International Journal of Data Warehousing and Mining
Volume 20 • Issue 1

Hybrid Inductive Graph Method for Matrix Completion

Jayun Yong, Sookmyung Women's University, South Korea
Chulyun Kim, Sookmyung Women's University, South Korea*
https://orcid.org/0000-0002-6471-5334

ABSTRACT

The recommender system can be viewed as a matrix completion problem, which aims to predict unknown values within a matrix. Solutions to this problem are categorized into two approaches: transductive and inductive reasoning. In transductive reasoning, the model cannot be applied to new cases unseen during training. In contrast, IGMC, the state-of-the-art inductive algorithm, only requires subgraphs for target users and items, without needing any other content information. While the absence of a requirement for content information simplifies the model and enhances transferability to new tasks, incorporating content information could still improve the model's performance. In this article, the authors introduce Hi-GMC, a hybrid version of the IGMC model that incorporates content information alongside users and items. They present a novel graph model to encapsulate the side information related to users and items and develop a learning method based on graph neural networks. This proposed method achieves state-of-the-art performance on the MovieLens-100K dataset for both warm and cold start scenarios.

KEYWORDS

Deep Learning, Graph Neural Networks, Hybrid Recommendation, Recommendation Systems

INTRODUCTION

In the vast ocean of information, navigating to the desired data has become a formidable challenge, thereby amplifying interest in recommender systems. These systems streamline the process for users, enabling them to discover the information they need more efficiently and swiftly, while also offering companies an avenue to enhance service engagement and foster business advantages (Zamanzadeh Darban & Valipour, 2022).

Recommender systems align predominantly with two main frameworks: content-based methods and collaborative filtering methods. Content-based methods suggest items that mirror the user's historical preferences. In contrast, collaborative filtering works by leveraging the aggregated preferences of other users to predict what an individual might like. While content-based strategies might limit suggestions to items closely related to those previously preferred by the user, thus narrowing

DOI: 10.4018/IJDWM.345361 *Corresponding Author

the breadth of recommendations, collaborative filtering broadens the spectrum of suggestions by reaching beyond similarities in content alone (Almazro et al., 2010).

Among various approaches to implementing collaborative filtering, matrix completion—organizing a matrix with users on one axis and items on the other—has gained prominence (Candès & Recht, 2009). By hypothesizing that the rating matrix is of low rank, numerous leading matrix completion algorithms employ factorization techniques, which have proven to be highly effective. However, matrix factorization faces an inherent limitation: it is transductive (Koren et al., 2009). This means that the latent features learned from users and items in the given dataset cannot be applied to new users or items not seen during training. Consequently, changes in the rating matrix, such as updates or the addition of new data, often necessitate a full retraining to generate new embeddings. To overcome this challenge, various studies have introduced methods to achieve inductive matrix completion (Michalski, 1983). One notable study in this area is the inductive graph-based matrix completion (IGMC) model (Zhang & Chen, 2020). IGMC leverages a graph neural network (GNN) that processes 1-hop subgraphs around user-item pairs from the rating matrix, associating these subgraphs with their respective ratings to enable inductive generalization effectively.

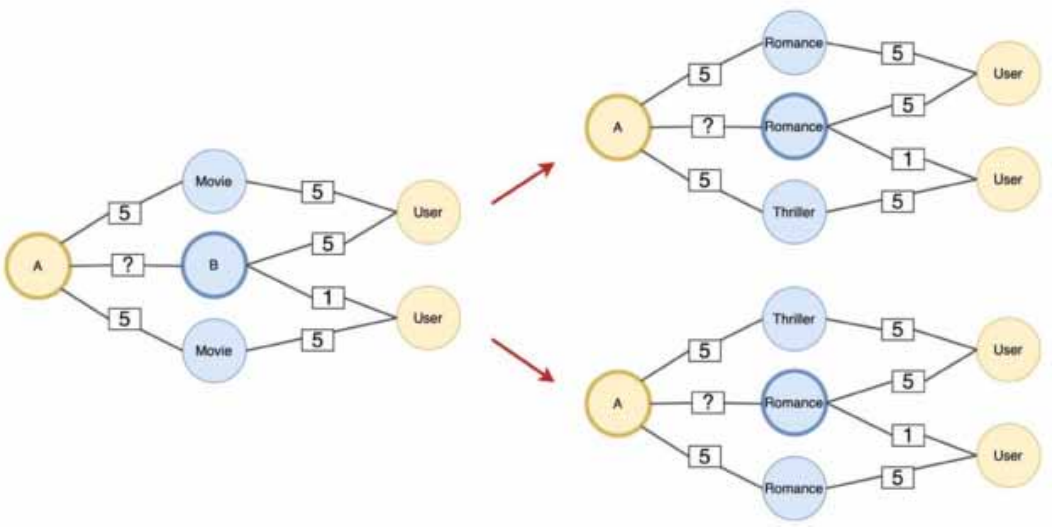
Although the IGMC model successfully introduced inductiveness into matrix completion, it also underscored a critical limitation inherent in collaborative filtering models by strategically avoiding content information to capitalize on its inductive strengths. A critical challenge faced by IGMC is the cold-start problem. This occurs when new users or items are introduced into the system without enough interaction data to form a reliable recommendation. This issue primarily affects collaborative filtering algorithms, which depend on past interactions to make their recommendations. In the absence of such data, a collaborative algorithm is unable to make a certain recommendation. Even with a minimal amount of interaction data, while a collaborative algorithm may still be able to offer recommendations, the reliability and quality of these recommendations are likely to be substandard.

Another key limitation of IGMC lies in its inability to finely integrate and weigh the preferences of neighboring users. Consider the scenario, depicted in Figure 1, where movie B might be recommended to user A. The graphs in Figure 1 illustrate interactions between users and movies with a 1-to-5 rating scale. In the graph on the left, since user A and all other users rate movies highly together except for movie B, the preferences of the other users influence the decision to recommend movie B to user A in collaborative filtering. However, in the graph, ratings for movie B diverge significantly among other users, and thus its recommendation to user A becomes less clear-cut. The addition of content information, like movie genres, can change this situation. The graphs on the right incorporate genre information, suggesting that recommendations for movie B, identified as a romance, could differ on the basis of the genres of movies that other users have rated. Assuming the scenario where ratings for the same genre have more weight than ratings for different genres, movie B might be strongly recommended to user A in the graph at upper right, since the neighbor user's rating for romance has more weight than that for thriller. Conversely, movie B might not be recommended to user A in the graph at lower right. Therefore, incorporating content information into collaborative filtering not only improves performance but also allows for more tailored recommendations.

In this study, we introduce a cutting-edge hybrid recommendation technique, termed hybrid inductive graph matrix completion (Hi-GMC), which enhances the accuracy of inductive matrix completion and its effectiveness in cold-start scenarios. Unlike IGMC, which foregoes the use of content information to optimize inductive generalization, Hi-GMC leverages additional content-related clues alongside user-item interaction data. On the MovieLens 100K dataset (Miller et al., 2003), Hi-GMC outperformed most contemporary leading models (Code, 2022), establishing a new state of the art in recommendation system performance.

In the subsequent sections of this paper, we first review literature pertinent to our study before detailing the methodologies and mathematical formulations underpinning IGMC. In the method section, we elaborate on the approaches and mathematical frameworks that form the foundation of IGMC, offering comprehensive insights into the mechanisms driving our proposed model. This

Figure 1. Collaborative filtering recommender systems without vs. with content information



account includes an in-depth exploration of the techniques and algorithms employed to enhance recommendation accuracy and address cold-start challenges in recommendation systems. In the experimental analysis, we conduct an exhaustive evaluation of Hi-GMC's performance in comparison to other leading methods using the MovieLens 100K dataset, followed by a discussion of the findings. We conclude by summarizing our research and highlighting its contributions.

RELATED WORK

Recommender System

Recommendation systems are generally divided into two primary methodologies: content-based and collaborative filtering. Content-based methods offer suggestions by aligning with the user's past preferences. For example, if a user shows interest in a specific item, the system will recommend similar items. This approach, while straightforward, tends to circumscribe the variety of suggestions, often overlooking the potential for serendipity (the joy of unexpected discoveries) and novelty (the introduction to previously unknown items) in recommendations (Maksai et al., 2015). In contrast, collaborative filtering expands recommendation horizons by leveraging the collective preferences of users with similar tastes. It operates on the assumption that if two users share similar interests, they are likely to appreciate each other's favorites, thus offering a broader range of recommendations. Despite the acclaim collaborative filtering has received for its role in enhancing recommendation diversity, it is imperative to recognize the value of content-specific data, which may be overlooked by models that are based solely on interaction. Consequently, the industry frequently adopts a hybrid approach, combining the strengths of both content-based and collaborative filtering, to deliver more comprehensive and nuanced recommendations (Almazro et al., 2010; Berg et al., 2017; Cai et al., 2021; Candès & Recht, 2009; Code, 2022; Zamanzadeh Darban & Valipour, 2022; Grover & Leskovec, 2016; Hamilton et al., 2018; He et al., 2020; Kim et al., 2021; Koren et al., 2009; Lam et al., 2008; Liu et al., 2020; Lops et al., 2011).

In the research conducted on recommenders, one area that has recently gained attention is fairness (Giap et al., 2022; Deldjoo et al., 2024). A drawback of recommendation systems is the issue of confirmation bias, which arises when diverse options that might be of interest are blocked,

and recommendations are limited mainly to popular items. Additionally, with the advancement of large language model (LLM) technologies, research on recommendation systems utilizing these technologies is also being actively conducted (Fan et al., 2023). This research includes not only studies that leverage the characteristics of LLM networks (Cui et al., 2023) but also research into the explainability of LLM's recommendation results (Gao et al., 2023).

Graph Neural Networks for Recommender System

Recent advancements in recommendation models have seen the integration of GNNs in various innovative ways (Berg et al., 2017; Monti et al., 2017). These models leverage the natural graph structure of user-item interactions, enhancing recommendation systems significantly. LightGCN simplifies the graph convolutional network (GCN) framework by removing nonessential components such as feature transformation and nonlinear activation functions, focusing instead on neighborhood aggregation for improved recommendations (He et al., 2020). On the other hand, the graph convolutional matrix completion (GCMC) employs the GCN approach, showing remarkable performance but falling short in capturing higher-order connectivity because of its single-layer structure (Liu et al., 2020). Moreover, it faces challenges with memory efficiency, as it requires loading the entire graph into memory simultaneously (Berg et al., 2017).

A common limitation among these models is their transductive nature, necessitating retraining to accommodate new, untrained nodes. In contrast, GraphSage extends GCN capabilities for segmental graph training, making it suitable for inductive settings where the model learns from parts of the graph (He et al., 2020). Similarly, the IGMC model adopts an inductive approach, learning local subgraph patterns through new node labeling, although it struggles to leverage user or item content information effectively. Content information, which encompasses a wealth of data beyond simple interaction details, plays a crucial role in enhancing recommendation quality. Attempts to incorporate content information into IGMC by converting it into feature vectors for final embedding have shown negligible performance improvements, indicating a challenge in utilizing content information effectively (Zhang & Chen, 2020).

PinSage represents a step forward by integrating node content into the GraphSage framework, tailored for platforms like Pinterest that have abundant node content information. However, its utility is limited in environments lacking rich node content (Ying et al., 2018). Another approach, CoRGi, adds content attention to the GNN model, enabling personalized recommendations through this mechanism, but it still requires substantial node content and retains the transductive model limitations (Kim et al., 2021). The challenges highlighted for existing inductive models, such as the ineffectiveness of IGMC in harnessing content information and the dependency on rich features for models like PinSage and CoRGi, underscore the ongoing quest for optimizing GNN-based recommendation systems (Ying et al., 2018; Zhang & Chen, 2020).

Cold Start Problem

A sufficient amount of data is required to make high-quality recommendations to users. However, not every organization possesses large datasets, and newly introduced users and items often lack enough interaction data to generate reliable recommendations. This dilemma is known as the cold-start problem, in which the absence of adequate data hampers the ability to offer accurate recommendations (Lam et al., 2008). Despite extensive research in recommendation systems, the cold-start issue persists as a fundamental challenge. Utilizing content information is a common strategy to mitigate this problem, but basic application methods have shown limited effectiveness.

The heterogeneous graph neural recommender (HGNR) model presents an innovative solution designed to address the cold-start challenge within heterogeneous graphs. It enhances the utilization of content information by expanding the traditional user-item matrix into a combined (user+item)-by-(user+item) matrix. This expansion includes additional connections between users based on social network insights and between items that share similar reviews, thereby enriching the graph's

structure. Such an approach enables the content information to be integrated more effectively than merely including it as a feature attribute, offering a more nuanced and comprehensive framework for overcoming the cold-start obstacle (Liu et al., 2020).

METHODS

Preliminaries

The IGMC model operates by predicting the relationship between a target user and an item through the analysis of local subgraph patterns. Given the inefficiency of training on the entire graph, it instead focuses on a 1-hop subgraph surrounding the target user and item. This process includes relabeling within the subgraph, a process in which target users and items are assigned labels 0 and 1, respectively, while their 1-hop neighbors receive labels 2—i.e., number of hops * 2—for users and 3—i.e., number of hops * 2 + 1—for items. A unique aspect of this approach is that the same user or item can receive different labels in different subgraphs, enabling the model to learn inductively and adapt its predictions to varying contexts (Zhang & Chen, 2020).

While IGMC shows strong performance as an inductive model, rivalling even some transductive models, it falls short in leveraging additional content information effectively. Attempts to enhance its performance by incorporating feature vectors directly into the final embedding yielded no substantial improvement. In the subsequent sections, considering that content information can greatly aid in accurately predicting user preferences, we will integrate content information into the IGMC framework more effectively, allowing for a nuanced use of node content to enrich the recommendation process.

Creation of Heterogeneous Graph

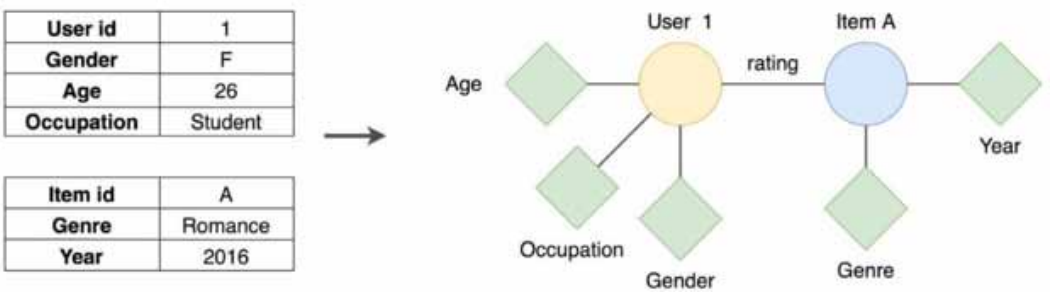
Graphs can be divided into two main types: homogeneous and heterogeneous. Homogeneous graphs contain only a single type of node and edge, akin to relationships in social networks. Conversely, heterogeneous graphs have nodes and edges of multiple types (Wang et al., 2020). Recommendation datasets typically fall into the latter category, being heterogeneous in nature (Zhao et al., 2020). These graphs comprise nodes that represent users and items. In the case of implicit data, there is only a single type of edge, but explicit data may contain various types of edges, each representing a different rating level. Despite their complexity, traditional graph-based recommendation systems often limit their focus to users, items, and ratings, without exploring beyond these elements, even in the context of heterogeneous graphs.

In developing our graph-based recommendation system, we initially create a graph that maps interactions between users and items. However, in contrast to traditional approaches, we enrich this graph with additional node content that provides deeper insights into the nodes, such as movie genres or user demographics. For instance, a movie recommendation graph may include nodes enriched with features like movie genre or user gender. An illustration of constructing a graph with this content information is provided in Figure 2. For categorical variables (e.g., genre, gender, occupation), the nodes can directly represent these categories. Numeric variables, however, need to be categorized on the basis of defined intervals for this purpose. During our experiments, we converted numerical variables into categorical ones based on specified ranges. The selection of features from node content is flexible; one may utilize all available features or selectively use only a few. Through extensive testing, we opted to use a single feature for each user and item. This choice was driven by the observation that employing the full array of features did not yield a sufficient improvement in performance to justify the time inefficiency.

Subgraph Extraction

Training an entire graph in memory is not resource-efficient. Therefore, our approach involves extracting 1-hop subgraphs, similar to the method used by IGMC, where a 1-hop subgraph is drawn

Figure 2. Representation of users, items, and all content information as nodes in a heterogeneous graph



between the target user and the target item. Although IGMC explored the potential of extending beyond 1-hop, Zhang and Chen (2020) found that the marginal gains in performance did not justify the additional time required.

In this context, items that the target user has rated become 1-hop neighbors to the target user, and conversely, users who have rated the target item become 1-hop neighbors to that item. This expansion mirrors the breadth-first search algorithm, extending the graph through layers of direct connections. Node content, which is defined by the characteristics of users and items, is inherently incorporated into the subgraph once the constituents of the subgraph are selected, seamlessly integrating relevant features into the model's learning process (Zhang & Chen, 2020). In Figure 3, an example subgraph is given. The subgraph is constructed from a specific target user-item pair, incorporating all 1-hop adjacent users and items. Furthermore, it includes all related feature nodes associated with these users and items. The user or item associated with a feature node is called the mother node of the feature node.

Edge Labeling

The rating serves as the edge value connecting users to items. The challenge lies in determining the edge value between a feature node and its mother node. Note that the edge value assigned to a feature node does not mean any rate of value judgement, but is just for the consistency of graph structure. We explored several alternative assignment methods, such as the average rating value or serial numbers,

Figure 3. Inclusion of all related feature nodes as well as 1-hop item and user nodes in a subgraph from the heterogeneous graph

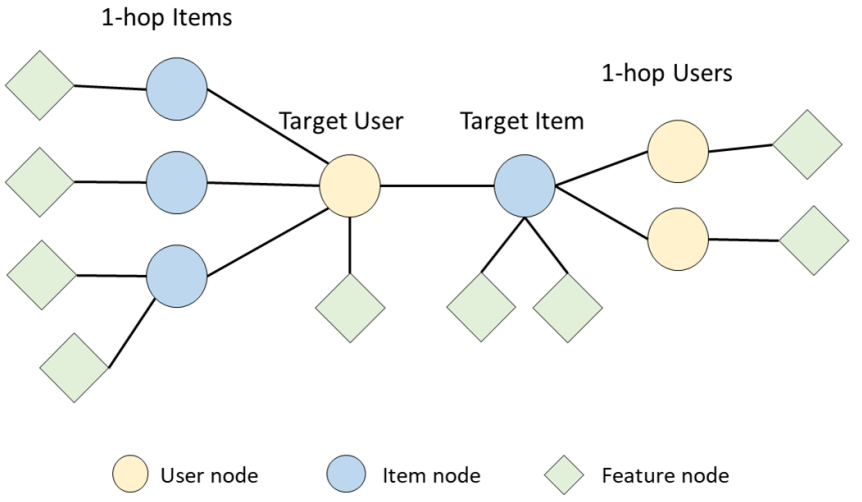


Table 1. Edge value assignment methods for feature nodes

Method	Description
Average rating scale	Use the average rating scale for all feature nodes
Number of features	Assign the edge value as the number of sibling feature nodes
Serial number	Assign a serial number to each distinct feature
Value replication	Replicate the edge value of mother node

as shown in Table 1. As verified in the experimental section, the most effective approach was value replication, and thus we adopted this value replication strategy for Hi-GMC.

Let us explain each method by using the examples illustrated in Figure 4. In the initial subgraph (a), only the target user and item nodes have the edge values. If the average rating value is applied, all feature nodes have the same edge value, which is determined by calculating the average rating scale. Supposing 1-5 Likert scale, this average rating scale is 3, and every feature node has the edge value of 3 in (b). The subgraph in (c) shows the case of applying the number of features. The target user has a single feature node, which has the edge value of 1, and the target item has two feature nodes, which have the edge values of 2. For the serial number method, we need to know all types of features. In the subgraph (d), we assume that there are only two types of features, gender and genre; the first serial number 1 is assigned to gender feature nodes, and the second serial number 2 is assigned to genre feature nodes. In the case of the value replication, the edge value of a mother node is replicated as those of its feature nodes, as shown in the subgraph (e). The feature nodes of the target user and item have the edge values as the replication of the edge value between the target user and item. The feature nodes of a 1-hop user (or item) have the replication of the edge value between the 1-hop user (or item) and the target item (or user).

Node Labeling

After extracting a subgraph, a new label is assigned to each node. The target user and item are assigned 0 and 1 are assigned, respectively, and 2 and 3 are used for 1-hop users and items, respectively. Here, note that the labels give only topological clues but not semantic ones. We assign labels to feature nodes as well, but intend to introduce minimal semantic distinction into labels. First, we distinguish the type of feature and its value. For example, if a user is female, the feature type is "gender," and the feature value is "female." Then, we divide all feature values into four groups. The first group contains feature values of the target user, and the second group contains those of the target item. The feature values of 1-hop users and those of 1-hop item are classified into the third and fourth groups,

Figure 4. Edge value assignment methods for feature nodes

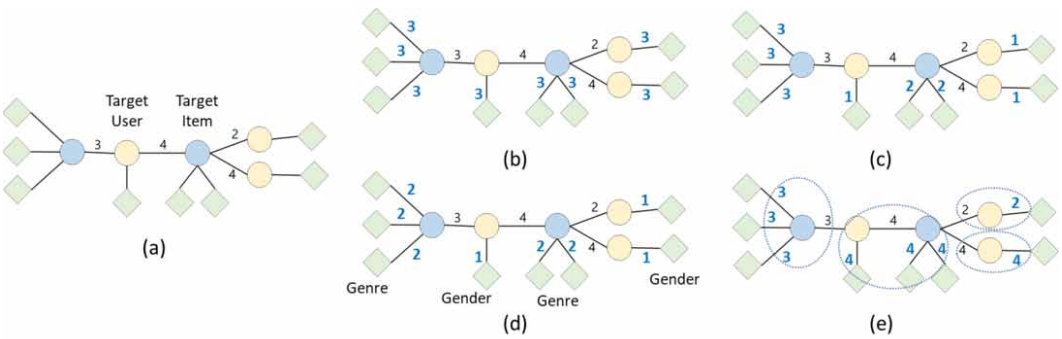
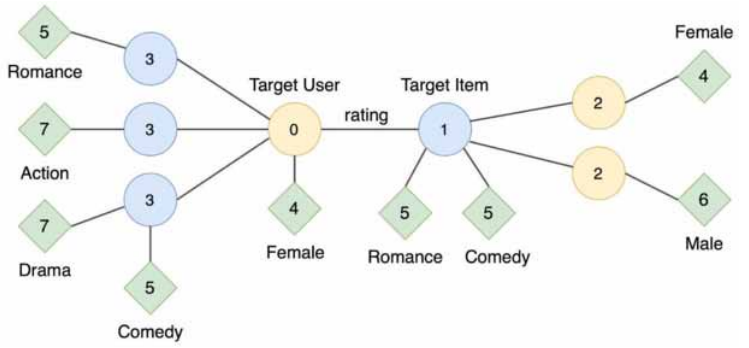


Figure 5. Example of feature node labeling



respectively. If a feature value can be classified into multiple groups, we define that the feature value is assigned to the first former group in order to make the four groups disjoint. Finally, assuming that the zero-base order is arbitrarily assigned to every feature type of user (or item), the label of a feature node is calculated as 3+(4i+j), where i is the order of feature type having the value of j-th group.

For example, let us consider the case given in Figure 5. There are six feature values in all: female, male, romance, comedy, action, and drama. Thus, the first group is {female}, the second group is {romance, comedy}, and the third and fourth groups are {male} and {action, drama}, respectively. Notice that although "female" is found in not only the target user's feature but also 1-hop user's feature, it is contained only in the first group. For the order of features, since user and item have only a single feature type in this example, the order of gender is 0, and that of genre is 0 as well. Therefore, the label of the target user's feature node is 4 (=3+(4*0+1)), and those of the target item's feature nodes are 5 (=3+(4*0+2)) together. However, feature nodes of 1-hop users have different labels. Since the feature value "female" of the upper 1-hop user is classified into the first group, its label is 4 (=3+(4*0+1)), but because the feature value "male" of the lower 1-hop user is classified into the third group, its label is 6 (=3+(4*0+3)). Similarly, feature nodes of 1-hop items having value of "romance" or "comedy" have labels as 5 (=3+(4*0+2)), and those having value of "action" or "drama" have labels as 7 (=3+(4*0+4)).

This node labeling method serves two goals: inductiveness and content-awareness. The inductiveness is achieved through simple numbering. Numbers do not depend on specific context, but indicate the topological relation. Even if a user or item has a new feature value unseen in training, its node label in a subgraph is determined simply by classifying this value into a group. Therefore, the trained model can be applied to the new feature value inductively without retraining. In addition, we should notice that the feature value grouping is a means for minimal semantic distinction. Even though the proposed labeling method does not consider specific feature values, the grouping of values reveals common features among 1-hop user (or item) nodes related to the target user (or item). As evaluated in the experimental section, this minimal semantic distinction enables the proposed model to significantly improve recommendation performance in cold-start cases.

Model Training

The model architecture is based on the design principles of IGMC (Zhang & Chen, 2020). It incorporates the relational graph convolutional network (R-GCN), a type of GCN layer, which is distinctive for its method of training with varying weights according to the type of edge involved. This approach offers a significant benefit: it enables the model to capture the complex patterns within graphs more effectively than it would by applying uniform weights. Therefore, Hi-GMC employs R-GCN as its core mechanism for the message-passing layer, as formulated in Equation (1):

$$h_i^{(l+1)} = \left(\sum_{r \in R} \sum_{j \in N_i^r} \frac{1}{c_{i,r}} W_r^l h_j^l \right) + W_0^l h_i^l \tag{1}$$

where h_i^l means the embedding of node i at layer l, and R denotes the set of edge values presented within the graph. For example, in MovieLens, R = {1, 2, 3, 4, 5}. The term N_i^r refers to node i's neighborhood with the edge value of r, and $c_{i,r}$ is the number of neighborhood in N_i^r. W_0^l and W_r^l are trainable parameter matrices. Equation (1) implies that the embedding of node i at layer $(l + 1)$ is calculated by aggregating the embeddings of node i 's neighbors at layer l using different weights for each rating.

The message passing layer processes through a total of L levels, where the output embedding from each level serves as the input for the subsequent level. Initially, the input for the first layer is derived from one-hot encoding of node labels, which are assigned within the extracted subgraph. The final node embedding is obtained by concatenating the embedding from each layer as shown in Equation (2) (Zhang & Chen, 2020; Zhang et al., 2018).

$$x_i = \text{concat}\left(h_i^0, h_i^1, \ldots, h_i^L\right) \tag{2}$$

To generate embeddings that represent an entire subgraph, we aggregate the embeddings of selected nodes as given in Equation (3). Specifically, Hi-GMC focuses on aggregating the embeddings of only the target users, target items, and target feature nodes within the subgraph. This approach is adopted because target nodes are deemed more critical than their neighboring nodes, making it more effective to concentrate solely on these nodes for aggregation (Zhang & Chen, 2020). The message passing layer, by its design, has demonstrated commendable performance by considering just the target user and target item embeddings, i.e., x_u and x_v, respectively. Nevertheless, to achieve marginally improved performance, we have also incorporated the embeddings of target feature nodes, i.e., x_{f1}, x_{f2}, … .

$$g = \text{concat}\left(x_u, x_v, x_{f_1}, x_{f_2}, \ldots\right) \tag{3}$$

The final subgraph representation g serves as the input to a linear layer, which is employed to compute the rating $\hat{r}$ for the recommendation. In this process, the ReLU activation function, σ, is utilized (Zhang & Chen, 2020). This approach is formally described in Equation (4).

$$\hat{r} = w^\top \sigma\left(Wg\right) \tag{4}$$

We employ mean squared error (MSE) as the loss function, where $R_{(u,v)}$ represents the true rating, and $\widehat{R}_{(u,v)}$ represents the predicted ratings. A pair $\left(u, v\right)$ signifies a user-item combination that exists within the rating matrix (Zhang & Chen, 2020), as shown in Equation (5).

$$\mathcal{L} = \frac{1}{\left|\left(u, v\right)\right|} \sum \left(R_{u,v} - \widehat{R}_{u,v}\right)^2 \tag{5}$$

The R-GCN layer operates with parameters tailored to various types of ratings. However, this approach has a limitation: it does not account for the relative values of the edges. For instance, although a rating of 4 is numerically closer to 5 than to 1, the R-GCN treats ratings 1, 4, and 5 as distinct categories without considering their numerical proximity. To address this issue, we normalize adjacent ratings to yield matrices that reflect their similarity (Zhang & Chen, 2020).

Feature Selection

Each dataset contains a vast array of feature information, making it crucial to discern which features to utilize for constructing the graph. Among these features, some are pivotal for predictions, while others are less important. Ideally, selecting the graph structure that delivers optimal performance would involve experimenting with every possible feature combination. However, given the extensive diversity of feature combinations, conducting such exhaustive experiments is impractical. Therefore, we assessed the simplest subset of feature combinations—the 1:1 user-item pairings—against the comprehensive approach where all features are included. Our findings, detailed in the experimental section, reveal instances where selected 1:1 combinations outperformed the all-inclusive approach. Moreover, limiting the scope to a subset of side information proves to be more memory-efficient. Thus, this paper introduces a weighted voting ensemble method tailored for viable 1:1 combinations, aiming to enhance accuracy.

Voting is an ensemble technique that determines the final output through a majority-vote principle. In weighted voting, each model is assigned a varying weight based on its confidence level. However, this voting method is applied predominantly in classification tasks and is not suitable for regression scenarios (Mendes-Moreira et al., 2012). Therefore, we have adapted the weighted voting approach as detailed in Equation (6).

$$\hat{r} = \sum_{i \in N} \frac{p_i}{p_{\text{sum}}} \hat{r}_i \tag{6}$$

where N represents the total number of models, each based on every possible 1:1 feature combination. p_i denotes the performance of each individual model, while p_{sum} is the cumulative performance of all models. Consequently, the final rating is calculated by allocating weights in proportion to the performance metrics of all models.

EXPERIMENTS

Dataset and Experimental Setup

We carried out studies utilizing the MovieLens 100K dataset to support the Hi-GMC project. This dataset is designed for movie recommendation purposes, comprising 100,000 ratings across 1,682 movies from 943 users. Out of the total 100,000 rating cases, we allocated 60,000 ratings for training, 20,000 for validation, and the remaining 20,000 for testing. Each rating is explicit, spanning from 1 to 5, with every user contributing a minimum of 20 ratings. Users are associated with supplementary demographic details, including gender, age, and occupation. Additionally, movies are cataloged with their titles, genres, and release years. The dataset classifies movies into 19 distinct genres (Miller et al., 2003).

We conducted experiments using a single Nvidia TITAN XP GPU. After the subgraphs are extracted by the CPU, the GPU focuses solely on training, allowing for efficient training with reduced memory requirements. Our training protocol included 30 epochs, a learning rate of 0.001, and the Adam optimization algorithm. We incorporated four message-passing layers into the model's architecture. To mitigate overfitting, we applied an edge dropout with a rate of 0.2 during the training process.

Edge Labeling Test

As mentioned in the Edge Labeling section, we carried out experiments to evaluate the edge values connecting user or item nodes with feature nodes, focusing on the genre feature from the MovieLens 100K dataset for performance comparison. As shown in Table 2, the efficacy of these experiments was evaluated on the basis of the root mean squared error (RMSE). RMSE represents the square root of the MSE, which itself is the average of the squared differences between actual and predicted values. This metric serves as an indicator of the model's accuracy in forecasting actual ratings, with a lower RMSE signifying higher prediction accuracy. The technique of replicating edge values demonstrated the best performance in comparison to other methodologies.

Feature Selection Test

The Hi-GMC model is capable of utilizing multiple pieces of content information, offering the flexibility to select different combinations. This process resembles the selection of a meta-path in a heterogeneous graph (Cai et al., 2021). We conducted experiments with various combinations of features on the MovieLens 100K dataset.

Utilizing the full set of features resulted in an RMSE of 0.861, indicating commendable performance. However, subsequent experiments revealed superior performance in terms of time and memory efficiency when a single feature was selected for each user and item. According to the results presented in Table 3, the optimal feature combination was identified as the pair of genre and gender, though other combinations also yielded satisfactory results. Furthermore, as discussed in Feature Selection section, we adapted the weighted voting technique to experiment with ensembles of 1:1 combination models. This modification led to an improved performance, lowering the RMSE to 0.851. It was observed that the graph structure is better understood when evaluating each feature combination individually, rather than aggregating all features together.

Table 2. Experiment Results of Edge Value Determination

Method	RMSE
Average rating scale	0.946
Number of features	0.950
Serial number	0.923
Value replication	**0.901**

Table 3. Performance comparison of Hi-GMC by changing feature combinations

Use Features	RMSE
gender, genre	**0.855**
gender, year	0.885
occupation, genre	0.860
occupation, year	0.880
age, genre	0.866
age, year	0.872
weighted ensemble	**0.851**
all features	0.861

Table 4. Test results on Movielens 100K

Model	Inductive	Content	RMSE
GC-MC	no	no	0.910
GC-MC+Feat	no	yes	0.905
GraphRec	no	no	0.904
GraphRec+Feat	no	yes	0.897
GHRS	no	yes	0.887
IGMC	yes	no	0.905
PinSage	yes	yes	0.951
Hi-GMC	yes	yes (gender, genre)	**0.855**
Hi-GMC	yes	yes (weighted ensemble)	**0.851**

Warm Start Test

Our model, Hi-GMC, was benchmarked against several notable graph-based recommendation models, including GC-MC, PinSage, GHRS, GraphRec, and IGMC, drawing from the works of Berg et al. (2017), Zamanzadeh Darban & Valipour (2022), Rashed et al. (2019), Ying et al. (2018), and Zhang & Chen (2020). GC-MC operates as a transductive model, utilizing one-hot encoding to differentiate nodes, but its efficiency is limited by the requirement to load the entire graph into memory for training. PinSage, in contrast, is an inductive model that leverages rich feature vectors and employs importance-based neighbor selection through random walks instead of simple k-hop subgraph extraction. However, PinSage's dependence on extensive side information (owing to its having been originally designed for image recommendations on Pinterest) may limit its effectiveness in other contexts. GHRS, a hybrid model, enhances feature extraction through the integration of autoencoders, demonstrating strong performance on the MovieLens dataset. GraphRec addresses the challenge of sparse user attributes due to privacy concerns by embedding users and items into a shared latent space, effectively operating without extensive side information. Finally, IGMC, the precursor to Hi-GMC, emphasizes memory efficiency by extracting 1-hop subgraphs and focuses on learning graph structures inductively, without relying on side information.

As detailed in Table 4, Hi-GMC outperformed all other models in our comparison, achieving a state-of-the-art RMSE of 0.855 on the MovieLens 100K dataset. This benchmark includes not only the models evaluated in this study but also those listed on the authoritative ranking platform, paperswithcode.com. When compared to the least effective combination of gender and year highlighted in Table 3, Hi-GMC's performance exceeds that of the existing SOTA. Furthermore, while models like GC-MC (Berg et al., 2017) and GraphRec (Rashed et al., 2019) show negligible performance variation with the inclusion of content information, Hi-GMC demonstrates a notable enhancement over IGMC, despite the structural similarities between the two. This result underscores the efficiency of Hi-GMC's graph structure in processing and learning from feature data.

Cold Start Test

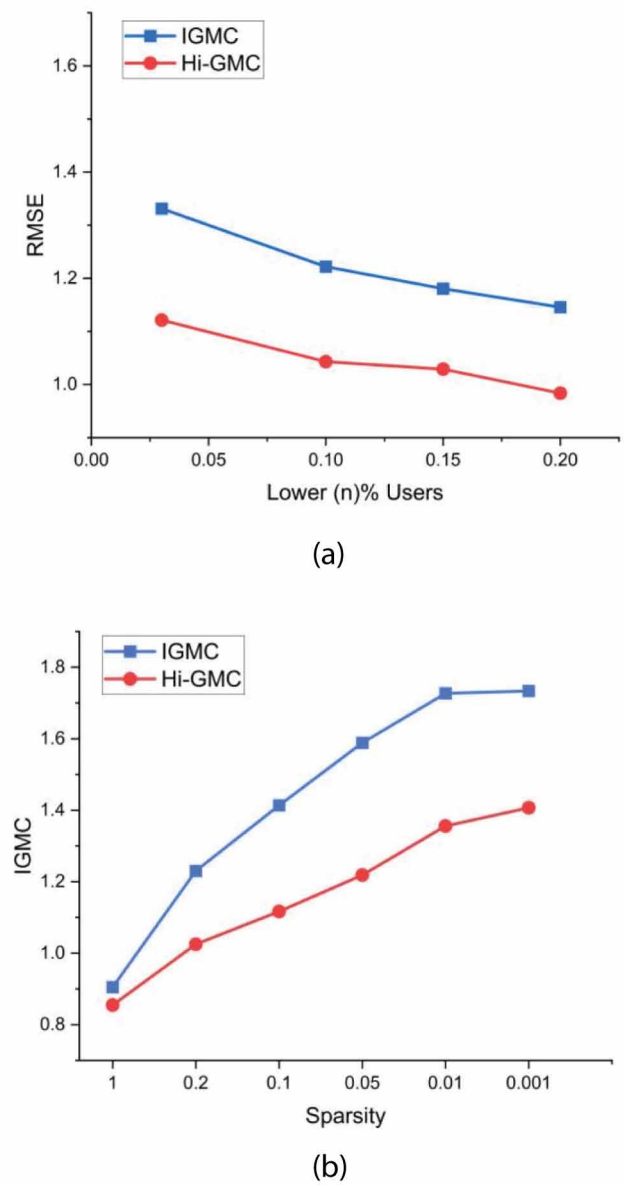
We carried out cold-start tests to evaluate Hi-GMC against IGMC, focusing on the challenges posed by the sparsity of interactions among sub-users with low activity. IGMC, a model that learns from graph patterns, does not address cold-start scenarios (Zhang & Chen, 2020). However, Hi-GMC performed better in cold-start situations. Our experiment was designed on the premise that no preference information is available for the target user within the MovieLens 100K dataset. Although learning

patterns from graphs is generally less effective for cold starts (Zhang & Chen, 2020), we showed that the inclusion of feature nodes in Hi-GMC could offer relative improvements (Prando et al., 2017).

First, we assessed the performance of the bottom 3 ~ 20% of users based on their interaction frequency. In the MovieLens 100K dataset, the minimum interaction count is 20, with 32 users (approximately 3%) having exactly 20 interactions. Figure 6(a) illustrates that the performance disparity between IGMC and Hi-GMC widens as the number of interactions decreases, approaching a cold-start scenario.

For the sparsity test, we generated a sparse graph by randomly eliminating edges. The performance impact was then evaluated across varying levels of sparsity, progressively set at 0.2, 0.1, 0.05,

Figure 6. (a) RMSE results for various ratios of lower users and (b) RMSE Results for various sparsity ratios



0.01, and 0.001. As depicted in Figure 6(b), the performance gap between IGMC and Hi-GMC widened in correlation with increased sparsity. These experiments demonstrate Hi-GMC's superior recommendation capability in sparse contexts compared to IGMC, due to its ability to leverage side information to compensate for the lack of interactions. This advantage is achieved by the relational benefits garnered from structuring the content information into the graph.

CONCLUSION

In this study, we introduced Hi-GMC, a model that encapsulates side information within nodes and utilizes inductive training. Despite sharing a structural similarity with the IGMC model, Hi-GMC demonstrates markedly enhanced performance. It outperforms both transductive models and other models that incorporate content information. Notably, Hi-GMC also exhibits superior handling of the cold-start problem in comparison to IGMC. A significant benefit of our approach is the ability to train feature nodes while preserving inductive properties, eliminating the need for retraining with the introduction of new feature nodes. This method, pioneered by Hi-GMC, holds potential applicability across various graph-based recommendation models.

Hi-GMC distinguishes itself by not merely treating side information as vectors but integrating it directly into the graph structure. This novel approach is expected to surpass traditional methods that are reliant on feature vectors. Furthermore, the inductive feature training opens avenues for transferring learning across distinctly different datasets, an aspect we aim to explore in future research. We hope Hi-GMC sets a new precedent for incorporating feature information into graph-based recommendation systems.

CONFLICTS OF INTEREST

The authors declare that there is no competing interest for this work.

FUNDING

This research was supported by the Culture, Sports and Tourism R&D Program through the Korea Creative Content Agency grant funded by the Ministry of Culture, Sports and Tourism in 2022 (Project Name: Development of Software Copyright Application Technology for Fair Trade and Distribution; Project Number: R2022020041) and by the Sookmyung Women's University Research Grants No. 1-1703-2041.

PROCESS DATES

Received: 12/7/2023, Revision: 3/25/2024, Accepted: 4/11/2024

REFERENCES

Candès, E. J., & Recht, B. (2009). Exact matrix completion via convex optimization. *Foundations of Computational Mathematics*, *9*(6), 717–772. doi:10.1007/s10208-009-9045-5

Cui, Y., Duan, Y., Zhang, Y., & Pan, L. (2023). A cross-domain recommender system for literary books using multi-head self-attention interaction and knowledge transfer learning. [IJDWM]. *International Journal of Data Warehousing and Mining*, *19*(1), 1–22. doi:10.4018/IJDWM.334122

Deldjoo, Y., Jannach, D., Bellogin, A., Difonzo, A., & Zanzonelli, D. (2024). Fairness in recommender systems: Research landscape and future directions. *User Modeling and User-Adapted Interaction*, *34*(1), 59–108. doi:10.1007/s11257-023-09364-z

Giap, C. N., Son, N. N., Giang, N. L., Chau, H. T. M., Tuan, T. M., & Son, L. H. (2022). A new approach for fairness increment of consensus-driven group recommender systems based on Choquet integral. [IJDWM]. *International Journal of Data Warehousing and Mining*, *18*(1), 1–22. doi:10.4018/IJDWM.290891

Grover, A., & Leskovec, J. (2016). Node2vec: Scalable feature learning for networks. In *KDD '16: Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 855–864). Association for Computing Machinery. doi:10.1145/2939672.2939754

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

Lam, X. N., Vu, T., Le, T. D., & Duong, A. D. (2008). Addressing cold-start problem in recommendation systems. In *ICUIMC '08: Proceedings of the 2nd international conference on ubiquitous information management and communication* (pp. 208–211). Association for Computing Machinery. doi:10.1145/1352793.1352837

Liu, S., Ounis, I., Macdonald, C., & Meng, Z. (2020). A heterogeneous graph neural model for cold-start recommendation. In *SIGIR '20: Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval* (pp. 2029–2032). Association for Computing Machinery. doi:10.1145/3397271.3401252

Lops, P., De Gemmis, M., & Semeraro, G. (2011). Content-based recommender systems: State of the art and trends. In F. Ricci, L. Rokach, B. Shapira, & P. Kantor (Eds.), *Recommender systems handbook* (pp. 73–105). Springer. doi:10.1007/978-0-387-85820-3_3

Maksai, A., Garcin, F., & Faltings, B. (2015). Predicting online performance of news recommender systems through richer evaluation metrics. In *RecSys '15: Proceedings of the 9th ACM conference on recommender systems* (pp. 179–186). Association for Computing Machinery. doi:10.1145/2792838.2800184

Mendes-Moreira, J., Soares, C., Jorge, A. M., & Sousa, J. F. D. (2012). Ensemble approaches for regression: A survey. *ACM Computing Surveys*, *45*(1), 1–40. doi:10.1145/2379776.2379786

Michalski, R. S. (1983). A theory and methodology of inductive learning. *Artificial Intelligence*, *20*(2), 111–161. doi:10.1016/0004-3702(83)90016-4

Miller, B. N., Albert, I., Lam, S. K., Konstan, J. A., & Riedl, J. (2003). MovieLens unplugged: Experiences with an occasionally connected recommender system. In *IUL '03: Proceedings of the 8th international conference on intelligent user interfaces* (pp. 263–266). Association for Computing Machinery. doi:10.1145/604045.604094

Monti, F., Bronstein, M., & Bresson, X. (2017). Geometric matrix completion with recurrent multi-graph neural networks. In *NIPS '17: Proceedings of the 31st international conference on neural information processing systems* (pp. 3700–3710). Curran Associates, Inc.

Prando, A. V., Contratres, F. G., de Souza, S. N. A., & de Souza, L. S. (2017). Content-based recommender system using social networks for cold-start users. In A. Fred & J. Filipe (Eds.), *Proceedings of the 9th International conference on knowledge discovery and information retrieval: Vol. OIC3K* (pp. 181–189). SciTePress. doi:10.5220/0006496301810189

Rashed, A., Grabocka, J., & Schmidt-Thieme, L. (2019). Attribute-aware nonlinear co-embeddings of graph features. In *Proceedings of the 13th ACM conference on recommender systems* (pp. 314–321). Association for Computing Machinery. doi:10.1145/3298689.3346999

Ying, R., He, R., Chen, K., Eksombatchai, P., Hamilton, W. L., & Leskovec, J. (2018). Graph convolutional neural networks for web-scale recommender systems. In *KDD '18: Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining* (pp. 974–983). Association for Computing Machinery. doi:10.1145/3219819.3219890

Zamanzadeh Darban, Z., & Valipour, M. H. (2022). GHRS: Graph-based hybrid recommendation system with application to movie recommendation. *Expert Systems with Applications*, *200*, 116850. doi:10.1016/j.eswa.2022.116850

Zhang, M., Cui, Z., Neumann, M., & Chen, Y. (2018, February). An end-to-end deep learning architecture for graph classification. *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*. AAAI. doi:10.1609/aaai.v32i1.11782

Zhao, Z., Zhang, X., Zhou, H., Li, C., Gong, M., & Wang, Y. (2020). HetNErec: Heterogeneous network embedding based recommendation. *Knowledge-Based Systems*, *204*, 106218. doi:10.1016/j.knosys.2020.106218

Jayun Yong is a software engineer at LINE PLUS. She graduated in IT engineering from Sookmyung Women's University. She received a master's degree in AI from Sookmyung Women's University.

Chulyun Kim received the B.S. degree in computer engineering, the M.S. degree in cognitive science, and the Ph.D. degree in electrical engineering and computer science from Seoul National University, South Korea, in 1996, 1998, and 2010, respectively. He received the Microsoft Research Asia Fellowship in 2004. He is currently an Associate Professor at Sookmyung Women's University, South Korea. Before that, he was an Assistant Professor at Gachon University, the Chief Technical Officer at NTROS, Ltd., South Korea, and a Research Assistant at the National Center for Supercomputing Applications, Urbana, IL, USA. He has been working in the area of data engineering focusing on machine learning, data mining, big data analysis, databases, artificial intelligence and interdisciplinary informatics. His writings have appeared in many professional conferences and journals, including VLDB and IEEE publications. He previously served as a Chair/PC member for ICDE, PAKDD, IEEE BigComputing, and WWW conferences.