REHG: A Recommender Engine Based on Heterogeneous Graph

Xiaoyang Xin[1], Zesheng Cheng[1(✉)], Tiankuan Wang[2], Mengqiu Yan[1], Ruixuan Zhao[1], Chong Peng[1], and Jianbo Li[1]

[1] College of Computer Science and Technology, Qingdao University, Qingdao 266071, China
czs_110@hotmail.com

[2] Faculty of Computing Science, University of Alberta, Edmonton, Canada

Abstract. Recommendation systems are extensively utilized across diverse domains with the aim of suggesting the most appropriate items to individual users. An effective recommendation engine provides valuable guidance for daily life.

Therefore, improving the efficiency of recommender systems has become a prominent subject of interest in both industrial and academic circles. The majority of high-performing models in recommendation tasks utilize binary homogeneous graph convolutional methods and collaborative filtering as their foundational structures. These models are simple in structure, accurate and efficient in recommendation tasks. However, owing to the constraints of the binary homogeneous graph's structure, these models are unable to incorporate supplementary information during operation, thereby hindering the full utilization of diverse and efficient data. To fill the research gaps, this study introduces a novel recommendation system named REHG, which is grounded on directed heterogeneous graphs. Additionally, the study devises two experiments to assess its performance across various testing datasets. These experiments offer compelling evidence to support the superior performance of the proposed REHG model over common baseline models across different evaluation criteria. Furthermore, the study explores the feasibility of enhancing interoperability among various recommender system models.

Keywords: Data Engineering · Heterogeneous Graph · Recommender System

1 Introduction

Recommender system problem can be academically defined as the objective of recommending the most appropriate items to specific users by predicting their interest in those items based on their historical interactions. As information technology progresses, recommender systems using machine learning techniques are now widely used in many businesses [1]. Advanced approaches to solving recommender system problems often involve the integration of graph convolutional neural networks (GCN) with collaborative filtering (CF) methods [2–5]. CF methods operate on the premise that individuals with comparable experiences and interests tend to share similar preferences. By assessing the "similarity" between known items or users, these methods facilitate the effective

Z. Cai et al. (Eds.): WASA 2024, LNCS 14997, pp. 347–359, 2025.
https://doi.org/10.1007/978-3-031-71464-1_29

recommendation of information to other users [6]. The inclusion of GCN in this fusion approach leads to superior performance compared to other method categories across diverse datasets [7].

However, traditional GCN-based collaborative filtering methods primarily examine the graphical attributes of users and items, neglecting additional details like product origin or user age. This information may have a great impact on the final choice of the users. Moreover, incorporating this information into the model could enhance interpretability. In essence, the recommender system would not only offer recommendations but also provide insights into the reasoning behind each recommendation. This may be useful for subsequent business activities. Therefore, in today's diverse and personalized society, considering such extra information can enhance recommender system performance [8].

To address the above issue, this study introduces a novel recommender system algorithm named Recommender Engine based on Heterogeneous Graph (REHG). The REHG algorithm represents the relationship between users, information, and items in the dataset using a heterogeneous graph, allowing for the evaluation of additional information. Relevant experiments suggest that the proposed method is highly accurate. As far as the authors are aware, REHG demonstrates leading performance across multiple public datasets, substantially outperforming previous optimal results in enhancing key metrics. Additionally, REHG doesn't necessitate intricate deep learning procedures, making it computationally efficient with significantly lower computational demands compared to deep learning-based recommendation models. Moreover, by incorporating the evaluation of various meta-paths (please refer to Sect. 3 for more details) among heterogeneous graphs, the proposed method offers a degree of interpretability, potentially aiding subsequent business activities.

2 Related Work

This section discusses related research, including early recommendation engines and CF methods using GCN. It also identifies research gaps towards the end.

2.1 Early Recommendation Engines

Early recommendation engines can be broadly grouped into three categories. The first one is the methods based on matrix decomposition [9, 10]. These methods accomplish the task of filling the unknown portions of the User-Item (U-I) matrix by utilizing singular value decomposition (SVD) and inner product operations on the known data, thereby facilitating the recommendation process. The core difficulty is to solve the problem of high time complexity of SVD ($O(n^3)$) and the high sparsity of U-I matrix. The second category is the auto encoder-based approach [11]. The fundamental principle involves embedding users and items into vectors that adhere to specific criteria using autoencoder machines, which then engage in subsequent operations. The last category is CF methods which principle has been introduced in Sect. 1 above.

The term "early" here means "before GCN-based CF methods came into use." Over time, these three categories have evolved and served different purposes in various real-life situations. However, the widespread use of graph theory in computer science has made GCN-based CF methods perform significantly better than the others in various aspects.

2.2 Collaborative Filtering Methods Based on GCN

Neural Graph Collaborative Filtering (NGCF) is considered to be a representative method among initial GCN based CF methods. It leverages the U-I interaction graph to propagate embedding as:

$$\boldsymbol{e}_u^{(k+1)} = \sigma(W_1\boldsymbol{e}_u^k + \sum_{i\in\aleph_u}\frac{1}{\sqrt{|\aleph_u||\aleph_i|}}(\boldsymbol{W}_1\boldsymbol{e}_i^k + \boldsymbol{W}_2\left(\boldsymbol{e}_i^k \odot \boldsymbol{e}_u^k\right))) \tag{1}$$

$$\boldsymbol{e}_i^{(k+1)} = \sigma(W_1\boldsymbol{e}_i^k + \sum_{i\in\aleph_i}\frac{1}{\sqrt{|\aleph_u||\aleph_i|}}(\boldsymbol{W}_1\boldsymbol{e}_u^k + \boldsymbol{W}_2\left(\boldsymbol{e}_u^k \odot \boldsymbol{e}_i^k\right))) \tag{2}$$

NGCF follows the standard GCN principles, using nonlinear activation functions and trainable weighted matrices. However, LightGCN suggests that by removing these elements, the model's training speed improves and achieves better performance. As a result, the embeddings of users and items are represented as follows:

$$\boldsymbol{e}_u^{(k+1)} = \sum_{i\in\aleph_u}\frac{1}{\sqrt{|\aleph_u||\aleph_i|}}\boldsymbol{e}_i^k \tag{3}$$

$$\boldsymbol{e}_i^{(k+1)} = \sum_{i\in\aleph_i}\frac{1}{\sqrt{|\aleph_u||\aleph_i|}}\boldsymbol{e}_u^k \tag{4}$$

It's crucial to note that higher-dimensional embeddings for users and items generally provide more comprehensive information and can improve model accuracy. This applies to both NGCF and LightGCN. A new method called Graph Filter based Collaborative Filtering (GF-CF) explores what happens when LightGCN's embeddings expand to infinite dimensions. It suggests that in this case, LightGCN can be approximated as a linear filter. Then, GF-CF adds an ideal low-pass model on top of this. The overall structure of the model is represented as follows:

$$\boldsymbol{S} = \boldsymbol{R}\left(\tilde{\boldsymbol{R}}^{\boldsymbol{T}}\tilde{\boldsymbol{R}} + \alpha\boldsymbol{D}_{\boldsymbol{I}}^{-\frac{1}{2}}\tilde{\boldsymbol{U}}\tilde{\boldsymbol{U}}^{\boldsymbol{T}}\boldsymbol{D}_{\boldsymbol{I}}^{-\frac{1}{2}}\right) \tag{5}$$

The latest BSPM model brings the diffusion model from image processing to recommender systems. This model consists of two stochastic processes: forward perturbation and backward recovery. BSPM shows that the process of the GF-CF model can be seen as a forward perturbation process (Eq. 6), and BSPM adds a backward recovery process (Eq. 7).

$$\boldsymbol{B}(1) = R + \int_0^1 \boldsymbol{B}(\boldsymbol{t})\left(\tilde{\boldsymbol{R}}^{\boldsymbol{T}}\tilde{\boldsymbol{R}} + \alpha\boldsymbol{D}_{\boldsymbol{I}}^{-\frac{1}{2}}\tilde{\boldsymbol{U}}\tilde{\boldsymbol{U}}^{\boldsymbol{T}}\boldsymbol{D}_{\boldsymbol{I}}^{-\frac{1}{2}} - \boldsymbol{I}\right)dt \tag{6}$$

$$\boldsymbol{S}(\boldsymbol{T_s}) = \boldsymbol{B}(1) + \int_0^{T_s} -\boldsymbol{S}(\boldsymbol{t})\tilde{\boldsymbol{R}}^{\boldsymbol{T}}\tilde{\boldsymbol{R}}\,dt \tag{7}$$

According to the authors, BSPM stands out as potentially the most accurate recommender system algorithm available. It has demonstrated state-of-the-art performance on three significant public datasets. Since the model doesn't require training, most of the computational load comes from large-scale matrix multiplication operations. As a result, BSPM and GF-CF can complete recommendation tasks much faster compared to models that need training. Other methods in this category are detailed in references [7, 12].

2.3 Analysis and Research Gaps

Based on the description of the cutting-edge recommender system algorithms above, it's clear that most researchers in related fields are primarily focused on studying the interactions between users and items on the graph. This focus often overlooks the specific attributes and information about the users or items themselves. Common algorithms that discuss this problem include [13–17] which are far fewer both in terms of number and the depth of discussion. However, using such information to improve method accuracy and data utilization efficiency is common in large data analysis projects. The experimental section also shows that considering such information improves recommender system performance. Therefore, this study aims to fill the research gaps in recommendation algorithms by applying additional information. To achieve this, we've developed a model called REHG based on directed heterogeneous graph structures for recommender systems.

3 Methodology

Table 1 is a dataset containing two job candidates, three jobs and their related information. Using this dataset as a toy example, this section introduces the structure and operation process of the REHG method in more details (Table 2).

Table 1. Dataset of toy example

Candidates & Jobs	Job #1 (Beijing, Age 30–35)	Job #2 (Shanghai, Age 25–30)	Job #3 (Beijing, Age 25–35)
Candidate #1 (Beijing, Age 30)	1	1	0
Candidate #2 (Shanghai, Age 35)	0	1	1

3.1 Preliminaries

Graph in Recommender System. As it is mentioned above, graph is commonly used in current recommender systems to represent the interaction between users and items.

Table 2. Definitions of symbols used in this paper

Symbol	Definition
$\boldsymbol{\sigma}$	Nonlinear activation function, ReLu is used in the paper
k	k is the length of the to-be-selected list when calculating Recall and Ndcg
$\tilde{\boldsymbol{R}}$	Normalized interaction scores matrix (U-I graph adjacent matrix)
$\tilde{\mathbf{A}}$	Normalized form of matrix A
$\boldsymbol{P}$	Similarity Matrix. The item-item and user-user similarity matrix can be expressed as: $P_{\mathrm{II}} = {R_{\mathrm{UI}}}^{\mathrm{T}} R_{\mathrm{UI}.}$ And $P_{\mathrm{UU}} = {R_{\mathrm{IU}}}^{\mathrm{T}} R_{\mathrm{IU}}$
$\boldsymbol{U}$	Top-k singular vectors of corresponding adjacent matrix
$\boldsymbol{D}$	Degree matrix for normalization
$\boldsymbol{S} / \boldsymbol{R}$	Predicted / Observed U-I interaction matrix
$\tilde{\boldsymbol{U}} / \tilde{\boldsymbol{V}}$	Top-K singular vectors of $\tilde{R}$ / $\tilde{R}^T$
$\boldsymbol{D_I} / \boldsymbol{D_U}$	Degree matrix of items / users
$\boldsymbol{W_1} / \boldsymbol{W_2}$	Feature transformation weight matrix (trainable)
$\mathbf{e_u}^{\mathbf{k}} / \mathbf{e_i}^{\mathbf{k}}$	Refined embedding of user *u* and item *i* after *k* layers propagation
$\aleph_{\mathbf{u}} / \aleph_{\mathbf{i}}$	Set of items interacted by user *u* / users interacted by item i
$\alpha / \beta / \gamma / \delta / \varepsilon$	Hyper-parameters

Such graphs can generally be represented as G = {U, I, E}, where U and I donate for user set and item set, E donate for edge set. In recommender systems, if there is an observed interaction between one user and one item, it is considered that there exists an edge in the set E connecting that user and item. Figure 1 below shows an undirected graphical representation of the toy example from the candidates' perspective. In this graph, U = {1, 2}, I = {1, 2, 3}. Since it is a candidate perspective, only whether candidates are interested in particular jobs is considered. Therefore, E = {(1, 1), (1, 2), (2, 2), (2, 3)}. This is the method that current state-of-the-art GCN based CF method (GF-CF and BSPM) applied to represent U-I interactions.

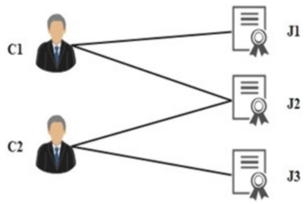
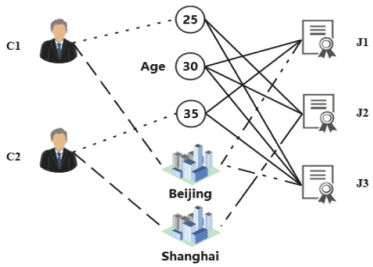


Fig. 1. Toy example, undirected graph on candidate's perspective

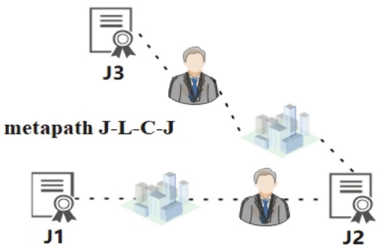
3.2 Heterogeneous Graph and Meta-Path

REHG utilizes heterogeneous graphs to represent user-information-item interactions, enabling it to assess additional information. A heterogeneous graph is characterized by various types of information attached to nodes and edges. Therefore, using a single tensor for node or edge features isn't sufficient due to variations in dimensionality or type. Instead, we need a set of types with different data tensors for specific nodes and edges. The feature transformation approach is adjusted accordingly to compute message and update functions based on node and edge type.

Meta-path is a common information transformation approach in heterogeneous graph applications. It is defined as a path that connects nodes of different categories. A heterogeneous graph can be transformed into its homogeneous subgraph by using a meta-path with the same categories of origin and destination nodes. Figure 2a shows the complete heterogeneous graph in the toy example, while Fig. 2b depicts the transformed homogenized subgraph using the meta-path job-location-candidate-job. The categories of nodes and edges remain the same.



(a) Heterogeneous graph with information



(b) Homogenized subgraph through meta-path J-L-C-J

Fig. 2. Toy example, heterogeneous graph and meta-path

By applying the method above, the similarity between node with same category through a specific meta-path can be calculated. Assuming that the meta-path is A_1-A_2-A_3-…-A_N-A_1, which the corresponding adjacent matrices are $R_{1,2}$, $R_{2,3}$, $R_{3,4}$, …, $R_{N\text{-}1,N}$, $R_{N,1}$, respectively. Then the similarity between nodes of category A_1 is:

$$\boldsymbol{P}_{A_1A_1}(A_1, A_2, A_3, \ldots, A_N, A_1) = \boldsymbol{R}_{1,2} \cdot \boldsymbol{R}_{2,3} \cdot \boldsymbol{R}_{3,4} \cdot \cdots \cdot \boldsymbol{R}_{N-1,N} \cdot \boldsymbol{R}_{N,1} \tag{8}$$

Hence in Fig. 2b:

$$\boldsymbol{P}_{JJ}(\text{job, location, candidate, job}) = \boldsymbol{R}_{JL} \cdot \boldsymbol{R}_{LC} \cdot \boldsymbol{R}_{CJ} \tag{9}$$

Same approach holds true for the normalized adjacent matrix.

It's important to note that even without additional information, the graph structure used in CF methods can be viewed as heterogeneous because users and items belong to different node categories. For instance, Eq. 8 can be seen as computing the similarity of items through the meta-path item-user-item.

3.3 Graph Signal Processing and De-Nosing

In signal processing, it's believed that signals consist of both information and noise. These signals may vary in strength across different frequencies in the spectral domain. Low-pass filtering is a widely used method for denoising, as it efficiently eliminates noise at higher frequencies while preserving information at lower frequencies [18].

Similar approach has been migrated to graphs and it is known as graph signal processing [19]. GF-CF [4] has demonstrated the importance of smoothness and applying lower pass filtering in CF method. The statement is also valid in mainstream CF methods. Since more information is contained in the REHG model, de-nosing process is believed to be necessary. Instead of low-pass filtering, graph signal processing applies singular value decomposition (SVD) on adjacent matrix to achieve low-pass filtering [20]:

$$R = U\Sigma V^T (10) \tag{10}$$

After applying the SVD process, the top-k singular values of the adjacency matrix are used in the REHG model to create a new adjacency matrix. This helps reduce noise in the graph signal. The selection of k, a hyper-parameter in REHG, is crucial. This process is similar to spectral decomposition, where information from the top k dimensions represents the original signal.

3.4 Unified Framework of REHG

With the above steps, for each known and reasonable meta-path of the heterogeneous graph, REHG method can provide two similarity expressing for the corresponding nodes, i.e., before and after the de-nosing process. For the toy example, the similarity between the job nodes through meta-path job-location-candidate-job in Fig. 2b can be represented as:

$$\widetilde{\boldsymbol{P}}_{\boldsymbol{JJ}}(\text{job, location, candidate, job}) = \widetilde{\boldsymbol{R}}_{\boldsymbol{JL}} \cdot \widetilde{\boldsymbol{R}}_{\boldsymbol{LC}} \cdot \widetilde{\boldsymbol{R}}_{\boldsymbol{CJ}} \tag{11}$$

$$\widetilde{\boldsymbol{P}}_{\boldsymbol{JJ}}^{\boldsymbol{Denoise}}(\text{job, location, candidate, job}) = \boldsymbol{D}_{\boldsymbol{J}}^{-1/2} \cdot \boldsymbol{U}_{\boldsymbol{JL}} \cdot \boldsymbol{U}_{\boldsymbol{LC}} \cdot \boldsymbol{U}_{\boldsymbol{CJ}} \cdot \boldsymbol{D}_{\boldsymbol{J}}^{-\frac{1}{2}} \tag{12}$$

Each meta-path in REHG contains semantic information and is normalized. Therefore, the magnitude and sign of the corresponding parameter can somewhat reflect the relevance and importance of the information on that meta-path. In other words, the model exhibits some interpretability. Although making these coefficients trainable could potentially enhance performance, this aspect is not the focus of this study. Instead, a set of weights with satisfactory performance is provided as a hyper-parameter through grid search.

Similar to mainstream methods, the proposed REHG selects the top-n options with the highest interaction scores as the recommended results, which are then used for subsequent assessments of method performance. All the statements above will be further discussed in Sect. 4 below.

4 Experiment

In order to illustrate the superiority and validity of the REHG model, two experiments are set up in this study. This section will describe the setups, provide the experimental results and discuss the relevant results for them respectively.

4.1 Experiment Setups

Dataset
Table 3 below shows the basic information of two datasets used in the experiments. In each experiment, 80% of the data are randomly selected to form the training set while the rest of data are selected to form the testing set.

Table 3. Information of two datasets

Name	User no	Item no	Interactions no	Density
Yelp [21]	27.717	22.835	631.911	0.10%
Movielens-25m [22]	16.983	12.319	2.402.778	1.14%

Experiment Environment
All the experiments in this study are conducted on a server with a single Intel I7-11700K@3.6 GHz CPU and one 28 RAM NVIDIA GeForce RTX 3090 GPU. The setup of hyper-parameters as well as model-specific structures are described separately in different subsections below.

Evaluation Indicators
For all experiments, the top 20 items in terms of predicted interactions score are taken as recommendations. In this study, Recall and Ndcg, which are commonly used in the field of recommender systems, are applied to evaluate the performance of each model.

4.2 Experiment with Additional Information

This experiment aims to demonstrate that considering additional information in a recommender system is able to improve the performance of the model. The experiment in conducted on two datasets, Movielens-25m and Yelp, containing additional information. On the movielens-25m dataset, three additional meta-paths: movie-year-movie; movie-genre-movie and movie-tag-user-movie as well as two original meta-path: movie-user-movie and user-movie-user has been applied. The model is finally represented as Eq. 13 below:

$$\boldsymbol{S} = \alpha \boldsymbol{R}\left(\tilde{\boldsymbol{R}}^{\boldsymbol{T}}\tilde{\boldsymbol{R}} + \mu \boldsymbol{D}_{\boldsymbol{I}}^{-\frac{1}{2}}\tilde{\boldsymbol{U}}\tilde{\boldsymbol{U}}^{\boldsymbol{T}}\boldsymbol{D}_{\boldsymbol{I}}^{-\frac{1}{2}}\right) + \beta \tilde{\boldsymbol{R}}^{\boldsymbol{T}}\left(\tilde{\boldsymbol{R}}\tilde{\boldsymbol{R}}^{\boldsymbol{T}} + \mu \boldsymbol{D}_{\boldsymbol{U}}^{-\frac{1}{2}}\tilde{\boldsymbol{V}}^{\boldsymbol{T}}\tilde{\boldsymbol{V}}\boldsymbol{D}_{\boldsymbol{U}}^{-\frac{1}{2}}\right)$$
$$+ \gamma \boldsymbol{R}\big[\tilde{\boldsymbol{P}}_{\boldsymbol{MM}}(\boldsymbol{movie}, \boldsymbol{year}, \boldsymbol{movie})$$

$$
\begin{aligned}
&+\mu\tilde{\boldsymbol{P}}_{\boldsymbol{MM}}^{\boldsymbol{Denoise}}(\boldsymbol{movie},\boldsymbol{year},\boldsymbol{movie})\Big]\\
&+\delta\boldsymbol{R}\Big[\tilde{\boldsymbol{P}}_{\boldsymbol{MM}}(\boldsymbol{movie},\boldsymbol{genre},\boldsymbol{movie})\\
&+\mu\tilde{\boldsymbol{P}}_{\boldsymbol{MM}}^{\boldsymbol{Denoise}}(\boldsymbol{movie},\boldsymbol{genre},\boldsymbol{movie})\Big]\\
&+\varepsilon\boldsymbol{R}\Big[\tilde{\boldsymbol{P}}_{\boldsymbol{MM}}(\boldsymbol{movie},\boldsymbol{tag},\boldsymbol{user},\boldsymbol{movie})\\
&+\mu\tilde{\boldsymbol{P}}_{\boldsymbol{MM}}^{\boldsymbol{Denoise}}(\boldsymbol{movie},\boldsymbol{tag},\boldsymbol{user},\boldsymbol{movie})\Big]
\end{aligned}
\tag{13}
$$

Again by applying grid search, the hyper-parameters α, β, γ, δ, ε and μ are set to [1, –0.35, 0.2, 0.2, 0.1, 0.3]. Through a similar approach, on the Yelp dataset, two additional meta-paths: item-postal code-item and item-category-item as well as two original meta-paths: item-user-item and user-item-user has been applied. The hyper-parameters are then set to [1, 0.2, 0.2, 0.2, 0.3]. The experiment applies GF-CF and BSPM as baselines. Additionally, an HRBA-WI model which do not consider additional information is tested. The experimental results are summarized in Table 4 below.

Table 4. Experimental Results with Additional Information

Model	Movielens-25m		Yelp	
	Recall@20	Ndcg@20	Recall@20	Ndcg@20
MF-CCL [24]	0.2364	0.3195	0.1343	0.0885
EASE [11]	0.2272	0.3076	0.1264	0.0854
YouTubeNet [23]	0.2257	0.3153	0.1320	0.0877
NGCF [2]	0.2021	0.2840	0.1114	0.0738
LightGCN [3]	0.2355	0.3326	0.1248	0.0820
GF-CF [4]	0.2471	0.3418	0.1385	0.0917
BSPM [5]	0.2512	0.3459	0.1258	0.0843
REHG-WI	0.2524	0.3502	0.1393	0.0922
REHG	**0.2561**	**0.3554**	**0.1445**	**0.0954**

In the baseline models of this experiment, GF-CF is the fundamental signal processing model. BSPM is considered to be the state-of-the-art recommendation engine. While REHG-WI can be considered as an ablation experiment for the meta-path structure and extra information in REHG model. Besides, it is worth noting that all models perform poorly on the yelp dataset. The reason for this is that the yelp dataset is more sparse. As it is shown in Table 4, REHG that has the ability to evaluate additional information may improve model performance by around 3–7%. This provides enough evidence for the validity of the heterogeneous graph and meta-path structure in the REHG model.

4.3 Comparison with Models that Evaluate Additional Information

This experiment aims to demonstrate that REHG model has the state-of-the-art performance among all recommender systems that have the ability to consider additional information. The experiment is also performed on two datasets, Movielens-25m and Yelp, with additional information. The setup of REHG is consistent with that in Subsect. 4.3. It applies PEAGNN, HeRec, Metapath2Vec, MultiGCCF, and KGAT as baseline models. The experimental results are summarized in Table 5 below.

Table 5. Comparison of models that evaluate additional information

Model	Movielens-25m		Yelp	
	Recall@20	Ndcg@20	Recall@20	Ndcg@20
HeRec [13]	0.0900	0.0645	0.0447	0.0054
Metapath2Vec [14]	0.1335	0.0990	0.0509	0.0366
MultiGCCF [15]	0.1494	0.1113	0.0567	0.0583
KGAT [16]	0.1507	0.2028	0.0708	0.0601
PEAGNN [17]	0.1825	0.2395	0.0988	0.0775
REHG	**0.2561**	**0.3554**	**0.1445**	**0.0954**

As it is shown in Table 5, compared to the model which is able to evaluate additional information, REHG proposed in the paper has a dramatically improvement in all indicators. The improvements on all indicators exceed 35%. It is believed that the application of latest graph signal processing theory had a significant impact on this. Due to the lack of sufficient process computation method, the performances of the recent baselines, such as PEAGNN which applies additional information in the model, are even worse than the methods that do not evaluate additional information into as shown in Table 5 above. And in summary, the experiments above provide enough evidence to show that REHG is currently the state-of-the-art recommendation engine that can evaluate additional information in the system.

4.4 Analysis

In order to ensure a fair comparison and to provide sufficient and reasonable evidence to the superiority of the REHG model in terms of performance, the study designs two experiments as described previously. Among them, Experiment 1 in Subsect. 4.2 illustrates that additional information such as age of the users and origin of the products can be used to improve the performance of recommendation engine. Then experiment 2 in Subsect. 4.3 demonstrates that the REHG models has a dramatically performance advantage compared to baseline models that can evaluate additional information.

Besides high accuracy than baseline models, the proposed REHG model also requires much less time for model computation since there is no deep learning process. Only matrix multiplication and top-k SVD process are required in the computation process, with time complexity of $O(n^3)$ and $O(kn^2)$ respectively. Due to this reason, REHG model can keep the overall computation time. To demonstrate this, we recorded the training time for each model. LightGCN was trained on the Movielens-25m and Yelp datasets in 3 and 5 h, respectively. PEAGNN was trained on the above two datasets in 7.5 and 14.7 h. REHG was trained on the two datasets in 5 and 4 min, respectively.

In addition, since the similarity matrices are normalized, the weights to some extend can reflect the correlation and importance between the information contained in each meta-path and the recommendation results in the weighted summation process. For example, in experiment 1, the weight of meta-path movie-year-movie is -0.35 and movie-genre-movie is 0.2. It is then reasonable to believe that the released year of a film is negatively correlated with whether or not people like the film. That is to say, people are more likely to watch a film from different released years. To the best knowledge of the authors, this is the first attempt to add interpretability to recommender system model. This allows recommender systems not only provide recommendation, but also be able to provide feedback on why the recommendation has been made which may be useful for subsequent business activities.

In summary, it should be possible to consider the REHG model proposed in the study as a reasonable, efficient and interpretable model. Its performance is better than the common types of recommendation engines in all aspects.

5 Conclusion

In conclusion, this study proposes a new recommendation system REHG based on directed heterogeneous graph. In this study, two experiments are designed to validate the effectiveness of the REHG model. These experiments demonstrate the effectiveness of the structure, while its recommendation accuracy on the test dataset is substantially improved compared to the common baseline models. Compared to models that cannot evaluate additional information, REHG is approximately 5–7% better on each indicator. Furthermore, compared to existing models that are able to evaluate additional information, REHG improves the performance by more than 30% on all indicators based on its structural superiority. In addition, this study discusses the arithmetic requirements and the interpretability of the model. Overall, it should be possible to consider the REHG model proposed in the study as a reasonable, efficient and interpretable model.

In further research, a more stable sharpening process could be considered to further improve the performance of the REHG model. At the same time, the study of the interpretability should be more intensive. An indicator similar to p-value may be designed to describe the significance of each variable. Thus, the complexity of the model can be effectively reduced and the computational efficiency of the model can be improved by filtering the relevant meta-paths and information.

Acknowledgments. This study is supported by the Natural Science Foundation of Shandong Province (Grant No. ZR2022QF144).

References

1. Lu, J., et al.: Recommender system application developments: a survey. Decis. Support. Syst. **74**, 12–32 (2015)
2. Wang, X., et al.: Neural graph collaborative filtering. In: Proceedings of the 42nd international ACM SIGIR Conference on Research and Development in Information Retrieval (2019)
3. He, X., et al.: 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 (2020)
4. Shen, Y., et al.: How powerful is graph convolution for recommendation? In: Proceedings of the 30th ACM International Conference on Information and Knowledge Management (2021)
5. Choi, J., et al.: Blurring-sharpening process models for collaborative filtering. In: Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval (2023)
6. Boratto, L., Carta, S.: State-of-the-art in group recommendation and new approaches for automatic identification of groups. In: Information Retrieval and Mining in Distributed Environments, pp. 1–20. Springer (2011)
7. Liu, F., et al.: Interest-aware message-passing GCN for recommendation. In: Proceedings of the Web Conference 2021 (2021)
8. Ricci, F., Rokach, L., Shapira, B.: Recommender systems: introduction and challenges. In: Recommender Systems Handbook, pp. 1–34 (2015)
9. Rendle, S., et al.: BPR: bayesian personalized ranking from implicit feedback. arXiv preprint arXiv:1205.2618 (2012)
10. Chen, C., et al.: Efficient neural matrix factorization without sampling for recommendation. ACM Trans. Inform. Syst. (TOIS) **38**(2), 1–28 (2020)
11. Steck, H.: Embarrassingly shallow autoencoders for sparse data. In: The World Wide Web Conference (2019)
12. Hu, J., et al.: MGDCF: Distance Learning via Markov Graph Diffusion for Neural Collaborative Filtering. arXiv preprint arXiv:2204.02338 (2022)
13. Shi, C., et al.: Heterogeneous information network embedding for recommendation. IEEE Trans. Knowl. Data Eng. **31**(2), 357–370 (2018)
14. Dong, Y., Chawla, N.V., Swami, A.: Metapath2vec: scalable representation learning for heterogeneous networks. In: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (2017)
15. Sun, J., et al.: Multi-graph convolution collaborative filtering. In: 2019 IEEE International Conference on Data Mining (ICDM). IEEE (2019)
16. Wang, X., et al.: KGAT: knowledge graph attention network for recommendation. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (2019)
17. Han, Z., et al.: Metapath-and entity-aware graph neural network for recommendation. CoRR, abs/2010.11793 (2020)
18. Yang, T., Yu, H., Wang, Y.: An efficient low-pass-filtering algorithm to de-noise global GRACE data. Remote Sens. Environ. **283**, 113303 (2022)
19. Xia, F., et al.: Graph learning: a survey. IEEE Trans. Artif. Intell. **2**(2), 109–127 (2021)
20. Chang, H., et al.: Not all low-pass filters are robust in graph convolutional networks. Adv. Neural. Inf. Process. Syst. **34**, 25058–25071 (2021)
21. Yu, M., Xue, M., Ouyang, W.: Restaurants Review Star Prediction for Yelp Dataset, Technical Report 17. UCSD (2015)
22. Harper, F.M., Konstan, J.A.: The movielens datasets: history and context. ACM Trans. Interact. Intell. Syst. (TIIS) **5**(4), 1–19 (2015)

23. Covington, P., Adams, J., Sargin, E.: Deep neural networks for YouTube recommendations. In: Proceedings of the 10th ACM Conference on Recommender Systems, pp. 191–198 (2016)
24. Ma, J., Zhou, C., Cui, P., Yang, H., Zhu, W.: Learning disentangled representations for recommendation. In: NeurIPS, vol. 32 (2019)