Recommendation Algorithm Based on Knowledge Graph to Propagate User Preference

Zhisheng Yang

Qilu University of Technology(Shandong Academy of Sciences) https://orcid.org/0000-0003-2458-5114

Jinyong Cheng (cjy@qlu.edu.cn)

School of Computer Science and Technology, Qilu University of Technology (Shandong Academy of Sciences), Jinan, 2500353, China https://orcid.org/0000-0003-3432-4831

Research

Keywords: recommendation algorithm, Knowledge Graph, Preference propagation

Posted Date: January 18th, 2021

DOI: https://doi.org/10.21203/rs.3.rs-139847/v1

Recommendation Algorithm Based on Knowledge Graph to Propagate User Preference

Zhisheng Yang,Jinyong,Cheng*[1]

School of Computer Science and Technology,

Qilu University of Technology (Shandong Academy of Sciences),

Jinan, 2500353, China

Abstract

In recommendation algorithms, data sparsity and cold start problems are always inevitable. In order to solve such problems, researchers apply auxiliary information to recommendation algorithms to mine and obtain more potential information through users' historical records and then improve recommendation performance. This paper proposes a model ST_RippleNet, which combines knowledge graph with deep learning. In this model, users' potential interests are mined in the knowledge graph to stimulate the propagation of users' preferences on the set of knowledge entities. In the propagation of preferences, we adopt a triple-based multi-layer attention mechanism, and the distribution of users' preferences for candidate items formed by users' historical click information is used to predict the final click probability. In ST_RippleNet model, music data set is added to the original movie and book data set, and the improved loss function is applied to the model, which is optimized by RMSProp optimizer. Finally, tanh function is added to predict click probability to improve recommendation performance. Compared with the current mainstream recommendation methods, ST_RippleNet recommendation algorithm has very good performance in AUC and ACC, and has substantial improvement in movie, book and music recommendation.

Key words: recommendation algorithm; Knowledge Graph; Preference propagation.

*Corresponding author: Jinyong Cheng (e-mail: cjy@qlu.edu.cn).

1 Introduction

In modern online platforms, recommendation systems play a vital role in making users pay attention to personalized content. Users expect to obtain personalized content on modern e-commerce, entertainment and social media platforms, but the effectiveness of recommendations is limited by existing user-project interaction and model capacity. The explosive growth of online content and services has provided users with a large number of choices, such as movies, music and books. In order to improve the recommendation effect, Researchers are always thinking of ways to improve recommendation performance, judging the similarity of preferences among users through the interaction between users, or recommending from high to low by collecting their preferences and building tables for multiple users with the same preferences, and solving the problem of recommendation accuracy through various ways. In the process of improving the recommendation algorithm, researchers have applied three-way neural network, meta-path, attribute reuse structure, CNN[1], GAN[26] and other methods to the recommendation algorithm, and achieved good results. Among them, collaborative filtering mines potential preference information by analyzing users' historical records and then makes recommendations. The sparsity of user-project interaction and cold start always interfere with the recommendation effect. For data sparse and cold start problems, researchers have put forward many ideas, and the idea of integrating auxiliary information such as social network[2], user/project attributes[3], images[4] and context[5] into CF has achieved good results.

For auxiliary information, the knowledge map contains various interrelationships between users and projects. Some recently proposed knowledge maps have been used in applications such as question answer [6], KG completion [9], text classification [8] and word embedding [7], such as Microsoft Satori's knowledge map, which has achieved good results. In order to better mine potential information through the knowledge map, we have made the knowledge map of the music recommendation system as shown in Figure1.

In this age of recommendation algorithms, The recommendation system not only plays a great role in shopping software such as Taobao and Jingdong, There are also video software such as Aiqiyi and Tencent, as well as music such as QQ Music and Netease Cloud Music. It is these online applications

that continuously promote the development of recommendation algorithms and even 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[12], mainly use historical user-project interaction records to simulate users' preferences for projects; There are also recommendations made through the similarity function, recommendation learning is carried out through human judgment on the similarity of objects[13], accurate similar neighbors between users or project requirements are captured according to their historical common evaluation, and then appropriate projects or items are recommended. The intelligent recommendation system[10] can make appropriate music, movies and books through different hobbies of users[14], and is widely used to realize accurate matching between users and various resources. Nowadays, various kinds of auxiliary data are becoming more and more 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 by introducing semantic associations between items[17], consisting of various types of interaction relationships, and linking user history with recommendation records.

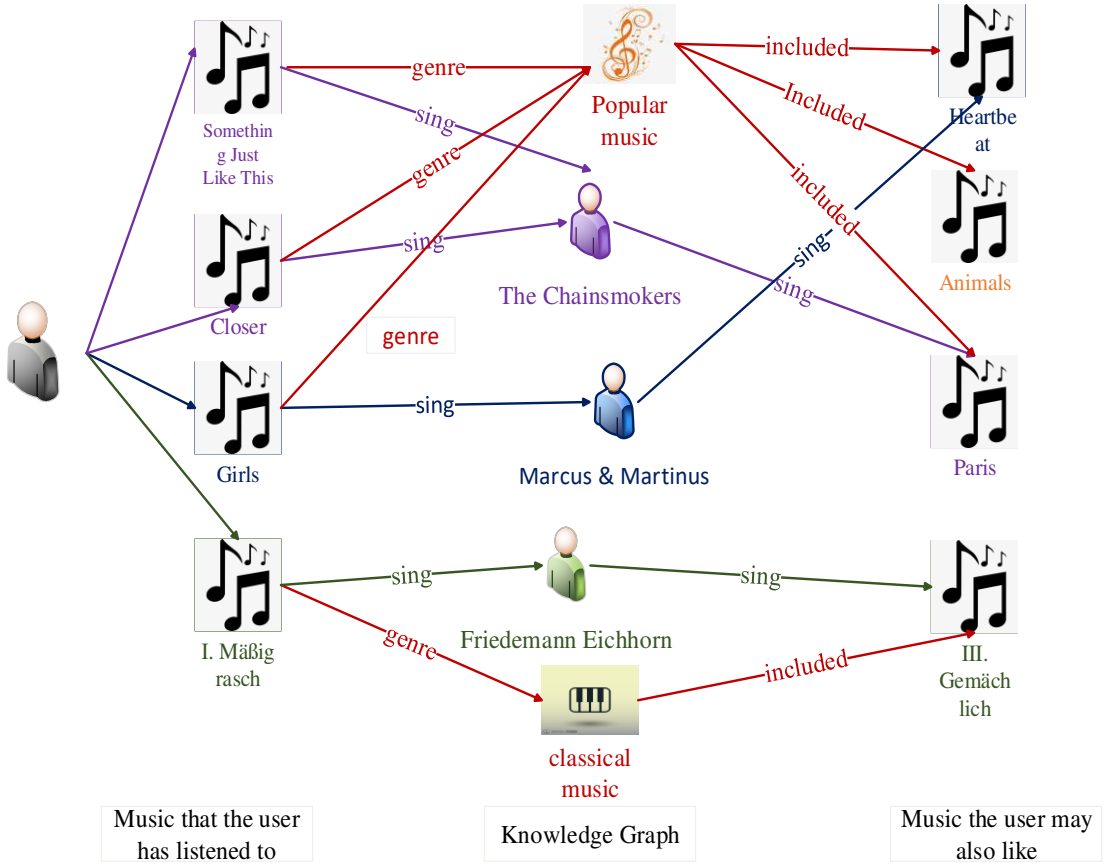

Fig.1: The knowledge graph of music recommendation system provides rich interaction between users and music, which is helpful to mine potential user preference information.

This paper proposes a knowledge graph recommendation model ST_RippleNet to deal with the defects of some current recommendation algorithms. ST_RippleNet can not only introduce KGE method into recommendation, but also mine potential information that users may choose.In this paper, when ST_RippleNet is applied to music, books and movies, it is found that compared with the current mainstream recommendation[28] 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%.

On the dataset, In addition to applying ST_RippleNet to the original dataset MovieLens-1M and Book-Crossing, We also added the music data set Last.FM, and found that ST_RippleNet has better effect than other baselines in music recommendation. In addition, we improved the loss function of the framework and applied the characteristics of tanh function when making the loss function, while we used RMSProp optimizer when optimizing and tanh function when predicting the click probability. In

the experiment, we found that RMSProp optimizer is more suitable for our framework ST_RippleNet than other optimizers.

Through experiments on Last.FM, Book-Crossin and MovieLens-1M data sets, the effectiveness of ST_RippleNet in recommendation system is proved according to its evaluation index and recommendation effect.

2 Materials and methods

2.1 Recommendation Algorithm

In this article, we use all project attributes project features for PER[18] (for example, "Music-Singer-Music").

SHINE[19] Use automatic encoders for user-project interaction and project profiles to predict click probability.

DKN regards entity embedding[20] and word embedding as multiple channels and combines them together in CNN[16] for CTR prediction.

CKE combines CF[21] with structure, text and visual knowledge in a unified recommendation mode].

LibFM is a feature-based decomposition model widely used in CTR scenarios.

Wide&Deep is a deep learning[23] model that combines (wide) linear channels with (deep) nonlinear channels.

RippleNet[12] is a method similar to memory network, which propagates user preferences on knowledge graphs[22] to make recommendations. The last super parameter setting[12] FM is d = 8, H = 2, $\lambda_1 = 106$, $\lambda_2 = 0.01$, $\eta = 0.02$.

2.2 Attention Mechanism

Attention mechanism is generated to replace the traditional CNN and RNN[11] structures and obtain the required information by paying attention to local important information, while

recommendation algorithm is to obtain accurate information for high-precision recommendation. The attention mechanism is mathematically understood as weighted summation. Formally speaking, it is a key-value query. From the perspective of the room, it can be understood as similarity measurement. ST_RippleNet uses a multi-level attention module based on knowledge triple[12] for preference propagation, in which tails are weighted evenly by the similarity between their related heads, tails and specific items[12].

2.3 Data Set and Evaluation Index

In this paper, Last.FM[24], Book-Crossing and MovieLens-1M data sets are used to verify ST_RippleNet model. Table 1 describes the characteristic data contained in these three data sets. The Last.FM[24] dataset 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.

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, $\lambda_1 = 10^{-7}$, $\lambda_2 = 0.01$, $\eta = 0.02$

Accuracy (ACC) and area under curve (AUC) are 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

$$ACC = \frac{I_r}{I_t} \tag{16}$$

Where I_r represents the number of records correctly predicted, and I_t represents the number of all test data.

AUC index is a quantification of ROC curve. Due to the problem of threshold value, ROC curve is not necessarily smooth, and it is difficult to judge the performance of the model at this time. Therefore, AUC is selected to evaluate the model, and the area formed by ROC curve and FPR axis is the value of AUC.

2.4 ST_RippleNet model

This paper proposes a recommendation algorithm based on knowledge graph to propagate user preferences, Its basic idea is: Taking music recommendation as an example, Firstly, the user-project knowledge graph is constructed, that is, the music recommendation knowledge graph, By using node2vec[25] to extract the structural features of music recommendation knowledge graph, Vectorized representation of its knowledge, Embedding the entities in the knowledge graph and the relationships among them into 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, i.e. The basic characteristics of different characteristics are taken as the weight value of transmission intensity. Furthermore, using the structural information of the music recommendation knowledge graph[26] 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 learned to generate top N recommendation list, and then the click rate is predicted.

This algorithm utilizes the integration of knowledge graph to multi-source heterogeneous data, The low-dimensional vector is introduced into the recommendation system, which makes full use of the knowledge information of the knowledge graph. 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 knowledge graph 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 information utilization problem of knowledge graph and improves the recommendation performance. The structure diagram is shown in Fig. 2.

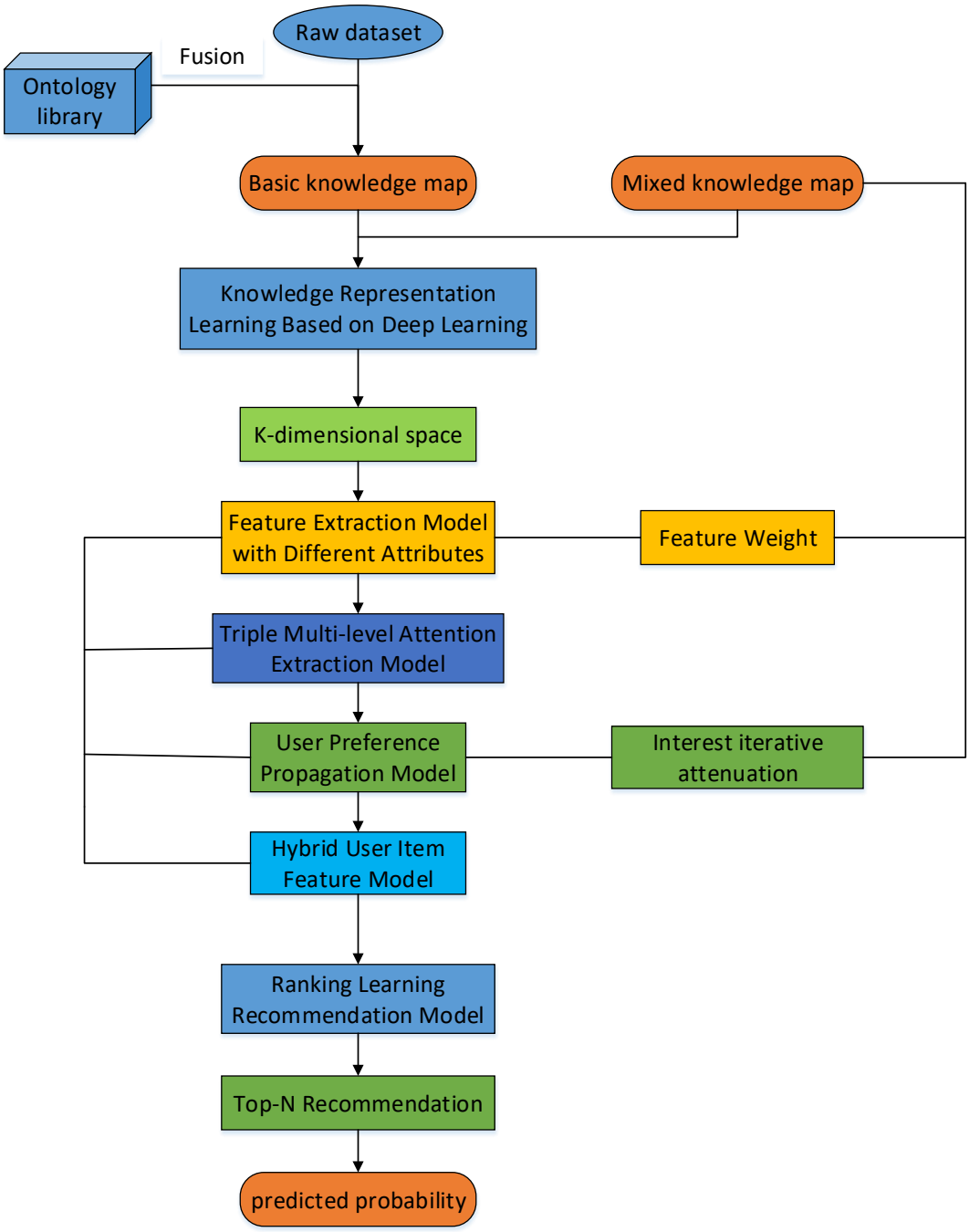

Fig. 2 Recommendation algorithm for spreading user preferences based on knowledge graph

ST_RippleNet network structure

In ST_RippleNet, given the interaction matrix Y and the knowledge graph G, the set of k-hop related entities we define for user u is

$$\xi_u^k = \{t \mid (h,r,t) \in G \ and \ h \in \xi_u^{k-1}\}, k = 1,2,\ldots,H. \tag{1}$$

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

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

$$M_u^k = \{(h,r,t) | (h,r,t) \in G \ and \ h \in \xi_u^{k-1}\}, k = 1,2,\ldots,H. \tag{2}$$

In the ST_RippleNet model, we use preference propagation technology[12] to mine hidden interests and hobbies in users' ripple sets, so as 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[12], attribute, word package or context information[27] 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:

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

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 the item v and the entity h_i measured in the space of the 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:

(4)

$$
\begin{aligned}
o_u^1 &= \sum\nolimits_{(h_i, r_i, t_i) \in M_u^1} p_i t_i \\
&= \sum\nolimits_{(h_i, r_i, t_i) \in M_u^1} \left(soft\max\left(v^T R_i h_i \right) \right) t_i \\
&= \sum\nolimits_{(h_i, r_i, t_i) \in \mathrm{M}_u^1} \left(\frac{\exp\left(v^T R_i h_i \right)}{\sum\nolimits_{(h, r, t) \in \mathrm{M}_u^1} \exp\left(v^T R_i h_i \right)} \right) t_i
\end{aligned}
$$

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 the user u relative to the music v is represented by a vector o_u^1. The second-order response o_u^2 of the user u is obtained by performing preference propagation and this process can be iterated on its ripple set M_u^i to obtain $\mathrm{i} = 1, \ldots, H$, 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 embedding is also calculated through all responses.

$$
u = o_u^1 + o_u^2 + \ldots + o_u^H \tag{5}
$$

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 tanh function to predict the click probability:

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

2.5 Model optimization

RMS (Root Mean Square) is an abbreviation of root mean square. Like momentum gradient reduction, it is all a way to accelerate gradient reduction by removing sloshing during the whole process of gradient reduction. Gradient Upgrade Formula Calculation:

When the weight value is upgraded, The way the Eradication number is applied, The large

gradient can be greatly reduced, while the smaller gradient can be reduced by a smaller margin, so that the fluctuation in the orientation of the large gradient can be reduced, and the shaking in the whole process of all gradient reduction will be smaller, so that a large learning-rate can be set, the pace of learning and training can be increased, and the purpose of accelerating learning and training can be achieved.

In the specific application, the weight value W or B is usually a combination of weight values of many levels, which is multi-dimensional. In the actual operation of eradicating numbers, the gradient of large levels will be greatly reduced, not to say that the trend analysis of weight value W is the same. The formula for the RMSProp optimizer is as follows:

γ : Power, usually set to 0.9.

η :The value is generally 0.001

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

$$g_t = \nabla w J(W) \tag{7}$$

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

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

RMSProp algorithm has long been proved to be a reasonable and easy-to-use deep neural network optimization algorithm in working experience. At present, it is one of the optimization methods often selected by deep neural network practitioners.

2.6 Loss function

In the ST_RippleNet model, for a given knowledge graph G, combined with its implicit feedback matrix Y, the following posterior probabilities of model parameters are maximized: we hope the posterior probabilities are as follows:

$$\max p(\Theta | G, Y) \tag{10}$$

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

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

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

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

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

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

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

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

Therefore, the loss function of ST_RippleNet is:

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

The embedding matrices of items and entities are respectively represented by V and E[12], which are slices of indicator tensor I of relation R in KG, and R is the embedding matrix of relation r[12].

3 Result

This section mainly describes the results and analysis of ST_RippleNet model and other mainstream advanced baseline recommendation models under three real data sets. Firstly, the results of ST_RippleNet model and other algorithms on three data sets are compared in

experiments, and then the performance of this model is compared and analyzed from different angles. Finally, the recommendation effects of music data set Last.fm, book-Crossing and movie data set MovieLens-1M on the model are analyzed in detail.

3. 1 Result Analysis of Advanced Recommendation Algorithm

In this paper, a comparative experiment is designed to verify the ST_RippleNet model. The comparative models include CKE, LibFM, DKN, SHINE, PER, Wide-Deep and RippleNet. Table 3 describes the evaluation index for each algorithm. From the chart, it can be seen that ST_RippleNet model has the best influence on music data set, followed by books and finally movies. Among the current mainstream recommendation algorithms, DKN model has the worst recommendation effect for music, books and movies, which indicates that it is necessary to mine the preference information of users and items in a deeper level. For movie recommendation and book recommendation, RippleNet model defeated several current mainstream recommendation models. However, it was defeated by LibFM model in music recommendation. Compared with RippleNet model, the two evaluation indexes AUC and ACC of LibFM model increased by 1.17% and 2.6% respectively. Among the current mainstream models, RippleNet model has the best recommendation effect for movies and books, while LibFM model has the best recommendation effect for music.

The model ST_RippleNet proposed in this paper not only has better recommendation performance than RippleNet model in movie recommendation and book recommendation, but also its evaluation index AUC is improved by 0.4% and 0.8% respectively, and ACC is improved by 0.9% and 4.7% respectively. Moreover, compared with LibFM model, which has a good effect on music recommendation, our model also has a better recommendation effect, and its evaluation indexes AUC and ACC have increased by 6.4% and 7.6% respectively. Therefore, it can prove the effectiveness of the recommended performance of ST_RippleNet model. In addition, we have also made histograms and line charts as shown in Fig.3 and Fig.4.

Table 3 Comparison of 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	**0.925**	**0.852**	**0.735**	**0.693**	**0.827**	**0.763**

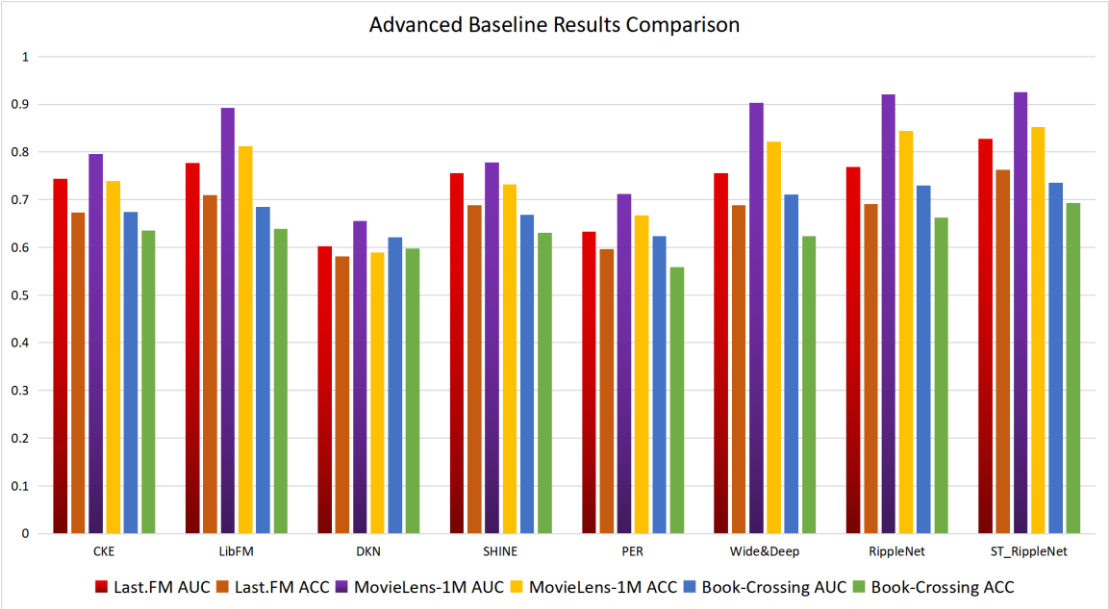

Fig.3. Advanced Baseline Results Comparison Histogram

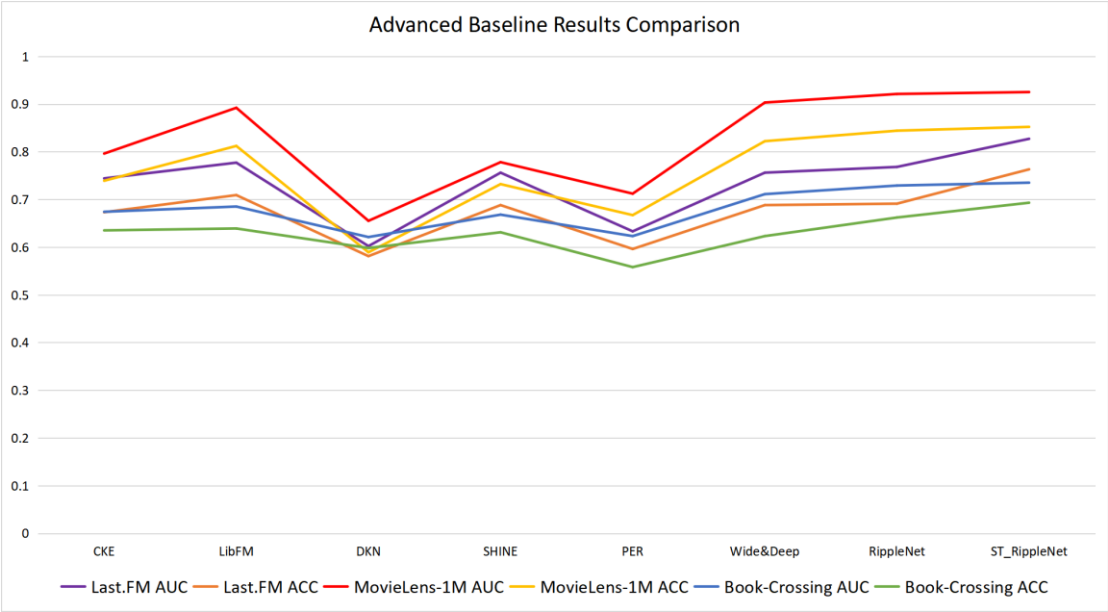

Fig.4. Advanced Baseline Results Comparison Line Chart

As can be seen from the column chart Fig. 3, it can be seen that DKN model has the worst recommendation performance for the three data sets among the current mainstream recommendation models, RippleNet model and LibFM model have the best effect, while ST_RippleNet model has better recommendation accuracy than RippleNet and LibFM. It can also be clearly seen from the line chart in FIG. 4 that the evaluation indexes AUC and ACC of ST_RippleNet model in the three recommendation scenarios are better than other mainstream recommendation models. See the evaluation index of other mainstream recommendation models. Therefore, it can be proved that the ST_RippleNet model is effective for the recommended performance of each data set.

3.2 Last.FM Result Analysis

In ST_RippleNet, we set the number of hops H = 2, the embedding dimension d = 16 and the learning rate 0.001 for the music dataset Last.FM. The experimental results show that for the music dataset, the increase of the number of hops will hardly improve the recommendation performance, but will cause more overhead. In order to compare the results better, the parameters of our recommended methods are the same.

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

When making music recommendations, ST_RippleNet model fully applies the interaction relationship between users and items to mining users' potential preferences. It not only obtains the similarity between users' items through knowledge graphs, but also obtains the correlation degree between users, music and music, and users and music, i.e. The basic characteristics of different characteristics, in order to mine users' potential preferences information. In the current mainstream recommendation algorithms, each algorithm will carry out information mining on the user-item interaction relationship for recommendation. Different recommendation algorithms have different recommendation effects on different data sets. For example, comparing LibFM and RippleNet, LibFM model is better than RippleNet on Last.FM, while RippleNet's recommendation performance is better than LibFM on MovieLens-1M and Book-Crossing. As can be seen from Table 4, LibFM model has the best recommendation performance for music among the current mainstream recommendation models.

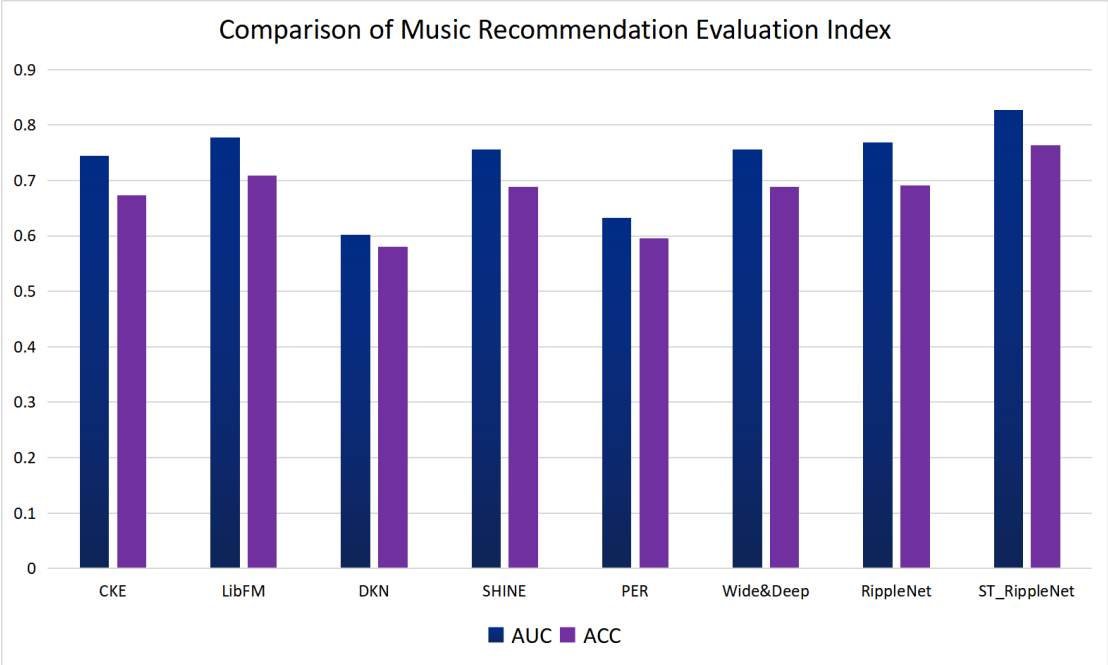

Fig.5. Music Recommendation Results Comparison Column Chart

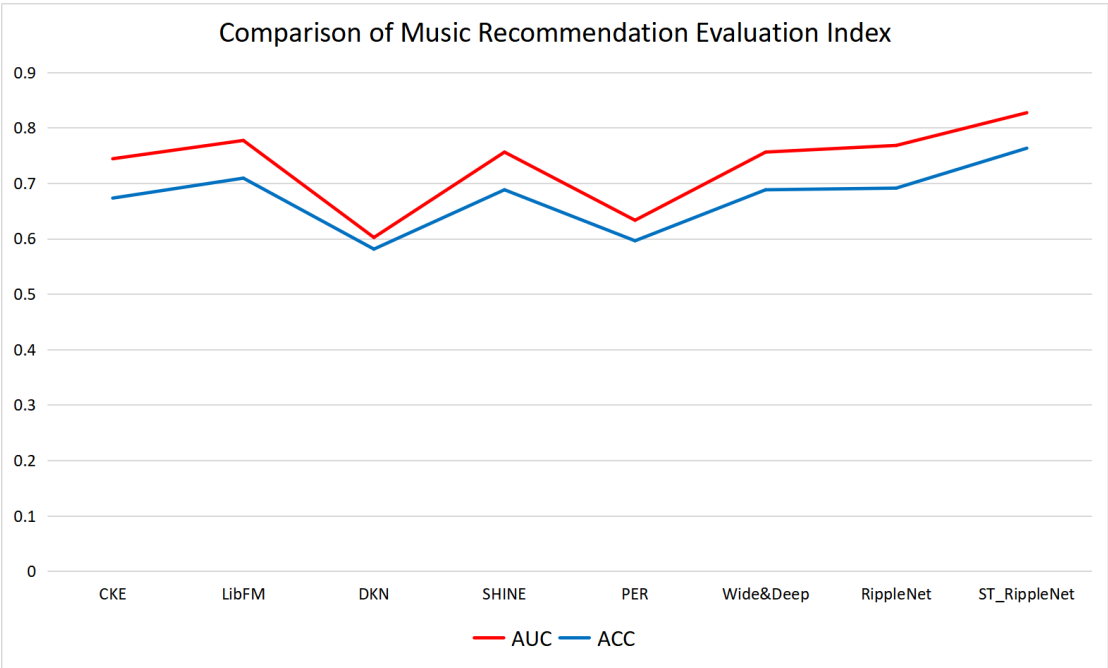

Fig.6. Comparison Line Chart of Music Recommendation Results

The ST_RippleNet model is very effective in recommending the music dataset Last.FM. As can be seen from the histogram shown in FIG. 5, the recommendation accuracy of ST_RippleNet model is better than that of other models. It can be observed on the histogram that the two evaluation indexes of

our model are much higher than that of other models. From the line chart depicted in FIG. 6, it can be seen that the AUC index of other mainstream recommendation models hardly exceeds 0.8, 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 shows the effectiveness of ST_RippleNet model for the recommended performance of Last.FM data set.

3.3 MovieLens-1M and Book-Crossing Result Analysis

In ST_RippleNet, we set the number of hops for movie and book recommendation as H=2 and H=3 respectively, the embedding dimension d of item and knowledge graph as 16 and 4 respectively, and the learning rate as 0.02 and 0.001 respectively. The experimental results show that the increase of the number of hops will hardly improve the recommendation performance, but will cause more overhead. Table 2 gives the complete super parameter settings.

Table 5 Comparison of Recommendation Results of 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

Among the current mainstream recommendation algorithms, RippleNet model has the best recommendation effect for movies and books, while LipFM model has the best recommendation effect for music. LibFM model is also very effective in movie and book recommendation, but RippleNet model has better recommendation performance. Compared with LipFM model, RippleNet's evaluation indexes AUC and ACC increased by 3.25% and 3.94% respectively in movie recommendation. On the recommendation of books, the evaluation indexes AUC and ACC increased by 6.42% and 3.60% respectively.

Among the current mainstream recommendation models, it can be seen from Table 5 that RippleNet model has the best recommendation performance for movies and books, while our model ST_RippleNet has increased 3.24%, 4.9%, 0.4% and 0.9% respectively in movie recommendation compared with LIbFM and RippleNet's evaluation indexes AUC and ACC, and similarly, has increased 7.3%, 8.5%, 0.8% and 4.7% respectively in book recommendation. This proves the effectiveness of ST_RippleNet for movie and book recommendation. In addition, we have also made histogram Fig.7 and line diagram Fig.8.

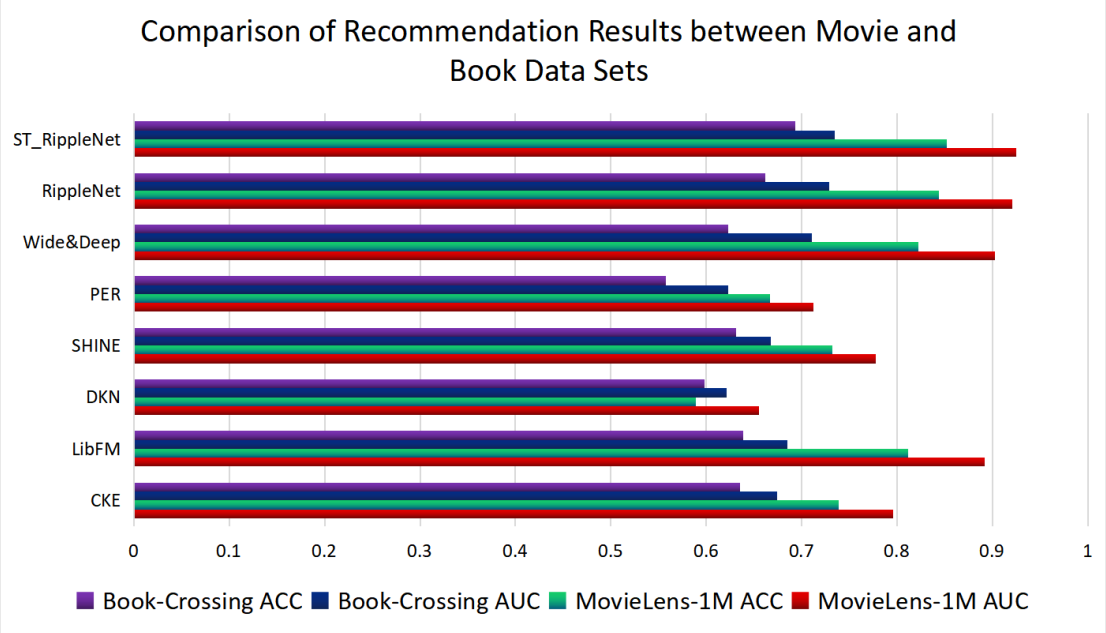

Fig.7. Comparison of Movie and Book Recommendation Results Bar Chart

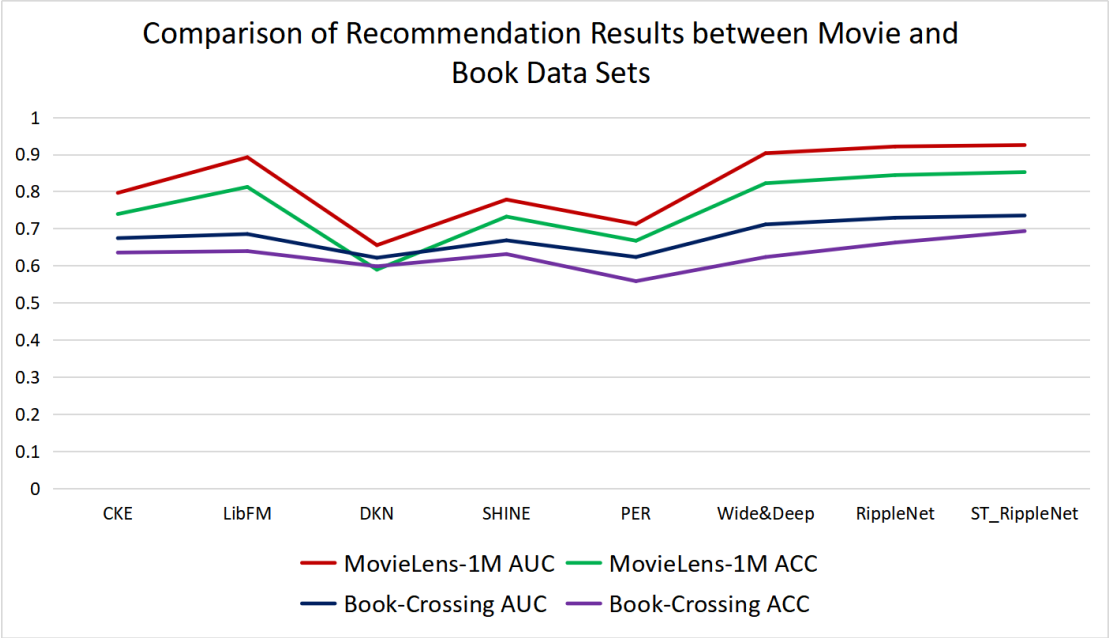

Fig.8. Comparison Line Chart of Recommendation Results of Movies and Books

The recommendation accuracy of ST_RippleNet model for MovieLens-1M and Book-Crossing is improved obviously. As can be seen from the bar chart shown in FIG. 7, the recommendation accuracy of ST_RippleNet model is better than that of other models. On the bar chart, it can be observed that the two evaluation indexes AUC and ACC of our model are much higher than that of other models. From the line chart depicted in FIG. 8, it can be seen that compared with the evaluation indexes of other mainstream recommendation models, ST_RippleNet model has more obvious improvement in ACC index. Therefore, it can be proved that the ST_RippleNet model is effective for the recommendation performance of MovieLens-1M and Book_Crossing data sets.

4 Discussion and Conclusion

This paper proposes a knowledge graph recommendation model ST_RippleNet to deal with the defects of some current recommendation algorithms. ST_RippleNet can not only introduce KGE method into recommendation, but also mine potential information that users may choose. In this paper, when ST_RippleNet is applied to music, books and movies, it is found that compared with the current mainstream recommendation 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%.

Through experiments on Last.FM, Book-Crossin and MovieLens-1M data sets, the effectiveness of ST_RippleNet in recommendation system is proved according to its evaluation index and recommendation effect.

In this article, The ST_RippleNet model proposed by us explores the potential interests of users in the knowledge graph and stimulates the propagation of users' preferences on the set of knowledge entities. In the process of preference propagation, we adopt a triple-based multi-layer attention mechanism, and use the user's preference distribution for candidate items formed by the user's historical click information to predict the final click probability. This model solves the limitations of the existing KG-aware recommendation methods based on embedding and path. This method can mine the potential interest of users and improve the recommendation effect. Moreover, we have carried out extensive experiments in the three recommendation scenarios of music, movies and books. The results prove that ST_RippleNet has significant advantages over the current mainstream recommendation algorithms, and this model can well solve the cold start problem. In the future, more detailed information of user interaction items, such as the time the user stays, the number of clicks, etc., will be considered to further model the user's preferences.

Abbreviations

CKE:Collaborative Knowledge base Embedding;

ACC:Accuracy;AUC:area under curve;

PER:Player Efficiency Ratings;

FM:Factorization Machines;

DKN:Deep Knowledge-aware Network;

KG:Knowledge Graph;

SHINE:Signed Heterogeneous Information Network Embedding;

RMS: Root Mean Square.

Acknowledgements

All the research steps were carried out in the School of Computer Ability and Technology of Qilu University of Technology (Shandong Academy of Sciences).

Authors' contributions

Zhisheng Yang collected data, contributed to the implementation of the proposed detection algorithm, conducted experiments and wrote papers. JinYong Cheng provided help in the research and design, and contributed to the analysis of results and the preparation of manuscripts. All the authors read and approved the final manuscript.

Funding

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

Availability of data and materials

Please contact corresponding author for data requests.

Consent for publication

Not applicable.

Competing interests

The authors declare that they have no competing interests.

References

1. Jingyu Chen,Shengjie Xiu,Xiang Chen,Hao Guo,Xiaohua Xie. Flounder-Net: An efficient CNN for crowd counting by aerial photography[J]. Neurocomputing,2021,420.
2. De-Anonymizing Social Networks With Overlapping Community Structure[J]. IEEE/ACM Transactions on Networking,2020,28(1).
3. Amin Mahmoudi,Mohd Ridzwan Yaakub,Azuraliza Abu Bakar. The Relationship between Online Social Network Ties and User Attributes[J]. ACM Transactions on Knowledge Discovery from Data (TKDD),2019,13(3).

4. Hyun Kwon,Hyunsoo Yoon,Ki-Woong Park. Acoustic-decoy: Detection of adversarial examples through audio modification on speech recognition system[J]. Neurocomputing,2020,417.

5. Fang Chen,Yongfeng Huang. Knowledge-enhanced neural networks for sentiment analysis of Chinese reviews[J]. Neurocomputing,2019.

6. Deepak Gupta,Asif Ekbal,Pushpak Bhattacharyya. A Deep Neural Network Framework for English Hindi Question Answering[J]. ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP),2019,19(2).

7. Wenqiang Liu,Jun Liu,Mengmeng Wu,Samar Abbas,Wei Hu,Bifan Wei,Qinghua Zheng. Representation Learning over Multiple Knowledge Graphs for Knowledge Graphs Alignment[J]. Neurocomputing,2018.

8. Qianli Ma,Liuhong Yu,Shuai Tian,Enhuan Chen,Wing W. Y. Ng. Global-Local Mutual Attention Model for Text Classification[J]. IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP),2019,27(12).

9. Xiaodan Zhang,Shengfeng He,Xinhang Song,Rynson W.H. Lau,Jianbin Jiao,Qixiang Ye. Image captioning via semantic element embedding[J]. Neurocomputing,2020,395.

10. Weina Zhang,Xingming Zhang,Haoxiang Wang,Dongpei Chen. A Deep Variational Matrix Factorization Method for Recommendation on Large Scale Sparse Dataset[J]. Neurocomputing,2019.

11. Mikel Canizo,Isaac Triguero,Angel Conde,Enrique Onieva. Multi-head CNN–RNN for multi-time series anomaly detection: An industrial case study[J]. Neurocomputing,2019,363.

12. Hongwei Wang, Fuzheng Zhang, Jialin Wang, Miao Zhao, Wenjie Li, Xing Xie,and Minyi Guo. 2018. RippleNet: Propagating User Preferences on the Knowledge Graph for Recommender Systems. In Proceedings of the 27th ACM International Conference on Information and Knowledge Management. ACM.

13. Wenpeng Lu, Xu Zhang, Huimin Lu, Fangfang Li, Deep hierarchical encoding model for sentence semantic matching. Journal of Visual Communication and Image Representation, 2020, 71: 102794.

14. Li Weidong,Peng Rong,Wang Yaqian,Yan Zhihuan. Knowledge graph based natural language generation with adapted pointer-generator networks[J]. Neurocomputing,2020,382(C).

15. Shuangyin Li,Heng Wang,Rong Pan,Mingzhi Mao. MemoryPath: A Deep Reinforcement Learning Framework for Incorporating Memory Component into Knowledge Graph Reasoning[J]. Neurocomputing,2020.

16. Liqing Huang,Youshen Xia. Joint blur kernel estimation and CNN for blind image restoration[J]. Neurocomputing,2020,396.

17. Guiming Zhu, Chenzhong Bin, Tianlong Gu, Wei Chen, Zhonghao Jia.Neural modeling framework of user preference based on knowledge graph [J]. Pattern Recognition and Artificial Intelligence, 2019, 32 (07): 661-668.

18. Xiao Lin,Min Zhang,Yiqun Liu,Shaoping Ma. Enhancing Personalized Recommendation by Implicit Preference Communities Modeling[J]. ACM Transactions on Information Systems (TOIS),2019,37(4).

19. Kumar Sunil,Thahsin Adheeba,M Manju,G Gopakumar. A Heterogeneous Information Network Model for Long Non-Coding RNA Function Prediction.[J]. IEEE/ACM transactions on computational biology and bioinformatics,2020,PP.

20. Desheng Cai,Gongqing Wu. Content-aware attributed entity embedding for synonymous named entity discovery[J]. Neurocomputing,2018.

21. Yutian Hu,Fei Xiong,Dongyuan Lu,Ximeng Wang,Xi Xiong,Hongshu Chen. Movie collaborative filtering with multiplex implicit feedbacks[J]. Neurocomputing,2020,398.

22. Basawaraj,Starzyk Janusz A,Horzyk Adrian. Episodic Memory in Minicolumn Associative Knowledge Graphs.[J]. IEEE transactions on neural networks and learning systems,2019,30(11).

23. W. Lu, Y. Zhang, S. Wang, H. Huang, Q. Liu and S. Luo, "Concept Representation by Learning Explicit and Implicit Concept Couplings," in IEEE Intelligent Systems, doi: 10.1109/MIS.2020.3021188.

24. Wang H, Zhang F, Zhao M, et al. Multi-task feature learning for knowledge graph enhanced recommendation [C]//The World Wide Web Conference. 2019: 2000-2010.

25. Aditya Grover and Jure Leskovec. 2016. node2vec: Scalable feature learning for networks. In KDD'16. ACM, 855–864.

26. Qi Wang,Yuede Ji,Yongsheng Hao,Jie Cao. GRL: Knowledge graph completion with GAN-based reinforcement learning[J]. Knowledge-Based Systems,2020,209.

27. J. Bernabé-Moreno,A. Tejeda-Lorente,J. Herce-Zelaya,C. Porcel,E. Herrera-Viedma. A context-aware embeddings supported method to extract a fuzzy sentiment polarity dictionary[J]. Knowledge-Based Systems,2020,190.

28. Zafar Ali,Guilin Qi,Khan Muhammad,Bahadar Ali,Waheed Ahmed Abro. Paper recommendation based on heterogeneous network embedding[J]. Knowledge-Based Systems,2020,210.

Biographical Sketch and Photo

First-Author ZHISHENG YANG received a bachelor's degree. Department of Software Engineering, North University of China, 2019. At present, I am studying for a master's degree in the School of Computer Science and Engineering of Qilu University of Technology (Shandong Academy of Sciences). His research directions include recommendation system, machine learning and in-depth learning.

Second-Author JINYONG CHENG received the M.S. degree in computer software and theory from the Qingdao University, Qingdao, China, in 2005. He is currently an Associate Professor of the School of Computer Science and Technology, Qilu University of Technology (Shandong Academy of Sciences). His research interests include artificial intelligence, image processing, and biomedical information processing. Until now, he has published more than 20 papers in conferences and journals. His research is sponsored by Natural Science Foundation of Shandong province.

Figures

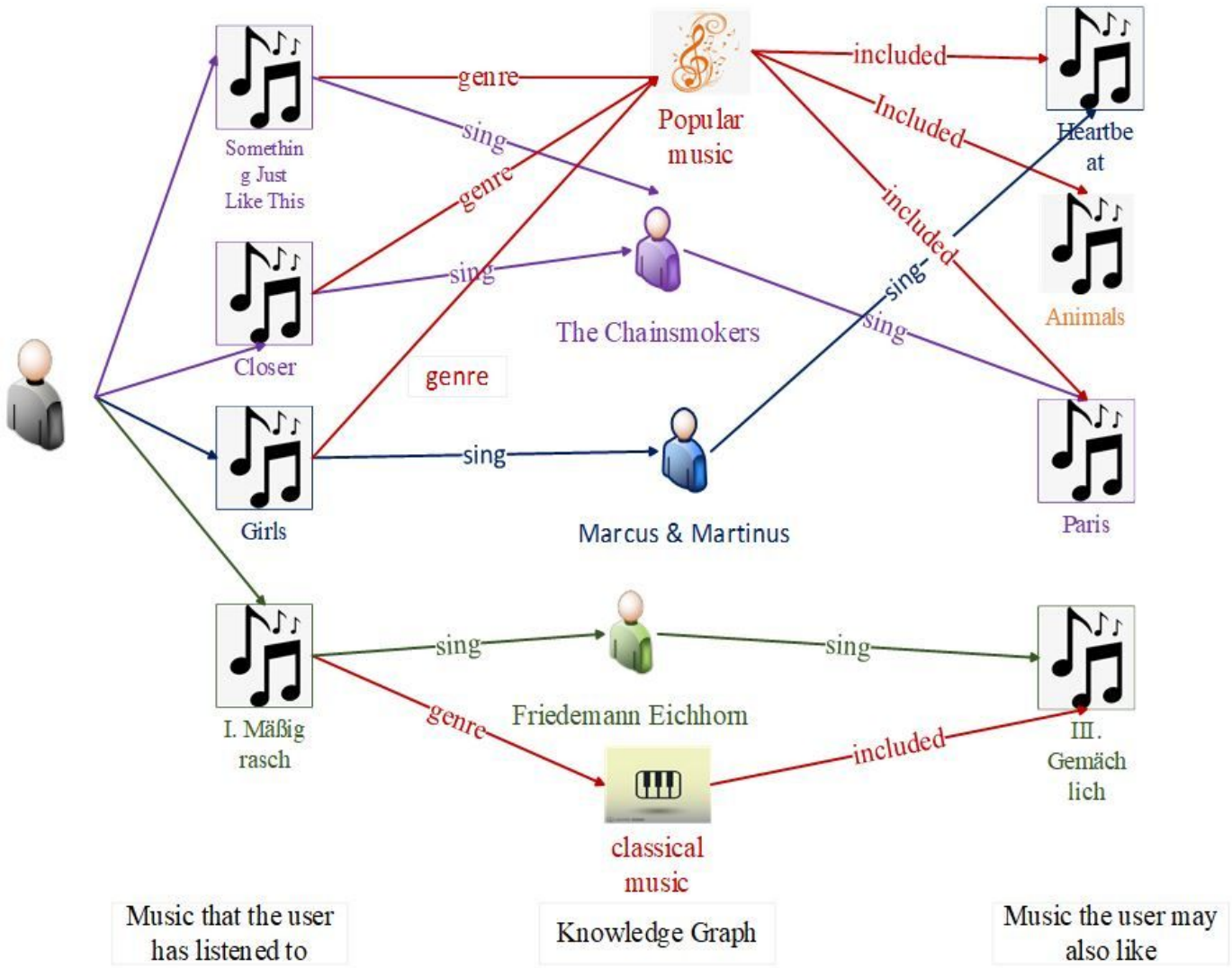

Figure 1

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

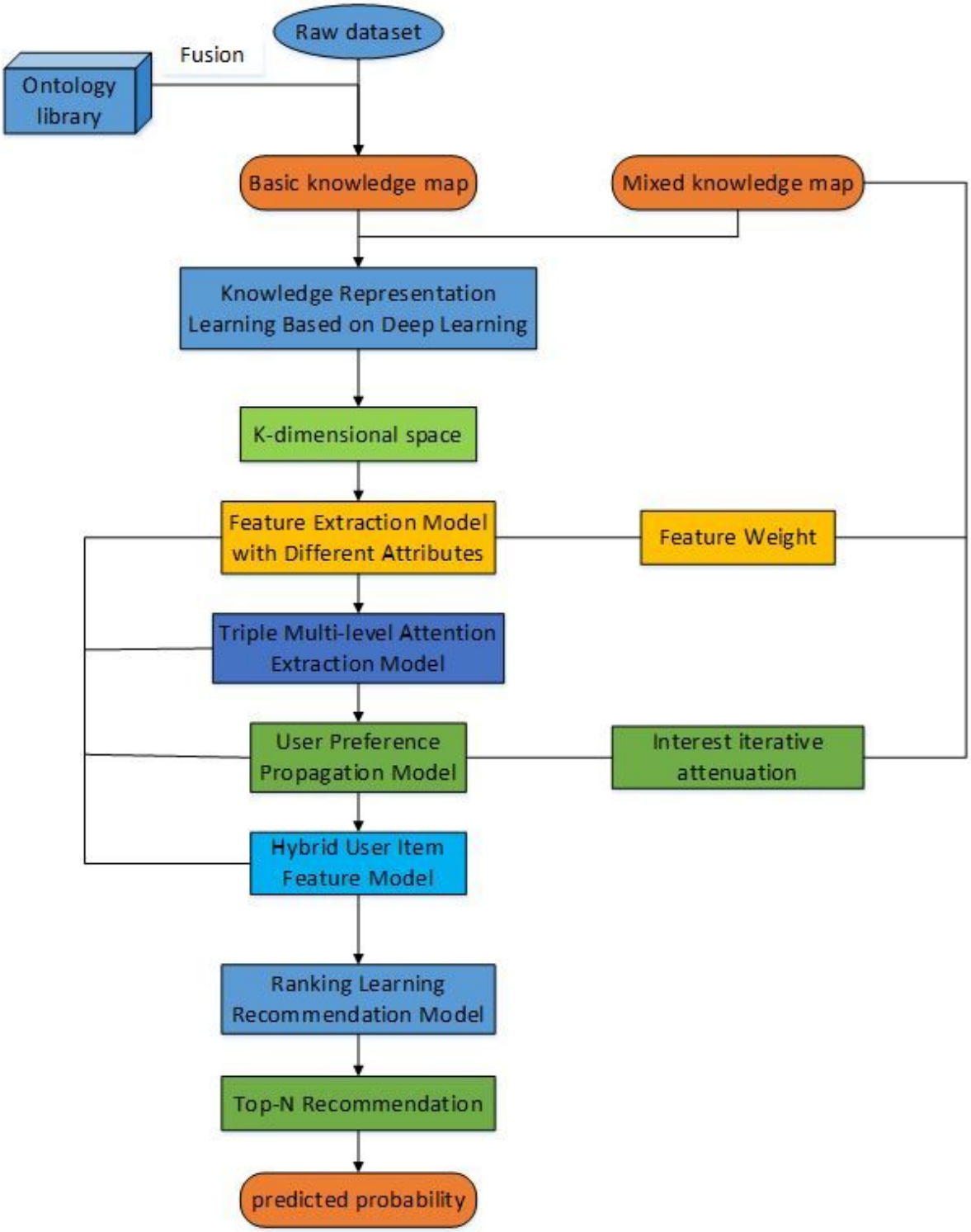

Figure 2

Recommendation algorithm for spreading user preferences based on knowledge graph

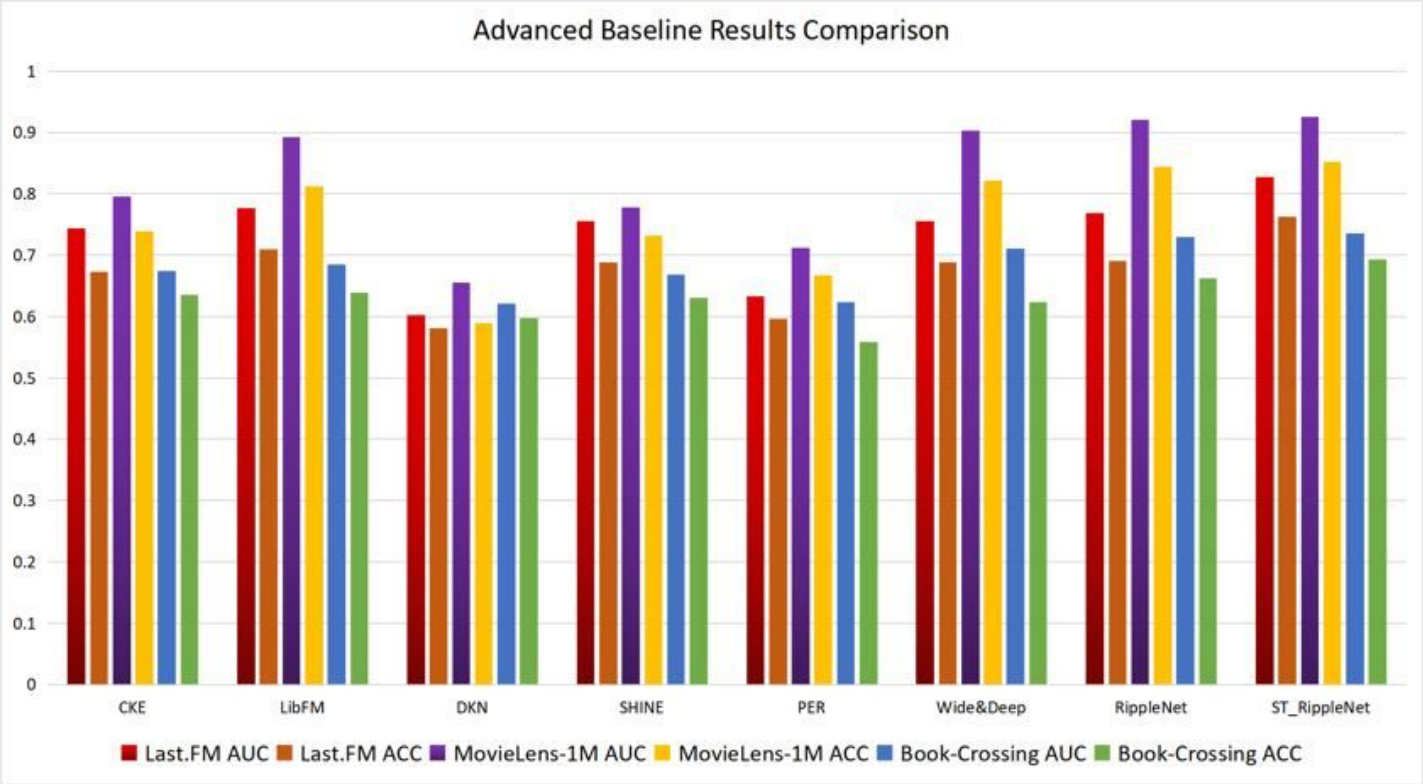

Figure 3

Advanced Baseline Results Comparison Histogram

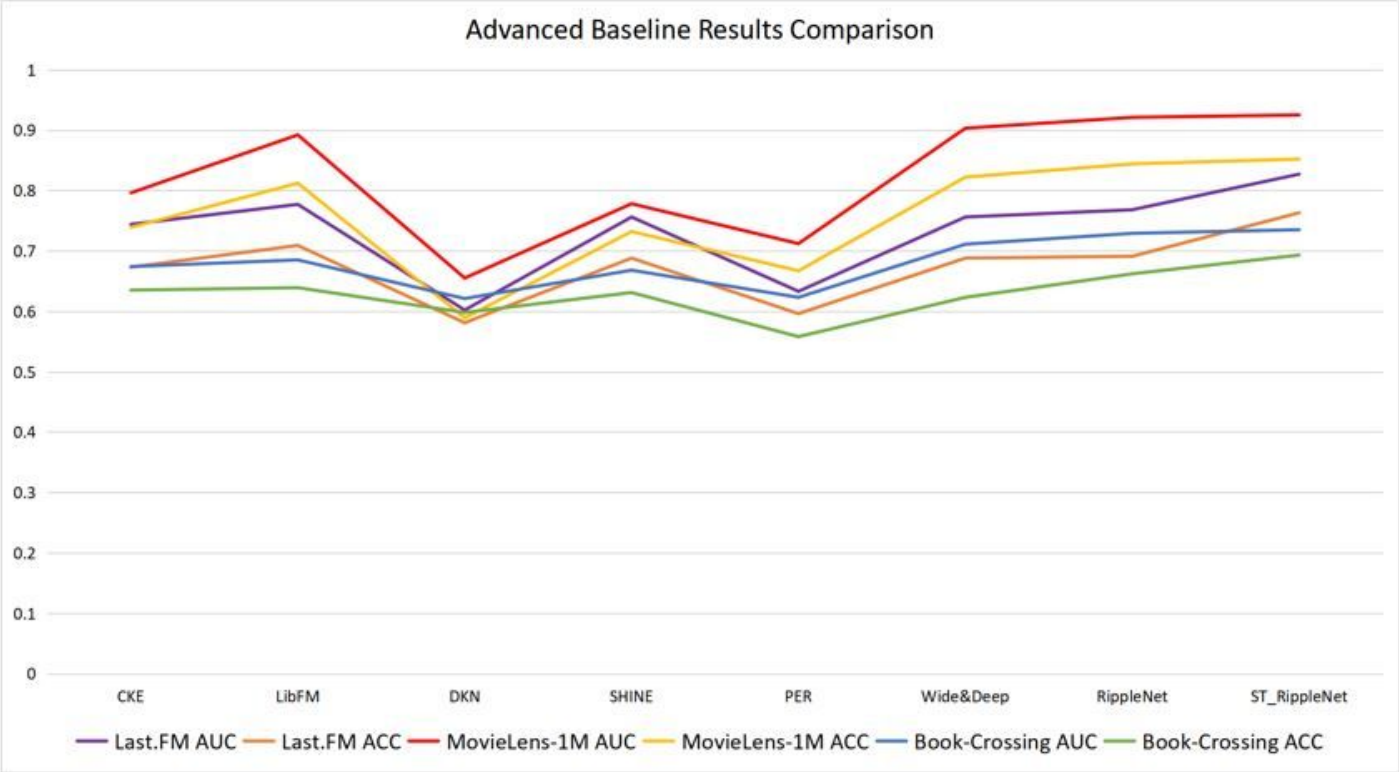

Figure 4

Advanced Baseline Results Comparison Line Chart

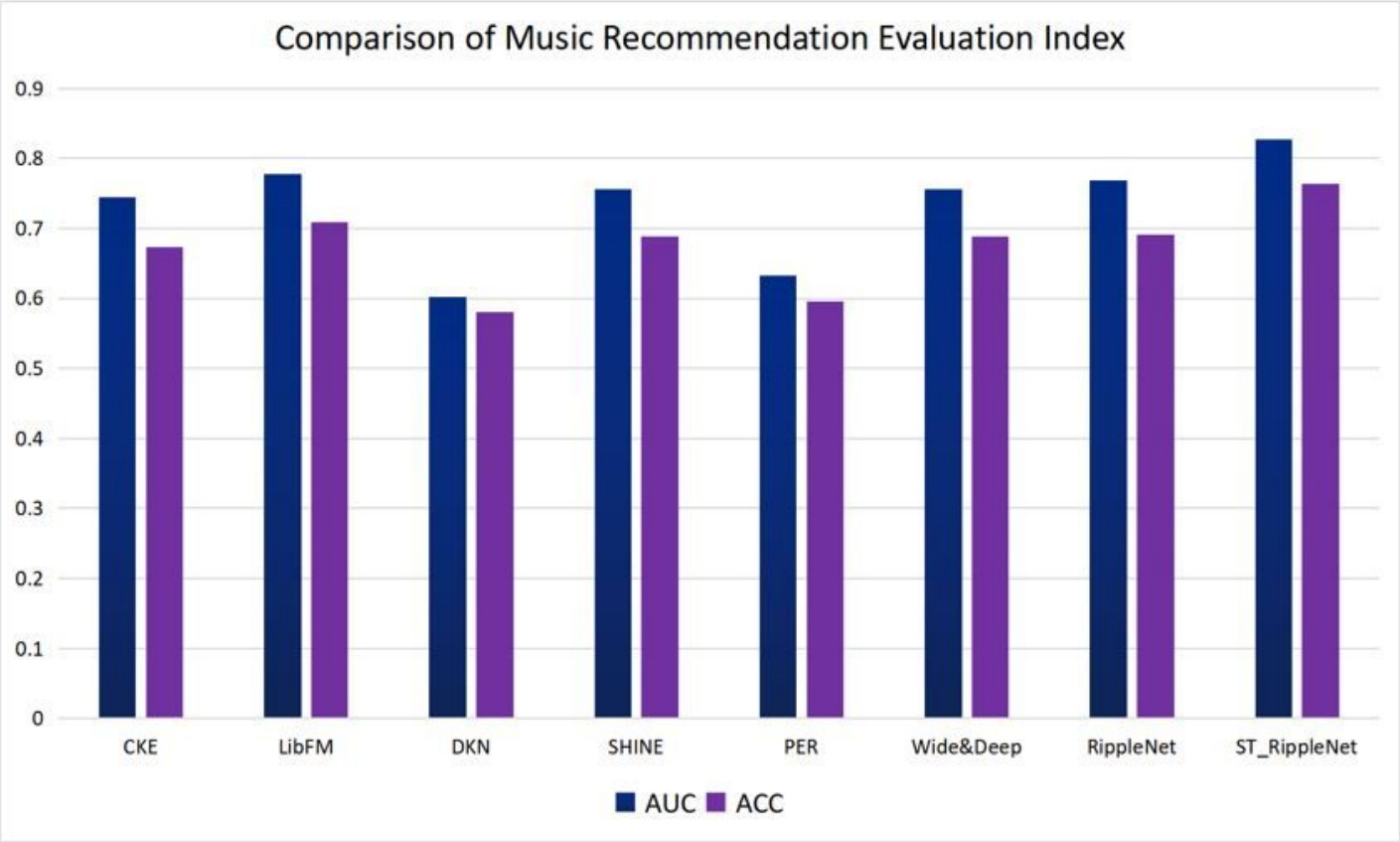

Figure 5

Music Recommendation Results Comparison Column Chart

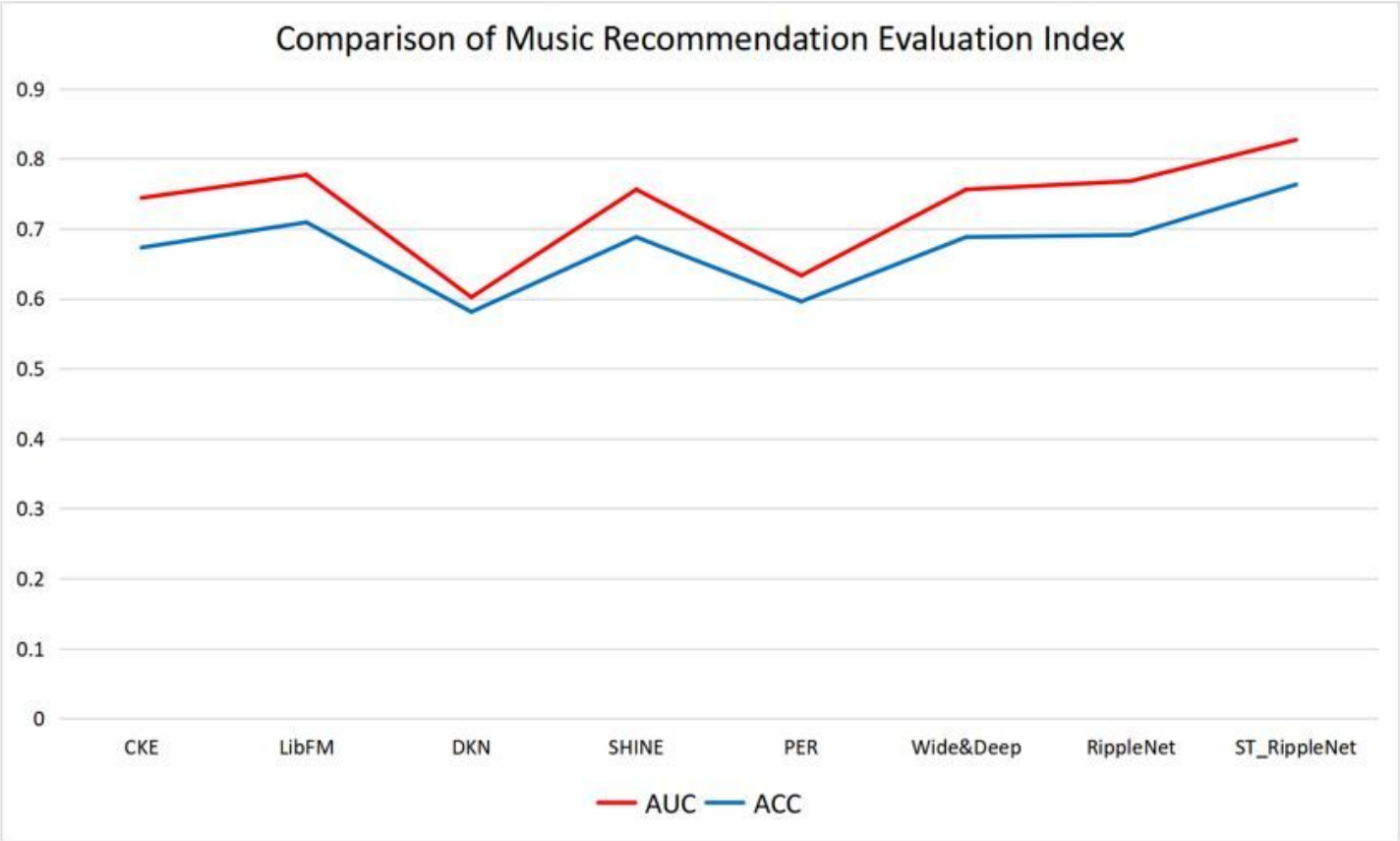

Figure 6

Comparison Line Chart of Music Recommendation Results

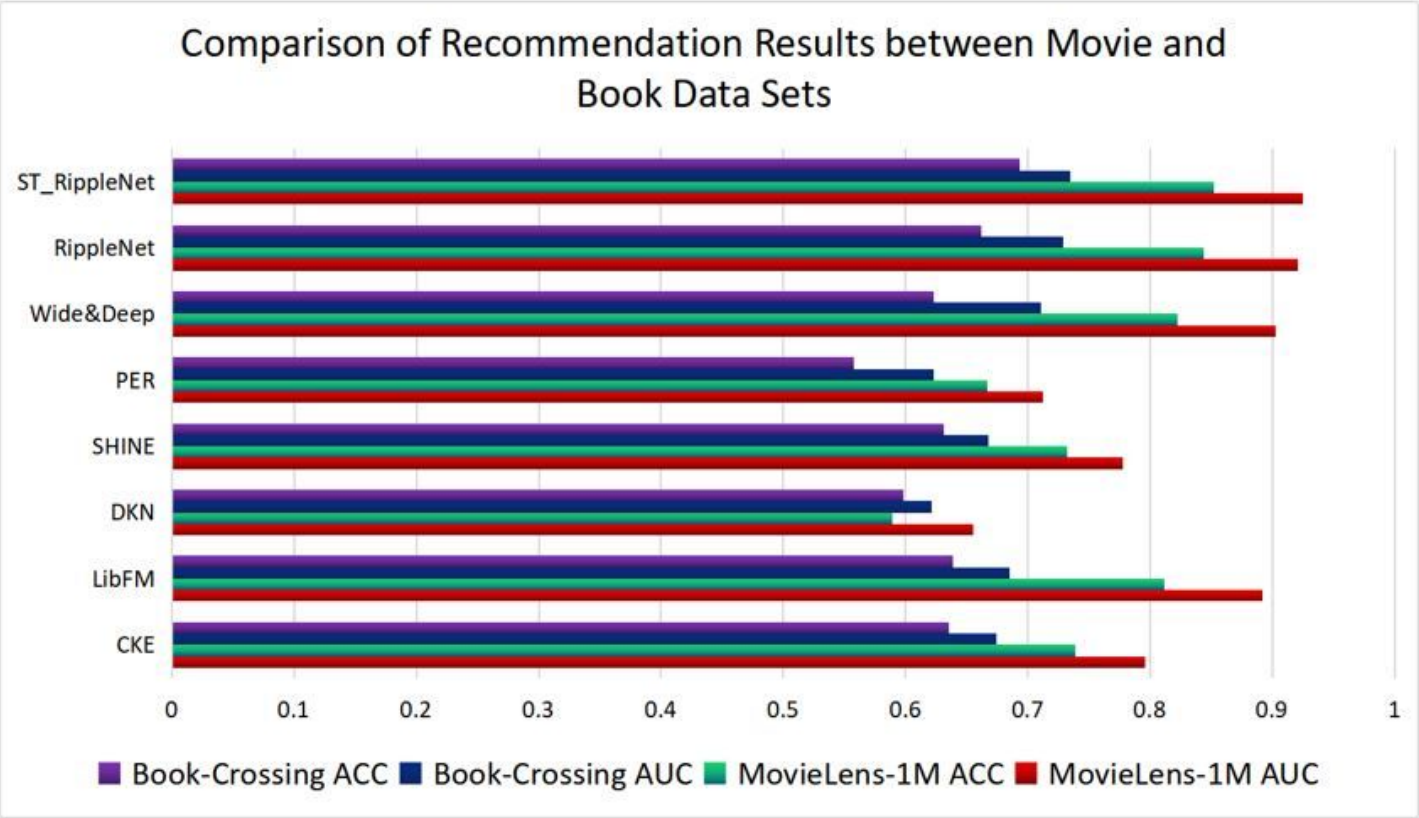

Figure 7

Comparison of Movie and Book Recommendation Results Bar Chart

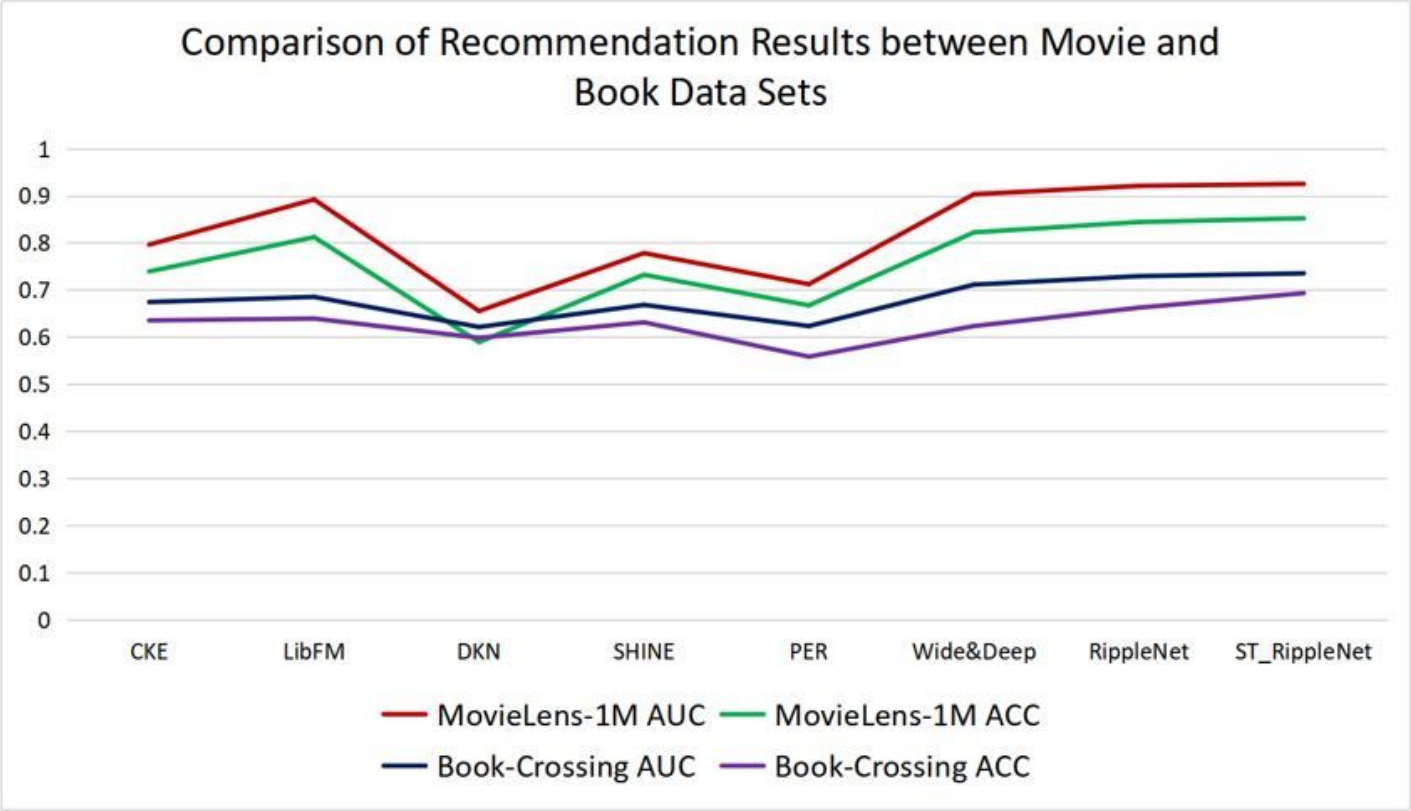

Figure 8

Comparison Line Chart of Recommendation Results of Movies and Books