A Course Recommendation Method Based on the Integration of Curriculum Knowledge Graph and Collaborative Filtering

Jingyi Hu
Big Data and Artificial Intelligence College
Anhui Xinhua University
Hefei, China
E-mail: hujingyi@axhu.edu.cn

Qingqing Wang
Big Data and Artificial Intelligence College
Anhui Xinhua University
Hefei, China
E-mail: 3119905948@qq.com

***Abstract*—To address the problems of data sparsity and cold start in collaborative filtering algorithms, this paper proposes an improved course recommendation method that integrates knowledge graphs and collaborative filtering. First, the RippleNet model is used to construct a knowledge graph based on course-attribute-relation triples and generate a recommendation list. Then, an item-based collaborative filtering algorithm utilizes users' historical interaction behavior to produce another recommendation list. Finally, a weighted linear method is employed to fuse the recommendation list generated by the RippleNet-based course knowledge graph and the one generated by collaborative filtering, resulting in the final course recommendation list. Experiments conducted on the public dataset MOOCCube demonstrate that the RippleNet-CF method improves precision, recall, and F1-score, while also effectively mitigating the issue of data sparsity.**

Keywords-Data Sparsity; Course Attributes; Knowledge Graph

I. INTRODUCTION

With the rapid development of information technology, there has been an explosion of data [1], and data mining technology has been widely applied in various fields such as education, communication and e-commerce [2]. In the field of education, how to recommend courses based on students' learning characteristics is a key focus of data mining [3]. In recommendation systems, domain-based recommendation is the most fundamental algorithm, which is generally divided into user-based collaborative filtering algorithms [4] and item-based collaborative filtering algorithms [5]. The user-based collaborative filtering algorithm recommends items to target users based on user similarity. When the target user has too few historical interactions with items, it cannot make accurate recommendations. This algorithm is more suitable for social recommendations such as news [6]. This algorithm is suitable for personalized user recommendations but also has the problem that too few interactions between users can lead to unsatisfactory recommendation results. In order to optimize the recommendation effect, scholars have considered introducing and expanding the sources of information. They can use auxiliary information such as the attributes of items themselves, users' social networks, and context to improve the accuracy of recommendations.

This paper proposes a RippleNet-CF model that combines the RippleNet model based on knowledge graphs and the collaborative filtering algorithm. The algorithm leverages course entities and the attributes of courses themselves to simulate the propagation of user course interests on the knowledge graph through ripple patterns. It also takes into account the interaction history between users and courses, such as viewing records and ratings, to uncover personalized recommendations for users. By expanding the sources of information and integrating the historical and current interests of target users, the accuracy of recommendation results is enhanced. The performance of the recommendation results is evaluated using three metrics: accuracy, recall, and F1.

DOI: 10.2478/ijanmc-2025-0020

II. Related Theories

A. Collaborative Filtering Recommendation Algorithm

The item-based collaborative filtering algorithm calculates the similarity between courses based on user preference data and then recommends a list of other courses that are similar to the ones the user likes [7]. However, it faces issues such as data sparsity and cold start. This paper chooses the course-based collaborative filtering algorithm for personalized course recommendations, and the implementation of this algorithm is divided into two steps:

1) Calculate the similarity between courses

Construct a student-course matrix:Let $U=\{u_1,u_2,u_3,..,u_m\}$ be the set of m students; $I=\{i_1,i_2,i_3,\cdots,i_n\}$ be the set of n courses, and Rm $\times$ n represent the rating matrix of students to courses as shown in formula (1):

$$R_{m\times n}=\begin{bmatrix} R_{11} & R_{12} & R_{13} & \cdots & R_{1n-1} & R_{1n} \\ R_{21} & R_{22} & R_{23} & \cdots & R_{2n-1} & R_{2n} \\ R_{31} & R_{32} & R_{33} & \cdots & R_{3n-1} & R_{3n} \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ R_{m-11} & R_{m-12} & R_{m-13} & & R_{m-1n-1} & R_{m-1n} \\ R_{m1} & R_{m2} & R_{m3} & & R_{mn-1} & R_{mn} \end{bmatrix} \quad (1)$$

Here, R_{ij} represents the rating of student U_i to course I_j, and the higher the value of R_{ij}, the more student u_i likes course I_j.

As an example, to measure how similar two courses are, all students' ratings for a given course are treated as an m $\times$ 1 vector. The ratings for course i are represented as $F_i = \{r_{1i}, r_{2i}, r_{3i}, ..., r_{mi}\}$, and the ratings for course j are recorded as $F_j = \{r_{1j}, r_{2j}, r_{3j}, ..., r_{mj}\}$. The formula for computing the similarity between courses i and j is provided in Equation (2).

$$W_{ij}=\frac{F_i\cdot F_j}{\|F_i\|\cdot\|F_j\|}=\frac{\sum_{u=1}^{m} r_{u_i}\cdot r_{uj}}{\sqrt{\sum_{u=1}^{m} r_{u_i}{}^2}\cdot\sqrt{\sum_{u=1}^{m} r_{u_j}{}^2}} \quad (2)$$

Among them, W_{ij} represents the cosine similarity value between course i and course j, with a corresponding range of [-1, 1]. The W_{ij} higher the value, the more similar courses i and j are, and the target user is expected to have similar behavior towards the course in the future.

2) Selecting Neighbors

When selecting neighbors, this paper chooses to rank them according to the similarity of courses. Then, several courses with the highest ranks from the sorted results are selected as neighbors.

B. Knowledge Graph Learning

Knowledge Graphs (KG) [8] can effectively map out vast amounts of disordered data through theoretical methods such as data mining and information processing, making it more convenient and accurate for people to obtain the information they need. A knowledge graph is a large-scale semantic network representing a complex web of relationships between entities, generally composed of (entity, relationship, entity) triples [9]. Incorporating knowledge graphs into recommendation systems can uncover deeper semantic relationships and more precisely identify the interests of target users. Currently, the application of knowledge graph feature learning [10] in recommendation systems is generally divided into: path-based recommendation algorithms [11]and embedding-based recommendation algorithms [12], with representative models including TransE, TransH, SME, NTN, etc.

C. RippleNet Model

Due to the limitations of knowledge graph perception reconstruction methods applied to recommendation systems, scholars have proposed another model, RippleNet [13].

The knowledge graph is constructed from the triple relationships corresponding to course entities $G=\{(h,r,t)\,|\,h,t\in R\}$. The goal of the RippleNet model is to construct a knowledge graph to utilize students' preferences for courses and calculate the click probability of student u for the target course v. The main implementation of its algorithm is as follows:

Deffnition 1: Item Embedding. Based on the characteristics, semantics, and attributes of items, the embedding is performed. Given the embedding vector v of a speciffed course and the 1-hop ripple

set, each expansion outward yields a triplet. The relevance score between item v and each (h_i, r_i, t_i) in the 1-hop set is calculated, and the linear relevance scores are normalized using the softmax function. Consequently, the head entity h_i and the relation R_i of the triplet are treated as an association probability P_i, as shown in formula (3):

$$p_i = soft\max(v^T R_i h_i) = \frac{\exp\left(v^T R_i h_i\right)}{\sum_{(h,r,t)\in S_u^1} \exp(v^T Rh)} \quad (3)$$

$$o_u^1 = \sum_{(h_i,r_i,t_i)\in S_u^1} p_i t_i \quad (4)$$

Here, v^T represents the item vector, t_i is the tail entity vector, h_i is the head entity vector, and r_i is the relation mapping matrix. S_u^1 the first-layer Ripple Set of a student (the first-hop Ripple Set, as shown in the figure1) is formed by selecting a certain number of items from the student ' s interaction history.Essentially, this process calculates the correlation and similarity between the seed node and its connected one-hop nodes in the knowledge graph, as represented by triples—illustrated in Figure 1.

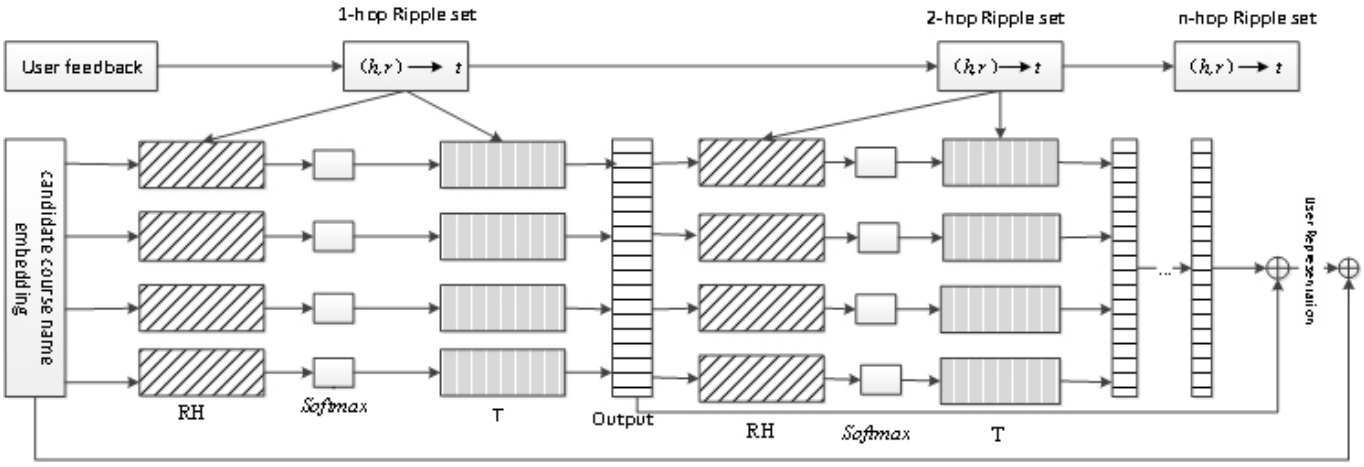

Figure 1. RippleNet Model Diagram.

By repeating the above process, the knowledge graph undergoes multi-hop propagation. The corresponding vectors obtained from each hop are then summed to generate the student's embedding vector (user embedding). After repeating the process H times, H output vectors o are obtained, and the final user embedding is calculated according to Equation (5).

$$u = o_u^1 + o_u^2 + \ldots + o_u^H \quad (5)$$

Finally, the likelihood of user u engaging with course v is computed by integrating their respective latent representations, as illustrated in Equation (6).

$$y_{uv} = \sigma(u^T v)$$
$$\sigma(x) = \frac{1}{1+e^{-x}} \quad (6)$$

III. Integration of the RippleNet Model and an Item-Oriented Collaborative Filtering Approach

Conventional item-level recommendation techniques algorithms only consider users' rating data on courses. After extracting the relationships between courses and their attributes, this paper proposes an algorithm that integrates course attribute information with user-course interaction data by combining the RippleNet model and collaborative filtering. The RippleNet model

leverages historical user-course rating records as implicit relationships between users and items. It constructs a knowledge graph based on the relationships among course attributes and extracts corresponding triples for each course. Using the ripple propagation mechanism through these triples, it computes user preferences. Meanwhile, the item-oriented filtering method estimates a user's interest in unvisited courses by analyzing past interactions between the user and various courses. By combining both approaches, a comprehensive course recommendation list is generated. This method fully utilizes the strengths of both algorithms by linearly fusing the results of the two recommendation lists. The fusion method is defined in Equation (7).

$$C = \beta * Y + (1 - \beta) * P \tag{7}$$

Here, β represents the weight within the range (0, 1). Y indicates the likelihood that the target user clicks on unseen courses as inferred by the RippleNet model, while P reflects the same likelihood as estimated through the collaborative filtering method.

By integrating the knowledge graph and collaborative filtering course recommendation algorithms from both direct and indirect perspectives, the limitations of using a single approach can be effectively mitigated. The knowledge graph also provides strong interpretability throughout the entire process. The corresponding flowchart of the integrated RippleNet and collaborative filtering recommendation algorithm (RippleNet-CF) is shown in Figure 2.

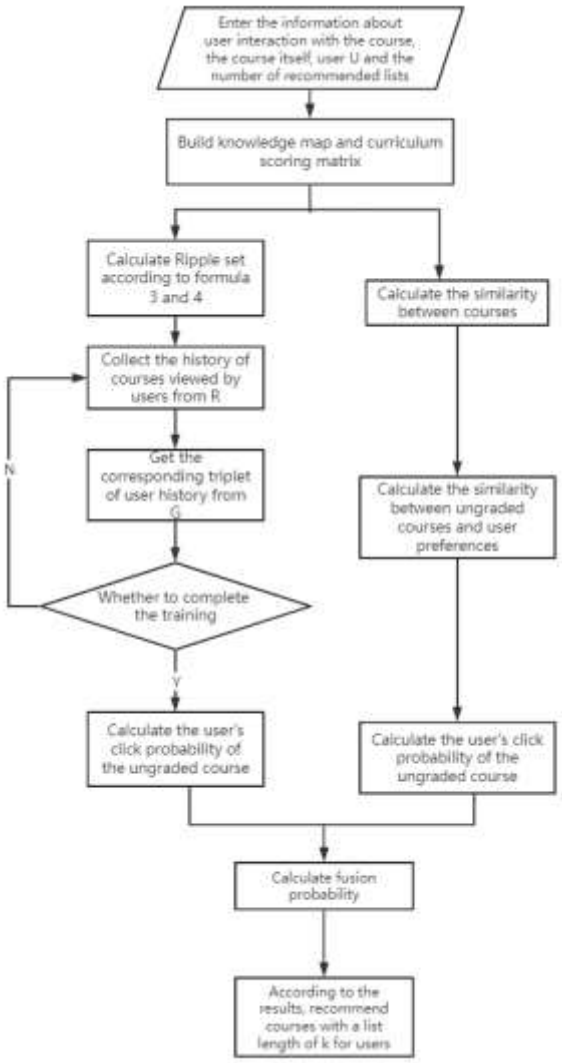

Figure 2. Flowchart of the Integrated Recommendation Algorithm

IV. Experimental Results and Analysis

A. Dataset and Preprocessing

The dataset used in this experiment is MOOCCube, which was collected by a research team from Tsinghua University from the XuetangX platform. They extracted entities such as courses, concepts, and students, and built a knowledge base based on the complex relationships among these entities. This educational resource database is large in scale and rich in data, especially with detailed records of student behavior, including learning duration, frequency, and video segments viewed. The dataset used in this experiment involves nearly 200,000 students and approximately 5 million

video viewing records [14]. Before conducting the experiment, the collected online student dataset needs to be preprocessed. The specific steps are as follows:

- Integrate the video viewing information of each student from the MOOCCube dataset, calculating the total duration of videos for the same course as well as the specific viewing details of the students.
- Handling of missing or duplicate values. For data that is missing or duplicated, it is directly removed.
- The learner's rating is determined by the ratio between their actual viewing time (t) and the total video length (T). That is, the rating score = t/T. Furthermore, these scores are categorized into five distinct levels, with the detailed classification criteria provided in Table 1.

TABLE I. COURSE RATING

Rating	***Score***
S<0.2	1
0.2≤S<0.4	2
0.4≤S<0.6	3
0.6≤S<0.8	4
S≥0.8	5

B. Constructing a Knowledge Graph

Based on the results of data preprocessing, mark the user-course interaction Yuv = 1 if the user's rating for the course is greater than or equal to 4, and mark Yuv = 0 for other scores. According to the courses that users have interacted with, extract the relationships between the attributes of the courses themselves to construct triples. Since there are too many entities in each course for constructing triples, to lower the cost of constructing the knowledge representation, each course is associated with only five extracted entities, as illustrated in Table 2.

TABLE II. EXTRACTION OF SOME COURSE ENTITIES

Course Name	***Entity***
Popular Java Framework	*Tsinghua University Press, October 2018, Lectured by Li Lian, Knowledge Points, Computer*
Data Structures	*People's Publishing House, February 2022, Yu Yun, Knowledge Points, Computer*
Database Principles	*Posts and Telecommunications Press, October 2018, Cao Lan, Knowledge Points, Computer*
Advanced Mathematics	*Tsinghua University Press, September 2018, Zhang Yu, Knowledge Points, Mathematics*

After determining the corresponding entities, construct the corresponding ternary relationships, a total of 5 types of entities are constructed as shown in Table 3.

TABLE III. TERNARY ENTITY RELATIONSHIPS

Entity	***Relationship***	***Entity***
Course Name	*Taught by*	*Teacher*
Course Name	*Published by*	*Specific Publisher*
Course Name	*Time*	*Specific Publication Time*
Course Name	*Belongs to*	*Specific Category*
Course Name	*Contains*	*Knowledge Points*

Based on the construction of ternary relationships for association: for instance, if a teacher teaches several courses, one of the courses can be associated with another by the common teacher who teaches them, as specifically shown in Figure 3: In this experiment, a total of 447,517 ternary relationships were constructed.

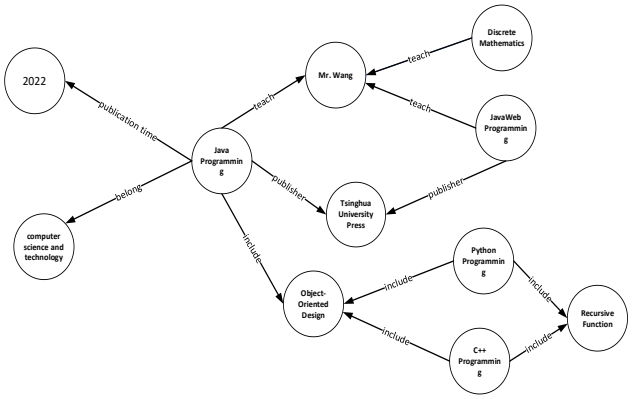

Figure 3. Partial View of the Knowledge Graph

C. Evaluation Metrics

The experimental results in this paper adopt a Top-N recommendation strategy for delivering personalized suggestions to target users. Performance is assessed through three evaluation indicators: precision, recall, and F1 score. In this

context, L(u) denotes the actual recommendation list for user U in the test dataset, while R(u) corresponds to the predicted list generated by the algorithm. Here, U refers to the set of users, and I signifies the collection of available courses.

- Precision: The calculation method is as shown in Formula (8).

$$Precision = \frac{\sum_u \epsilon U |L(u) \cap R(u)|}{\sum_u \epsilon U |L(u)|} \quad (8)$$

- Recall: The calculation method is shown in Equation (9).

$$Recall = \frac{\sum_u \epsilon U |L(u) \cap R(u)|}{\sum_u \epsilon U |R(u)|} \quad (9)$$

- F1 Score (F-Measure): The calculation method is shown in Equation (10).

$$F1 = \frac{2Precision * Recall}{Precision + Recall} \quad (10)$$

D. Experimental Results Analysis

In this paper's RippleNet-CF algorithm, the weight β in equation (9) needs to be trained with corresponding parameters, and the results are shown in Figure 4:

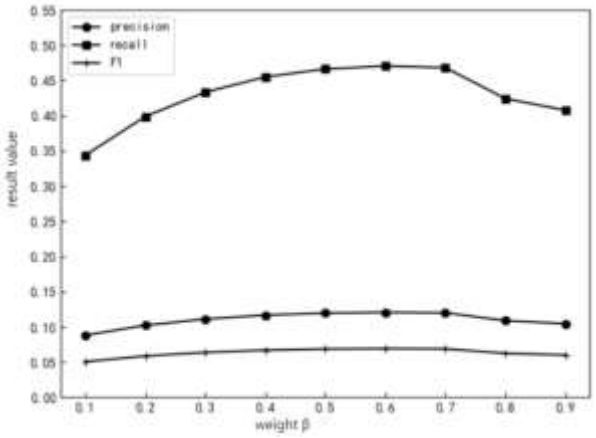

Figure 4. RippleNet-CF Results Chart

From Figure 4, it can be concluded that both accuracy and recall increase as the weight value increases within the range of [0.1, 0.6], corresponding to higher probability values. The accuracy and recall reach their maximum when the weight β equals 0.6. However, the coverage rate is highest at 0.4 and then decreases as the weight value increases.

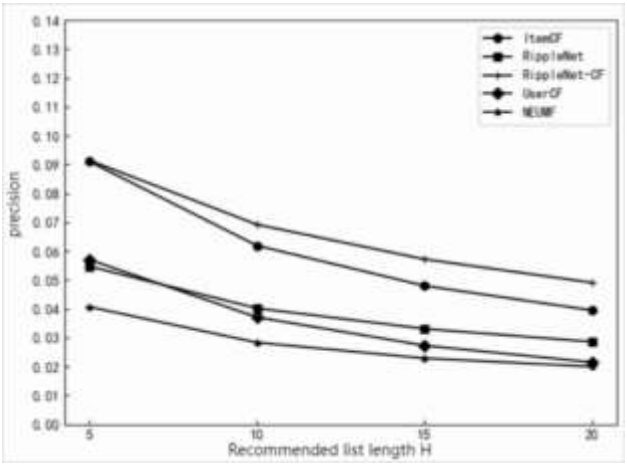

Figure 5. Accuracy Results Chart

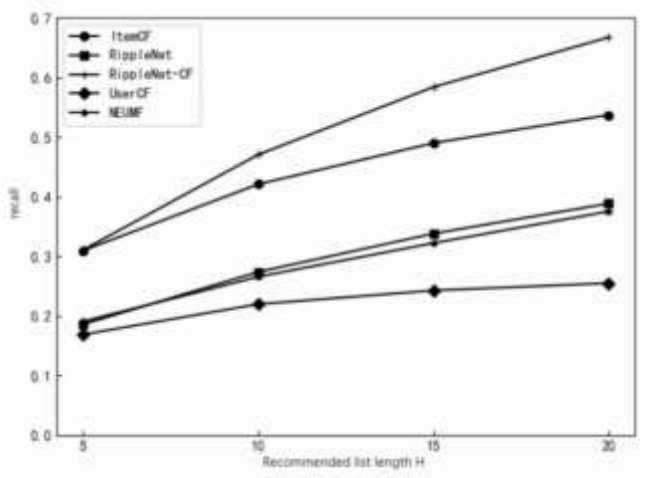

Figure 6. Recall Results Chart

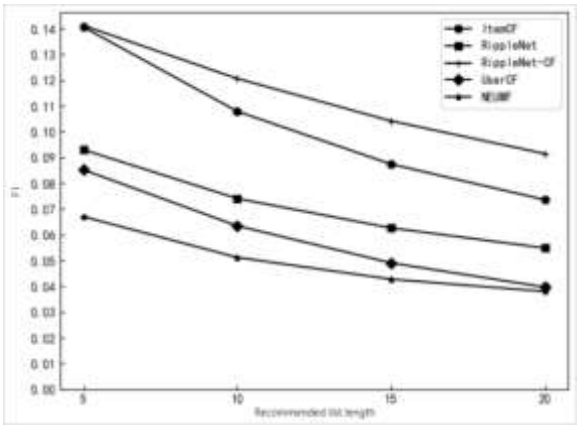

Figure 7. F1 Score Chart

From Figures 5, 6, and 7, it can be seen that at β= 0.6 and h = 2 as the recommendation list increases, the RippleNet-CF method has the best accuracy, recall, and F1 scores compared to the other four algorithms. This is because RippleNet-CF not only uses the interaction information between users and items but also mines the potential connections between courses to expand the information source, thereby improving the optimization effect.

V. Summary and Future Work

In response to the traditional item-based collaborative filtering algorithm, which does not fully utilize the attribute information of items themselves, this paper proposes the RippleNet-CF method using course attribute knowledge graphs and interaction information. This method uses knowledge graphs to explore the potential connections between courses and collaborative filtering to explore existing user connections, thereby improving the issues of data sparsity and cold start problems. However, courses are offered according to semesters and have strong practical sequential characteristics. Future work will consider incorporating time series feature information to further improve accuracy.

VI. Acknowledgment

The authors would like to express their gratitude to Anhui Xinhua University (China) for the support the University-level Scientific Research Project of Anhui Xinhua University (2024zr012). Additionally, we appreciate the support from. Provincial Innovation and Entrepreneurship Program for College Students (S202412216185)

References

[1] S. Wu, F. Sun, W. Zhang, et al., "Graph neural networks in recommender systems: a survey," ACM Computing Surveys, vol. 55, no. 5, May 2022, pp. 1–37, doi:10.1145/3519724.

[2] S. Wang, L. Cao, Y. Wang, et al., "A survey on session-based recommender systems," ACM Computing Surveys (CSUR), vol. 54, no. 7, Aug. 2021, pp. 1–38, doi:10.1145/3460951.

[3] J. Li, Z. Ye, "Course recommendations in online education based on collaborative filtering recommendation algorithm," Complexity, vol. 2020, Apr. 2020, Article ID 8813370, doi:10.1155/2020/8813370.

[4] P. K. Singh, R. Ahmed, I. S. Rajput, et al., "A comparative study on prediction approaches of item-based collaborative filtering in neighborhood-based recommendations," Wireless Personal Communications, vol. 121, no. 6, Nov. 2021, pp. 857–877, doi:10.1007/s11265-021-01696-1.

[5] G. Piao, J. G. Breslin, "A study of the similarities of entity embeddings learned from different aspects of a knowledge base for item recommendations," in Proceedings of the European Semantic Web Conference (ESWC 2018), Springer, Cham, June 2018, pp. 345–359, doi:10.1007/978-3-319-93417-4_21.

[6] M. J. Pazani, D. Billsus, "Content-based recommendation systems," in The Adaptive Web: Methods and Strategies of Web Personalization, Springer, Berlin, Heidelberg, May 2007, pp. 325–341, doi:10.1007/978-3-540-72079-9_10.

[7] H. Wang, F. Zhang, J. Wang, et al., "Ripplenet: Propagating user preferences on the knowledge graph for recommender systems," in Proceedings of the 27th ACM International Conference on Information and Knowledge Management (CIKM 2018), ACM Press, Oct. 2018, pp. 417–426, doi:10.1145/3269206.3271764.

[8] W. Jiang, Y. Sun, "Social-RippleNet: Jointly modeling of ripple net and social information for recommendation," Applied Intelligence, vol. 53, no. 3, Mar. 2023, pp. 3472–3487, doi:10.1007/s10489-021-03214-7.

[9] Y. Q. Wang, L. Y. Dong, Y. L. Li, et al., "Multitask feature learning approach for knowledge graph enhanced recommendations with RippleNet," Plos One, vol. 16, no. 5, May 2021, e0251162, doi:10.1371/journal. pone. 0251162.

[10] H. Wang, F. Zhang, X. Xie, et al., "DKN: Deep knowledge-aware network for news recommendation," in Proceedings of the 2018 World Wide Web Conference (WWW 2018), ACM Press, Apr. 2018, pp. 1835–1844, doi:10.1145/3178876.3186143.

[11] H. Wang, F. Zhang, M. Hou, et al., "Shine: Signed heterogeneous information network embedding for sentiment like prediction," in Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining (WSDM 2018), ACM Press, Feb. 2018, pp. 592–600, doi:10.1145/3159652.3159668.

[12] X. Yu, X. Ren, Y. Sun, et al., "Personalized entity recommendation: A heterogeneous information network approach," in Proceedings of the 7th ACM International Conference on Web Search and Data Mining (WSDM 2014), ACM Press, Feb. 2014, pp. 283–292, doi:10.1145/2556195.2556222.

[13] Y. Cao, X. Wang, X. He, et al., "Unifying knowledge graph learning and recommendation: Towards a better understanding of user preferences," in The World Wide Web Conference (WWW 2019), ACM Press, May 2019, pp. 151–161, doi:10.1145/3308558.3313433.

[14] F. M. Harper, J. A. Konstan, "The MovieLens Datasets: History and Context," ACM Transactions on Interactive Intelligent Systems, vol. 5, no. 4, Dec. 2016, Article No. 19, doi:10.1145/2827872.

.