacian regularizer.

Observe that ImprovedGCN can do more better by fine-tuning the α_ℓ, whereas here consistent setup of $\frac{1}{L+1}$ is only utilized to circumvent overtuning. Between these models, Mult-VAE has more performance than GRMF and NGCF. If Laplacian regularizer is employed with normalization, GRMF-norm executes well than GRMF on Gowalla, whereas incorporates no advantages when executed on Amazon-Book and Yelp2018.

5.9. Ablation and effectiveness analysis

Several investigations are carried out on ImprovedGCN by demonstrating how to layer concatenation and symmetric normalization affect its efficiency. To justify the rationality of ImprovedGCN, the effect of embedding smoothness which is the significant cause of ImprovedGCN's effectiveness is also investigated.

5.9.1. *Influence of layer concatenation*

The outcomes of ImprovedGCN and its variation ImprovedGCN-single, which doesn't employ layer concatenation, are shows in Fig. 10. The outcomes on Yelp2018 have not been shown due to limited space. The work depicts outcomes of ImprovedGCN and its variation ImprovedGCN-single that never utilize level concatenation (i.e., $Z(L)$ is employed for final prediction for an L-layer ImprovedGCN).

Concentrating on ImprovedGCN-single, it is observed that its performance first enhances and then decreases when the number of layers is increased from 1 to 4. This shows that considering first-order and second-order neighbors for smoothing node embedding is very advantageous for CF, but when higher-order neighbors are considered, the problem of over smoothness arises.

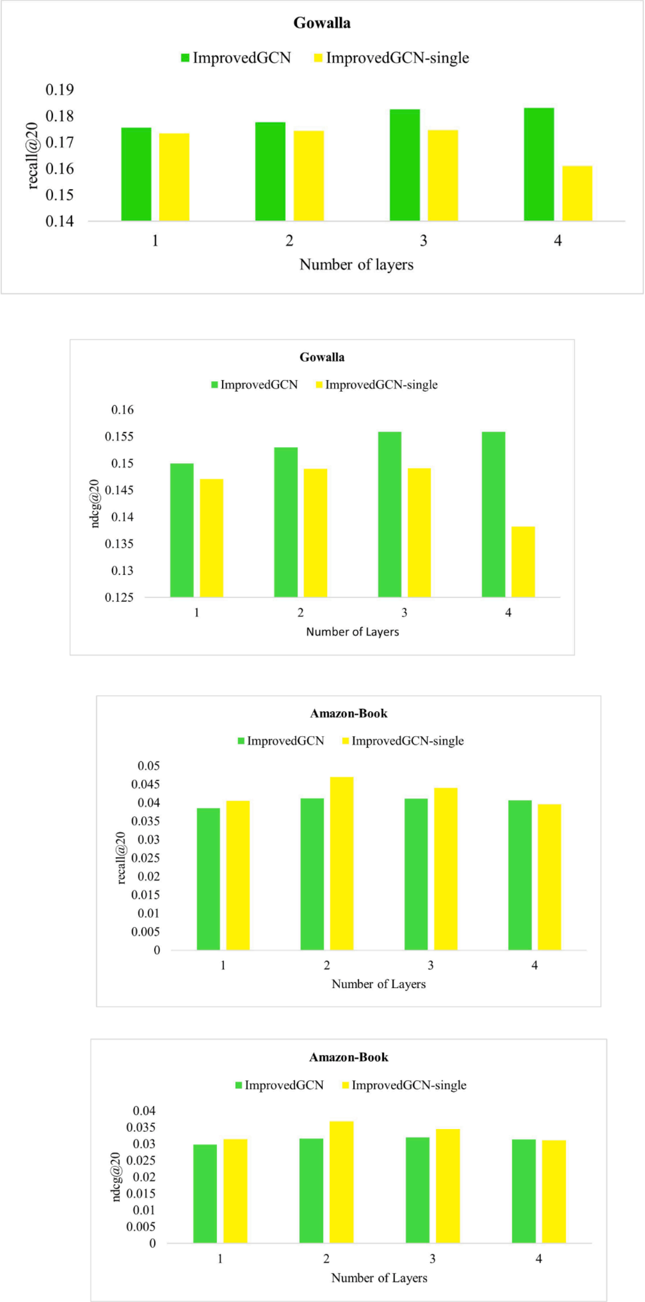


Fig. 10. Outcomes of ImprovedGCN and its variation which doesn't employ layer concatenation (i.e., ImprovedGCN-single) at diverse layers on Gowalla and Amazon-Book dataset (Yelp2018 dataset exhibits same behavior and has been excluded for space).

The maximum performance is on layer 2 in many scenarios, whereas after it the performance degrades rapidly up to layer 4. Concentrating on ImprovedGCN, it has been noticed that its implementation consistently enhances with an increase in the count of layers, and despite it, the performance does not deteriorate even with four layers. This proves the efficiency of layer concatenation. When the two approaches are compared, it can be observed that ImprovedGCN gradually performs better than ImprovedGCN-single on Gowalla, but not on Yelp2018 and Amazon-Book (here the two-layer ImprovedGCN-single does well). Pertinent to this, two aspects require to be observed to derive the conclusion:

1) ImprovedGCN-single is a special version of ImprovedGCN that assigns the value of 1 to α_L and 0 to other α_ℓ.

2) The parameter α_ℓ is not tuned and for simplicity value of $\frac{1}{L+1}$ is assigned to it consistently to ImprovedGCN. The effectiveness of ImprovedGCN can further be enhanced by tuning α_ℓ.

5.9.2. *Effect of symmetric sqrt normalization*

In ImprovedGCN, when neighborhood aggregation is carried out, the symmetric sqrt normalization $\frac{1}{\sqrt{|N_u|}\sqrt{|N_i|}}$ on every neighborhood embedding is utilized. To examine its reasonability, different variations are explored. Observe that if the normalization is removed, instability occurs during training, and problems like *not-a-value* (NAN) are encountered, therefore, this scenario is not shown.

The utilization of normalization at the leftward portion (i.e., the objective node's coefficient) and the right portion (i.e., the neighbor node's coefficient) is employed. L_1 normalization, i.e., eliminating the square root is used. The best arrangement in particular is employing sqrt regularization at both boundaries (i.e., the current model of ImprovedGCN). Eliminating any of these sides will degrade and deteriorate the performance by a large factor.

The second-best arrangement is employing L_1 normalization at the leftward boundary only (i.e., ImprovedGCN-L_1-L). It is the same as normalizing the adjacency matrix as a stochastic matrix through the in-degree. It is advantageous for the sqrt regularization if regularization is applied symmetrically on two sides, but will deteriorate the effectiveness of L_1 normalization.

Another arrangement can be at the rightward boundary only as depicted in Table 9 (i.e., ImprovedGCN-L_1-R).

5.9.3. *Embedding smoothness analysis*

As it has been analyzed in the earlier sections, a two-layer ImprovedGCN flattens an individual's embedding centered on the individuals who have an overlay on collaborated products, it is assumed that our ImprovedGCN is effective due to such smoothing of embeddings. It is

Table 9
Effectiveness of the three-layer ImprovedGCN with diverse alternatives of normalization structures.

Dataset	Amazon-Book		Gowalla		Yelp2018	
Approach	ndcg	recall	ndcg	recall	ndcg	recall
ImprovedGCN	**0.0311**	**0.0404**	**0.1548**	**0.1828**	**0.0529**	**0.0645**
ImprovedGCN-L	0.0298	0.0382	0.1316	0.1588	0.0508	0.0618
ImprovedGCN-R	0.0195	0.0251	0.1155	0.1419	0.0400	0.0520
ImprovedGCN-L_1-L	0.0319	0.0418	0.1413	0.1723	0.0510	0.0629
ImprovedGCN-L_1-R	0.0258	0.0333	0.1347	0.1577	0.0476	0.0586
ImprovedGCN-L_1	0.0274	0.0360	0.1318	0.158	0.0464	0.0572

proved by defining the smoothness of user embeddings as underneath:

$$S_U = \sum_{u=1}^{m} \sum_{w=1}^{m} c_{w \to u} - \left(\frac{z_u}{||z_u||^2} - \frac{z_w}{||z_w||^2} \right)^2 \tag{26}$$

wherever the L_2 norm is employed on embeddings to remove the effect of the measure of the embedding. Likewise, the definition for item embeddings can be achieved.

Observe that two-layer ImprovedGCN-single performs better than MF in terms of accuracy of the recommendation by a big factor. It can be observed that the loss in the smoothness of ImprovedGCN-single is less than in MF by a large factor. This shows that carrying out graph convolution in this manner leads to embeddings being smoother and appropriate for a recommendation. Table 10 illustrates the smoothness of two frameworks, matrix factorization and the ImprovedGCN-single and lower values indicate larger smoothness.

Since in matrix factorization, smoothness loss is small as compared to several techniques, however, the proposed model performs even better when compared to MF when it comes to smoothness loss (smaller numbers mean good i.e. lower values indicate larger smoothness).

5.10. *Investigation of hyper-parameter*

The utmost significant hyperparameter to fine-tune is the L_2 regularization coefficient apart from the standard hyperparameter termed as learning rate when ImprovedGCN is applied to a new dataset. It can be observed that ImprovedGCN is comparatively unaffected when λ is changed, even when a value of 0 is assigned to λ, ImprovedGCN performs well as compared to NGCF, where dropout is utilized to circumvent overfitting. The change in the performance of ImprovedGCN concerning λ is examined here. The ideal score for Amazon-Book, Gowalla, and Yelp2018 is $10^{-4}, 10^{-4}$ and 10^{-3} correspondingly. The abovesaid depicts that ImprovedGCN is lesser susceptible to overfitting because the trainable constraints in ImprovedGCN are ID embeddings of the 0th level, the complete approach is easier to regularize and train. When a value bigger than 10^{-3} is assigned to λ, the performance deteriorates rapidly, which depicts that too robust regularization will hurt method training and is never recommended.

6. Conclusion

ImprovedGCN has been suggested in this research which comprises two required regions i.e., improved graph convolution and layer concatenation. In this study, the unnecessary complexity of GCN for CF is argued, and carried out experimental investigations to prove the same.

In improved graph convolution, non-linear activation and feature transformation have been removed which are two standard tasks in GCN, but they enhance the intricacy of training. Several empirical studies have been carried out to exhibit the strong points of ImprovedGCN i.e., its simplicity, its easiness of training, and more robustness.

In layer concatenation, a node's concluding embedding is built as the weighted total of its embeddings on all levels so that self-connections are included and is advantageous to control over smoothing. It is assumed that these intuitions of ImprovedGCN are motivational to build more robust RSs in near future. The perception of ImprovedGCN in these approaches is exploited.

Other future work is to individualize the layer concatenation weights α_ℓ, to force adaptive order flattening for diverse individuals (e.g., sparse users may need an additional signal from multi-hop neighbors whereas active individuals may need low).

Due to the existence of linked graph information in real-world scenarios, graph centered approaches are becoming significant in recommendation i.e., by exploring the associations between objects in the predictive method, these are beneficial for conventional supervised learning approaches, for instance, factorization machines (He and Chua, 2017; Rendle et al., 2011) which depict the associations implicitly.

Table 10
Smoothness loss of the embeddings learned by ImprovedGCN and MF.

Dataset	Amazon-Book	Gowalla	Yelp201
		Product Embeddings' Smoothness	
ImprovedGCN-single	16865.0	**5828.0**	**6459.7**
Matrix Factorization	28307.8	12106.6	16632.0
		Customer Embeddings' Smoothness	
ImprovedGCN-single	32191.0	12872.6	10091.6
Matrix Factorization	38034.1	15449.2	16258.1

For instance, a modern approach is to explore secondary data like item knowledge graphs (Wang et al., 2019a), multimedia content (Wei et al., 2019), and social network (Wu et al., 2019b) for recommendations where GCN is employed broadly. But these approaches may also encounter similar problems to NGCF because the item-user collaboration graph is affected by similar neural tasks which are not required.

In the end, the strong points of ImprovedGCN's simpleness, examining if a quick solution occurs for non-sampling regression loss (He et al., 2019) can be exploited. The research has shown in Dhawan et al. (2021b), Dhawan et al. (2020), and Dhawan et al. (2021a) that how the aspect of link prediction and community discovery can help in the task of recommendation.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

The authors are thankful to the editor-in-chief, associate editor and the three anonymous reviewers whose constructive comments led to a significant improvement of this paper.

References

Berg, R.v.d., Kipf, T.N., Welling, M., 2017. Graph convolutional matrix completion. In: KDD Workshop on Deep Learning Day. pp. 1–9. arXiv:1706.02263.

Bruna, J., Zaremba, W., Szlam, A., LeCun, Y., 2014. Spectral networks and locally connected networks on graphs. In: International Conference on Learning Representations (ICLR2014). pp. 1–14. arXiv:1312.6203v3.

Cao, J., Sheng, J., Cong, X., Liu, T., Wang, B., 2022. Cross-domain recommendation to cold-start users via variational information bottleneck. arXiv preprint arXiv: 220316863.

Chang, L., Chen, W., Huang, J., Bin, C., Wang, W., 2021. Exploiting multi-attention network with contextual influence for point-of-interest recommendation. Appl. Intell. 51 (4), 1904–1917. https://doi.org/10.1007/s10489-020-01868-0.

Chen, L., Xia, M., 2021. A context-aware recommendation approach based on feature selection. Appl. Intell. 51 (2), 865–875. https://doi.org/10.1007/s10489-020-01835-9.

Chen, J., Zhang, H., He, X., Nie, L., Liu, W., Chua, T.S., 2017. Attentive collaborative filtering: Multimedia recommendation with item-and component-level attention. In: Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval, pp. 335–344. https://doi.org/10.1145/3077136.3080797.

Chen, Y., Chen, B., He, X., Gao, C., Li, Y., Lou, J.G., Wang, Y., 2019. λ Opt: learn to regularize recommender models in finer levels. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. ACM. pp. 978–986. doi: 10.1145/3292500.3330880.

Chen, C.M., Wang, C.J., Tsai, M.F., Yang, Y.H., 2019.. Collaborative similarity embedding for recommender systems. The World Wide Web Conference 2637–2643. https://doi.org/10.1145/3308558.3313493.

Chen, L., Wu, L., Hong, R., Zhang, K., Wang, M., 2020. Revisiting graph based collaborative filtering: A linear residual graph convolutional network approach. In: Proceedings of the AAAI conference on artificial intelligence. vol. 34. pp. 27–34. arXiv:2001.10167.

Chen, L., Xie, T., Li, J., Zheng, Z., 2022. Graph enhanced neural interaction model for recommendation. Knowl.-Based Syst. 108616.

Chen, L., Cao, J., Wang, Y., Liang, W., Zhu, G., 2022b. Multi-view graph attention network for travel recommendation. Expert Syst. Appl. 191, 116234.

Cheng, Z., Ding, Y., Zhu, L., Kankanhalli, M., 2018. Aspect-aware latent factor model: Rating prediction with ratings and reviews. In: Proceedings of the 2018 world wide web conference, pp. 639–648. https://doi.org/10.1145/3178876.3186145.

Chu, Y., Yao, J., Zhou, C., Yang, H., 2022. Graph neural networks in modern recommender systems. Graph Neural Networks: Foundations, Frontiers, and Applications. Springer 423–445.

Covington, P., Adams, J., Sargin, E., 2016. Deep neural networks for youtube recommendations. In: Proceedings of the 10th ACM Conference on Recommender Systems. New York, NY, USA, pp. 191–198. https://doi.org/10.1145/2959100.2959190.

Defferrard, M., Bresson, X., Vandergheynst, P., 2016. Convolutional neural networks on graphs with fast localized spectral filtering. Advances in neural information processing systems 29, 3844–3852. arXiv:1606.09375.

Dhawan, S., Singh, K., Batra, A., 2020. Defining and evaluating network communities based on ground-truth in online social networks. In: The International Conference on Recent Innovations in Computing. Springer, pp. 151–163. https://doi.org/10.1007/978-981-15-8297-4_13.

Dhawan, S., Singh, K., Batra, A., 2021a. An analysis and comparison of community detection algorithms in online social networks. Intelligent Computing and Communication Systems. Springer 363–370. https://doi.org/10.1007/978-981-16-1295-4_37.

Dhawan, S., Singh, K., Batra, A., 2021. Community discovery and behavior prediction in online social networks employing node centrality. In: International Conference on Mobile Radio Communications and 5G Networks. pp. 73–85. doi: 10.1007/978-981-15-7130-5_6.

Ding, J., Quan, Y., He, X., Li, Y., Jin, D., 2019. Reinforced negative sampling for recommendation with exposure data. IJCAI 2230–2236. https://doi.org/10.24963/ijcai.2019/309.

Duan, Z., Xu, W., Chen, Y., Ding, L., 2022. ETBRec: a novel recommendation algorithm combining the double influence of trust relationship and expert users. Appl. Intell. 52, 282–294. https://doi.org/10.1007/s10489-021-02419-x.

Ebesu, T., Shen, B., Fang, Y., 2018. Collaborative memory network for recommendation systems. In: The 41st international ACM SIGIR conference on research & development in information retrieval, pp. 515–524. https://doi.org/10.1145/3209978.3209991.

Feng, S., Zhang, H., Cao, J., Yao, Y., 2018. Merging user social network into the random walk model for better group recommendation. Appl. Intell. 49, 2046–2058. https://doi.org/10.1007/s10489-018-1375-z.

Feng, F., He, X., Wang, X., Luo, C., Liu, Y., Chua, T.S., 2019. Temporal relational ranking for stock prediction. ACM Trans. Inf. Syst. 37 (2), 1–30. https://doi.org/10.1145/3309547.

Fu, H.M., Poirson, P., Lee, K.S., Wang, C., 2022. Revisiting neighborhood-based link prediction for collaborative filtering. arXiv preprint arXiv:220315789.

Glorot, X., Bengio, Y., 2010. Understanding the difficulty of training deep feedforward neural networks. In: Proceedings of the thirteenth international conference on artificial intelligence and statistics, pp. 249–256. URL: https://proceedings.mlr.press/v9/glorot10a/glorot10a.pdf.

Gori, M., Pucci, A., 2007. ItemRank: a random-walk based scoring algorithm for recommender engines. In: IJCAI'07: Proceedings of the 20th International Joint Conference on Artifical Intelligence. Morgan Kaufmann Publishers Inc. pp. 2766–2771. URL: https://www.aaai.org/Papers/IJCAI/2007/IJCAI07-444.pdf.

Hamilton, W.L., Ying, R., Leskovec, J., 2017. Inductive representation learning on large graphs. In: Proceedings of the 31st International Conference on Neural Information Processing Systems. pp. 1025–1035. arXiv:1706.02216.

Haveliwala, T.H., 2002. Topic-sensitive pagerank. In: Proceedings of the 11th International Conference on World Wide Web. ACM; WWW '02, pp. 517–526. https://doi.org/10.1145/511446.511513.

He, X., Chua, T.S., 2017. Neural factorization machines for sparse predictive analytics. In: Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval, pp. 355–364. https://doi.org/10.1145/3077136.3080777.

He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 770–778. https://doi.org/10.1109/CVPR.2016.90.

He, X., Liao, L., Zhang, H., Nie, L., Hu, X., Chua, T.S., 2017. Neural collaborative filtering. In: Proceedings of the 26th international conference on world wide web, pp. 173–182. https://doi.org/10.1145/3038912.3052569.

He, X., He, Z., Song, J., Liu, Z., Jiang, Y.G., Chua, T.S., 2018. Nais: Neural attentive item similarity model for recommendation. IEEE Trans. Knowl. Data Eng. 30 (12), 2354–2366. https://doi.org/10.1109/TKDE.2018.2831682.

He, X., Tang, J., Du, X., Hong, R., Ren, T., Chua, T.S., 2019. Fast matrix factorization with nonuniform weights on missing data. IEEE Trans. Neural Networks Learn. Syst. 31 (8), 2791–2804. https://doi.org/10.1109/TNNLS.2018.2890117.

Kabbur, S., Ning, X., Karypis, G., 2013. Fism: factored item similarity models for top-n recommender systems. In: Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 659–667. https://doi.org/10.1145/2487575.2487589.

Kingma, D.P., Ba, J., 2014. Adam: A method for stochastic optimization. arXiv preprint arXiv:14126980.

Kipf, T.N., Welling, M., 2016. Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:160902907.

Klicpera, J., Bojchevski, A., Günnemann, S., 2018. Predict then propagate: Graph neural networks meet personalized pagerank. In: International Conference on Learning Representations (ICLR). pp. 1–15. arXiv:1810.05997.

Koren, Y., 2008. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In: Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 426–434. https://doi.org/10.1145/1401890.1401944.

Koren, Y., Bell, R., Volinsky, C., 2009. Matrix factorization techniques for recommender systems. Computer 42 (8), 30–37. https://doi.org/10.1109/MC.2009.263.

Liang, D., Krishnan, R.G., Hoffman, M.D., Jebara, T., 2018. Variational autoencoders for collaborative filtering. In: Proceedings of the 2018 world wide web conference, pp. 689–698. https://doi.org/10.1145/3178876.3186150.

Li, Q., Han, Z., Wu, X.M., 2018. Deeper insights into graph convolutional networks for semi-supervised learning. In: Thirty-Second AAAI conference on artificial intelligence. pp. 3538–3545. arXiv:1801.07606.

Lin, Z., Tian, C., Hou, Y., Zhao, W.X., 2022. Improving graph collaborative filtering with neighborhood-enriched contrastive learning. pp. 2320–2329. arXiv preprint arXiv:220206200. doi: 10.1145/3485447.3512104.

Liu, F., Liao, J., Zheng, J., Wang, S., Wang, X., 2022. Gcn recommendation model based on the fusion of dynamic multiple-view latent interest topics. Preprint submitted to Expert Systems with Applications.

Liu, K., Xue, F., He, X., Guo, D., Hong, R., 2022. Joint multi-grained popularity-aware graph convolution collaborative filtering for recommendation. IEEE Trans. Comput. Soc. Syst.

Ni, J., Huang, Z., Hu, Y., Lin, C., 2022. A two-stage embedding model for recommendation with multimodal auxiliary information. Inf. Sci. 582, 22–37.

Qiu, J., Tang, J., Ma, H., Dong, Y., Wang, K., Tang, J., 2018. Deepinf: Social influence prediction with deep learning. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 2110–2119. https://doi.org/10.1145/3219819.3220077.

Rao, N., Yu, H.F., Ravikumar, P., Dhillon, I.S., 2015. Collaborative filtering with graph information: Consistency and scalable methods. In: Advances in Neural Information Processing Systems 28. vol. 2. pp. 2107–2115. https://dl.acm.org/doi/10.5555/2969442.2969475.

Rendle, S., Freudenthaler, C., 2014. Improving pairwise learning for item recommendation from implicit feedback. In: Proceedings of the 7th ACM international conference on Web search and data mining, pp. 273–282. https://doi.org/10.1145/2556195.2556248.

Rendle, S., Freudenthaler, C., Gantner, Z., Schmidt-Thieme, L., 2009. BPR: bayesian personalized ranking from implicit feedback. In: Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence. AUAI Press. pp. 452–461. arXiv:1205.2618.

Rendle, S., Gantner, Z., Freudenthaler, C., Schmidt-Thieme, L., 2011. Fast context-aware recommendations with factorization machines. In: Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval, pp. 635–644. https://doi.org/10.1145/2009916.2010002.

Tay, Y., Anh, Tuan L., Hui, S.C., 2018. Latent relational metric learning via memory-based attention for collaborative ranking. In: Proceedings of the 2018 World Wide Web Conference, pp. 729–739. https://doi.org/10.1145/3178876.3186154.

Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., Bengio, Y., 2017. Graph attention networks. arXiv preprint arXiv:171010903.

Wang, J., De Vries, A.P., Reinders, M.J., 2006. Unifying user-based and item-based collaborative filtering approaches by similarity fusion. In: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval, pp. 501–508. https://doi.org/10.1145/1148170.1148257.

Wang, X., He, X., Cao, Y., Liu, M., Chua, T.S., 2019. Kgat: knowledge graph attention network for recommendation. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. pp. 950–958. doi: 10.1145/3292500.3330989.

Wang, X., He, X., Wang, M., Feng, F., Chua, T.S., 2019. Neural graph collaborative filtering. In: Proceedings of the 42nd international ACM SIGIR conference on Research and development in Information Retrieval. pp. 165–174. doi: 10.1145/3331184.3331267.

Wang, F., Li, Y., Zhang, Y., Wei, D., 2022. Klgcn: Knowledge graph-aware light graph convolutional network for recommender systems. Expert Syst. Appl. 195, 116513 https://doi.org/10.1016/j.eswa.2022.116513.

Wei, Y., Wang, X., Nie, L., He, X., Hong, R., Chua, T.S., 2019. MMGCN: multi-modal graph convolution network for personalized recommendation of micro-video. In: Proceedings of the 27th ACM International Conference on Multimedia, pp. 1437–1445. https://doi.org/10.1145/3343031.3351034.

Wei, W., Huang, C., Xia, L., Xu, Y., Zhao, J., Yin, D., 2022. Contrastive meta learning with behavior multiplicity for recommendation. In: Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining. WSDM '22, pp. 1120–1128. https://doi.org/10.1145/3488560.3498527.

Wen, P., Yuan, W., Qin, Q., Sang, S., Zhang, Z., 2021. Neural attention model for recommendation based on factorization machines. Appl. Intell. 51 (4), 1829–1844. https://doi.org/10.1007/s10489-020-01921-y.

Wu, F., Souza, A., Zhang, T., Fifty, C., Yu, T., Weinberger, K., 2019. Simplifying graph convolutional networks. In: International conference on machine learning. pp. 6861–6871. arXiv:1902.07153.

Wu, L., Sun, P., Fu, Y., Hong, R., Wang, X., Wang, M., 2019. A neural influence diffusion model for social recommendation. In: Proceedings of the 42nd international ACM SIGIR conference on research and development in information retrieval. pp. 235–244. doi: 10.1145/3331184.3331214.

Wu, Y., Zhao, Y., Wei, S., 2020. Collaborative filtering recommendation algorithm based on interval-valued fuzzy numbers. Appl. Intell. 50(9), 2663–2675. doi: 10.3233/IDA-150438.

Wu, B., Zhong, L., Yao, L., Ye, Y., 2022. Eagcn: An efficient adaptive graph convolutional network for item recommendation in social internet of things. IEEE Internet Things J. 1. https://doi.org/10.1109/JIOT.2022.3151400.

Xiao, T., Shen, H., 2019. Neural variational matrix factorization for collaborative filtering in recommendation systems. Appl. Intell. 49 (10), 3558–3569. https://doi.org/10.1007/s10489-019-01469-6.

Xing, S., Wang, Q., Zhao, X., Li, T., et al., 2019. Content-aware point-of-interest recommendation based on convolutional neural network. Appl. Intell. 49 (3), 858–871. https://doi.org/10.1007/s10489-018-1276-1.

Xu, K., Hu, W., Leskovec, J., Jegelka, S., 2018. How powerful are graph neural networks? arXiv preprint arXiv:181000826.

Yang, J.H., Chen, C.M., Wang, C.J., Tsai, M.F., 2018. HOP-rec: high-order proximity for implicit recommendation. In: Proceedings of the 12th ACM Conference on Recommender Systems, pp. 140–144. https://doi.org/10.1145/3240323.3240381.

Yang, Y., Rao, Y., Yu, M., Kang, Y., 2022. Multi-layer information fusion based on graph convolutional network for knowledge-driven herb recommendation. Neural Networks 146, 1–10.

Ying, R., He, R., Chen, K., Eksombatchai, P., Hamilton, W.L., Leskovec, J., 2018. Graph convolutional neural networks for web-scale recommender systems. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 974–983. https://doi.org/10.1145/3219819.3219890.

Yuan, F., He, X., Karatzoglou, A., Zhang, L., 2020. Parameter-efficient transfer from sequential behaviors for user modeling and recommendation. In: Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 1469–1478. https://doi.org/10.1145/3397271.3401156.

Zhang, P., Zhang, Z., Tian, T., Wang, Y., 2019. Collaborative filtering recommendation algorithm integrating time windows and rating predictions. Appl. Intell. 49 (8), 3146–3157. https://doi.org/10.1007/s10489-019-01443-2.

Zhang, X., Luo, H., Chen, B., Guo, G., 2020. Multi-view visual bayesian personalized ranking for restaurant recommendation. Appl. Intell. 50 (9), 2901–2915. https://doi.org/10.1007/s10489-020-01703-6.

Zhao, C., Li, C., Fu, C., 2019. Cross-domain recommendation via preference propagation graphnet. In: Proceedings of the 28th ACM International Conference on Information and Knowledge Management, pp. 2165–2168. https://doi.org/10.1145/3357384.3358166.

Zhu, H., Feng, F., He, X., Wang, X., Li, Y., Zheng, K., Zhang, Y., 2020. Bilinear graph neural network with neighbor interactions. arXiv preprint arXiv:200203575.