

## Research Article

# Recommendation Algorithm Based on Knowledge Graph to Propagate User Preference

Zhisheng Yang<sup>ID</sup>, Jinyong Cheng<sup>\*</sup><sup>ID</sup>

*School of Computer Science and Technology, Qilu University of Technology (Shandong Academy of Sciences), Jinan, 250353, China*

## ARTICLE INFO

### Article History

Received 18 Mar 2021

Accepted 29 Apr 2021

### Keywords

Recommendation algorithm

Knowledge graph

Preference

## ABSTRACT

In recommendation algorithms, data sparsity and cold start problems are inevitable. To solve such problems, researchers apply auxiliary information to recommendation algorithms, mine users' historical records to obtain more potential information, and then improve recommendation performance. In this paper, ST\_RippleNet, a model that combines knowledge graphs with deep learning, is proposed. This model starts by building the required knowledge graph. Then, the potential interest of users is mined through the knowledge graph, which stimulates the spread of users' preferences on the set of knowledge entities. In preference propagation, we use a triple multi-layer attention mechanism to obtain triple information through the knowledge graph and use the user preference distribution for candidate items formed by users' historical click information to predict the final click probability. Using ST\_RippleNet model can better obtain triple information in knowledge graph and mine more useful information. In the ST\_RippleNet model, the music data set is added to the movie and book data set; additionally, an improved loss function is used in the model, which is optimized by the RMSProp optimizer. Finally, the tanh function is added to predict the click probability to improve recommendation performance. Compared with current mainstream recommendation methods, ST\_RippleNet achieves very good performance in terms of the area under the curve (AUC) and accuracy (ACC) and substantially improves movie, book and music recommendations.

© 2021 The Authors. Published by Atlantis Press B.V.

This is an open access article distributed under the CC BY-NC 4.0 license (<http://creativecommons.org/licenses/by-nc/4.0/>).

## 1. INTRODUCTION

In modern online platforms, recommendation systems play a vital role in making users pay attention to personalized content. Users expect personalized content on modern e-commerce, entertainment and social media platforms, but the effectiveness of recommendation is limited by existing user-project interaction and model capacity. The explosive growth of online content and services has provided users with many choices, such as movies, music and books. Researchers are devising methods for improving recommendation performance, judging the preference similarity between users through the interaction between users, providing ranked recommendations by collecting information regarding user hobbies, building tables among multiple users with the same preferences and using various methods to solve the problem of low recommendation accuracy. Some of the methods used to improve recommendation performance include three-way neural networks, meta-paths, attribute reuse structures, convolutional neural networks (CNN) [1] and generative adversarial networks (GAN), achieving good results. Among these methods, collaborative filtering (CF) mines potential preference information by analyzing users' historical records and using them to make recommendations. The sparsity of user-project interaction and cold start always interfere with the recommendation effect. Researchers have proposed many ideas about data sparsity and cold start, and the idea of integrating

auxiliary information such as social networks [2], user/project attributes [3], images [4] and context [5] into CF has achieved good results.

Regarding auxiliary information, the knowledge graph (KG) contains various interrelationships between users and projects. Some recently proposed knowledge graphs have been used in applications such as answering questions [6], KG completion [7], text classification [8] and word embedding [9]. For example, Microsoft Satori has achieved good results.

In this age of recommendation algorithms, recommendation systems play a great role in not only shopping software such as Taobao and Jingdong but also video software such as Aiqiyi and Tencent, as well as music such as QQ Music and Netease Cloud Music. These online applications [10] continuously promote the development of recommendation algorithms and recommendation systems. The greater the demand and the more investment, the stronger the research on recommendation systems will certainly be. Classical recommendation methods, such as matrix factorization [11], mainly use historical user-project interaction records to simulate users' preferences for projects. There are also recommendations made using the similarity function. Using this function, recommendation learning is carried out through human judgment on the similarity of objects [12], after which accurate similar neighbors between users or project requirements are captured according to their historical common evaluation, and then appropriate projects

\*Corresponding author. Email: [cjy@qlu.edu.cn](mailto:cjy@qlu.edu.cn)

or items are recommended. An intelligent recommendation system [13] can recommend appropriate music, movies and books based on user hobbies [14] and is widely used to realize accurate matching between users and various resources. Currently, various kinds of auxiliary data are becoming increasingly available in online services. Many methods further suggest using this contextual information to improve recommendation performance. Due to the heterogeneity and complexity of auxiliary data, it is still challenging to effectively utilize this context information in recommendation systems. KG [15] can improve recommendation performance [16] by introducing semantic associations between items [17] consisting of various types of interaction relationships and linking user history with recommendation records.

Current recommendation algorithms are inevitably required to solve the problems of data sparsity and cold start. In recommendation systems, the problem of data sparsity will directly affect the recommendation accuracy. While data containing null values in the recommendation data set provide some information, at information is not complete enough. If such null value information can be accommodated or supplemented, the recommendation effect will undoubtedly be greatly improved. In addition, methods of obtaining more useful potential information from users as well as mining the specific data needed from such a large volume of information will also affect the final recommendation result.

In this paper, our proposed KG recommendation model, ST\_RippleNet, represents a network structure for joint training of KGs and recommendation systems where the S represents a triple attention mechanism, and the T represents a tanh function. Built to address the defects of some current recommendation algorithms, ST\_RippleNet is a network model that integrates the characteristics of the tanh function and the advantages of the triple attention mechanism for joint training of KGs and recommendation systems. It not only introduces knowledge graph embedding (KGE) method into recommendations but also mines potential user choices. This model first builds a KG to obtain more potential information between users and projects. In preparation for solving the problem of data sparsity, the triple information in the KG is then mined by the triple attention mechanism. Through the triple attention mechanism, we mine more potential information from users, and this information can be added to null values in the case of data sparsity, providing more useful information to improve the recommendation effect. By constructing a triple attention model to reason and mine more potential information from the triple, this model also mines more potential user preference information in the KG and then efficiently uses the relationship structure information between entities through the user preference propagation model, taking into account the propagation intensity while propagating preferences, thus effectively improving the entity recommendation effect. In this paper, ST\_RippleNet is applied to music, book and movie recommendations, and it is found that on the music data set Last.FM [18], the area under the curve (AUC) evaluation index reaches 0.827 and the accuracy (ACC) evaluation index reaches 0.763. On MovieLens-1M, the AUC reaches 0.925 and the ACC reaches 0.852. On the Book-Crossing data set, the AUC reaches 0.735 and the ACC reaches 0.693. Compared with current mainstream recommended methods, ST\_RippleNet achieves AUC gains of 6.4% to 37.4%, 0.8% to 18.4% and 0.4% to 41.2% and ACC gains of 7.6% to 31.9%, 4.7% to 24.2% and 0.9% to 44.7%, respectively.

This paper proposes a KG recommendation model ST\_RippleNet to address the defects of some current recommendation algorithms. ST\_RippleNet not only introduces the KGE method into recommendations but also mines potential user information. In this paper, when ST\_RippleNet is applied to music, books and movies, it is found that compared with the current mainstream recommendation [19] methods, ST\_RippleNet achieves AUC gains of 6.4% to 37.4%, 0.8% to 18.4% and 0.4% to 41.2%, respectively, and ACC gains of 7.6% to 31.9%, 4.7% to 24.2% and 0.9% to 44.7%.

In addition to applying ST\_RippleNet to the original data sets, MovieLens-1M and Book-Crossing, we added the music data set Last.FM, and found that ST\_RippleNet achieves better results than other baselines in music recommendation. Additionally, we improved the loss function of the framework by applying the characteristics of the tanh function, while we used the RMSProp optimizer when optimizing and the tanh function when predicting the click probability. In the experiment, we found that the RMSProp optimizer is more suitable for our framework than other optimizers.

Through experiments on the Last.FM, Book-Crossing and MovieLens-1M data sets, the effectiveness of ST\_RippleNet is proven according to its evaluation index results and recommendation effect.

## 2. CONSTRUCTING THE KG

A KG is a source of auxiliary information, and to construct a KG from the original data set, it must go through information extraction, knowledge fusion, data processing, among other steps, as shown in Figure 1. First, the data in the music/book/movie data sources are divided into structured data, semi-structured data and unstructured data according to the degree to which they are structured. Second, the semi-structured and unstructured data with unclear data structures are extracted by attribute extraction, entity extraction and joint extraction of entities and relationships. Then, knowledge integration is performed. The information in the structured database, the third-party database and the information database obtained by extraction of semi-structured data and unstructured data are integrated into a knowledge base. The primary problem to be solved in the integration process is entity alignment, that is, entity disambiguation and co-reference resolution. Its purpose is to integrate different descriptions of each entity in different databases to obtain a complete description of the entity, that is, the knowledge base is finally obtained. After that, a top-down approach is adopted to construct the data organization framework of the KG, that is, the data model. Then, the potential relationship information between users and projects is obtained through knowledge reasoning, that is, data mining through logical systems according to reasoning rules. Finally, the final data obtained are optimized, and the qualified data that meet our needs are put into the KG; thus, the KG is constructed.

Using the music data set as an example, we constructed the music recommendation KG required by our recommendation algorithm, as shown in Figure 2. In the music KG, data are mainly depicted by triples, and triples are mainly expressed in the forms of (entity-relationship-entity) and (entity-attribute-entity). In Figure 2, Closer is an entity and Paris is an entity, so Closer-The Chainsmokers-Paris is a sample (entity-relationship-entity) triple. The Chainsmokers

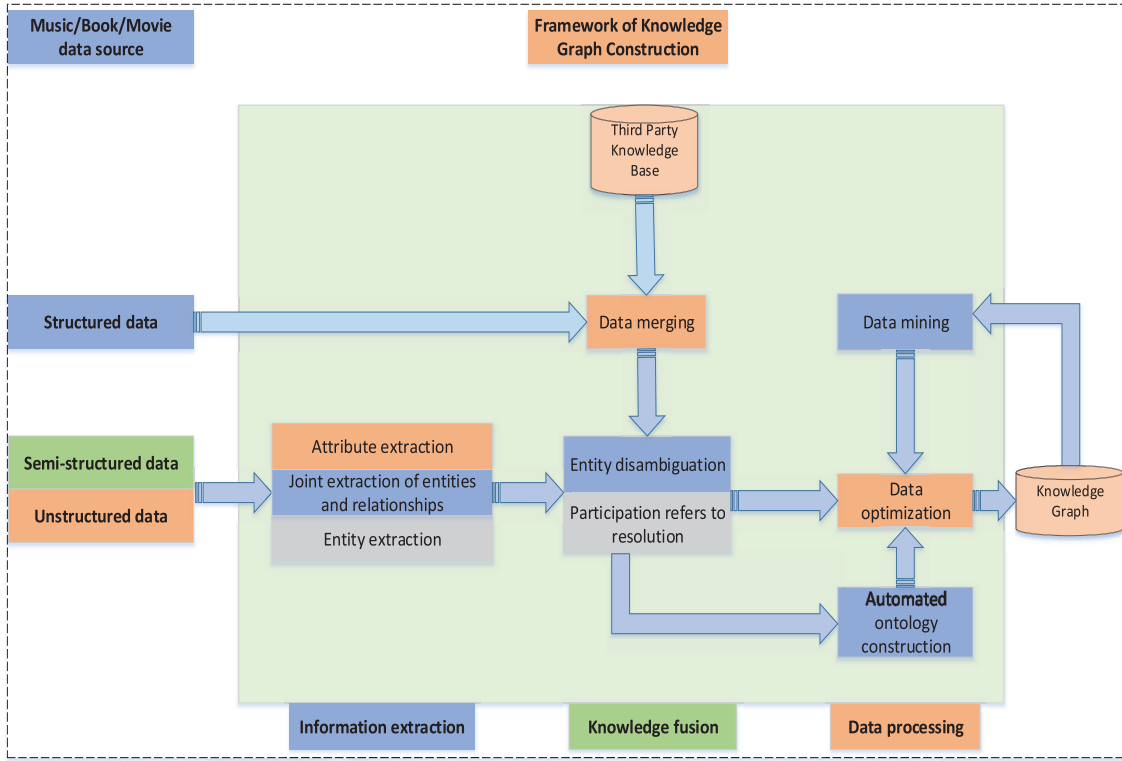


Figure 1 | Knowledge graph construction.

is an attribute and Popular music is an attribute class, so Something Just Like This-The Chainsmokers-Popular music constitutes a sample triple (entity-attribute-attribute value). Therefore, we can mine the potential relationship between more users and projects by obtaining triple samples in the constructed KG. Then, through the triple multi-layer attention mechanism to obtain triple information, a triple attention model is constructed to reason and mine more potential information from the triple, which is also used to mine more potential user preference information from the KG, aiming at recommending appropriate music for users more accurately. The same is true for the knowledge graph construction of MovieLens-1M, a movie data set, and Book-Crossing, a book data set.

### 3. MODEL DESCRIPTION

#### 3.1. ST\_RippleNet Model

This paper proposes a recommendation algorithm based on KGs to propagate user preferences. Its basic idea is as follows: Taking music recommendation as an example, first, the user-project KG is constructed, that is, the music recommendation KG. The graph is constructed by using node2vec [20] to extract the structural features of the music recommendation KG; then, the vectorized representation of its knowledge, the entities in the KG and the relationships among them, are embedded in the KG as dense low-dimensional vectors. Then, by calculating the similarity between user items, the correlation degree between user and user, music and music, and between user and music is obtained. The basic different characteristics are used as the weight value of transmission intensity. Furthermore, using the structural information from the music recommendation KG [21] to carry out iterative calculation through the

user preference transmission model combined with the transmission intensity, the structural features are extracted and the training model is used as input to adjust the importance of different features through the objective function to achieve the optimal result. The entities are sorted and taught to generate top N recommendation lists, and then the click rate is predicted.

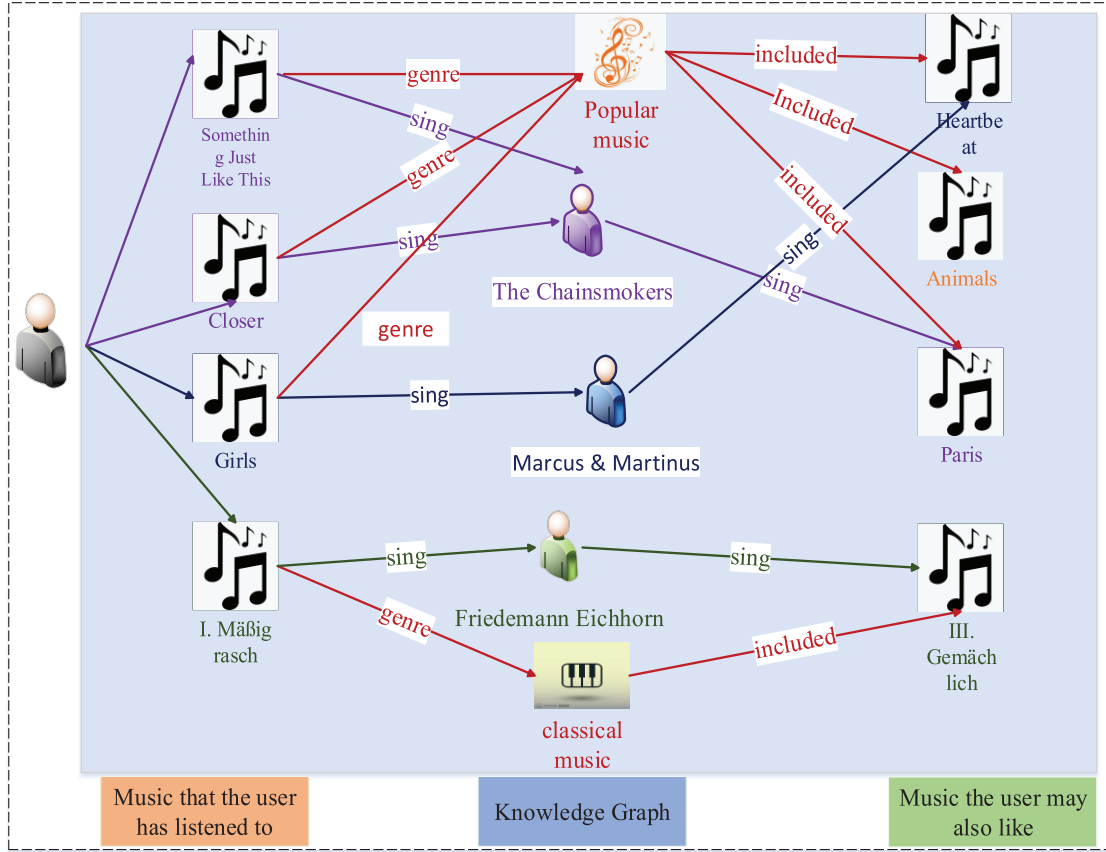
This algorithm utilizes the integration of KGs to multi-source heterogeneous data. The low-dimensional vector is introduced into the recommendation system, which makes full use of the information in the KG. In addition, it does not need to manually design a specific meta-path. Through the user preference propagation model, the relationship structure information between entities in the KG is efficiently utilized, and the propagation intensity is taken into account while the preference is propagated, thus effectively improving the entity recommendation effect. The mixed user item feature model obtained by our method is more accurate, which improves the KG information utilization problem and improves the recommendation performance. The structure diagram is shown in Figure 3.

#### 3.2. ST\_RippleNet Network Structure

In ST\_RippleNet, given the interaction matrix  $Y$  and the KG  $G$ , the set of  $k$ -hop related entities that we define for user  $u$  is

$$\xi_u^k = \{t | (h, r, t) \in G \text{ and } h \in \xi_u^{k-1}\}, k = 1, 2, \dots, H. \quad (1)$$

In Collection,  $\xi_u^0 = V_u = \{v | v_{uv} = 1\}$  is a collection of historical items clicked by the user.



**Figure 2** | The knowledge graph of the music recommendation system provides rich interaction between users and music, which is helpful to mine potential user preference information.

Its Ripple Set is a set of knowledge triples starting from  $\xi_u^{k-1}$ , and a maximum value is set here to prevent the Ripple Set from being too large.

$$M_u^k = \{(h, r, t) \mid (h, r, t) \in G \text{ and } h \in \xi_u^{k-1}\}, k = 1, 2, \dots, H. \quad (2)$$

In the ST\_RippleNet model, we use preference propagation technology [11] to mine hidden interests and hobbies in users' ripple sets to obtain more information for subsequent recommendations and more accurate prediction of click rates.

Taking music recommendation as an example, the music embedded item  $v \in R^n$  may combine the single hot identification [11], attribute, word package or context information [22] of the music item  $v$ . Given the set of 1-hop ripples  $M_u^1$  of the embedded term  $v$  and the user  $u$ , each triple  $(h_i, r_i, t_i)$  in  $M_u^1$  is assigned a correlation probability using the softmax function through the music comparison term  $v$  and the header  $h_i$  and the interaction relation  $r_i$  in the triple:

$$p_i = \text{softmax}(v^T R_i h_i) = \frac{\exp(v^T R_i h_i)}{\sum_{(h_i, r_i, t_i) \in M_u^1} \exp(v^T R_i h_i)} \quad (3)$$

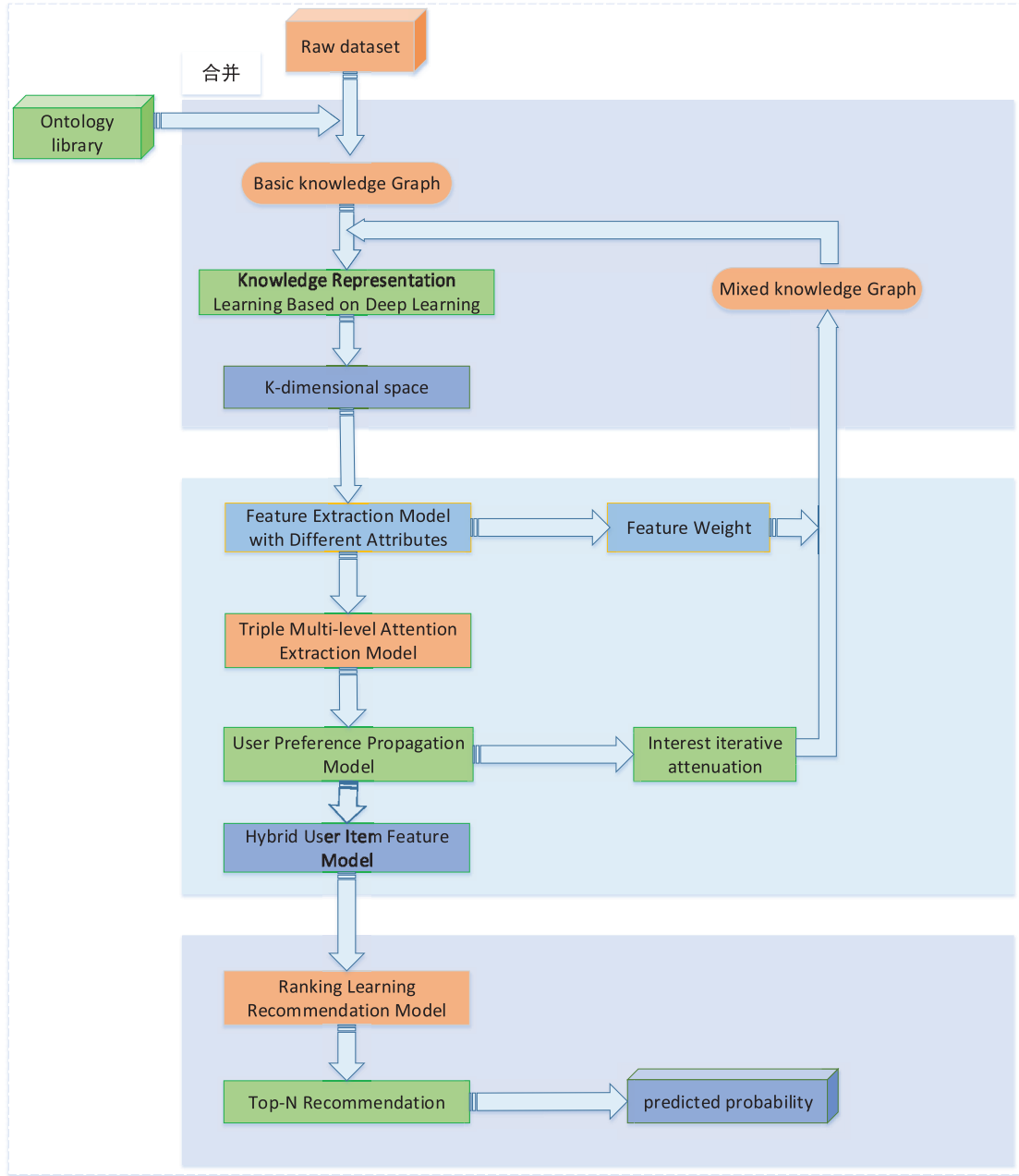
The embedding of relation  $r_i$  is  $R_i \in R^{n \times n}$ , while the embedding of header  $h_i$  is  $h_i \in R^d$ . The association probability  $p_i$  can be regarded as the similarity between item  $v$  and entity  $h_i$  measured

in the space of relation  $R_i$ . Please note that when calculating the correlation between item  $V$  and entity  $h_i$ , it is necessary to consider the embedding matrix  $R_i$  because when measured by different relationships, item–entity pairs may have different similarities. For example, Something Just Like This and Paris are highly similar when considering singers or music genres, but from the perspective of composers, the two are far from each other.

After obtaining the correlation probability, we take the sum of the tails in  $M_u^1$  multiplied by the corresponding correlation probability to return vector  $o_u^1$ :

$$\begin{aligned} o_u^1 &= \sum_{(h_i, r_i, t_i) \in M_u^1} p_i t_i \\ &= \sum_{(h_i, r_i, t_i) \in M_u^1} (\text{softmax}(v^T R_i h_i)) t_i \\ &= \sum_{(h_i, r_i, t_i) \in M_u^1} \left( \frac{\exp(v^T R_i h_i)}{\sum_{(h_i, r_i, t_i) \in M_u^1} \exp(v^T R_i h_i)} \right) t_i \end{aligned} \quad (4)$$

The embedding of tail  $t_i$  is used for  $t_i \in R^n$ . A section response of the music click record  $V_u$  of user  $u$  relative to music  $v$  is represented by a vector  $o_u^1$ . The second-order response  $o_u^2$  of user  $u$  is obtained by performing preference propagation, and this process can be iterated on its ripple set  $M_u^1$  to obtain  $i = 1, \dots, H$ . In this process, the response of a plurality of music click records of user  $U$  can be observed, and the user's preference is also propagated to the position where the most  $H$ -hops of the click records are recorded. Its



**Figure 3** | Recommendation algorithm for spreading user preferences based on knowledge graphs.

embedding is also calculated through all responses.

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

Finally, it should be noted that although the user response of the last hop  $O_u^H$  theoretically contains all the information from the previous hop, since some information may be diluted in  $O_u^H$ , the calculation of user embedding must be combined with  $O_u^k$  of small hop  $K$ . Then, user embedding and item embedding are combined to output the predicted click probability. In this process, we use the tanh function to predict the click probability:

$$\hat{y}_{uv} = \frac{\exp(u^T v) - \exp(-u^T v)}{\exp(u^T v) + \exp(-u^T v)} \quad (6)$$

The tanh function is more conducive to the training model and more easily reaches the optimal value when generating the user's music/movie/book hobby preference model through training before predicting the click rate, and a more accurate user preference recommendation model can be obtained.

### 3.3. Model Optimization

The root mean square (RMS), like momentum gradient reduction, is a way to eliminate sloshing as a means to accelerate gradient reduction. The RMSProp algorithm, using RMS, is a reasonable and easy-to-use deep neural network optimization algorithm. In application, the weight value  $W$  or  $B$  is usually a combination of weight values from many levels, which is multi-dimensional. In the actual



operation of the eradication number, the gradient at the larger level will be greatly reduced, while the trend analysis of the weight value  $W$  will not be the same. The calculation formula for the RMSProp optimizer is as follows:

$\gamma$ : Power, usually set to 0.9.

$\eta$ : The value is generally 0.001

$E[g^2]_t$ : Represents the average value of the square of the gradient for the first  $t$  times.

$$g_t = \nabla wJ(W) \quad (7)$$

$$E[g^2]_t = \gamma E[g^2]_{t-1} + (1 - \gamma) g_t^2 \quad (8)$$

$$W_{t+1} = W_t - \frac{\eta}{\sqrt{E[g^2]_t} + \xi} \odot g_t \quad (9)$$

Because the RMSProp optimizer contains an attenuation factor to control the amount of history information, in the process of building the user preference model, our model also needs to obtain the potential information from users by obtaining the users' historical information to provide accurate recommendations. Therefore, the attenuation coefficient of the RMSProp optimizer, which controls the amount of user history information, can help us to obtain more information from users to recommend and optimize the model better when constructing the user preference model through a KG.

### 3.4. Loss Function

In the ST\_RippleNet model, for a given KG  $G$ , combined with its implicit feedback matrix  $Y$ , the following posterior probabilities of model parameters are maximized: we calculate the posterior probabilities as follows:

$$\max p(\Theta|G, Y) \quad (10)$$

which represents maximizing model parameters. After the posterior probability is expanded, it is as follows:

$$p(\Theta|G, Y) = \frac{p(\Theta|G, Y)}{p(G, Y)} \propto p(\Theta) \cdot p(G, \Theta) \cdot p(Y|\Theta, G) \quad (11)$$

However, the prior probability of its parameters must obey the normal distribution of 0 mean:

$$p(\Theta) = N(0, \lambda_1^{-1}I) \quad (12)$$

The likelihood function of the second term is composed as follows:

$$\begin{aligned} p(G|\Theta) &= \prod_{(h,r,t) \in \xi \times R \times \xi} p((h, r, t) | \Theta) \\ &= \prod_{(h,r,t) \in \xi \times R \times \xi} N(I_{h,r,t} - h^T R t, \lambda_2^{-1}) \end{aligned} \quad (13)$$

The likelihood function of the third term is expressed as follows:

$$\begin{aligned} p(Y|\Theta, G) &= \prod_{(u,v) \in Y} \left( \frac{\exp(u^T v) - \exp(-u^T v)}{\exp(u^T v) + \exp(-u^T v)} \right)^{y_{uv}} \\ &\quad \cdot \left( 1 - \frac{\exp(u^T v) - \exp(-u^T v)}{\exp(u^T v) + \exp(-u^T v)} \right)^{1-y_{uv}} \end{aligned} \quad (14)$$

In constructing the third likelihood function, because the features contained in the recommended data set are obviously different, we have introduced the characteristics of tanh function, which can continuously expand the relevant feature effects contained in the recommendation data set in the cycle process. The tanh function is beneficial to the training model, makes it easier to reach the optimal value, and can also ensure the recommendation model can drastically reduce losses and make recommendations more accurately.

Therefore, the loss function of ST\_RippleNet is as follows:

$$\begin{aligned} \min L &= -\log(p(Y|\Theta, G) \cdot p(G|\Theta) \cdot p(\Theta)) \\ &= \sum_{(u,v) \in Y} - \left( y_{uv} \log \left( \frac{\exp(u^T v) - \exp(-u^T v)}{\exp(u^T v) + \exp(-u^T v)} \right) \right. \\ &\quad \left. + (1 - y_{uv}) \log \left( 1 - \frac{\exp(u^T v) - \exp(-u^T v)}{\exp(u^T v) + \exp(-u^T v)} \right) \right) \\ &\quad + \frac{\lambda_2}{2} \sum_{r \in R} \|I_r - E^T R E\|_2^2 \\ &\quad + \frac{\lambda_1}{2} \left( \|V\|_2^2 + \|E\|_2^2 + \sum_{r \in R} \|R\|_2^2 \right) \end{aligned} \quad (15)$$

The embedding matrices of items and entities are represented by  $V$  and  $E$  [11], respectively, which are slices of indicator tensor  $I$  of relation  $R$  in KG, and  $R$  is the embedding matrix of relation  $r$  [11].

Through many experiments on music, movies and book data sets, it is found that using our loss function has better accuracy for evaluation indexes ACC and AUC, so using our loss function has a better recommendation effect on music, movies and books.

## 4. EXPERIMENTS

This section mainly describes the results and analysis of the ST\_RippleNet model and other mainstream advanced baseline recommendation models on three real data sets. First, the three data sets and evaluation indexes used in the experiment are introduced. Then, the existing advanced recommendation algorithms are analyzed and the performance of this model is compared and analyzed from different angles. Finally, the recommendation effects of the Last.FM data set, Book-Crossing data set and MovieLens-1M data set on this model are analyzed in detail.

## 4.1. Experimental Preparation

### 4.1.1. Data sets

In this paper, the Last.FM [18], Book-Crossing and MovieLens-1M data sets are used to verify the ST\_RippleNet model. Table 1 describes the characteristic data contained in these three data sets. The Last.FM [18] data set contains music records from 2000 users. The Book-Crossing dataset contains the scoring data of different books by each user. MovieLens-1M contains the scoring data of different movies by each user.

In addition, in Table 2, the super-parameter settings of each data set in the ST\_RippleNet model are given.

### 4.1.2. Evaluation index

The ACC and AUC were used to evaluate the ST\_RippleNet model. The ACC indicator is used to describe the classification ratio that is correctly predicted for the whole, namely,

$$ACC = \frac{I_r}{I_t} \quad (16)$$

where  $I_r$  represents the number of records correctly predicted and  $I_t$  represents the number of all test data.

The AUC index is a quantification of the ROC curve. Due to the threshold value problem, the ROC curve is not necessarily smooth and it is difficult to judge the performance of the model at this time. Therefore, the AUC index was selected to evaluate the model, and the area formed by the receiver operator characteristic curve Receiver Operator characteristic Curve (ROC) curve and false positive rate False Positive Rate (FPR) axis was used as the AUC value.

## 4.2. Mainstream Recommendation Models

The PER [23] model represents the connectivity between users and projects by extracting features based on meta-paths. In this article, we use “User-Musician-Genre-Musician,” “User-Actor-Movie Type-Actor” and “User-Book-Age-Book.”

The SHINE [24] model uses automatic encoders for user-project interaction and project profiles to predict click probabilities.

The DKN [25] model regards entity embedding and word embedding as multiple channels and combines them together in a CNN [16] for CTR prediction. In this article, we use the music name as the text input to the DKN. The dimension of word embedding and

entity embedding is 64, and the number of filters is 128 for each window size of 1, 2 and 3.

CKE combines CF [26] with structure, text and visual knowledge in a unified recommendation model [26]. In this paper, CKE is implemented as a CF + structure knowledge module. The user and item embedding dimensions of the music data set are set to 32. The dimension embedded in the entity is 32.

LibFM [27] is a feature-based decomposition model widely used in CTR scenes.

Wide & Deep [28] is a deep learning model that combines (wide) linear channels with (deep) nonlinear channels. Width and depth inputs are the same as LibFM. The dimension of users, projects and entities is 64. We use a two-layer depth channel with dimensions of 100 and 50, and a wide channel.

RippleNet [11] is a memory network-like approach that propagates user preferences on a KG [27] for recommendations.

## 4.3. Evaluation Results

### 4.3.1. Comparison of advanced baseline results

To illustrate the effectiveness of the recommendation algorithm based on KG propagation of user preferences, we compare the performance of the proposed method [29] with seven previously developed advanced recommendation algorithms, including CKE, LibFM, DKN, SHINE, PER, Wide & Deep and RippleNet. Table 3 shows the results obtained by the ST\_RippleNet model on the evaluation indices AUC and ACC, and its comparison to the other methods tested on the Last.FM [18], Book-Crossing, and MovieLens-1M data sets. As can be seen from the results, our model obtains the best results. On the music data set Last.FM [18], the AUC reaches 0.827 and the ACC reaches 0.763. On MovieLens-1M, the AUC reaches 0.925 and the ACC reaches 0.852. On the Book-Crossing data set, the AUC reaches 0.735 and the ACC reaches 0.693.

In addition, Figure 3 shows that the ST\_RippleNet model obtains the best recommendations from the music data set, followed by books and movies. Among the current mainstream recommendation algorithms, the DKN model has the worst recommendation results for music, books and movies, which indicates that it is necessary to mine the preference information of users and items at a deeper level. For movie and book recommendations, the RippleNet model outperforms several current mainstream recommendation models; however, it did not perform as well as the LibFM model in music recommendation. Compared with the RippleNet model, the two evaluation indexes, the AUC and ACC, the LibFM model

**Table 1** Data sets.

Dataset	#users#	#items#	#interactions#	#KGtriples#
Last.FM	1872	3846	42346	15518
Book-Crossing	17860	14910	139746	19793
MovieLens-1M	6036	2347	753772	20195

**Table 2** Data set super-parameter settings.

Last.FM	$d = 16, H = 2, \lambda_1 = 10^{-7}, \lambda_2 = 0.01, \eta = 0.001$
Book-Crossing	$d = 4, H = 3, \lambda_1 = 10^{-5}, \lambda_2 = 0.01, \eta = 0.001$
MovieLens-1M	$d = 16, H = 2, \lambda_1 = 10^{-7}, \lambda_2 = 0.01, \eta = 0.02$

increased by 1.17% and 2.6%, respectively. Among the current mainstream models, the RippleNet model has the best recommendation effect for movies and books, and the LibFM model has the best recommendation effect for music. The ST\_RippleNet model proposed in this paper not only addresses the problem of null values caused by data sparsity in data set by constructing KG, it also uses triples to describe data through music/book/movie recommendation KG. The triple is in the form of (entity-relationship-entity) or (entity-attribute-entity). By taking triple samples to mine the potential relationship between the user and the project, 3-tuple information is obtained through a triple multi-layer attention mechanism to more accurately recommend suitable music, movies or book for that user. It not only solves the problems of data sparsity, how to obtain more useful potential information from users and how to extract specific data from a large volume of information. Therefore, our ST\_RippleNet model not only has better recommendation performance than the RippleNet model in movie recommendation and book recommendation, its AUC evaluation index is increased by 0.4% and 0.8%, respectively, and the ACC is increased by 0.9% and 4.7%, respectively. Moreover, compared with the LibFM model, which has a good effect on music recommendation, our model has a better recommendation effect, and its AUC and ACC evaluation

indexes have increased by 6.4% and 7.6%, respectively. These results prove the effectiveness of the recommendation performance of the ST\_RippleNet model. In addition, we also made bar charts and line charts, as shown in Figures 4 and 5.

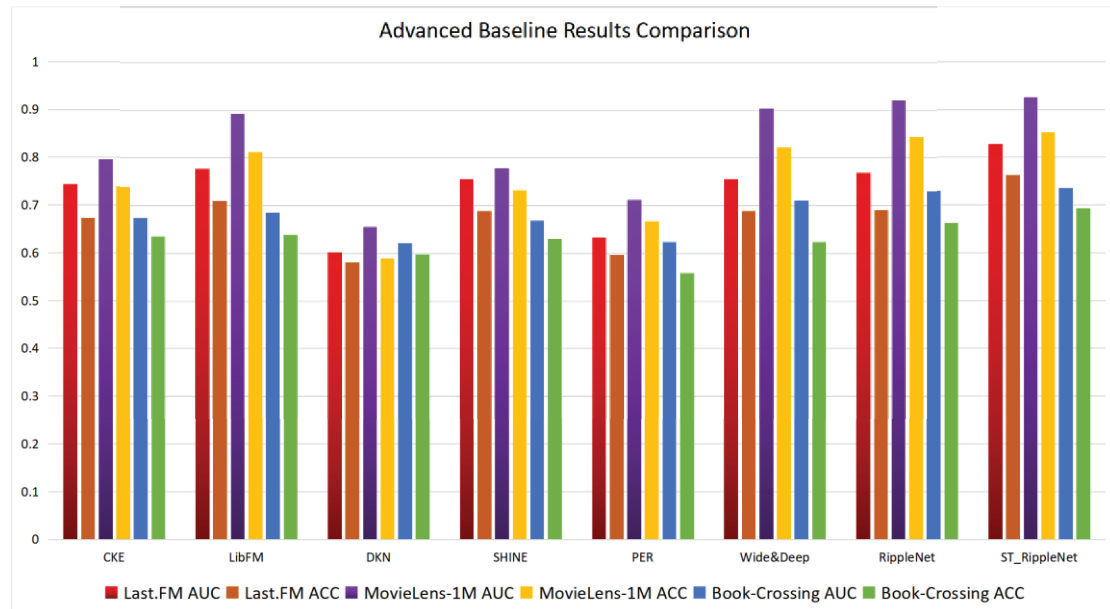
As seen from the column chart in Figure 4, the DKN model has the worst recommendation performance for the three data sets among the current mainstream recommendation models. The RippleNet model and LibFM model have the best effect, while the ST\_RippleNet model has better recommendation accuracy than RippleNet and LibFM. It can also be clearly seen from the line chart in Figure 5 that the AUC and ACC evaluation indexes for the ST\_RippleNet model in the three recommendation scenarios are better than those of the other mainstream recommendation models, as indicated by the evaluation indices of other mainstream recommendation models. Therefore, it can be proven that the ST\_RippleNet model is effective when making recommendations for each data set.

#### 4.3.2. Analysis of last.FM results

In ST\_RippleNet, we set the number of hops  $H = 2$ , the embedding dimension  $d = 16$  and the learning rate to 0.001 for the Last.FM

**Table 3** Comparison of the advanced baseline data results.

Model	MovieLens-1M		Book-Crossing		Last.FM	
	AUC	ACC	AUC	ACC	AUC	ACC
CKE	0.796(−16.2%)	0.739(−15.3%)	0.674(−9.1%)	0.635(−9.1%)	0.744(−11.2%)	0.673(−13.4%)
LibFM	0.892(−3.24%)	0.812(−4.9%)	0.685(−7.3%)	0.639(−8.5%)	0.777(−6.4%)	0.709(−7.6%)
DKN	0.655(−41.2%)	0.589(−44.7%)	0.621(−18.4%)	0.598(−15.9%)	0.602(−37.4%)	0.581(−31.3%)
SHINE	0.778(−18.9%)	0.732(−16.4%)	0.668(−9.1%)	0.631(−9.8%)	0.756(−9.4%)	0.688(−10.9%)
PER	0.712(−29.9%)	0.667(−27.7%)	0.623(−18.0%)	0.558(−24.2%)	0.633(−30.6%)	0.596(−28.0%)
Wide&Deep	0.903(−2.4%)	0.822(−3.6%)	0.711(−3.4%)	0.623(−11.2%)	0.756(−9.4%)	0.688(−10.9%)
RippleNet	0.921(−0.4%)	0.844(−0.9%)	0.729(−0.8%)	0.662(−4.7%)	0.768(−7.7%)	0.691(−10.4%)
ST_RippleNet	<b>0.925</b>	<b>0.852</b>	<b>0.735</b>	<b>0.693</b>	<b>0.827</b>	<b>0.763</b>



**Figure 4** Advanced baseline results comparison histogram.



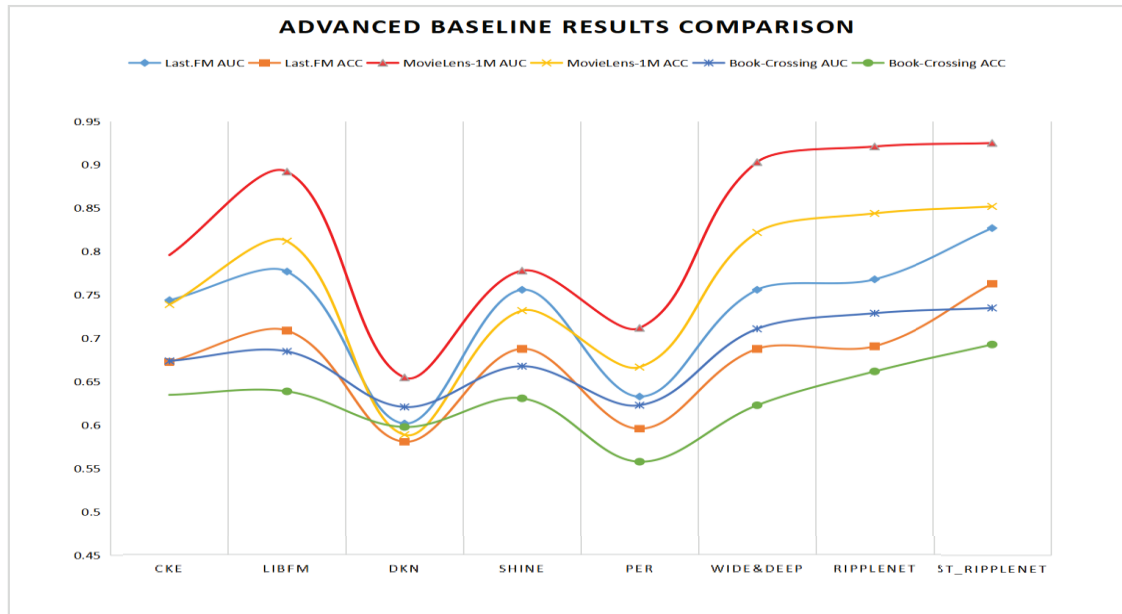


Figure 5 | Advanced baseline results comparison line chart.

Table 4 | Comparison of music recommendation results.

Model	Last.FM	
	AUC	ACC
CKE	0.744	0.673
LibFM	0.777	0.709
DKN	0.602	0.581
SHINE	0.756	0.688
PER	0.633	0.596
Wide&Deep	0.756	0.688
RippleNet	0.768	0.691
ST_RippleNet	0.827	0.763

dataset. The experimental results show that for the music dataset, an increase in the number of hops will barely improve the recommendation performance but will cause more overhead. To better compare the results, the parameters of our recommended methods are the same.

For the current mainstream recommendation algorithms, each algorithm will mine information about user-item interactions for recommendations, and different recommendation algorithms have different recommendation effects for different data sets. As seen from Table 4, the LibFM model has the best recommendation performance for music among the current mainstream recommendation models. When making music recommendations, the ST\_RippleNet model obtains more potential information from users by constructing music KGs. This information can be used not only to fill null values in the Last.FM dataset, it is also able to fully tap the interaction between users and projects. Through the triple attention mechanism, the triple information from the music KG is obtained and the potential information is mined. Moreover, the basic different characteristics, that is, the correlation degree between users and music are obtained in order to dig out the potential preference information of users. Then, it promotes the propagation of user preferences on the set of entities from the KG,

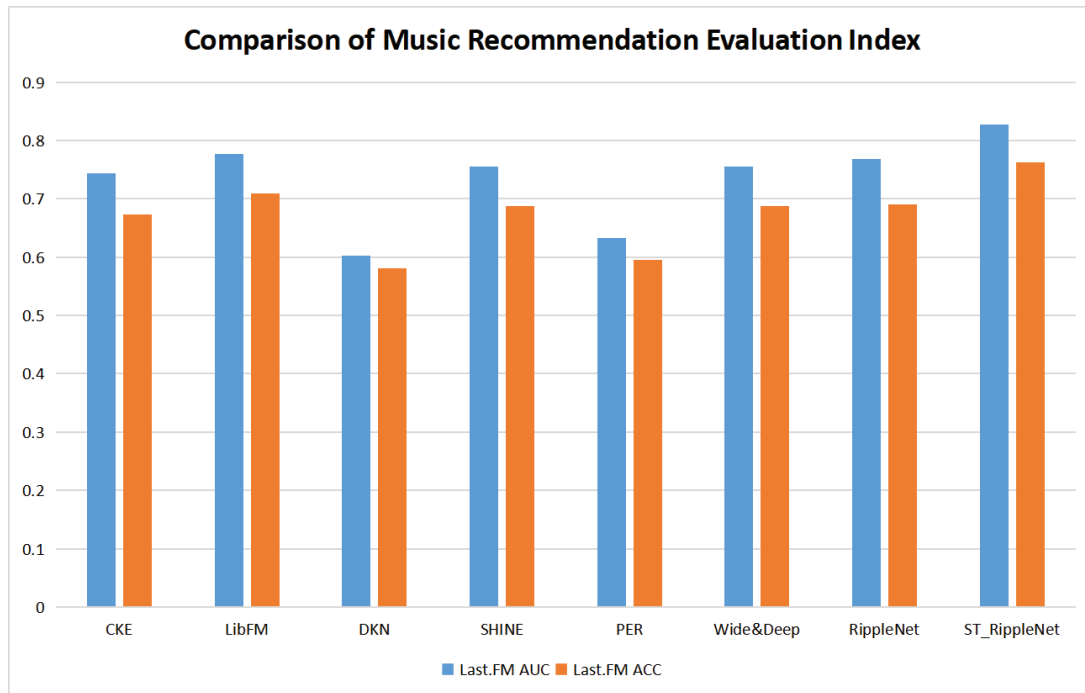
Preference propagation is realized by the triple attention mechanism. Finally, the user's preference distribution for candidate items formed by the user's historical click information is used to predict the final click prediction rate, better and more accurately acquire the connectivity between the user and music in the KG, and use it as a reference to provide appropriate music for the user, and the recommendation effect is excellent. Its AUC and ACC evaluation indices on the music data set reached 0.827 and 0.763, respectively.

For the Last.FM data set, this paper compares the ST\_RippleNet model with CKE, LibFM, DKN, SHINE, PER, Wide&Deep and RippleNet, and verifies that its AUC evaluation index has increased by 11.2%, 6.4%, 37.4%, 9.4%, 30.6%, 9.4% and 7.7%, respectively, with the highest increase being 37.4%. The ACC evaluation index also increased by 13.4%, 7.6%, 31.3%, 10.9%, 28.0%, 10.9% and 10.4%, with a maximum increase of 31.4% and a minimum increase of 7.6%. Therefore, the ST\_RippleNet model has excellent recommendation performance when recommending music. In addition, we also made bar charts and line charts, as shown in Figures 6 and 7.

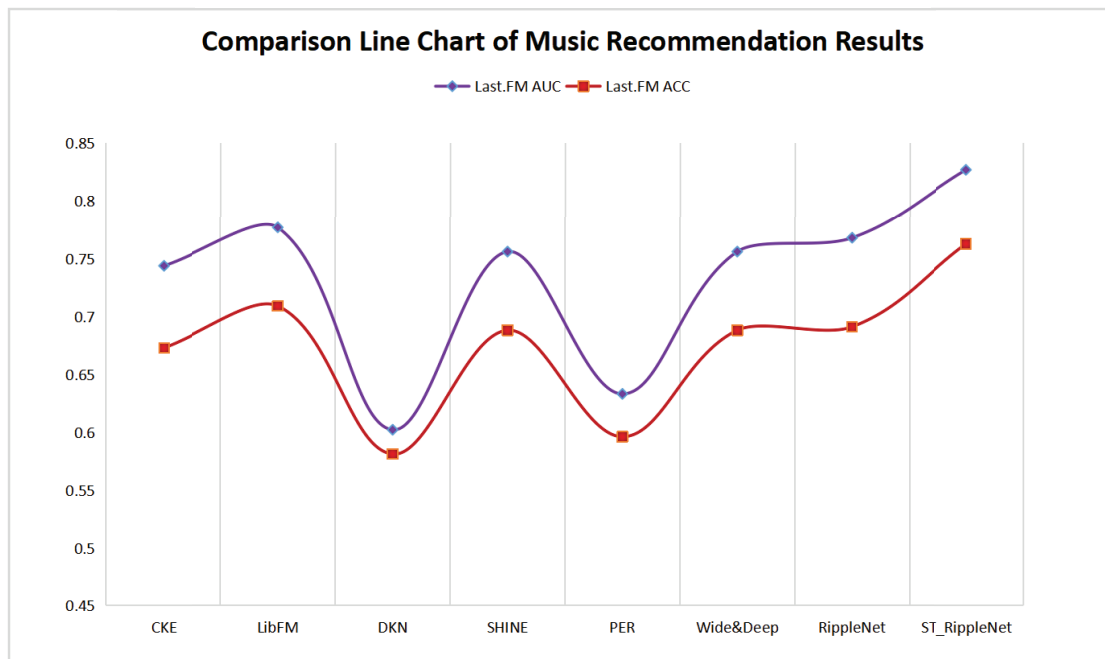
The ST\_RippleNet model is very effective when making recommendations using the Last.FM dataset. As seen from the histogram shown in Figure 6, the recommendation accuracy of the ST\_RippleNet model is better than that of the other models. It can be observed on the histogram that the two evaluation indices of our model are much higher than those of other models. From the line chart depicted in Figure 7, it can be seen that the AUC index of other mainstream recommendation models hardly exceeds 0.8, and that only the ST\_RippleNet model exceeds 0.8. Similarly, the ACC index of other mainstream models hardly exceeds or approaches 0.7, while that of ST\_RippleNet model exceeds 0.7 and approaches 0.8. This demonstrates the effectiveness of the ST\_RippleNet model.

#### 4.3.3. Analysis of MovieLens-1M and book-crossing data set results

In ST\_RippleNet, we set the number of hops for movie and book recommendations to  $H = 2$  and  $H = 3$ , respectively, the



**Figure 6** | Music recommendation results comparison column chart.



**Figure 7** | Comparison line chart of music recommendation results.

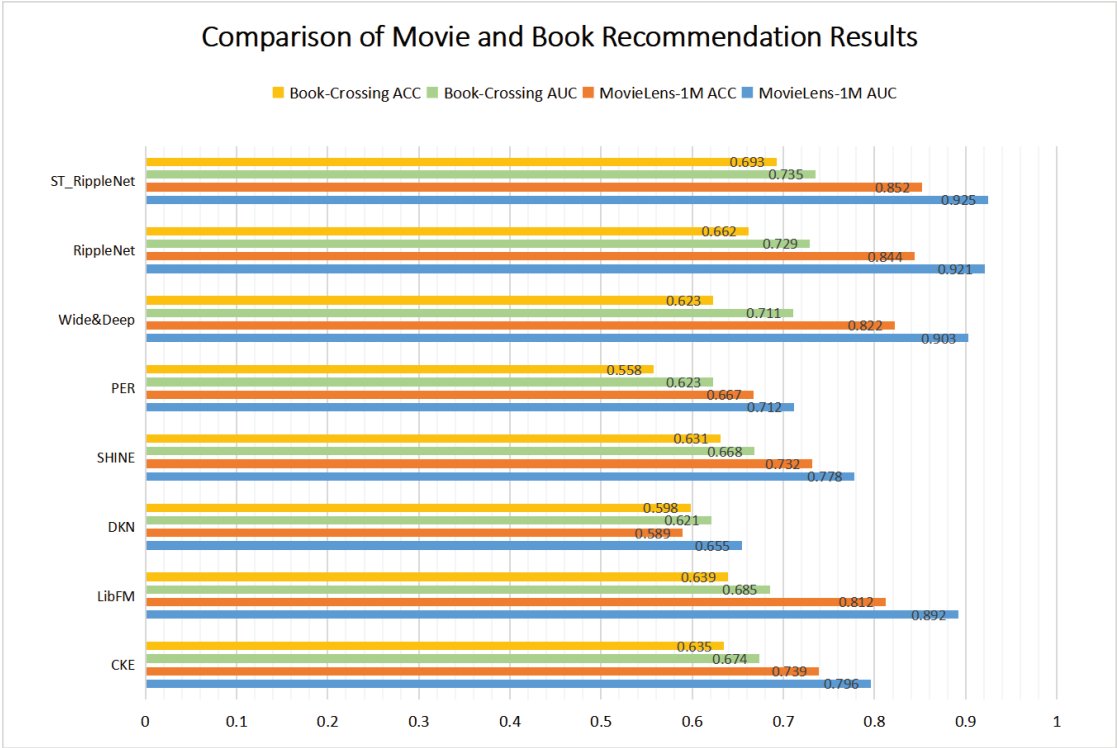
embedding dimension  $d$  of the item and KG as 16 and 4, respectively, and the learning rate as 0.02 and 0.001, respectively. The experimental results show that an increase in the number of hops will barely improve the recommendation performance but will cause more overhead. Table 2 gives the complete super-parameter settings.

For the current mainstream recommendation algorithms, each model obtains more potential interests of users by obtaining more potential information from users and the interaction between users

and projects. The ultimate goal is to provide more accurate recommendations for users through their potential interests and hobbies. In the current recommendation algorithms, for the MovieLens-1M data set and the Book-Crossing data set, the model with the best recommendation effect is RippleNet. From Table 5, it can be concluded that the AUC evaluation index for the movie data set and the book data set reach 0.921 and 0.844, respectively, and the ACC reaches 0.729 and 0.662, respectively. Although the LibFM model is also very effective in movie and book recommendations, the RippleNet

**Table 5** Comparison of recommendation results for movies and books.

Model	MovieLens-1M		Book-Crossing	
	AUC	ACC	AUC	ACC
CKE	0.796	0.739	0.674	0.635
LibFM	0.892	0.812	0.685	0.639
DKN	0.655	0.589	0.621	0.598
SHINE	0.778	0.732	0.668	0.631
PER	0.712	0.667	0.623	0.558
Wide&Deep	0.903	0.822	0.711	0.623
RippleNet	0.921	0.844	0.729	0.662
ST_RippleNet	0.925	0.852	0.735	0.693



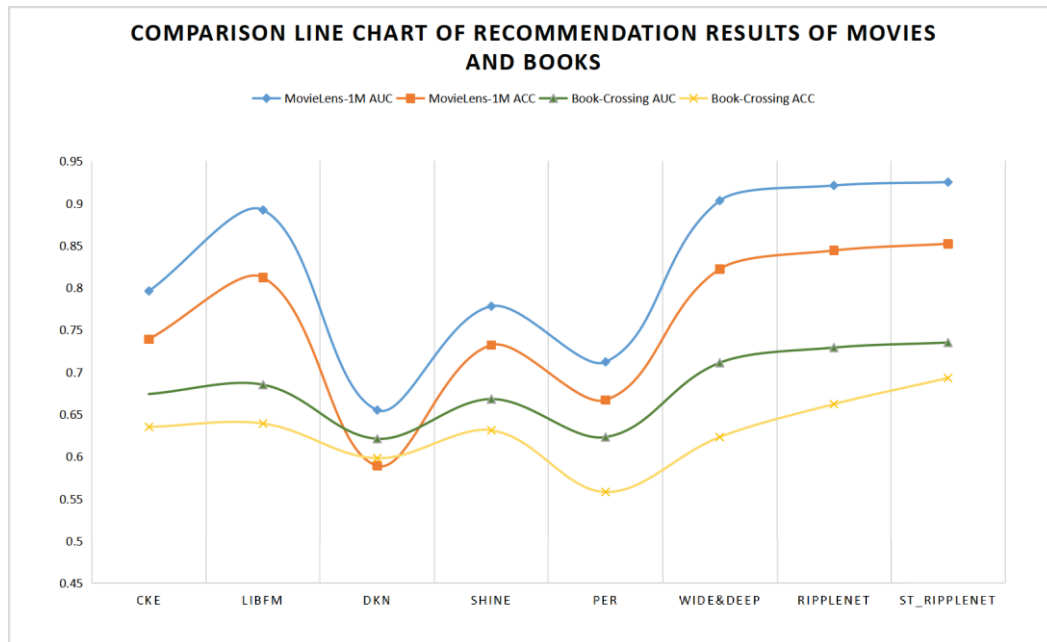
**Figure 8** Comparison of movie and book recommendation results.

model has better recommendation performance. Compared with those of the LibFM model, RippleNet’s AUC and ACC evaluation indices increased by 3.25% and 3.94%, respectively. When making book recommendations, the AUC and ACC evaluation indices increased by 6.42% and 3.60%, respectively.

Our ST\_RippleNet model not only solves the problem of null values caused by sparse data in data sets but also helps us obtain more potential interests and hobbies from users. Therefore, our ST\_RippleNet model has an excellent recommendation effect for the MovieLens-1M data set and Book-Crossing data set, and its AUC evaluation index reaches 0.925 and 0.732, respectively, while the ACC reaches 0.852 and 0.693, respectively. Its recommendation effect exceeds the performance of most current recommendation algorithms. Compared with the evaluation indexes of LibFM and RippleNet, the AUC and ACC increased by 3.24%, 4.9%, 0.4% and 0.9%, respectively, for film recommendations, and similarly, they

increased by 7.3%, 8.5%, 0.8% and 4.7%, respectively, for book recommendations. Therefore, ST\_RippleNet is effective when recommending movies and books. In addition, we also made bar chart Figure 8 and line chart Figure 9.

The recommendation accuracy of the ST\_RippleNet model for MovieLens-1M and Book-Crossing is clearly improved. As seen from the bar chart shown in Figure 8, the recommendation accuracy of the ST\_RippleNet model is better than that of the other models. On the bar chart, it can be observed that the two evaluation indices for our model are much higher than those of other models. From the line chart depicted in Figure 9, it can be seen that when compared with the evaluation indices of other mainstream recommendation models, the ST\_RippleNet model’s improvement is most clearly shown by the ACC index.. Therefore, it can be proven that the ST\_RippleNet model is effective when making recommendations from the MovieLens-1M and Book-Crossing data sets.



**Figure 9** | Comparison line chart of recommendation results of movies and books.

## 5. CONCLUSION

In this article, the ST\_RippleNet model we propose first obtains more potential information from users and projects by constructing a KG. In preparation for solving the problem of data sparsity, after that, the triple information in the KG is mined by the triple attention mechanism. Through the triple attention mechanism, we mine more potential information from users. This information can also be added to null values, provide more useful information to improve the recommendation effect, and construct a triple attention model to reason and mine more potential information from the triple. It also mines more potential user preference information from the KG and then efficiently uses the relationship structure information between entities in the KG through the user preference propagation model, taking into account the propagation intensity while propagating preferences, thus effectively improving the entity recommendation effect. In addition, the ST\_RippleNet model can better obtain the connectivity between users and projects in the recommendation system. Using book recommendation as an example, more information from users can be obtained through the connectivity between users and projects such as “user-writer-book type-writer,” “user-book-book type-book” and “user-writer-age-writer’s home” from the KG, to improve the accuracy of book recommendations. This approach addresses the limitations of the existing KG perception recommendation methods based on embedding and paths, and can accurately mine the potential interests of users, which improves the recommendation effect. We conducted extensive experiments in the three domains of music, movies and books. On the Last.FM music data set [18], the AUC evaluation index reaches 0.827 and the ACC evaluation index reaches 0.763. On the MovieLens-1M movie data set, the AUC reaches 0.925 and the ACC reaches 0.852. On the Book-Crossing data set, the AUC reaches 0.735 and the ACC reaches 0.693. The results show that ST\_RippleNet has obvious advantages over the current mainstream recommendation algorithms, and this model

can effectively solve the cold start problem as well. In the future, more detailed information on user interaction items, such as user stay time and number of clicks will be considered to further model user preferences.

## CONFLICTS OF INTEREST

The author declare they have no conflicts of interest.

## AUTHORS’ CONTRIBUTIONS

Zhisheng Yang: completed the realization of the model and completed many experiments. He also completed the analysis and preparation of the manuscript. He also performed well in data analysis and manuscript writing; Jinyong Cheng: provided the research, provided many valuable opinions in the improvement process of the model, and provided the environment needed for the experiment, and provided valuable opinions on the writing of the paper.

## ACKNOWLEDGMENTS

This work was supported in part by the National Key R&D Program (2019YFB2102600), China, and the Key Research and Development Project of Shandong Province (2019JZZY020124), China.

## REFERENCES

- [1] J. Chen, S. Xiu, X. Chen, H. Guo, X. Xie, Flounder-Net: an efficient CNN for crowd counting by aerial photography, *Neurocomputing*. 420 (2021), 82–89.
- [2] L. Fu, J. Zhang, X. Wu, *et al.*, De-anonymizing social networks with overlapping community structure, *IEEE/ACM Trans. Netw.* 28 (2020), 360–375.

- [3] A. Mahmoudi, M.R. Yaakub, A. Abu Bakar, The relationship between online social network ties and user attributes, *ACM Trans. Knowl. Discov. Data.* 13 (2019), 1–15.
- [4] H. Kwon, H. Yoon, K.-W. Park, Acoustic-decoy: detection of adversarial examples through audio modification on speech recognition system, *Neurocomputing.* 417 (2020), 357–370.
- [5] F. Chen, Y. Huang, Knowledge-enhanced neural networks for sentiment analysis of Chinese reviews, *Neurocomputing.* 368 (2019), 51–58.
- [6] D. Gupta, A. Ekbal, P. Bhattacharyya, A deep neural network framework for English Hindi question answering, *ACM Trans. Asian Low-Resour. Lang. Inf. Process.* 19 (2019), 1–22.
- [7] X. Zhang, S. He, X. Song, R.W.H. Lau, J. Jiao, Q. Ye, Image captioning via semantic element embedding, *Neurocomputing.* 395 (2020), 212–221.
- [8] Q. Ma, L. Yu, S. Tian, E. Chen, W.W.Y. Ng, Global-local mutual attention model for text classification, *IEEE/ACM Trans. Audio Speech Lang. Process.* 27 (2019), 2127–2139.
- [9] W. Liu, J. Liu, M. Wu, S. Abbas, W. Hu, B. Wei, Q. Zheng, Representation learning over multiple knowledge graphs for knowledge graphs alignment, *Neurocomputing.* 320 (2018), 12–24.
- [10] M. Canizo, I. Triguero, A. Conde, E. Onieva, Multi-head CNN-RNN for multi-time series anomaly detection: an industrial case study, *Neurocomputing.* 363 (2019), 246–260.
- [11] H. Wang, F. Zhang, J. Wang, M. Zhao, W. Li, X. Xie, M. Guo, RippleNet: propagating user preferences on the knowledge graph for recommender systems, in *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, Turin, Italy, ACM, 2018.
- [12] W. Lu, X. Zhang, H. Lu, F. Li, Deep hierarchical encoding model for sentence semantic matching, *J. Visual Commun. Image Represent.* 71 (2020), 102794.
- [13] W. Zhang, X. Zhang, H. Wang, D. Chen, A deep variational matrix factorization method for recommendation on large scale sparse dataset, *Neurocomputing.* 334 (2019), 206–218.
- [14] L. Weidong, P. Rong, W. Yaqian, Y. Zhihuan, Knowledge graph based natural language generation with adapted pointer-generator networks, *Neurocomputing.* 382 (2020), 174–187.
- [15] S. Li, H. Wang, R. Pan, M. Mao, MemoryPath: a deep reinforcement learning framework for incorporating memory component into knowledge graph reasoning, *Neurocomputing.* 419 (2021), 273–286.
- [16] L. Huang, Y. Xia, Joint blur kernel estimation and CNN for blind image restoration, *Neurocomputing.* 396 (2020), 324–345.
- [17] G. Zhu, C. Bin, T. Gu, W. Chen, Z. Jia, Neural modeling framework of user preference based on knowledge graph, *Pattern Recognit. Artif. Intell.* 32 (2019), 661–668.
- [18] H. Wang, F. Zhang, M. Zhao, *et al.*, Multi-task feature learning for knowledge graph enhanced recommendation, in the *World Wide Web Conference*, San Francisco, CA, USA, 2019, pp. 2000–2010.
- [19] Z. Ali, G. Qi, K. Muhammad, B. Ali, W.A. Abro, Paper recommendation based on heterogeneous network embedding, *Knowl.-Based Syst.* 210 (2020), 106438.
- [20] A. Grover, J. Leskovec, node2vec: scalable feature learning for networks, in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'16)*, ACM, San Francisco, CA, USA, 2016, pp. 855–864.
- [21] Q. Wang, Y. Ji, Y. Hao, J. Cao, GRL: knowledge graph completion with GAN-based reinforcement learning, *Knowl.-Based Syst.* 209 (2020), 106421.
- [22] J. Bernabé-Moreno, A. Tejada-Lorente, J. Herce-Zelaya, C. Porcel, E. Herrera-Viedma, A context-aware embeddings supported method to extract a fuzzy sentiment polarity dictionary, *Knowl.-Based Syst.* 190 (2020), 105236.
- [23] X. Lin, M. Zhang, Y. Liu, S. Ma, Enhancing personalized recommendation by implicit preference communities modelling, *ACM Trans. Inf. Syst.* 37 (2019), 1–32.
- [24] K. Sunil, T. Adheeba, M. Manju, G. Gopakumar, A heterogeneous information network model for long non-coding RNA function prediction, *IEEE/ACM Trans. Comput. Biol. Bioinform.* 0 (2020), 1–1.
- [25] D. Cai, G. Wu, Content-aware attributed entity embedding for synonymous named entity discovery, *Neurocomputing.* 329 (2019), 237–247.
- [26] Y. Hu, F. Xiong, D. Lu, X. Wang, X. Xiong, H. Chen, Movie collaborative filtering with multiplex implicit feedbacks, *Neurocomputing.* 398 (2020), 485–494.
- [27] Basawaraj, J.A. Starzyk, H. Adrian, Episodic memory in mini-column associative knowledge graphs, *IEEE Trans. Neural Netw. Learn. Syst.* 30 (2019), 3505–3516.
- [28] W. Lu, Y. Zhang, S. Wang, H. Huang, Q. Liu, S. Luo, Concept representation by learning explicit and implicit concept couplings, *IEEE Intell. Syst.* 36 (2020), 6–15.
- [29] M. Aljanabi, M.A. Ismail, V. Mezhyuev, H. Garg Improved TLBO-JAYA algorithm for subset feature selection and parameter optimisation in intrusion detection system, *Complexity.* 2020 (2020), 1–18.