Group Intelligence Recommendation System based on Knowledge Graph and Fusion Recommendation Model

Chengning Huang[1*], Bo Jing[2], Lili Jiang[3], Yuquan Zhu[4]
School of Computer and Communication Engineering, Nanjing Tech University Pujiang Institute, Nanjing, 211222, China[1, 3]
School of Computer Science, Nanjing Audit University, Nanjing, 211815, China[2]
School of Computer Science and Communication Engineering, Jiangsu University, Zhenjiang, 212013, China[4]

***Abstract*—The challenge of how to further improve the accuracy of the system's recommendations in a data-limited environment is crucial as the use of group intelligence recommendation systems in everyday life increases. Through the fusion of different types of auxiliary information, this study develops a multi-feature fusion model based on the conventional recommendation model by introducing knowledge graphs. It also considers the homogeneity of push results caused by graph convolutional network smoothing when using knowledge graphs, and designs a fusion label propagation algorithm and graph convolution. The multi-feature fusion model had a maximum hit rate of over 80% and a normalised discount gain of up to 43% running time much lower than the conventional graph convolution recommendation model in the representation dimension interval [2, 32], while the fusion label propagation algorithm and graph convolution network model maintained a hit rate and normalised discount gain higher than the conventional model by 2 to 1 under 10 consecutive epochs. With a hit rate and normalised discount gain 2 to 10 percentage points higher than the conventional model, the coverage rate increased to 49.8%. This study is useful for research on group intelligence recommendation systems and can serve as a technical guide for improving the ability of group intelligence systems to make recommendations quickly.**

***Keywords*—Knowledge graphs; recommendation system; graph convolutional networks; label propagation algorithms**

I. Introduction

Group Intelligence Recommendation (GIR) System research is expanding along with the field of group intelligence technology. The common GIR systems predict the user's past choice data using neural network-like algorithms to create suggestions [1]. There is a pressing need to lessen the reliance of the recommendation model (RM) on users' previous data as standard recommendation systems have a tendency to produce highly biassed outcomes when data information is limited [2]. Since in the actual recommendation process, in addition to information about the interaction between users and items, there is also information about user profiles, items, some relevant environmental conditions, etc., knowledge graphs (KGs) containing a wide variety of information data have been noticed. Multiple pieces of information can be combined thanks to the properties of KGs, which also improve the scalability of the recommendation system (RS) [3]. This is because, as a directed heterogeneous graph, KG uses edges to represent relationships between entities and nodes to represent entities that can represent both users and items. As a result, interactions between users and other users as well as interactions between users and items can be fused into the graph as auxiliary information, which can be used to make up for the information deficit brought on by a data-scarce environment [4]. In general, conventional suggestion models possess certain limitations whereby they may encounter the issue of data sparsity. This refers to a scenario in which there is minimal interaction data between users and items. As a result, the accurate modelling of the relationship between users and items is challenging and may impede the correctness and customisation of suggestion outcomes. When the recommendation system begins operating, it lacks adequate user behavior data or item attribute information, over-relies on users' historical behavior data, and disregards their present interests and requirements. Therefore, it struggles to offer dependable personalized recommendations for new users or items. To design a multi-feature fusion model based on the traditional RM, this study attempts to improve the traditional RM. It also attempts to introduce KG while taking into account the problems with graph convolutional smoothing and homogenization of recommendation results that are easily encountered when applying KG. Finally, it attempts to introduce Label propagation algorithms (LPA) to Graph convolutional networks (GCN) in order to further optimise the multi-feature model. This research content has the potential to advance the field of recommendation systems and enhance the quality of recommendation services for practical applications.

The study is broken up into six sections: The second section gives a summary of the most recent research findings; the third section describes the study's methodology and design elements; the fourth section presents the experimental findings and an analysis based on those findings; and the fifth section summarises the study's findings and the prospect is given in sixth section.

II. Related Works

People use GIR systems frequently, and with the growth of the web sector, customised RS has become increasingly important. According to Zhan et al, current recommendation models (RMs) primarily consider item compatibility modelling and do not consider user profiles, resulting in a

system where knowledge is pushed in a one-dimensional way without linking to user preferences. By adding attention to attribute-aware KG, Zhan et al. subsequently created an association between users and things, and created user-relationship-aware attention layers and goal-aware attention layers to capture user preferences. The results demonstrate the superiority of the model over other models for capturing user preferences [5]. Although Chen et al. argue that the current usage of GCNRM typically involves recursive aggregation with neighbouring nodes and their subsets, there is uncertainty in determining whether said neighbours can provide vital information after graph convolution. The introduction of KG in GCNRM is indeed beneficial in handling diverse multi-information tasks. Chen et al. proposed the Neighbour Enhanced Graph Convolutional Network (NEGCN) to enhance graph refinement process based on GCNRM and designed the neighbour evaluation method for critical information assessment. NEGCN demonstrated significantly improved model performance compared to the traditional graph convolutional RM [6]. Jiang et al. claim that the current RS approach to information exploitation is still limited and often only considers neighbourhood-specific information. To improve the conventional RS recommendation model, Jiang et al. propose a social aggregation neural network model based on attention mechanisms (AM). The model enables optimal user model embedding by propagating global social influence and capturing heterogeneous influences through AM. According to the results, using multi-layered perception to simulate the interaction between users and things is more flexible and successful than using conventional linear interaction algorithms to produce recommendation results [7].

Sang proposes a new knowledge graph-enhanced neural collaborative RM that can operate on information aggregation from multi-hop neighbours, and can use AM to grade the importance of relationships, as well as to model users and items in the embedding by modelling them in the embedding dimensional connections. On the other hand, the use of KGs in RM has always been hampered by the difficulty of modelling higher-order connectivity in large KGs using traditional models. The results showed that the model somewhat reduced the challenge of applying KG to RM [8]. Zhang et al. proposed a new knowledge-aware representation of the Graph Convolutional Recommendation Network model, arguing that in real recommendation environments, data is often sparsely distributed and the use of neighbourhood information alone is not sufficient to support accurate recommendation prediction results. The model can fuse item information through the propagation of links between nearby nodes in the KG and quickly capture correlations between people and items. This allows the model to predict likely user choices over time. This model, which creates user profiles by establishing neighbourhood associations between people and user objects by sorting features with high similarity between different users, has been shown to significantly outperform classical RM on a large dataset [9]. Graph neural networks, on the other hand, are favourable in dealing with complicated transitions between entity interactions in the environment of limited information input, according to Gwadabe et al. As a result, Gwadabe et al. proposed a graph neural network-based RM that may use graph neural networks to learn the ordered interactions first, followed by the unordered interactions, in the current session. Numerous studies have revealed that this model performs noticeably better than other models in addressing unpredictable user behaviour in a data-limited environment [10].

In summary, experts have studied both the improvement of conventional RM and the implementation of KG, but have focused on optimising RM to increase the accuracy of push results to users, relying on the substantial information compensation found in KG itself. In reality, another problem with KG in RM is that a significant proportion of graph nodes are susceptible to convolutional smoothing, which can lead to homogenization of push results. It is still important to conduct research on how to use the entities' own information as much as possible in a data coefficient environment, and how to solve problems with the use of KG.

III. Design of the GIR Model for the Integration of KG

The principle of traditional RM is to get user behaviour prediction by analysing historical data of users, but the results derived from this model will be more deviant in a data sparse environment, so KG, which can fuse various kinds of auxiliary information, needs to be introduced to make up for the lack of interaction information. However, the KG itself is large in scale and is prone to the problem of excessive smoothing of a large number of graph nodes in the GCN, which can lead to homogenisation of the model's recommendations to users [11]. To address these problems, a multi-feature fusion model based on KG and AM and a model that fuses LPA and GCN are proposed.

A. Design of KGARA Model for Multi-Feature Fusion based on KG and AM

To address the limitations imposed on RM by data sparse environments, this research proposes the KGARA model. The core principle of this model is to incorporate relationship-aware structures to enrich the preference relationships between users and objects, and users and users. Based on joint AM, the model uses KG to fuse adjacent objects with different relationship types to obtain rich feature information. A graphical neural network is also used as a deep learning algorithm in RS.

The semantic information is first modelled using the representation-based KG recommendation algorithm, which vector embeds the input user features to obtain the initial representation, and then uses AM to complete the portrayal of user preferences. In this process, the representation is generated by the KG encoding operation on its entity, which requires the use of Knowledge Graph Embedding (KGE) [12]. KG, the interaction matrix of the relationship between the user and the item, and other information are fused to generate the user representation u, and the item representation v, and then they are inner-producted to obtain the probability of the user choosing the item, which is described by (1) is described.

$$\hat{y}_{uv} = u^T v \tag{1}$$

T in (1) denotes the inner product operation and $\hat{y}_{uv}$ is the user selection probability. Fig. 1 shows the structure of the model. From left to right, the first layer is an embedding vector layer, from which the unique hot codes of user, relationship and item are input and formed into an initial representation by vector embedding operation; then the unweighted KG formed by the initial representation is transformed into a weighted KG by the attention layer and stored in the adjacency matrix; the next layer is the feature propagation layer, where the item representation is trained by GCN and fused with the domain representation using an aggregator; finally, the probability of the user selecting the item is obtained by inner product operation on the obtained item representation e_v.

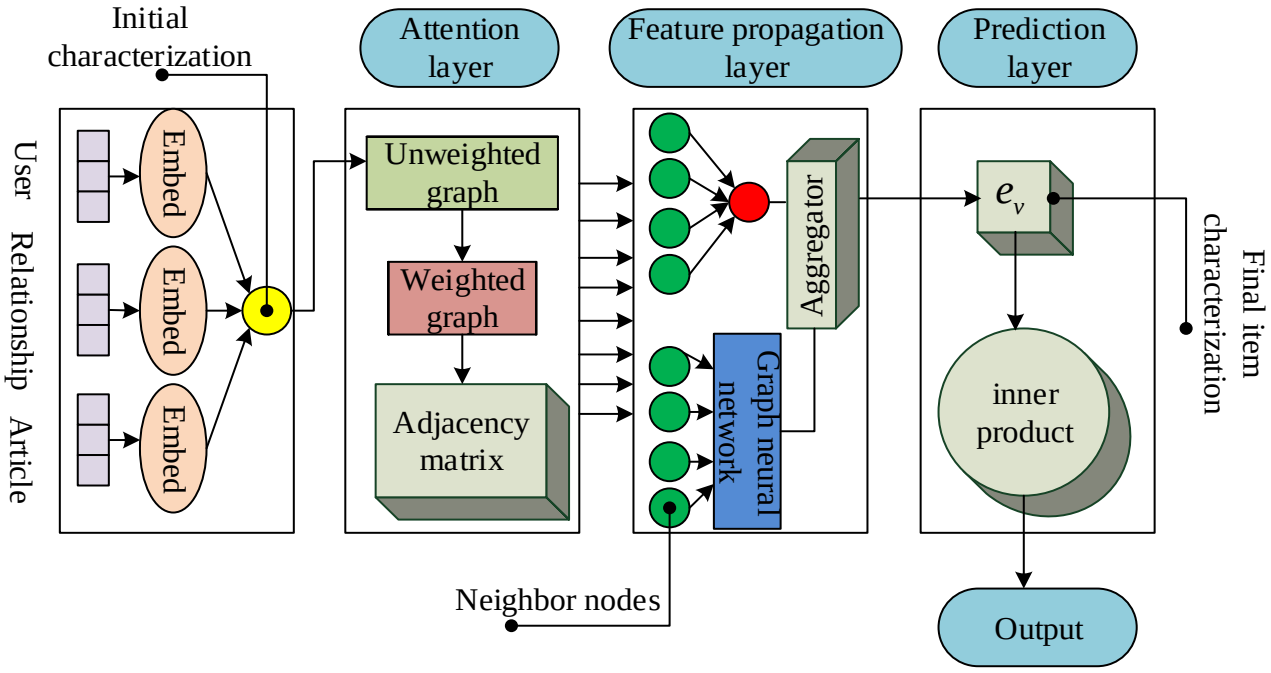


Fig. 1. Structure of the KGARA model.

In the relational attention network layer, the degree of importance between the relational representation and the user representation is obtained by doing an inner product operation on the two, described by (2).

$$c_u^r = e_u^T e_r \tag{2}$$

where e_r is the relational representation, e_u is the user representation and T denotes the inner product operation done on both. However, in the initial KG the edges can only describe the relationship but not the weight values, so the initial unweighted KG needs to be transformed into a weighted graph by (3).

$$y_{uv} \begin{cases} 1, \text{If } u, v \text{ interact;} \\ 0, \text{If } u, v \text{ do not interact;} \end{cases} \tag{3}$$

y_{uv} in (3) is a parameter in the user-item interaction matrix. The adjacency matrix resulting from the transformation into a weighted graph is denoted A_u and the relationship weights of entity i and entity j in row i and column j of this matrix are expressed in (4).

$$A_u^{i,j} = c_u^{r_{i,j}} \tag{4}$$

$c_u^{r_{i,j}}$ in (4) represents the entity relationships in the unweighted graph. Since the weighted graph itself can lead to an excessive computational burden, the KGARA model prioritises the nodes by attention weight values, which in turn yields the important nodes. The important nodes are weighted and summed by (5), which in turn gives the target entity representation.

$$e_{N(v)}^u = \sum_{\alpha \in N(v)} \hat{c}_u^r e_\alpha \tag{5}$$

The set of target node v and neighbouring nodes in equation (5) is denoted by $N(v)$, α is a parameter taken from this set, and the normalised relational attention score is $\hat{c}_u^r$. This indicates that nodes with a high relational attention score will be filtered out, as defined by (6) for them.

$$\hat{c}_u^r = \frac{\exp(c_u^r)}{\sum_{e \in N(v)} \exp(c_u^r)} \tag{6}$$

Next, at the feature propagation layer, the relational attention information is fused on the basis of the obtained weighted graph, and after all the rows in the matrix have been calculated, the individual entity representations are obtained,

denoted as h_k. The process is represented by equation (7).

$$\begin{cases} H_1 = \sigma(D_u^{-\frac{1}{2}} A_u D_u^{-\frac{1}{2}} H_0 W_0) \\ H_2 = \sigma(D_u^{-\frac{1}{2}} A_u D_u^{-\frac{1}{2}} H_1 W_1) \\ ..... \\ H_k = \sigma(D_u^{-\frac{1}{2}} A_u D_u^{-\frac{1}{2}} H_{k-1} W_{k-1}) \end{cases} \quad (7)$$

In (7), the representation matrix in row k is represented by A_u, W_k is the parameter matrix in row k, and σ is the activation function. where there is a logarithmic matrix relationship between $D_u^{-\frac{1}{2}}$ and A_u, described by (8).

$$D_u^{ii} = \sum_j A_u^{ij} \quad (8)$$

The item representation and the domain representation are aggregated by (8) to obtain the final item representation e_v. The final step of the model performs an inner product operation on the user representation and the item representation obtained from (8) to finally arrive at a probability value for that user to accept the recommended item, a process described by (9).

$$\hat{y}_{uv} = e_u^T e_v \quad (9)$$

B. Design of A GCNLP Model Incorporating LPA and GCN

When applied to RM, the KG technique is computationally intensive due to the presence of tens of billions of edges and billions of nodes [13]. In GCNs, the large number of nodes can also cause the problem of smooth graph convolution and thus homogeneous recommendation results [14]. Therefore, in this study, the KGARA model is used to adjust the weight value of the edges of the GCN, and the attention network is used to filter the user's maximum weight on the items.

The structure of the GCNLP model is shown in Fig. 2. The structure of the feature propagation layer is improved from the structure of the KGARA model in Fig. 1, and the rest of the embedding vector layer, attention layer, and prediction output layer are structured in the same way as the KGARA model. In the improved feature propagation layer, GCN operates on neighbouring nodes to derive item representations, while LPA is introduced to adjust the weight values for graph edges.

Specifically in the representation propagation layer, the model uses the GCN to make basic predictions and then uses the LPA to assist in adjusting the edge weights. As a multilayer feedforward neural network, the GCN is able to perform transformation and propagation operations on nodes in the graph, as described by (10).

$$P^{(k+1)} = \sigma(D^{-1} A P^k W^{(k)}) \quad (10)$$

In (9), σ is the activation function, σ is the parameter matrix of layer k, and the resulting $P^{(k+1)}$ is the node representation of layer k. Present at nodes σ and v_j, the GCN to v_i update process is described by (11).

$$P_i^{(k+1)} = \sigma(\sum_{v_j \in N(v_i)} \alpha_{ij} p_j^{(k)} W^{(k)}) \quad (11)$$

In (11), $p_j^{(k)}$ is the k th level node representation of the target node, α_{ij} is the value of the i th row and j th column in the adjacency matrix, i.e. the weight value between nodes, and the i th node in the set of neighbours is represented by $N(v_i)$,. After the domain nodes of the target node are aggregated to obtain the domain representation of the target node, they are then transformed into the next-order representation $P_i^{(k+1)}$ of the target node, and (12) and Equation (13) provide a description of these two steps.

$$s_i^{(k)} = \sum_{v_j \in N(v_i)} \alpha_{ij} p_j^{(k)} \quad (12)$$

(13) σ is domain representation.

$$P_i^{(k+1)} = \sigma(s_i^{(k)} W^{(k)}) \quad (13)$$

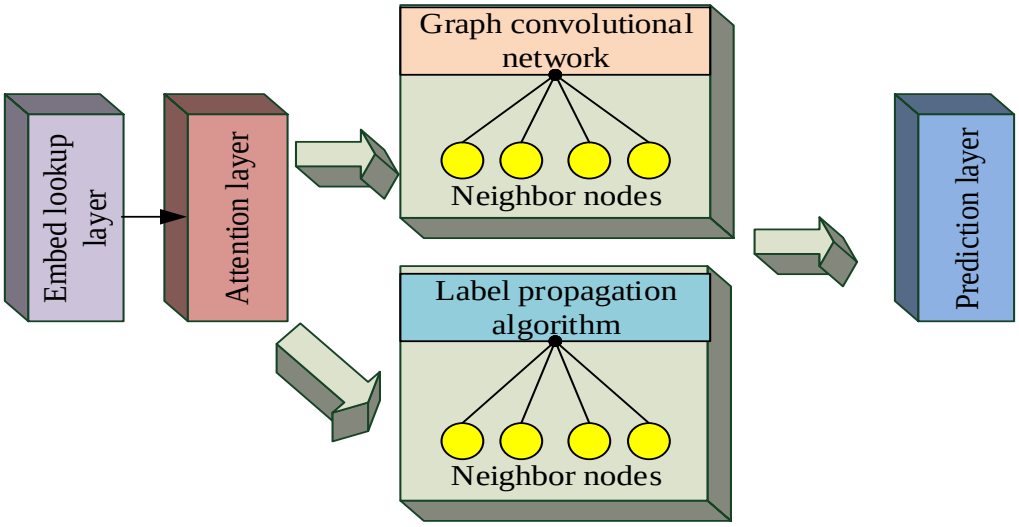


Fig. 2. GCNLP model structure.

The distance between v_i and v_j is defined as $Q(p^{(k)})$, and the distance between its domain representation s_i and s_j is defined as $Q(s^{(k)})$. After the aggregation operation on nodes v_i and v_j, $Q(s^{(k)})$ will be smaller than $Q(p^{(k)})$ and similar nodes will be grouped into the same class, i.e. in the GCNLP model GCN places items of user attention in the same class to improve recommendation performance.

The propagation process of LPA at each level, i.e. according to the normalised edge weights, all nodes are subjected to label propagation by their neighbouring nodes and all nodes that already have labels are subjected to an initialisation operation by themselves to prevent the labels from disappearing [15]. The simulated label propagation process is shown in Fig. 3, assuming that the propagation process is performed only three times, with the goal of propagating from node a to node b. Red dots indicate with labels and colourless dots indicate without labels. In the first execution (green line), node a passes the label to its neighbours node 3 and node 1, but node 1 already has a label, so the propagation path is inaccessible; in the second execution, node 3, which has already been propagated with a label, passes the label to its neighbours node b and node 4 (yellow line) and the first execution is completed, so node a initialises and propagates the label again (purple line); in the third execution, the nodes that already had labels in the previous execution also perform the initialization operation and continue to propagate the labels (blue line). Finally, the LPA must find all paths from node a to node b that are no longer than 3, as expressed by (14).

$$x_i^{\infty} = \sum_{j \in N(i)} a_{ij} x_j^{\infty} \tag{14}$$

In (13), x_i^{∞} denotes the node in the label matrix, x_i^{∞} is the value of column j of row i in the adjacency matrix, and $N(i)$ is the set of neighbours of the node.

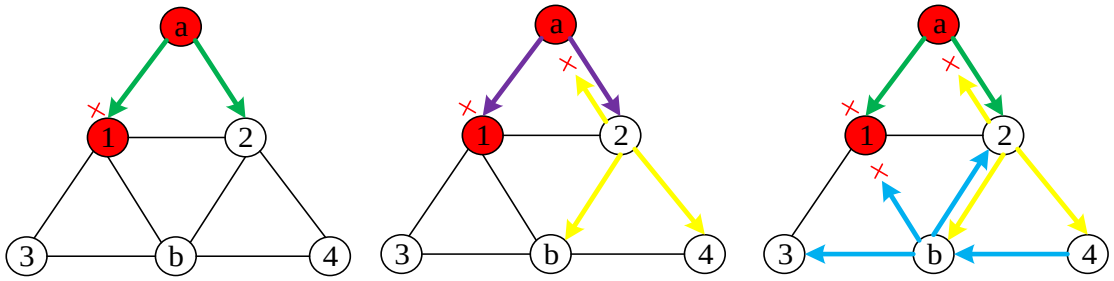


Fig. 3. Label propagation process.

The evaluation indicators introduced in this study were Hit Rate (HR), Normalized Discounted Cumulative Gain (NDCG) and Coverage, described by (15), (16) and (17) [16].

$$HR@N = \frac{Number\ of\ Hits@N}{|GT|} \tag{15}$$

$Number\ of\ Hits@N$ in equation (15) is the number of positive samples in the item sorting list in the recommendation task, and $|GT|$ is the number of total samples in the test set.

$$NDCG@N = \frac{\sum_{u \in U} NDCG_u@N}{|U|} \tag{16}$$

Where $|U|$ represents the number of users and $\sum_{u \in U} NDCG_u@N$ is the process of accumulating the normalised discounted gain in the test, resulting in a mean value of $NDCG@N$ [17].

$$Coverage@N = \frac{|U_{u \in U} R(u)|}{|I|} \tag{17}$$

In equation (17) $|I|$ is the set of items, U is the set of users, and RS is the list of items recommended by users denoted by $R(u)$.

IV. Experimental Results and Analysis

Performance testing experiments on the proposed KGARA model and the GCNLP model are conducted in the Book-Crossing environment, a book dataset, Movielens-1M, a film dataset, and Last FM, a music dataset [18]. For the model, the higher the value of HR and NDCG, the higher the quality. The Generalised Matrix Factorisation (GMF), Neural Matrix Factorisation (NeuMF) and Long Short-Term Memory R-GCN (LRGCN) were selected for the KGARA performance detection experiments [19]; the LRGCN, Ripple Net and Neural Graph Collaborative Filtering (NGCF) were selected for the GCNLP performance testing experiments [20]. An early termination strategy is implemented if HR@20 and NDCG@20 do not increase for 20 consecutive epochs on the test set, where an epoch is the process of training once using all samples in the training set.

A. Experimental Results and Analysis of the KGARA Model

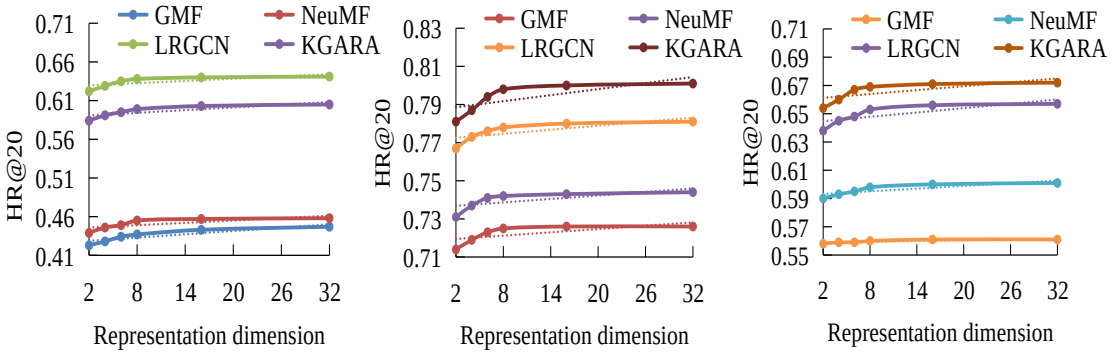
First, parametric experiments were conducted to explore the effect of different representational dimensions on the model, followed by a comparison of the effect of the data sparse environment on the model performance. The parameter settings for the datasets in the experiments are given in Table I.

TABLE I. DATASET PARAMETER SETTINGS

Aggregator	BI-Interaction			
Data set Configured parameters	*Learning rate*	*Batch size*	*Number of neighbors*	*Jump count*
Book-Crossing	0.0002	32	8	1
Movielens-1M	0.02	2048	4	2
LastFM	0.0004	128	8	1

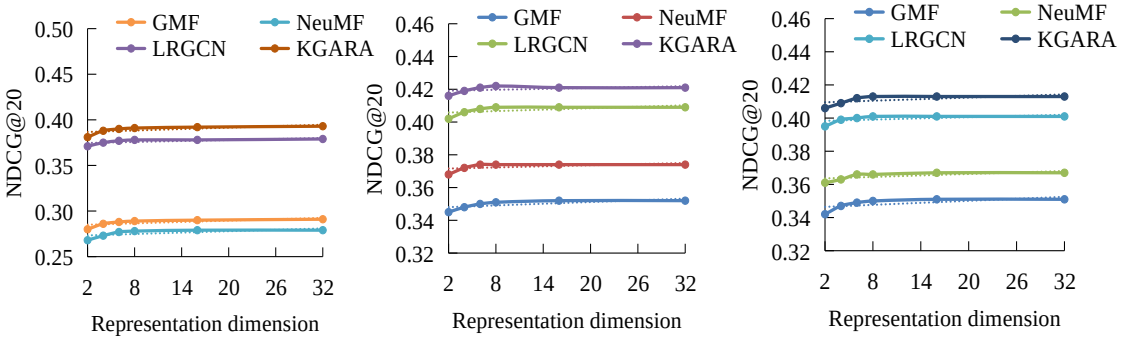
Fig. 4 shows the hit rate variation of the GMF, NeuMF, LRGCN and KGARA models on the three datasets under the representation dimension interval [2,32]. As can be seen from the figure, the overall hit rate of each model tends to increase as the representation dimension increases. However, compared to the Book-Crossing dataset and the Movielens-1M dataset, the increase for all models on the LastFM dataset varies between 1 and 3 percentage points, due to the fact that this dataset is less informative and can perform almost with sufficient information at lower representation dimensions. And in each dataset, compared to other models, the hit rate of KGARA proposed in this study is on average the highest, up to over 80% in the Movielens-1M dataset, which is on average 5 to 10 percentage points higher than the traditional graph neural network-based GMF and NeuMF; but in the Book-Crossing dataset, the average hit rate of LRGCN is slightly higher than that of KGARA, which is also due to the simpler structural design of LRGCN than KGARA.

Fig. 5 shows the changes in the normalised discount gain of the GMF, NeuMF, LRGCN and KGARA models as the representation dimension increases on the three datasets. As can be seen in Fig. 5, the NDCG values of all models increase as the representation dimension increases, and although the increase in the NDCG metric is smaller, KGARA performs best when comparing both the initial NDCG values and the NDCG values at representation dimension 32: the initial NDCG value on the Movielens-1M dataset is around 42%, and as the representation dimension increases to 32, its NDCG value reaches around 43%. The average NDCG values of the KGARA model are much higher than those of the traditional graph convolution models GMF and NeuMF, ranging from 4 to 10 percentage points higher on average.



(a) The change in hit rate of different models in the Book-Crossing dataset (b) The change in hit rate of different models in the Movielens-1M dataset (c) The change in hit rate of different models in the LastFM dataset

Fig. 4. Changes in hit rates of various models on different datasets.



(a) The Change in HDCG of Different Models in the Book-Crossing Dataset (b) The Change in HDCG of Different Models in the Movielens-1M Dataset (c) The Change in HDCG of Different Models in the LastFM Dataset

Fig. 5. Normalized loss gain changes of each model on different datasets.

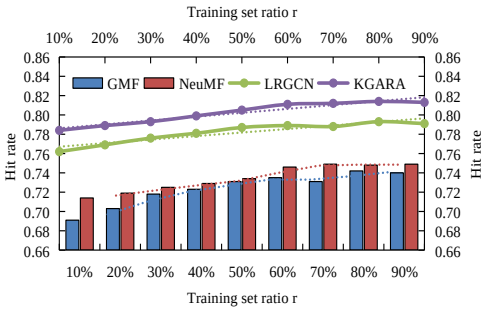


Fig. 6. Changes in hit rates of various models in data scarcity environments.

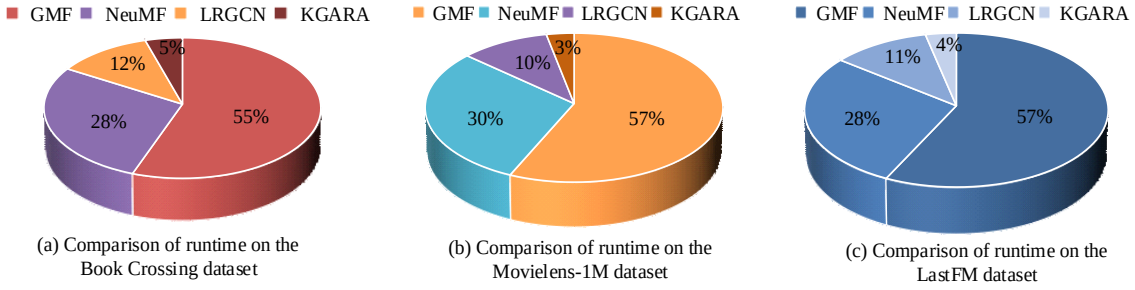
To test the performance of the KGARA model with the introduction of the KG feature fusion structure in a data sparse environment, the GMF, NeuMF, LRGCN and KGARA models were tested on the Movielens-1M dataset, and the hit rate variation of all models was compared by adjusting the proportion of the training set to the test set for this dataset. The experimental results are shown in Fig. 6, where r represents the proportion of the training set. As can be seen from Fig. 5, the hit rate of KGARA reached more than 78% when the proportion of the training set was only 10%, which was 2 to 10 percentage points higher than the other models, and as the proportion of the training set increased, the hit rate of KGARA also increased, stabilizing at 81.6% when the proportion of the training set reached 60%, with the highest hit rate reaching 81.8%. The other models basically stabilized after the training set percentage increased to 70%, and the final hit rate was still lower than KGARA by 2 to 8 percentage points.

Next, the runtimes of all models were examined on the three datasets at 1600MHz core frequency, RTX2080 graphics card and 8G video memory. Fig. 7 shows the results of the runtime comparison for all models. As can be seen from Fig. 7, although the runtime ratios of the models varied from dataset to dataset, the largest average runtime ratio was for the traditional graph convolution model GMF, which accounted for 56% of the four models; the model with the lowest runtime ratio was KGARA, which accounted for 4% on average, almost a tenth of the GMF runtime.

B. *Experimental Results and Analysis of the GCNLP Model Doing the Results Analysis*

In addition to testing the basic performance of GCNLP in different data dimensions, this study also needs to test whether GCNLP can improve the problem that GCN is prone to homogeneous recommendation results due to graph convolution smoothing, so 10 consecutive epochs were analyzed for the GCNLP model in terms of three metrics: hit rate, normalized discount gain and coverage.

Fig. 8 shows the hit rate variation of the four models GCNLP, LRGCN, Ripple Net and NGCF on different datasets in the environment with representation dimensions {2,4,6,8,16,32}. As can be seen from the figure, although the hit rate on the Book-Crossing dataset is on average 2 percentage points higher than that of the GCNLP model due to the simple structure of LRGCN, the GCNLP hit rate is 2 to 4 percentage points higher than the other models on all other datasets, with the GCNLP hit rate on the Movielens-1M dataset being on average the highest, averaging in the upper 65% range with a maximum of 66%.

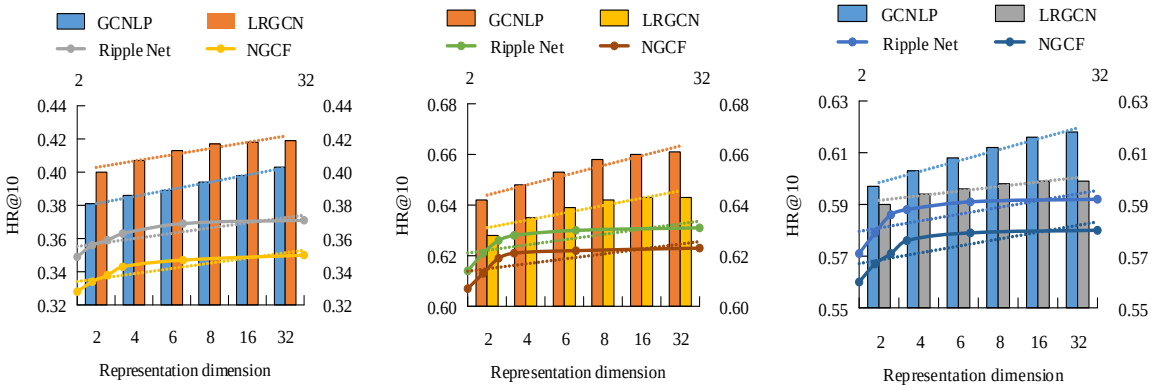


(a) Comparison of runtime on the Book Crossing dataset

(b) Comparison of runtime on the Movielens-1M dataset

(c) Comparison of runtime on the LastFM dataset

Fig. 7. The proportion of running time of each model on different datasets.

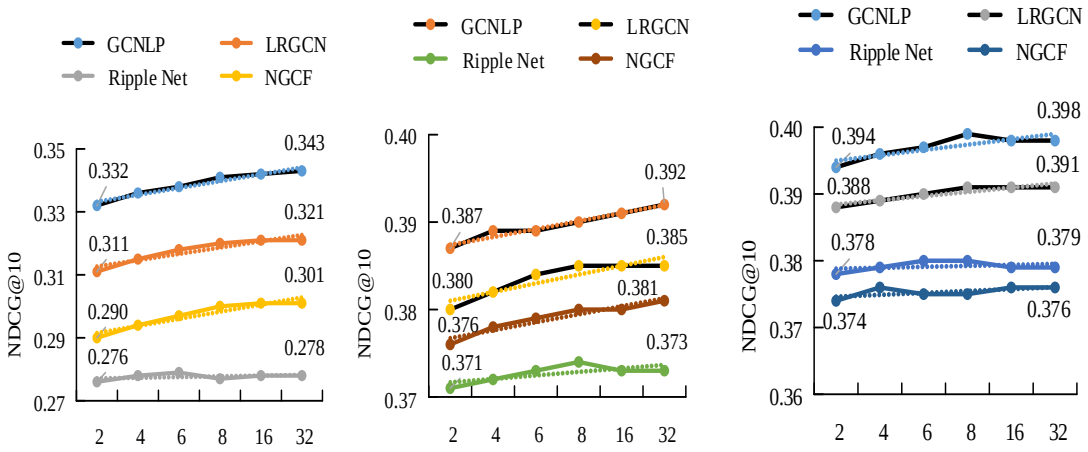


(a) The change in hit rate of different models in the Book-Crossing dataset

(b) The change in hit rate of different models in the Movielens-1M dataset

(c) The change in hit rate of different models in the LastFM dataset

Fig. 8. Changes in hit rates of various models on different datasets.



(a) The change in HDCG of different models in the Book-Crossing dataset

(b) The change in HDCG of different models in the Movielens-1M dataset

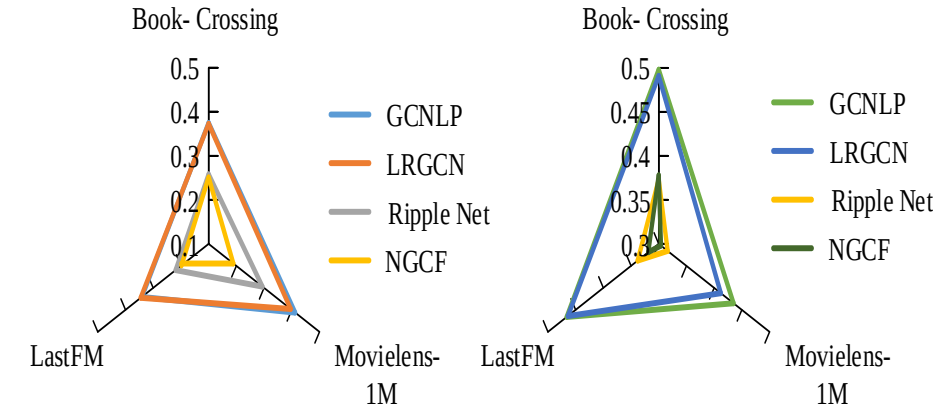
(c) The change in HDCG of different models in the LastFM dataset

Fig. 9. NDCG changes of each model on different datasets.

Fig. 9 shows the variation in normalised discount gain for each model on the different datasets, and it can be seen that the NDCG values for all four models increase as the characterisation dimension increases on all three datasets, but the NDCG values for GCNLP are higher than the initial and final NDCG values for the other models compared to the other models. GCNLP had the highest NDCG values overall on the LastFM dataset, averaging around 36.5% and up to 40% over the period, while being one to two percentage points higher than the other models.

For GCNLP improvements made on the basis of the KGARA model, the metric coverage (Coverage) can be targeted to detect GCNLP performance. The higher the value of $Coverage@N$, the higher the coverage of the model and the less likely the problem of homogeneous recommendation results will occur. Fig. 10 provides a comparison between the mean coverage of all models under the low representation dimension and the mean coverage under the high representation dimension based on three distinct datasets. As observed in Fig. 10, the coverage of each model increases as the representation dimension increases. The GCNLP and LRGCN show comparable coverage in the low representation dimension. However, in the high representation dimension, GCNLP exhibits a higher average coverage than all other models.

Table II displays comprehensive average coverage information for each model in both high and low representation dimensions. It is evident that, in the high representation dimension, the Book-Crossing dataset showed the highest coverage of GCNLP at 49.8%, while, in the low representation dimension, the Movielens-1M dataset presented the highest coverage of GCNLP at 41.2%. Overall, GCNLP's coverage was greater than the other models, indicating that the enhancements made to GCN using LPA improved the homogeneity of RM push results.



(a) Comparison of average coverage of different models under low representation dimensions

(b) Comparison of average coverage of different models under high representation dimensions

Fig. 10. Comparison of average coverage of various models.

TABLE II. DETAILED DATA ON AVERAGE COVERAGE OF EACH MODEL

nder low feature dimensions	Book-Crossing	Movielens-1M	LastFM
GCNLP	0.375	0.412	0.341
LRGCN	0.373	0.396	0.345
Ripple Net	0.258	0.294	0.219
NGCF	0.251	0.189	0.198
Under high feature dimensions	Book-Crossing	Movielens-1M	LastFM
GCNLP	0.498	0.435	0.466
LRGCN	0.491	0.412	0.464
Ripple Net	0.373	0.316	0.338
NGCF	0.378	0.304	0.319

V. CONCLUSION

The aim of this study was to address the problem of large deviations in recommendation results in traditional GIR models within a data sparse environment. To achieve this goal, the study developed a multi-feature fusion model, KGARA, and improved its GCN structure by combining with LPA to obtain the GCNLP model. This approach enabled a more accurate and reliable model, thus mitigating the issue mentioned above. The study also introduced KG and AM based on traditional RM to compensate for the lack of interaction information. The results of 20 consecutive epoch trials show that the KGARA model has a maximum hit rate of over 80% and a normalised discount gain of over 43% when the characterisation dimension interval [2,32] is shifted, which is higher than other models. Furthermore, the proportion of the Movielens-1M training set was altered from 10% to 90%, with a maximum hit rate of 81.8%, ultimately illustrating the KGARA model's superior performance compared to all other comparative models in the data-scarce setting. Owing to its efficient nature and the shortest running duration among its counterparts, the KGARA model proves to be the most effective. Based on the results from 10 consecutive epochs of experimentation on the GCNLP model, the hit rate and normalised discount gain surpass other comparison models when the representation dimension is altered within the range of [2,32]. Specifically, the hit rate reaches up to 66% and the normalised discount gain reaches up to 40%, both of which are higher than those of other models. Moreover, the GCNLP model attained an average coverage of 41.2%, which outstripped the conventional graph convolution model's average coverage by a significant margin. This suggests that the addition of LPA to the GCN structure was a fruitful improvement and could potentially resolve the problem of homogenization in push outcomes. The study successfully enhanced the classic push model; however, it only considered user preferences' constant conditions and omitted dynamic shifts in user preferences. This aspect warrants future investigation.

VI. PROSPECT

The KGARA and GCNLP models exhibit higher hit rates, normalized discount gains, and coverage performance when compared to other models. These outcomes suggest that the improved models are anticipated to outperform recommendation systems, enhancing the standard and personalization of recommendation outcomes. Such accomplishments can offer motivation and establish benchmarks for the betterment and refinement of recommendation system models. One potential avenue for future research could involve the development of recommendation algorithms that rely on temporal data. Such algorithms could utilize users' past behavior and historical data to predict their future interests. This could involve the application of techniques including time series analysis, sequence modeling, and deep learning.

ACKNOWLEDGMENT

The research is supported by: National Natural Science Foundation Program of China: Research on Distributed Progressive Classification Mining Method for Big Data Based on Reuse of Existing Knowledge(61702229); National Natural Science Foundation Program of Jiangsu: Key Technology Implementation of Smart Medical Platform in the Context of Big Data Cloud Computing(18KJD520001); Research Key Cultivation Project of Pujiang College, Nanjing University of Technology: Research on Multiple Personality Recommendation by Integrating Knowledge Graph and Attention Mechanism(njpj2022-1-07); Nanjing University of Technology Pujiang College Youth Teacher Development Fund (PJYQ03).

REFERENCES

[1] B. Hu, Y. Ye, Y. Zhong, J. Pan, and M. Hu, "TransMKR: Translation-based knowledge graph enhanced multi-task point-of-interest recommendation," Neurocomputing, vol. 474, no. 14, pp. 107-114, 2022.

[2] J. Chen, B. Li, J. Wang, Y. Zhao, L. Yao, and Y. Xiong, "Knowledge Graph Enhanced Third-Party Library Recommendation for Mobile Application Development," IEEE Access, vol. 8, pp. 42436-4244, 2020.

[3] Z. Hu, J. Wang, Y. Yan, P. Zhao, J. Chen, and J. Huang, "Neural graph personalized ranking for Top-N Recommendation," Knowledge-Based Systems, vol. 213, no. 8, pp. 106426.1-106426.9, 2020.

[4] Z.Y. Zhang, L. Zhang, D.Q. Yang, and L. Yang, "KRAN: Knowledge Refining Attention Network for Recommendation," ACM Transactions on Knowledge Discovery from Data (TKDD), vol. 16, no. 39, pp. 1-20, 2022.

[5] H. Zhan, J. Lin, K.E. Ak, B. Shi, L. Duan, and A.C. Kot, "A3-FKG: Attentive Attribute-Aware Fashion Knowledge Graph for Outfit Preference Prediction," IEEE Transactions on Multimedia, vol. 24, pp. 819-831, 2022.

[6] H. Chen, Z. Huang, Y. Xu, Z. Deng, F. Huang, P. He, and Z. Li, "Neighbor enhanced graph convolutional networks for node classification and recommendation," Knowledge-based systems, vol. 246, no. 21, pp. 108594.1-108594.101, 2022.

[7] N. Jiang, L. Gao, F. Duan, J. Wen, T. Wan, and H.L. Chen, "SAN: Attention-based social aggregation neural networks for recommendation system," International Journal of Intelligent Systems, vol. 37, no. 6, pp. 3373-3393, 2021.

[8] L. Sang, M. Xu, S. Qian, and X. Wu, "Knowledge graph enhanced neural collaborative recommendation," Expert Systems with Applications, vol. 164, no. 12, pp. 113992.1-113992.13, 2021.

[9] L. Zhang, Z. Kang, X. Sun, H. Sun, B. Zhang, and D. Pu, "KCRec: Knowledge-aware representation Graph Convolutional Network for Recommendation," Knowledge-Based Systems, vol. 230, no. 27, pp. 107399.1-107399.13, 2021.

[10] T.R. Gwadabe and Y. Liu, "Improving graph neural network for session-based recommendation system via non-sequential interactions," Neurocomputing, 2022, 468(Jan.11):111-122.

[11] J. Chen, J. Yu, W. Lu, Y. Qian, and P. Li, "IR-Rec: An Interpretive Rules-guided Recommendation over Knowledge Graph," Information Sciences, vol. 563, no. 10, pp. 326-341, 2021.

[12] Z. Yang and S. Dong, "HAGERec: Hierarchical Attention Graph Convolutional Network Incorporating Knowledge Graph for Explainable Recommendation," Knowledge-Based Systems, vol. 204, no. 27, pp. 106194.1-106194.11, 2020.

[13] L. Zhang, Z. Kang, X. Sun, H. Sun, B. Zhang, and D. Pu, "KCRec: Knowledge-aware representation Graph Convolutional Network for Recommendation," Knowledge-Based Systems, vol. 230, no. 27, pp. 107399.1-107399.13, 2021.

[14] H. Yang, H. He, W. Zhang, and Y. Bai, "MTGK: Multi-Source Cross-Network Node Classification via Transferable Graph Knowledge," Information Sciences, vol. 589, no. 1, pp. 395-415, 2022.

[15] A. Salamat, X. Luo, and A. Jafari, "HeteroGraphRec: A heterogeneous graph-based neural networks for social recommendations," Knowledge-Based Systems, vol. 217, no. 4, pp. 106817.1-106817.10, 2021.

[16] Y. Lin, B. Xu, J. Feng, H. Lin, and K. Xu, "Knowledge-enhanced recommendation using item embedding and path attention," Knowledge-Based Systems, 2021, 233(Dec.5):107484.1-107484.11.

[17] S. Tao, R. Qiu, Y. Ping, and H. Ma, "Multi-modal Knowledge-aware Reinforcement Learning Network for explainable recommendation," Knowledge-Based Systems, vol. 227, no. 5, pp. 107217.1-107217.11, 2021.

[18] S. Oslund, C. Washington, and A. So, et al., "Multiview Robust Adversarial Stickers for Arbitrary Objects in the Physical World," Journal of Computational and Cognitive Engineering, vol. 1, no. 4, pp.152-158, 2021.

[19] N. Nassar, A. Jafar, and Y. Rahhal, "A novel deep multi-criteria collaborative filtering model for recommendation system," Knowledge-Based Systems, vol. 187, pp. 104811.1-104811.7, 2020.

[20] S. Qianna, "Evaluation model of classroom teaching quality based on improved RVM algorithm and knowledge recommendation," Journal of Intelligent and Fuzzy Systems, vol. 40, no. 2, pp. 2457-2467, 2021.