electronics

Article

Embedding Enhancement Method for LightGCN in Recommendation Information Systems

Sangmin Lee, Junho Ahn * and Namgi Kim *

Department of Computer Science, Kyonggi University, Suwon 16227, Republic of Korea; d9249@kyonggi.ac.kr
* Correspondence: jha@kyonggi.ac.kr (J.A.); ngkim@kyonggi.ac.kr (N.K.); Tel.: +82-31-249-9662 (N.K.)

Abstract: In the modern digital age, users are exposed to a vast amount of content and information, and the importance of recommendation systems is increasing accordingly. Traditional recommendation systems mainly use matrix factorization and collaborative filtering methods, but problems with scalability due to an increase in the amount of data and slow learning and inference speeds occur due to an increase in the amount of computation. To overcome these problems, this study focused on optimizing LightGCN, the basic structure of the graph-convolution-network-based recommendation system. To improve this, techniques and structures were proposed. We propose an embedding enhancement method to strengthen the robustness of embedding and a non-combination structure to overcome LightGCN's weight sum structure through this method. To verify the proposed method, we have demonstrated its effectiveness through experiments using the SELFRec library on various datasets, such as Yelp2018, MovieLens-1M, FilmTrust, and Douban-book. Mainly, significant performance improvements were observed in key indicators, such as Precision, Recall, NDCG, and Hit Ratio in Yelp2018 and Douban-book datasets. These results suggest that the proposed methods effectively improved the recommendation performance and learning efficiency of the LightGCN model, and the improvement of LightGCN, which is most widely used as a backbone network, makes an important contribution to the entire field of GCN-based recommendation systems. Therefore, in this study, we improved the learning method of the existing LightGCN and changed the weight sum structure to surpass the existing accuracy.

Keywords: recommendation system; graph convolution network; LightGCN; learning methods

check for updates

Citation: Lee, S.; Ahn, J.; Kim, N. Embedding Enhancement Method for LightGCN in Recommendation Information Systems. *Electronics* **2024**, *13*, 2282. https://doi.org/10.3390/electronics13122282

Academic Editor: Domenico Ursino

Received: 30 April 2024
Revised: 31 May 2024
Accepted: 4 June 2024
Published: 11 June 2024

Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Today, with the rapid development of the digital age, a lot of content and information is created per second in the modern digital environment, and Internet users are exposed to such large amounts. This forces users to filter massive amounts of data to find the information or products that they want.

Recommendation systems act as a key tool to present the information or content desired by users in this information overload state, and are used in various areas such as e-commerce, entertainment, and social media; moreover, they are used to provide the content desired by users on platforms such as YouTube, Netflix, and Amazon. In order to recommend user-specific content, the most relevant information or products are effectively presented by analyzing the user's behavioral patterns, such as the user's past behavior, preferences, and interests. This provides opportunities for new discoveries and increases user loyalty to the content platform. In addition, by continuously providing relevant and interesting content to users, it allows users to spend more time on the platform, thereby increasing its profitability.

However, the existing recommendation system mainly uses the matrix factorization (MF) approach and the collaborative filtering (CF) approach, but the amount of data used for recommendation increases. In identifying similarities, a scalability problem occurs, in which the amount of computer calculations increases enormously, and the need for a method to solve problems such as a slow learning speed and an increased inference time has emerged due to this problem. As a way to effectively solve this problem, a recommendation system using deep learning based on the CF approach is being proposed. Typically, the graph convolution network (GCN) captures the correlation between users and items better than the existing methods, and, through this, studies are being proposed to improve the learning speed, inference speed, and accuracy of the existing methods [1–18].

Therefore, in this study, we have conducted research to optimize and improve the accuracy of the LightGCN model, which is mainly used as a backbone network in the latest GCN-based recommendation system. We propose a weight forwarding technique to improve learning speed and accuracy and use this method to improve the existing LightGCN model. Since the LightGCN model has been widely used as a backbone network in recent research, it is possible to improve the latest model by improving that model. A study was conducted to improve the layer combination structure.

2. Related Work

The MF approach, which is mainly used in existing recommendation systems, is a method of decomposing a matrix representing interactions and potential factors between users and items into two low-dimensional matrices. Hidden features (latent factors) such as user preferences or item characteristics are reflected in this decomposed matrix. In the MF approach, the collaborative signal, which refers to the latent feature vector of users who consumed items similar to mine, is used as the learning data, and the learning is performed using an optimization algorithm, such as gradient descent. After completing the learning process, a specific item in the matrix, which is reconstructed by multiplying the two low-dimensional matrices, represents the rating or preference that the user is expected to have for that item. However, it is computationally very complex to perform matrix decomposition on large-scale datasets with this method, and this process requires a lot of time and resources. Therefore, it is difficult for the MF approach to reflect dynamic changes in the system as user preferences or item characteristics change over time. In addition, the CF approach, which is mainly used in existing recommendation systems, is a recommendation method that recommends items preferred by neighbors with similar tastes to those of the target user to the target user themself. This methodology views user–item interaction data as a matrix and makes recommendations by restoring the entire matrix, like the MF methodology. Through this process, the similarity between the users or between the items is calculated based on evaluations of the items performed by past users, and recommendations are provided between users and items, allowing 'neighbors' with similar tastes to interact with the target user. Items that the user has not interacted with can be recommended as high-ranking items [19–22].

However, since the same problems as those seen in the MF methodology still occur in the CF methodology, deep learning technology is being proposed as a way to improve, starting with the neural graph collaborative filtering (NGCF) [23] model and the graph convolution network (GCN). A variety of recommended systems are being studied.

The NGCF model is used in the GCN-based CF model by explicitly encoding collaborative signals using GCN to consider high-order connectivity through user–item interaction data in a graph converted to a bipartite graph (Figure 1a). Here, high-order connectivity refers to high-order connectivity that cannot be expressed as a bipartite graph, and high-order connectivity generally refers to a path that reaches u_1 from all nodes with a path length greater than one. These high-order connections contain rich semantics that convey collaborative signals. As shown in Figure 1b, for example, the path $u_1 \leftarrow i_2 \leftarrow u_2$ indicates the behavioral similarity between u_1 and u_2, because both users have interacted with i_2. Looking at the longer path $u_1 \leftarrow i_2 \leftarrow u_2 \leftarrow i_4$, we can see that u_1 is likely to adopt i_4, because u_2, a similar user, has used i_4 before. Since i_4 has two connected paths and i_5 has one connected path, from the holistic perspective of $<i_4,u_1> <i_5,u_1> - 1 = 3$, i_4 is more likely to be of interest to u_1 than i_5. These high-order connections can be used to provide more accurate recommendations by capturing collaborative signals between the users and items.

A collaborative signal reflecting high-order connections is a concept proposed in the NGCF model, which includes the target users and neighboring users whose interaction history overlaps with the item's embedding vector, or the user who interacted with the neighbors but did not interact with the target. This is the embedding vector of the item. In addition, while the MF methodology and CF methodology implicitly reflected this collaborative signal, the NGCF explicitly reflected this, and, through this method, showed improvement in recommendation performance compared to the existing MF and CF methodologies.

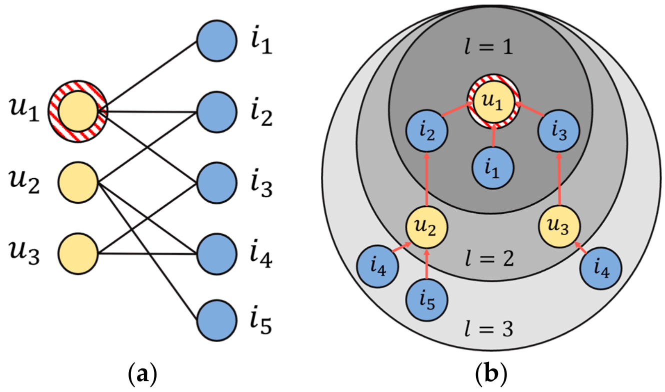


Figure 1. (**a**) Bipartite graph representing user–item interactions. (**b**) Higher-order connectivity based on u_1.

Equations (1) and (2) are the propagation rule and prediction layer of the NGCF model, respectively, and learning is performed using only the initial layer (shallow embedding) of Equation (2), $e_u^{(0)}$ and $e_i^{(0)}$. Here, hop refers to the distance between the nodes in the graph, and the layer of the GCN algorithm serves to aggregate the information of the surrounding nodes. Therefore, when proceeding with the propagation rule, when passing through the first layer, the information of the target node's 1-hop neighbors is aggregated; moreover, in the case of layer 2, the information of the 1-hop's neighbors, that is, up to the 2-hop, is aggregated. If it goes through N-layers, the information up to N-hops is aggregated. The author of NGCF first named this aggregated information a collaboration signal and proposed a process of disseminating and learning the collaboration signal containing information of such high-order connectivity to the GCN, leading to the use of deep learning in the recommendation system. $\mathcal{N}_u$ and $\mathcal{N}_i$ in Equation (1) refer to the connected nodes of the user u and the item i, respectively, and the embedding value when performing the convolution operation according to the symmetric normalization term (=$1/\sqrt{|\mathcal{N}_u|}\sqrt{|\mathcal{N}_i|}$). It serves to normalize in order to prevent it from becoming large, which means dividing by the number of neighboring nodes (users, items). The reason for normalization is that the more items connected to user u, the larger the collaborative signal becomes. The corresponding term ($\frac{1}{\sqrt{|\mathcal{N}_u|}\sqrt{|\mathcal{N}_i|}}\left(W_1 e_i^{(k)} + W_2\left(e_i^{(k)} \odot e_u^{(k)}\right)\right)$) shows that it is a process of performing message construction, and, as more messages are delivered in the process, the expression ability and recommendation performance increase.

$e_u^{(k)}$ is the embedding value in the kth layer of user u, and $e_i^{(k)}$ is the embedding value in the kth layer of item i. σ means non-linear activation, and $W_1 e_u^{(k)}$ and $W_1 e_i^{(k)}$ mean self-connection, where W_1 and W_2 are learnable weight matrices. To obtain the embedding value of the $k+1$th layer containing the kth hop in the graph, the previous embedding values of the neighbors of user u and item i are added to the normalized weight.

$$\begin{aligned} e_u^{(k+1)} &= \sigma\left(W_1 e_u^{(k)} + \sum_{i \in \mathcal{N}_u} \frac{1}{\sqrt{|\mathcal{N}_u|}\sqrt{|\mathcal{N}_i|}}\left(W_1 e_i^{(k)} + W_2\left(e_i^{(k)} \odot e_u^{(k)}\right)\right)\right), \\ e_i^{(k+1)} &= \sigma\left(W_1 e_i^{(k)} + \sum_{i \in \mathcal{N}_i} \frac{1}{\sqrt{|\mathcal{N}_u|}\sqrt{|\mathcal{N}_i|}}\left(W_1 e_u^{(k)} + W_2\left(e_u^{(k)} \odot e_i^{(k)}\right)\right)\right) \end{aligned} \tag{1}$$

$$e_u^* = e_u^{(0)} \| \cdots \| e_u^{(L)}, e_i^* = e_i^{(0)} \| \cdots \| e_i^{(L)} \tag{2}$$

LightGCN [24], an improved NGCF model, simplifies complex weight learning by removing feature transformation, non-linear activation, and self-connection from the propagation layer (Equation (3)). The prediction layer is shown in Equation (4). In addition, by changing the existing NGCF method to a weighted sum, all layers are summed and the last embedding values of the user and item, $e\prime_u$ and $e\prime_i$, are calculated through their average ($\alpha_k = 1/k$). Here, the weighted sum process used to obtain the last embedding value includes the initial layer embedding value. Score $\hat{y}_{ui}$, which represents the preference between user u and item i, is calculated by inner producing e_u and e_i, as shown in Equation (5). The matrix obtained through the inner product is used for recommendation. The basic learning process is shown in Figure 2. LightGCN uses an interaction graph between users and items to strengthen interactions between the nodes. As a result, the recommendation performance is improved by generating the node embedding value using only the connection information between the user and the item. LightGCN generates node embeddings using the interaction graph between the nodes instead of the weight learning of the GCN algorithm; therefore, the learning and inference time can be significantly reduced compared to NGCF. Additionally, its architecture is simpler than that of NGCF, so it can be usefully used in various recommendation scenarios and large-scale recommendation systems. Due to these various advantages, the LightGCN model is the most widely used backbone network in follow-up research.

$$e'^{(k+1)}_u = \sum_{i \in \mathcal{N}_u} \frac{1}{\sqrt{|\mathcal{N}_u|}\sqrt{|\mathcal{N}_i|}} e'^{(k)}_i; \;\; e'^{(k+1)}_i = \sum_{i \in \mathcal{N}_i} \frac{1}{\sqrt{|\mathcal{N}_i|}\sqrt{|\mathcal{N}_u|}} e'^{(k)}_u \tag{3}$$

$$\mathrm{e}'_{\mathrm{u}} = \sum_{k=0}^{n} \alpha_k \mathrm{e}'^{(k)}_{\mathrm{u}}; \;\; e'_{\mathrm{i}} = \sum_{k=0}^{n} \alpha_k \mathrm{e}'^{(k)}_{\mathrm{i}} \tag{4}$$

$$\hat{y}_{ui} = e'^{T}_{u} e'_{\mathrm{i}} \tag{5}$$

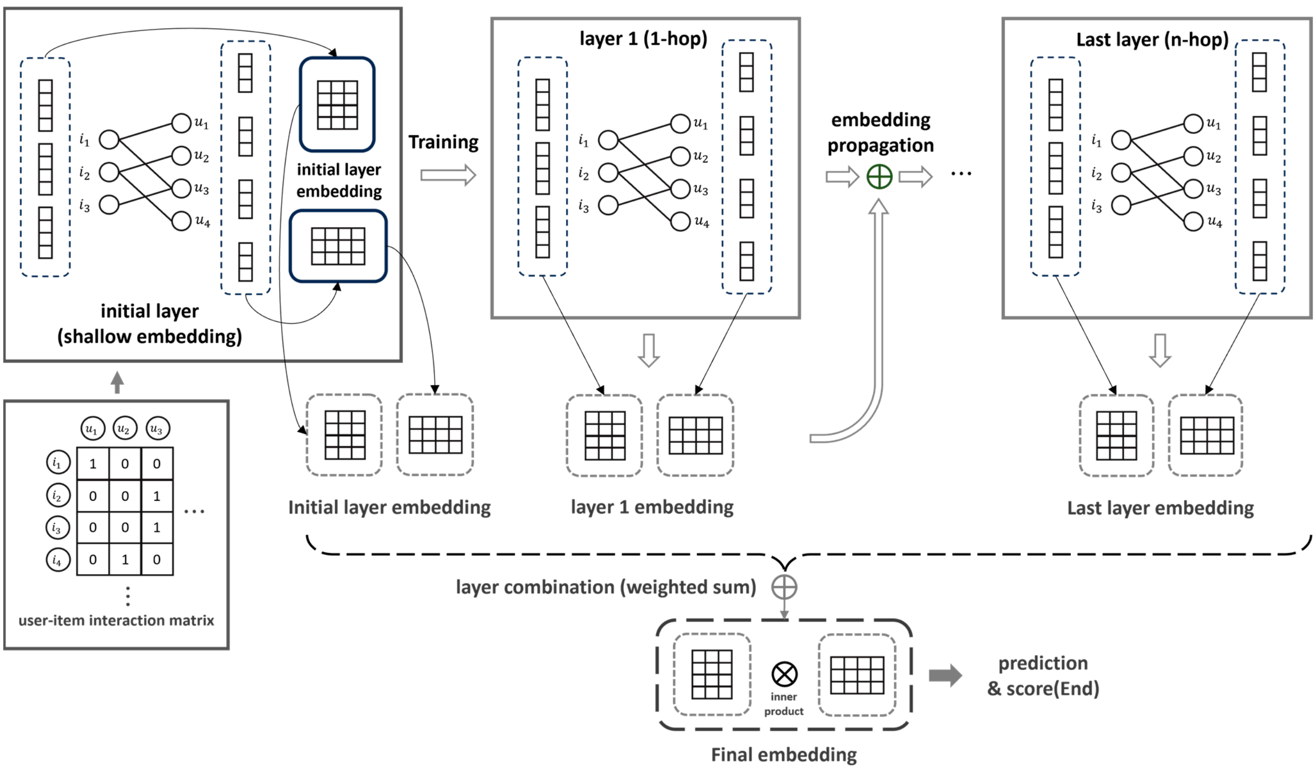


Figure 2. LightGCN's basic learning structure.

However, NGCF and LightGCN repeatedly perform a symmetric normalization term to obtain the embedding of the next layer and transmit the value to the next layer. They experience an underfitting problem, in which the embedding value decreases each time that it is transmitted to the next layer. These problems impede LightGCN's learning process and degrade its recommendation performance, preventing the original purpose of the recommendation system from being achieved. Additionally, the embedding value used for the recommendation of the LightGCN model is used after averaging the averages, as shown in Equation (4). Layer 1 learns only the 1-hop relationship and repeats it by increasing the length of the hop. The learning results of each hop are passed on to the next layer. The relationship data with a short previous hop length are the items that are most relevant to the user. The most related relationships are averaged and overlapped with the less related ones. The stronger the embedding value for this most relevant relationship, the lower the recommendation accuracy and recommendation diversity in the recommendation system. To overcome these problems, we improve the learning speed of the GCN algorithm and propose techniques and improvement structures to increase its recommendation performance.

3. Proposal Method (Embedding Enhancement Method)

LightGCN, whose embedding values continued to decrease during the learning process, robustly learns only the item features that the user has continuously viewed, allowing the users to discover new and relevant content that they would not have encountered otherwise, which is the essential reason for using a recommender system. With this, the probability decreases further. As a way to overcome this issue, in the process of passing the embedding of the previous layer to the next layer (as shown in Equation (6)), a scalar multiplication is performed on the embedding value that has performed a symmetric normalization term, resulting in enhancement embedding $e''^{(k+1)}_u, e''^{(k+1)}_i$ being passed to the next layer. Through this process, you will obtain enhancement embedding. Through the obtained embedding value, the user not only learns the characteristics of the item that he or she has continuously viewed, but also learns the relationship between other users and items, so that new items that have a relationship between the user and the item, but were not previously recommended, can be recommended. From this, the probability increases. Additionally, in the existing LightGCN structure, the self-connected effect of graph convolution is maintained by weight summing the embedding value derived from the previous hop layer, leading to the extraction of comprehensive representation. The embedding calculated by taking the weight sum overlaps the embedding value in the order of short hop length, due to the summation process, which has the effect of being further emphasized. However, the embedding value obtained through the proposed embedding enhancement method robustly contains the information of all hops, and, instead of using the enhancement embedding to perform a weight sum process (as shown in Equation (7)), it is recommended to use only the last layer embedding value, as shown in Equation (8). This makes it possible to provide more diverse recommendations than the existing LightGCN. The proposed non-combination structure specifically addresses LightGCN's weight sum issue by avoiding the averaging of embedding values from each layer and instead utilizing only the embedding value from the final layer. The reason for adopting this method is that the interference from the initial layer embeddings can actually degrade the recommendation performance, and increasing the layer depth beyond a certain point can also lead to performance degradation. Therefore, it is crucial to identify the optimal layer for each dataset or requirement and use only the results from this optimal layer for inference, which enhances the performance of the recommendation system. This approach minimizes the interference from the initial layers, allowing the model to learn deeper and more meaningful representations. By leveraging the embedding from the final layer, the model captures the most relevant and significant features for making recommendations. Furthermore, using the final layer embedding value helps us to provide more diverse recommendations, as it robustly includes the information from all hops without the overlapping effect seen in the weight-summed embeddings. This method increases the likelihood of recommending new items that have a relationship with the user but were not previously recommended, thus improving the overall recommendation quality. By implementing this non-combination structure, the recommender system not only learns the characteristics of the items that the user has continuously viewed, but also understands the relationships between the other users and items, facilitating the recommendation of previously unseen but relevant content. This ensures that the proposed method outperforms the existing LightGCN in providing diverse and accurate recommendations. A schematic diagram of the proposed method is shown in Figure 3.

$$e''^{(k+1)}_{u} = w_k \sum_{i \in \mathcal{N}_u} \frac{1}{\sqrt{|\mathcal{N}_u|}\sqrt{|\mathcal{N}_i|}} e''^{(k)}_{i};\ \ e''^{(k+1)}_{i} = w_k \sum_{i \in \mathcal{N}_i} \frac{1}{\sqrt{|\mathcal{N}_i|}\sqrt{|\mathcal{N}_u|}} e''^{(k)}_{u} \tag{6}$$

$$\mathrm{e}''_{\mathrm{u}} = \mathrm{e}''^{(k)}_{\mathrm{u}};\ \ e''_{\mathrm{i}} = \mathrm{e}''^{(k)}_{\mathrm{i}} \tag{7}$$

$$\hat{y}'_{ui} = e''^{T}_{u} e''_{\mathrm{i}} \tag{8}$$

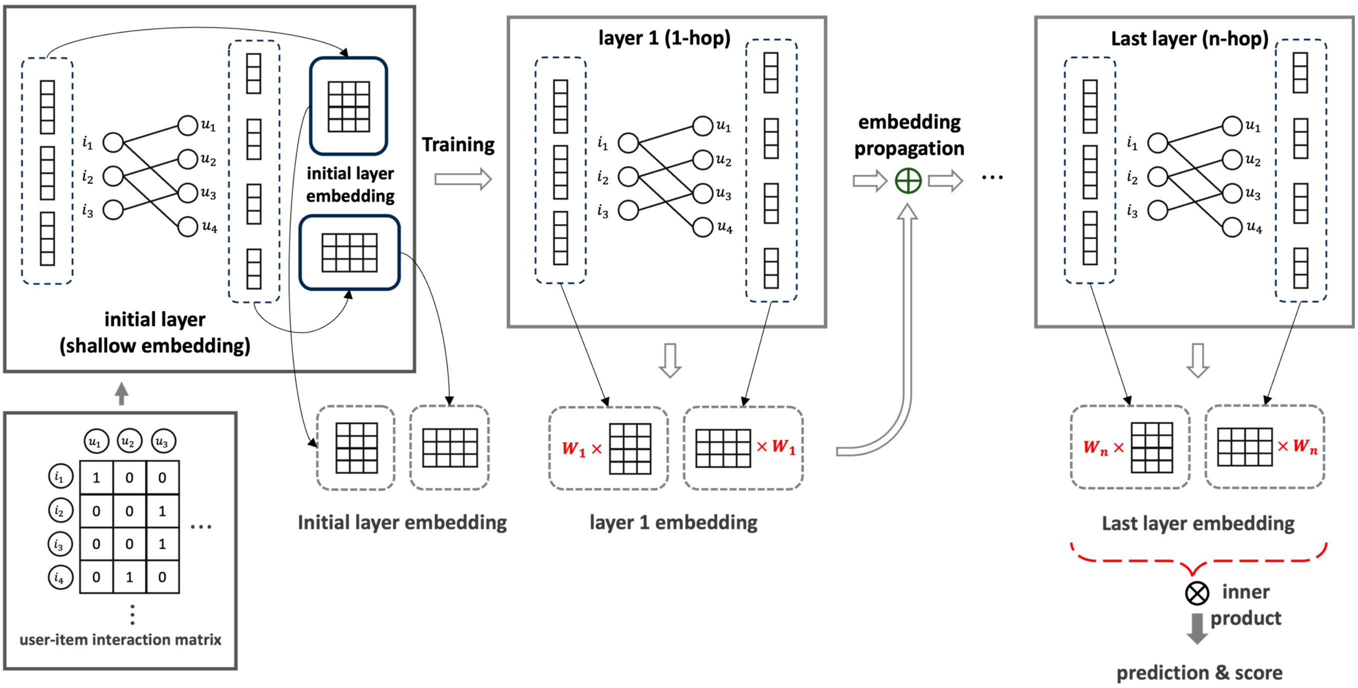


Figure 3. Non-combination structure of LightGCN that does not perform weight sum using the embedding enhancement method.

4. Experiment

In this study, a non-combination structure was proposed through the embedding enhancement method as a method to improve the problems that hinder the recommendation performance of LightGCN, as well as learning efficiency. The weight of the embedding enhancement of the proposed method is 2 of 0. To verify the proposed technique and structure, an experiment was conducted by applying the proposed method to LightGCN by increasing the weight from the power of 2 to the 11th power of 2. Since LightGCN is widely used as a backbone network in follow-up research, we believe that, if the performance and learning efficiency of LightGCN increases, the network proposed in follow-up research will also be able to produce meaningful results. SELFRec [25] was used in this experiment, which is an open-source library for GCN-based recommendation system algorithms. This library implements various recommendation algorithms based on PyTorch and facilitates recommendation system research and development. Additionally, several subsequent studies have used the same library as this experiment. Section 4.1

shows the experimental environment and data, and Section 4.2 shows the analysis results of the overall experiment. The link to the code written referring to SELFRec is as follows: Github, https://github.com/d9249/OptGCN (accessed on 22 February 2024). The learning records are as follows and have been posted on the Internet: Train log: https://wandb.ai/d9249/OptGCN (accessed on 22 February 2024).

4.1. Experimental Data and Environment

In experiments conducted to verify the proposed method, four types of datasets were used, as shown in Table 1. The MovieLens dataset is a dataset that is widely used in movie recommendation systems and contains interaction information about users' ratings of movies. Here, the interaction data refers to the viewing history between the user and the item, and this interaction relationship is expressed as a bipartite graph; furthermore, the interaction is learned in order to predict the mapping between the user and the item. The size of the dataset is divided based on the number of user viewing records. Likewise, the FilmTrust dataset is a dataset created by crawling users' movie evaluation information on the FilmTrust website, and, like the MovieLens dataset, it is often used in movie recommendation systems. In this experiment, only the rating information of the MovieLens dataset was used for learning between 4 and 5 points, but the FilmTrust dataset used rating information from 0.5 to 4 points for learning. The Douban-book dataset is a dataset created by crawling from Douban, a Chinese book review website, and includes book ratings and user information. The rating range for the dataset used in this experiment was four to five points. The Yelp2018 dataset comes from the Yelp Challenge 2018 and records ratings for local businesses, such as restaurants and bars, in 10 metropolitan areas across two countries, showing only those for which users have interacted with items.

Table 1. Datasets used in the experiment.

	Yelp2018		**MovieLens-1M** [26]	
	Train	**Test**	**Train**	**Test**
User	1499	812	12,638	10,882
Item	2033	340	22,222	19,075
Interaction	33,750	1747	478,730	119,690
Rating Range	0.5~4	0.5~4	4~5	4~5
Sparsity	98.89%	99.37%	99.83%	99.94%
	FilmTrust [27]		**Douban-book** [28]	
	Train	**Test**	**Train**	**Test**
User	6038	5989	31,668	31,668
Item	3492	3190	38,048	36,073
Interaction	460,359	114,922	1,237,259	324,147
Rating Range	4~5	4~5	None	None
Sparsity	97.82%	99.40%	99.90%	99.97%

The datasets used in the experiments of this study were the same as those used in the LightGCN follow-up study, and no preprocessing was performed on the data. These datasets are widely used as benchmark datasets for evaluating the performance of recommender system algorithms. To avoid overfitting problems, we implemented several strategies. First, we limited the training to 300 epochs and monitored the performance metrics during the training process. In particular, the epochs showing the highest performance metrics were selected to evaluate and compare the results. This approach prevents overfitting by preventing the model from continuing to learn beyond its optimal performance point. Second, we implemented early stopping based on the performance of the validation set. If the validation loss did not improve over a period of epochs, we stopped training to avoid overfitting the model to the training data. Finally, all accuracy analyses were performed using a separate validation dataset that the model did not see during training. This ensured that the reported performance metrics reflected the model's ability to generalize to unknown data. These strategies ensured the robustness of the results and prevented overfitting problems. By using early stopping and monitoring the performance of a separate validation dataset, we can provide strong support that our model maintains its ability to generalize across different datasets.

Information about the experiment environment of this study is shown in Table 2, and the parameters used for learning LightGCN are also shown in the table.

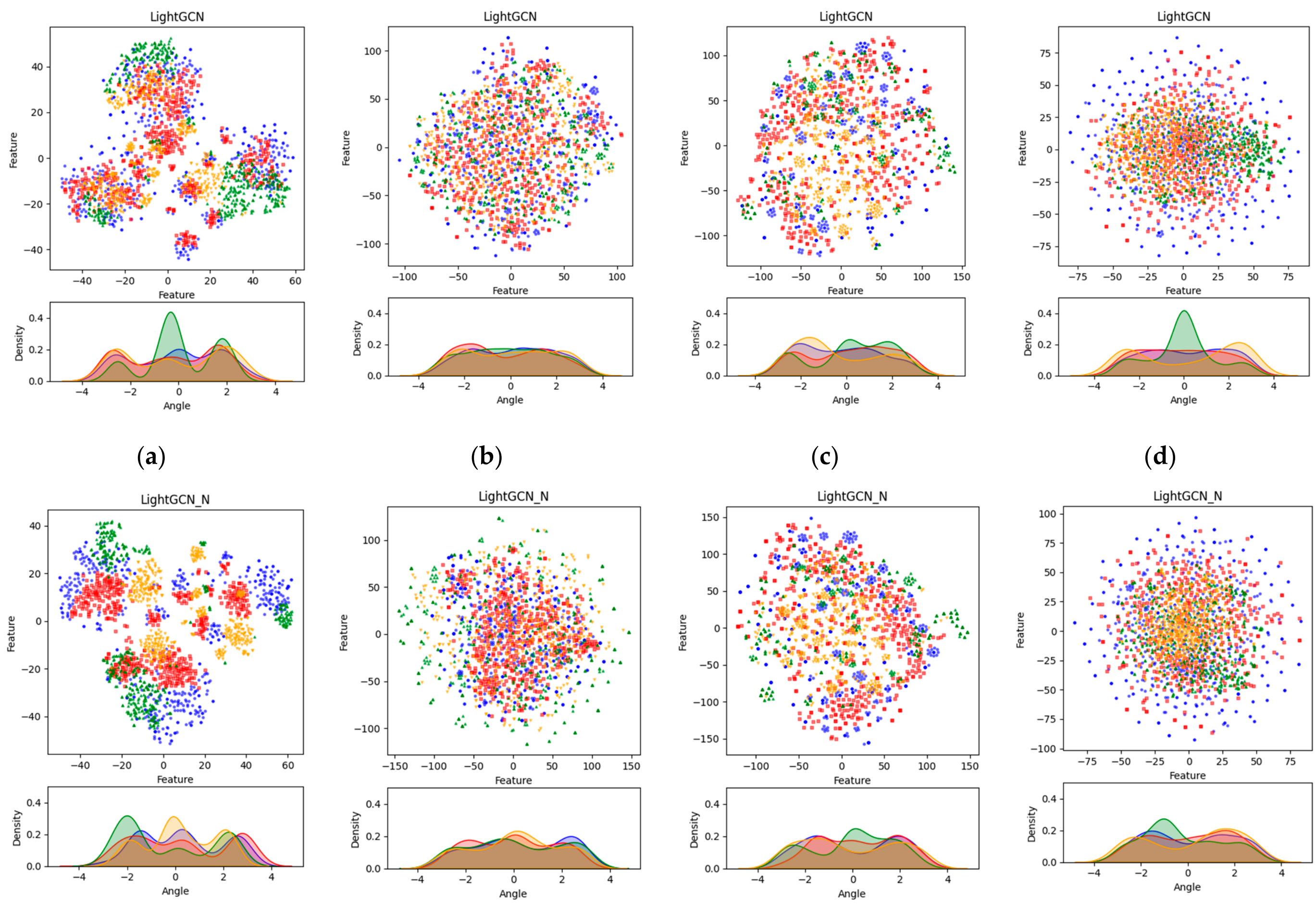
Table 2. Experimental environment and parameters.

Resource	Property
OS	Ubuntu 18.04.06 LTS
Nvidia Driver	525.78.01
CPU	Intel i9-12900K
GPU	GeForce RTX 3090
RAM	Samsung DDR4 16 GB (4EA)
CUDA	12.0
CuDNN	11.6
Python	3.7.11
Pytorch	1.10.0
Optimizer	Adam
Learning Epoch	300
Batch size	2048
Embedding size	64
Learning Rate	0.001
Reg lambda	0.0001
Layer	3

4.2. Experimental Results and Analysis

Figure 4 shows the analysis results achieved through dimensionality reduction techniques (DRT) using embedding values and datasets. In this analysis, principle component analysis (PCA) and Gaussian kernel density estimation (GKDE) were performed for each dataset to analyze the distribution and density of the embeddings. Additionally, for visualization purposes, we performed a data sampling process to rank the users and items by popularity, and then randomly selected 500 hot users (blue) and hot items (green) from the top 5% of users and item groups. Here, 500 cold users (red) and cold items (orange) were randomly sampled from the bottom 80% of the user and item groups. The expression learned through the sampling process was mapped to a two-dimensional space using t-SNE, and PCA visualization was performed through the expression mapped to the two-dimensional space. Additionally, the GKDE visualization results are shown below, and each point (x,y) was visualized using GKDE of atan(x,y).

The visualizations shown in Figure 4 allow us to evaluate the distribution and density of the embeddings, providing insights into factors such as separability, density, uniformity, clustering, noise, outliers, and stability. The visualizations represent the best-performing proposed method for each dataset. For the existing LightGCN, multiple clusters appeared in the PCA visualization, but the boundaries between the clusters were relatively less distinct compared to those of the proposed method. This suggests that the embeddings are not clearly separated and are intermingled. A similar pattern was observed in the GKDE visualization, with peaks indicating that specific feature values are concentrated in certain areas. In contrast, the embeddings resulting from the proposed algorithm showed relatively distinct clustering and wider dispersion in the PCA visualization. This indicates that the proposed method captures the diversity and features of the data more effectively. The GKDE results also showed relatively smoother curves, with the peaks being more subdued, indicating that the embeddings are more uniformly distributed. The use of t-SNE and PCA for dimensionality reduction and visualization is appropriate for several reasons. The t-SNE is particularly effective at preserving local structures in high-dimensional data, making it well-suited for visualizing the clusters and relationships within the data. By mapping the high-dimensional embeddings to a 2D space, t-SNE helps us to reveal natural groupings and separations that might not be apparent in the original high-dimensional space. PCA, on the other hand, is a linear dimensionality reduction technique that captures the global variance structure of the data. By projecting the embeddings onto the principal components, PCA helps us to understand the overall distribution and spread of the data. It provides insights into the major directions of variance and helps us to identify the primary factors contributing to the data structure. Combining t-SNE for initial dimensionality reduction with PCA for subsequent visualization leverages the strengths of both methods. t-SNE ensures that the local relationships are maintained, making the clusters more apparent, while PCA provides a broader view of the data's variance and distribution. This combination allows for a more comprehensive analysis of the embeddings, making it easier to evaluate their quality in terms of clustering, dispersion, and uniformity. These visualization results suggest that the proposed method learns the distribution of the embeddings more effectively than the existing method, better reflecting the diverse features of the data. Consequently, the proposed method enhances the quality of the embeddings by achieving clearer clustering, wider dispersion, and more uniform distribution, thus better representing the complexity of the data. The embedding values changed through the proposed method are shown in Table 3.



(a) (b) (c) (d) (e) (f) (g) (h)

Figure 4. Visualization of each dataset and the embedding results of the base model and proposed method: (**a**) Yelp2018 dataset; (**b**) MovieLens-1M dataset; (**c**) FilmTrust dataset; (**d**) Douban-book dataset; (**e**) LightGCN_N at Yelp2018 dataset; (**f**) LightGCN_N at MovieLesn-1M dataset; (**g**) LightGCN_N at FilmTrust dataset; (**h**) LightGCN_N at Douban-book dataset. (**e**–**h**) show visualizations based on the best performance of the applied methods for each dataset.

Table 3. Statistical analysis values of the existing LightGCN and the proposed method.

		FilmTrust		**Yelp2018**	
		Existing Average Value	**Proposed Method Average Value**	**Existing Average Value**	**Proposed Method Average Value**
User Embedding	Mean	-5.57×10^{-3}	-3.11×10^{0}	-1.67×10^{-2}	-5.19×10^{-2}
	Standard Deviation	4.94×10^{-1}	2.60×10^{1}	6.78×10^{-1}	4.78×10^{-1}
	Standard error	3.47×10^{-4}	4.86×10^{-2}	7.54×10^{-4}	5.32×10^{-4}
	Variance	2.44×10^{-1}	1.27×10^{2}	4.59×10^{-1}	2.29×10^{-1}
	Midian	-8.12×10^{-3}	-3.86×10^{0}	-1.90×10^{-2}	-2.62×10^{-2}
Item Embedding	Mean	-2.73×10^{-4}	-9.37×10^{-1}	-1.35×10^{-2}	-3.83×10^{-2}
	Standard Deviation	4.17×10^{-1}	1.20×10^{1}	5.50×10^{-1}	3.82×10^{-1}
	Standard error	2.67×10^{-4}	3.31×10^{-2}	4.62×10^{-4}	3.20×10^{-4}
	Variance	1.74×10^{-1}	1.33×10^{2}	3.03×10^{-1}	1.46×10^{-1}
	Midian	9.80×10^{-4}	-5.17×10^{-1}	-1.29×10^{-2}	-2.52×10^{-2}
		MovieLens-1M		**Douban-book**	
		Existing average value	Proposed method average value	Existing average value	Proposed method average value
User Embedding	Mean	6.55×10^{-3}	-6.70×10^{-3}	4.93×10^{-3}	-6.27×10^{-3}
	Standard Deviation	3.23×10^{-1}	3.49×10^{-1}	3.72×10^{-1}	3.06×10^{-1}
	Standard error	3.59×10^{-4}	3.88×10^{-4}	5.98×10^{-4}	4.93×10^{-4}
	Variance	1.04×10^{-1}	1.22×10^{-1}	1.38×10^{-1}	9.39×10^{-2}
	Midian	6.83×10^{-3}	-6.99×10^{-3}	6.57×10^{-3}	-1.03×10^{-2}
Item Embedding	Mean	-7.72×10^{-5}	-1.96×10^{-3}	3.82×10^{-3}	-2.51×10^{-2}
	Standard Deviation	2.40×10^{-1}	2.64×10^{-1}	4.78×10^{-1}	4.60×10^{-1}
	Standard error	2.02×10^{-4}	2.21×10^{-4}	1.01×10^{-3}	9.74×10^{-4}
	Variance	5.77×10^{-2}	6.96×10^{-2}	2.28×10^{-1}	2.12×10^{-1}
	Midian	-2.03×10^{-3}	-1.96×10^{-3}	4.15×10^{-3}	-1.95×10^{-2}

Four evaluation functions were used to verify the entire experiment, as follows: Precision, Recall, NDCG, and Hit Ratio, which were used to evaluate the quality of the model from various aspects. The Precision@K represents the ratio of items of interest to actual users among the recommendations proposed by the recommendation system. In other words, it measures how accurately the system actually recommends the preferred items to the user, and high precision indicates that the recommendation system accurately recommends the items that are highly relevant to the user. The Recall@K represents the ratio of items suggested by the recommendation system among the items that the actual user is interested in. In other words, it measures how many related items the recommender system recommends, and a high recall rate means that it suggests many related items to the user without missing them. Normalized discounted cumulative gain (NDCG)@K is a function that evaluates the relevance and ranking of the recommended items. The higher the user's preferred items are, the higher the value is. Therefore, a high NDCG indicates that the recommendation system accurately suggests items of interest to the user with a high ranking. The Hit Ratio@K represents the rate at which the recommendation system successfully includes the user's preferred items in the recommendation list. This measures whether the recommendation system recommends at least one item of interest to the user, and a high hit rate means that the recommendation system includes the user's preferred items well, without missing them.

Through these various evaluation indicators, the performance of the recommendation system was evaluated and compared from various aspects. Precision and Recall evaluate the accuracy and diversity aspects of the recommendations, NDCG evaluates whether the items of interest to the user are accurately recommended at the top, and Hit Ratio measures how successful the recommendation system is in suggesting at least one related item to the user. The entire experiment was conducted with K = 20, and the quantitative results of the entire experiment are shown in Table 4.

Table 4. Quantitative results of the entire experiment with both the traditional model and the proposed method per dataset.

Dataset	Weight	Best Epoch	Model	Precision	Recall	NDCG	Hit Ratio
Yelp2018	-	101	MF	0.02235	0.04998	0.04055	0.04367
	-	231	LightGCN	0.02688	0.05949	0.04887	0.05253
	8, 8, 8	56	LightGCN_W	0.02809	0.06191	0.05086	0.05488
	8, 8, 8	271	**LightGCN_N (Propose method)**	**0.03093**	**0.06873**	**0.05656**	**0.06044**
MovieLens-1M	-	46	MF	0.189	0.249	0.2831	0.197
	-	261	LightGCN	0.19987	0.27196	0.30274	0.20838
	64, 64, 64	11	LightGCN_W	0.19937	0.27716	0.30852	0.20786
	128,128,128	271	**LightGCN_N**	**0.20214**	**0.27748**	**0.3083**	**0.21074**
FilmTrust	-	296	MF	0.0868	0.848	0.6169	0.8178
	-	276	LightGCN	0.08761	0.85376	0.61566	0.82547
	8, 8, 8	76	LightGCN_W	0.08817	0.86131	0.6197	0.83481
	32, 32, 32	226	**LightGCN_N**	**0.08824**	**0.85874**	**0.62453**	**0.83137**
Douban-book	-	66	MF	0.05407	0.1247	0.1039	0.09662
	-	266	LightGCN	0.06209	0.14783	0.12508	0.11095
	16, 16, 16	26	LightGCN_W	0.06315	0.15127	0.12867	0.11284
	16, 16, 16	181	**LightGCN_N**	**0.07103**	**0.16701**	**0.14632**	**0.12692**

The analysis of the experimental results for each dataset was performed using the results shown in Table 4. On the Yelp2018 dataset, the proposed method achieved the best performance, showing remarkable improvement in all evaluation indicators. This dataset achieved an increase of 15.07% in Precision, 15.53% in Recall, 15.74% in NDCG, and 15.06% in Hit ratio. These numbers indicate a significant improvement in the model's ability to accurately recommend relevant items and encompass a wider range of user interests, and consistent improvements in all of these metrics suggest that LightGCN_N was particularly effective for the Yelp2018 dataset.

In the case of the MovieLens-1M dataset, the performance improvement was relatively small compared to that of Yelp2018. An improvement of approximately 2% was achieved in both the Recall and NDCG metrics, and this increase indicates a slight improvement in the model's ability to capture relevant items for user recommendations. Precision and Hit Ratio also showed a slight improvement, with an increase of about 1%. This confirmed that LightGCN_N improved in recommendation quality, but the impact was not as clear as that observed for the Yelp2018 dataset. These results occurred due to inherent differences between the datasets.

The FilmTrust dataset showed an enhanced performance across all metrics, each showing an increase of approximately 1%. This slight improvement observed in the results after applying the proposed method indicates limited improvement compared to the basic LightGCN model. The relatively low increase compared to the other datasets suggests that the proposed technique may have a limited impact on datasets with similar characteristics to those of FilmTrust.

The Douban-book dataset showed the most notable improvement among all datasets, especially for NDCG, with an increase of 16.98%. Additionally, both Precision and Hit Ratio increased by about 14.4%, and Recall improved by 12.97%. This suggests that LightGCN_N is very effective on the Douban-book dataset. This increase in all metrics indicates that the proposed method not only improved the overall recommendation accuracy, but also better identified a wider range of relevant items for users. It can be seen that the proposed method is suitable for the characteristics of the Douban-book dataset, including item types and user interaction patterns.

5. Conclusions

In the digital age, users are overwhelmed by the vast amount of content and information available online, and, as a solution to this, recommendation systems play an important role in helping users to find relevant content. Traditional recommendation systems utilizing matrix factorization and collaborative filtering approaches have scalability issues, due to increasing data volumes. This causes scalability problems and a decrease in the learning and inference speed of the existing methods. Therefore, this study aimed to optimize and improve the accuracy of the LightGCN model, which is a widely used backbone network in GCN-based recommendation systems. For this purpose, the following method was proposed: As a way to improve the robustness of the embeddings, the robustness between the layers was maintained by increasing the embeddings through weights after symmetric normalization. Using the enhancement embedding obtained through this process, learning was completed without going through the existing weighted sum, and only the last layer's embedding was used for recommendation; however, it surpassed the accuracy of the existing recommendation performance. What can be seen from these experiments is that the proposed method does not recommend only the items with a small number of hops—although there was a user–item relationship in the past—but is capable of recommending new and diverse items that had longer hops and were not otherwise recommended. The entire experiment was conducted on the MovieLens, FilmTrust, Douban-book, and Yelp2018 datasets using the SELFRec library, and Precision, Recall, NDCG, and Hit Ratio were evaluated as performance indicators for model verification. The analysis of the entire experiment showed notable improvements in all datasets, especially the Yelp2018 and Douban-book datasets. There were notable increases in these four indicators.

The detailed experimental results for each dataset are as follows: In most datasets, the proposed LightGCN_N showed improvements in all evaluation indicators. In the Yelp2018 dataset, Precision increased by 15.07%, Recall increased by 15.53%, NDCG increased by 15.74%, and Hit Ratio increased by 15.06%. This means that the proposed LightGCN_N can cover the user's interests more broadly and recommend related items more accurately than the existing LightGCN model. In the MovieLens-1M dataset, Precision increased by 1.14%, Recall increased by 2.03%, NDCG increased by 1.84%, and Hit Ratio increased by 1.13%. This suggests that, although the LightGCN_N model improves in recommendation quality, the effect is relatively less pronounced compared to that of the Yelp2018 dataset. In the FilmTrust dataset, Precision increased by 0.72%, Recall increased by 0.58%, NDCG increased by 1.44%, and Hit Ratio showed a performance improvement of 0.71%. The impact of the proposed technique was found on the datasets with similar characteristics to those of FilmTrust. This means that it may be limited. The Douban-book dataset showed the most notable performance improvement, especially for NDCG, which increased by 16.98%. Additionally, Precision and Hit Ratio improved by 14.4%, and Recall increased by 12.97%. This indicates that the proposed method improved the recommendation accuracy overall and provided the users with a wider range of related items.

The comprehensive results show that the proposed methods effectively improved the recommendation performance and learning efficiency of LightGCN, especially showing notable improvements in the Yelp2018 and Douban-book datasets. This can be seen as an important contribution that significantly improves the performance of the LightGCN model in the field of recommender systems. In the future, we plan to verify the proposed methods by applying LightGCN (which was proposed in a follow-up study) to algorithms [29–33] that use the backbone network, and to learn to overcome the problem of having to manually search for weights to obtain enhancement embedding. In the process, we plan to perform a learning process in which the optimal weight is calculated. These methods will be applied to the latest recommendation model.

Author Contributions: Conceptualization, S.L., J.A. and N.K.; methodology, N.K.; software, S.L.; validation, S.L. and N.K.; formal analysis, N.K.; investigation, S.L.; resources, N.K.; data curation, S.L.; writing—original draft preparation, S.L.; writing—review and editing, S.L.; visualization, S.L.; supervision, N.K.; project administration, N.K.; funding acquisition, N.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research was conducted with the support of the Korea Research Foundation and was funded by the Ministry of Education of Korea in 2020 (No. 2020R1A6A1A03040583) and the MSIT, Korea, under the National Program for Excellence in SW, supervised by the IITP (2021-0-01393).

Data Availability Statement: The data that support the findings of this study are available from the corresponding author upon reasonable.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Goldberg, D.; Nichols, D.; Oki, B.M.; Terry, D. Using Collaborative Filtering to Weave an Information Tapestry. *Commun. ACM* **1992**, *35*, 61–70. [CrossRef]
2. Resnick, P.; Iacovou, N.; Suchak, M.; Bergstrom, P.; Riedl, J. GroupLens: An open architecture for collaborative filtering of netnews. In Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work, Chapel Hill, NC, USA, 22–26 October 1994; pp. 175–186.
3. Herlocker, J.L.; Konstan, J.A.; Borchers, A.; Riedl, J. An Algorithmic Framework for Performing Collaborative Filtering. In Proceedings of the 22nd Annual iIternational ACM SIGIR Conference on Research and Development in Information Retrieval, Berkley, CA, USA, 15–19 August 1999; pp. 230–237.
4. Goyani, M.; Chaurasiya, N. A review of movie recommendation system: Limitations, Survey and Challenges. *ELCVIA Electron. Lett. Comput. Vis. Image Anal.* **2020**, *19*, 18–37.
5. Wang, H.; Le, Z.; Gong, X. Recommendation System Based on Heterogeneous Feature: A Survey. *IEEE Access* **2020**, *8*, 170779–170793. [CrossRef]
6. Ying, R.; He, R.; Chen, K.; Eksombatchai, P.; Hamilton, W.L.; Leskovec, J. Graph Convolutional Neural Networks for Web-Scale Recommender Systems. In Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, London, UK, 19–23 August 2018; pp. 974–983.
7. Sarwar, B.; Karypis, G.; Konstan, J.; Riedl, J. Item-Based Collaborative Filtering Recommendation Algorithms. In Proceedings of the WWW, Hong Kong, China, 1–5 May 2001; pp. 285–295.
8. Rajchakit, G.; Agarwal, P.; Ramalingam, S. *Stability Analysis of Neural Networks*; Springer: Singapore, 2021.
9. Roy, D.; Dutta, M. A systematic review and research perspective on recommender systems. *Big Data* **2022**, *9*, 59. [CrossRef]
10. Blanco-Fernandez, Y.; Pazos-Arias, J.J.; Gil-Solla, A.; Ramos-Cabrer, M.; Lopez-Nores, M. Providing entertainment by content-based filtering and semantic reasoning in intelligent recommender systems. *IEEE Trans. Consum. Electron.* **2008**, *54*, 727–735. [CrossRef]
11. Isinkaye, F.O.; Folajimi, Y.O.; Ojokoh, B.A. Recommendation systems: Principles, methods and evaluation. *Egypt. Inf. J.* **2015**, *16*, 261–273. [CrossRef]
12. Li, C.; Wang, Z.; Cao, S.; He, L. WLRRS: A new recommendation system based on weighted linear regression models. *Comput. Electr. Eng.* **2018**, *66*, 40–47. [CrossRef]
13. He, M.; Wang, B.; Du, X. HI2Rec: Exploring knowledge in heterogeneous information for movie recommendation. *IEEE Access* **2019**, *7*, 30276–30284. [CrossRef]
14. Afolabi, A.O.; Toivanen, P. Integration of recommendation systems into connected health for effective management of chronic diseases. *IEEE Access* **2019**, *7*, 49201–49211. [CrossRef]
15. Kang, S.; Jeong, C.; Chung, K. Tree-based real-time advertisement recommendation system in online broadcasting. *IEEE Access* **2020**, *8*, 192693–192702. [CrossRef]
16. Wei, J.; He, J.; Chen, K.; Zhou, Y.; Tang, Z. Collaborative filtering and deep learning based recommendation system for cold start items. *Expert Syst. Appl.* **2017**, *69*, 29–39. [CrossRef]
17. Li, H.; Cui, J.; Shen, B.; Ma, J. An intelligent movie recommendation system through group-level sentiment analysis in microblogs. *Neurocomputing* **2016**, *210*, 164–173. [CrossRef]
18. Chen, M.H.; Teng, C.H.; Chang, P.C. Applying artificial immune systems to collaborative filtering for movie recommendation. *Adv. Eng. Inform.* **2015**, *29*, 830–839. [CrossRef]
19. Koren, Y.; Bell, R.; Volinsky, C. Matrix Factorization Techniques for Recommender Systems. *Computer* **2009**, *42*, 30–37. [CrossRef]
20. Zhang, S.; Tong, H.; Xu, J.; Maciejewski, R. Graph convolutional networks: A comprehensive review. *Comput. Soc. Netw.* **2019**, *6*, 11. [CrossRef] [PubMed]
21. You, J.; Leskovec, J.; He, K.; Xie, S. Graph Structure of Neural Networks. In Proceedings of the 37th International Conference on Machine Learning (ICML), Vienna, Austria, 12–18 July 2020; pp. 10881–10891.

22. Wu, Z.; Pan, S.; Chen, F.; Long, G.; Zhang, C.; Yu, P.S. A Comprehensive Survey on Graph Neural Networks. *IEEE Trans. Neural Netw. Learn. Syst.* **2021**, *32*, 4–24. [CrossRef] [PubMed]
23. Wang, X.; He, X.; Wang, M.; Feng, F.; Chua, T.S. Neural Graph Collaborative Filtering. In Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, Paris, France, 21–25 July 2019; pp. 165–174.
24. He, X.; Deng, K.; Wang, X.; Li, Y.; Zhang, Y.; Wang, M. LightGCN: Simplifying and Powering Graph Convolution Network for Recommendation. In Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, Xi'an, China, 25–30 July 2020; pp. 639–648.
25. Yu, J.; Yin, H.; Xia, X.; Chen, T.; Li, J.; Huang, Z. Self-Supervised Learning for Recommender Systems: A Survey. *IEEE Trans. Knowl. Data Eng.* **2022**, *36*, 335–355. Available online: https://github.com/Coder-Yu/SELFRec (accessed on 7 February 2024). [CrossRef]
26. Harper, F.M.; Konstan, J.A. The MovieLens Datasets: History and Context. *ACM Trans. Interact. Intell. Syst.* **2015**, *5*, 19. [CrossRef]
27. Guo, G.; Zhang, J.; Thalmann, D.; Yorke-Smith, N. ETAF: An Extended Trust Antecedents Framework for Trust Prediction. In Proceedings of the 2014 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, Beijing, China, 17–20 August 2014; pp. 540–547.
28. Ma, H.; Zhou, D.; Liu, C.; Lyu, M.R.; King, I. Recommender systems with social regularization. In Proceedings of the Fourth ACM International Conference on Web Search and Data Mining, Hong Kong, China, 9–12 February 2011; pp. 287–296.
29. Huang, T.; Dong, Y.; Ding, M.; Yang, Z.; Feng, W.; Wang, X.; Tang, J. MixGCF: An Improved Training Method for Graph Neural Network-based Recommender Systems. In Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Singapore, 14–18 August 2021.
30. Wu, J.; Wang, X.; Feng, F.; He, X.; Chen, L.; Lian, J.; Xie, X. Self-supervised Graph Learning for Recommendation. In Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, Montréal, QC, Canada, 11–15 July 2021; pp. 726–735.
31. Yu, J.; Yin, H.; Xia, X.; Chen, T.; Cui, L.; Nguyen, Q.V.H. Are Graph Augmentations Necessary? Simple Graph Contrastive Learning for Recommendation. In Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, Madrid, Spain, 11–15 July 2022; pp. 1294–1303.
32. Yu, J.; Xia, X.; Chen, T.; Cui, L.; Hung, N.Q.V.; Yin, H. XSimGCL: Towards Extremely Simple Graph Contrastive Learning for Recommendation. *IEEE Trans. Knowl. Data Eng.* **2023**, *36*, 913–926. [CrossRef]
33. Lee, D.; Kang, S.; Ju, H.; Park, C.; Yu, H. Bootstrapping user and item Representations for One-Class Collaborative Filtering. In Proceedings of the SIGIR, Montréal, QC, Canada, 11–15 July 2021.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.