GPRE: Multi-Path Guided Recommender System Based on Graph Neural Networks

Evan Webster, Julie Hill, Tyson Shimada, Braedon Smyth,

Abstract—Recommender systems as an effective information filtering system can be used to obtain information through the user's explicit or implicit behavior. On the one hand finding items that may be of interest to the user. On the other hand, the recommendation facilitates the interaction between the user and the item to increase the revenue. Recommender systems have been widely used in various fields, such as e-commerce, travel recommendation, online books and movies, social networks, etc, which can satisfy the intrinsic implicit needs of users through personalized services. In recent years, the development of deep learning has further improved the performance of recommendation systems. Although these methods improve the performance of the recommendation system, when the number of users and products increases, the recommendation system may face sparsity and cold start problems, and thus cannot achieve personalized recommendations. Knowledge graphs, which are structured data, have become the choice of many algorithms due to the high quality and wide scale of the data, and therefore many recommendation algorithms combined with knowledge graphs have emerged as a popular new direction in recommendation systems. These algorithms are able to preserve the rich connections between different entities. Moreover, when constructing the features of an entity, the entities that are far away from the central entity can also be utilized. Entities are no longer only directly connected to each other. To address the shortcomings of existing recommendation algorithms, this paper designs the recommendation algorithm GPRE using graph neural networks. GPRE focuses on expressing the user's features. The graph neural network provides GPRE with a strong generalization capability for modeling, which can provide long-range semantics between users and entities, as well as selective entity selection in the auxiliary graph neural network. Explicit semantic links are established between remote and central nodes to reduce the introduction of noise. In this paper, experiments are conducted on real-world datasets and the results are compared with baselines. The experimental results show that GPRE performs well on the experimental dataset.

Index Terms—Recommendation; GNN, Feature; Path; Embedding.

1 Introduction

With the development of web information technology, recommendation systems are widely used In the field of social media networks, news delivery, shopping platforms and so on. Recommender systems as an effective information filtering system can be used to obtain information through the user's explicit or implicit behavior [1]. On the one hand finding items that may be of interest to the user. On the other hand, the recommendation facilitates the interaction between the user and the item to increase the revenue. Recommender systems have been widely used in various fields, such as e-commerce, travel recommendation, online books and movies, social networks, etc [2], which can satisfy the intrinsic implicit needs of users through personalized services.

Collaborative filtering is one of the most widely used recommendation algorithms. The main idea of collaborative filtering is to discover the correlation between users based on their preferences for products and make recommendations based on the correlation, i.e., users with high similarity tend to have similar preferences [3]. In general, collaborative filtering is modeled by the historical interaction information between users and products (e.g., ratings, click-through rates, etc.), and their embedding representations are obtained. For example, matrix decomposition [4] maps users and products into the same dimensional hidden space, and uses inner product to calculate users' preferences for products.

The advantage of CF algorithm is very obvious, it does not need to know the details of users and items, only requires a record of user-item interactions to complete the entire recommendation process [5]. Therefore, the algorithm does not need to processing complex text, image and other information, and can be used in a variety of scenarios, such as video recommendation, product and music recommendations. The greatest advantage of CF is its simplicity and universality. Because of this advantage, CF algorithms have been widely successful in many fields. However, despite the excellent performance of CF algorithms, the algorithms do not perform well when dealing with cold-start problems [6]. The cold start problem refers to when the number of users is small and the user-item interaction is low in the early stages of a recommendation system. Since the CF algorithm essentially recommends a new item to a user based on the user's history of interaction with the item, in a cold start situation, the history of user-item interactions is very sparse, so it is difficult for the CF algorithm to play its role.

In recent years, the development of deep learning has further improved the performance of recommendation systems [7]. The NCF model proposed by He et al. [8] further optimizes the traditional model by integrating the traditional matrix decomposition and MLP. The NGCF

*Evan Webster, and Julie Hill are the corresponding author.

- *Evan Webster, Julie Hill, Tyson Shimada, and Braedon Smyth are with University of Manitoba, Ca. (e-mail: Evanwebster.ca@hotmail.com, juliehillresearch@umanitoba.ca*

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 29 November 2022 doi:10.20944/preprints202211.0534.v1

model proposed by Wang et al. [9] constructs a bipartite graph of user-goods and embeds the historical interactions between users and goods into the feature vector by graph model. The NMCL proposed by Wei et al. [10] split the consistent component and the complementary component by a novel relation-aware attention mechanism. Although these methods improve the performance of the recommendation system, when the number of users and products increases, the recommendation system may face sparsity and cold start problems, and thus cannot achieve personalized recommendations. To fundamentally solve the cold-start problem, many algorithms have started to use external resources, such as users' and items' own attributes, to alleviate the over-reliance of collaborative filtering a lgorithms o n u sers' h istorical b ehaviors. This approach can be understood as using knowledge and common sense to infer user interests.

Knowledge graphs, which are structured data, have become the choice of many algorithms due to the high quality and wide scale of the data, and therefore many recommendation algorithms combined with knowledge graphs have emerged as a popular new direction in recommendation systems. The knowledge graph embedding model represented by TransE [11] models the knowledge graph structural relationships in a way that the embeddings of entities are obtained using the well-established knowledge graph embedding model. And the embeddings of the entities corresponding to the items in the knowledge graph are used as a new input feature, either by using existing mature algorithms [12] or by designing new algorithms [13] for recommendation prediction. The most distinctive feature of metapath-based approaches is the use of expert-constructed metapaths followed by recommendation prediction using similarity based metapaths [14], as in PER [15], or embeddings of metapath-generating entities [16]. The biggest problem with these types of approaches using metapaths is that their effectiveness is highly dependent on the quality of the metapaths themselves, which are often constructed by humans in order to ensure quality, which requires considerable expertise of the constructors. For multiple path features between users and items, RKGE [17] only considers the most significant f eatures i n e ach d imension o f these path features, which undoubtedly results in information loss. KPRN [18] considers all paths and selects each path for utilization based on different importance through an update strategy with weights, which is theoretically reasonable. This is theoretically reasonable, since obviously not all rules should work, or work equally. However, if some of the low-quality rules can be removed by a suitable filtering strategy, it can both reduce the computational burden of the model and improve its predictive power. Methods such as GCN [19], GAT [20], MMGCN [21] KGAT [22] and KGCN [23] sequentially aggregate neighboring entities around an entity at different distances to that entity so that the entity can obtain information about the surrounding nodes. These algorithms are able to preserve the rich connections between different entities. Moreover, when constructing the features of an entity, the entities that are far away from the central entity can also be utilized. Entities are no longer only directly connected to each other.

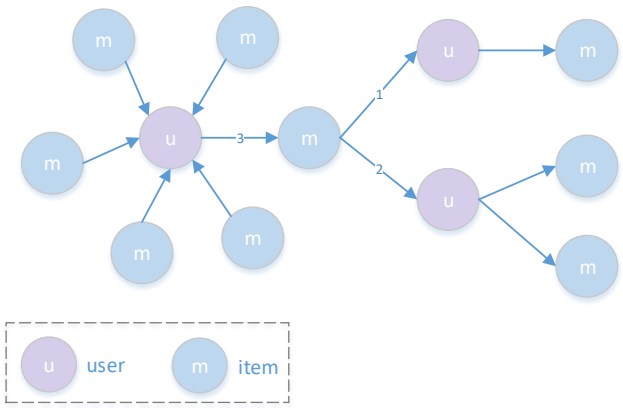


Fig. 1: Schematic diagram of the information propagation.

To address the shortcomings of existing recommendation algorithms, this paper designs the recommendation algorithm GPRE using graph neural networks. GPRE focuses on expressing the user's features. The graph neural network provides GPRE with a strong generalization capability for modeling, which can provide long-range semantics between users and entities, as well as selective entity selection in the auxiliary graph neural network. Explicit semantic links are established between remote and central nodes to reduce the introduction of noise. In this paper, experiments are conducted on real-world datasets and the results are compared with baselines. The experimental results show that GPRE performs well on the experimental dataset.

2 Related Work

2.1 Graph Neural Networks

Collaborative Filtering is a very classical algorithm that has been widely successful in recommender systems. Its basic idea is very simple. Taking music as an example, when recommending a song to a user, the CF algorithm searches for users who like the same songs and artists as this user, and then recommends the songs liked by the searched users to the target user. In terms of implementation, the classical SVD algorithm [24] constructs an interaction matrix with users as rows and items as columns, and obtains the feature representations of users and items through matrix decomposition, and then makes recommendation prediction of items. In recent years, some works such as NeuACF [25] have introduced deep neural networks into collaborative filtering and also achieved excellent results. The biggest problem of collaborative filtering algorithms is the cold start problem [26]. Collaborative filtering requires rich interaction between the user and the item to be effective, but when there is too little interaction between the user and the item, the collaborative filtering algorithm becomes less effective. For example, in the early stage of a commercial music software, there is very little interaction between users and songs, which makes it very difficult to use collaborative filtering algorithms. One way to solve this problem is to provide additional information about users and items

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 29 November 2022 doi:10.20944/preprints202211.0534.v1

to the recommendation system. Knowledge graphs, as high-quality structured data, are increasingly chosen by algorithms to solve the cold-start problem and improve the recommendation capability of recommendation systems due to their advantages of high data quality.

2.2 Graph Neural Networks

With the rapid development of the Internet, large-scale graph structured data has spread across various fields. Unlike images and text, graph structures are complex and variable irregular domains. The key aspect of graph analysis using machine learning algorithms and mathematical models is the graph representation. Graph neural networks are able to learn graph structure information and node features while saving computational space and time, and are suitable for for tasks such as node classification, edge prediction, and graph classification [27, 2 8]. G raph neural networks have received increasing attention as a deep learning method for analyzing graph structures. Since the node ordering of non-Euclidean data is irregular, traditional neural networks [29] need to traverse all possible orders of nodes and can hardly afford such a large computational effort. Also inspired by network embedding [30], the nodes and edges of the graph are represented as low-dimensional vectors. Defferrard et al. [31] proposed ChebNet, which defined t he fi lter as a Ch ebyshev po lynomial, verifying the ability to learn local features on graphs. Gao et al. [32] retained the node sequences by biasing random wandering, focusing only on the structural information and ignoring the attribute information of the nodes. However, recommendation tasks are heterogeneous and contain large and multiple types of information. Graph network based recommendation tasks consider users and items as nodes and need to integrate the connections between them and other auxiliary information to improve the recommendation quality [33]. MKR [13] designed a separate cross-compress unit to make the recommendation task better by sharing the feature parameters of recommendation and knowledge graph embedding. CKE [12] also inputs visual knowledge related to items, textual knowledge, and structured knowledge from the knowledge graph as features into the CF algorithm to improve the predictive power. KGCN and KGCNLS [23] use graph convolutional neural networks to clustered around an entity to that entity. These two works use the Attention mechanism to construct the relationships between entities The explicit long-range semantics between users and entities are ignored, and it is difficult t o c larify w hat s emantic c onnections a re made between points that are distant from the central node and the central node. Meanwhile, the KGCN and KGCNLS sample the points for aggregation randomly, which may cause information loss. This may cause information loss.

3 Methodology

3.1 Embedding Layer

The embedding layer transforms the input sparse vectors into dense vectors, randomly initializes the vector matrix of users and items$E \in \mathbb{R}^{(M+N)\times d}$, where the number of users is M, the number of items is N, and d is the dimension representing the size.

$$E = [e_{u1}, e_{u2}, \ldots, e_{uM}, e_{i1}, e_{i2}, \ldots, e_{iN}] \tag{1}$$

3.2 Propagation Layer

For any node pair (u, i) of a user, define the information transfer function from item i to user u as shown in equation (2). Where e_i is the feature vector of the item, e_u is the feature vector of the user, $\frac{1}{\sqrt{|N_i||N_u|}}$ is the weight coefficient between user u and item i, N_i and N_u represent the number of neighbor nodes of user and item respectively, and $W_1, W_2 \in \mathbb{R}^d$ is the weight matrix parameter.

$$m_{u\leftarrow i} = frac1\sqrt{|N_i||N_u|}(W_1 e_i + W_2 e_i \odot e_u) \tag{2}$$

Since the user may have more than one neighbor node, the final vector representation of the user is the fusion of all its neighbor nodes. Through the information transfer in layer l, the user can obtain the information of its neighbors of order l. The feature vector representation in layer l is shown in equation (3). It can be seen that the user not only fuses the features of its neighboring nodes, but also keeps the features of the user itself during the information transfer process.

$$e_u^{(l)} = LeakyReLU(m_{u\leftarrow i}^{(}l) + \sum_{i\in N_u} m_{u\leftarrow u}^{(}l)) \tag{3}$$

$$m_{u\leftarrow i}^{(}l) = p_{ui}(W_1^l e_i^{(l-1)} + W_2^l e_i^{(l-1)} \odot e_u^{(l-1)}) \tag{4}$$

$$m_{u\leftarrow u}^{(}l) = W_1^l e_u^{(l-1)} \tag{5}$$

where $_{ui}$ is the weight coefficient, W_1 and W_2 are matrix parameters, $e_i^{(l-1)}$ is the eigenvector of the items in layer l-1, and $e_u^{(l-1)}$ is the eigenvector of the users in layer l-1. The final representation of the user and product feature vectors is shown in equation (6), where $||$ denotes the concatenation between the vectors.

$$e_u^* = e_u^{(l)}|| \cdot ||e_u^{(l)}, e_i^* = e_i^{(l)}|| \cdot ||e_i^{(l)} \tag{6}$$

3.3 User Profile Construction

GPRE extends along path to the user. Then, GPRE reverse aggregates the extended entity nodes to the user node. This process constructs the feature representation of the user under r1 path. The set of k-hop extended entities of user u on path r is obtained from the following equation.

$$D_u^k(r) = \{o|(s, p_k, o), s \in D_u^{k-1}(r)\} \tag{7}$$

where $k \in [1, h]$, h is the length of the path r.

When this extends u along path r, if the (k-1)-hop extended entity set $D_u^{k-1}(r)$ cannot generate a k-hop extended entity set using the k-th predicate in path r. This point should receive a negative feedback indicating that path r is not suitable for user u to some extent.

In the aggregation process, the set of entities to which the aggregation operation is applied is represented by the following equation.

$$J_i = \{D_u^0(r) \cup D_u^1(r) \cup \cdots \cup D_u^{h-i}(r)\} \tag{8}$$

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 29 November 2022 doi:10.20944/preprints202211.0534.v1

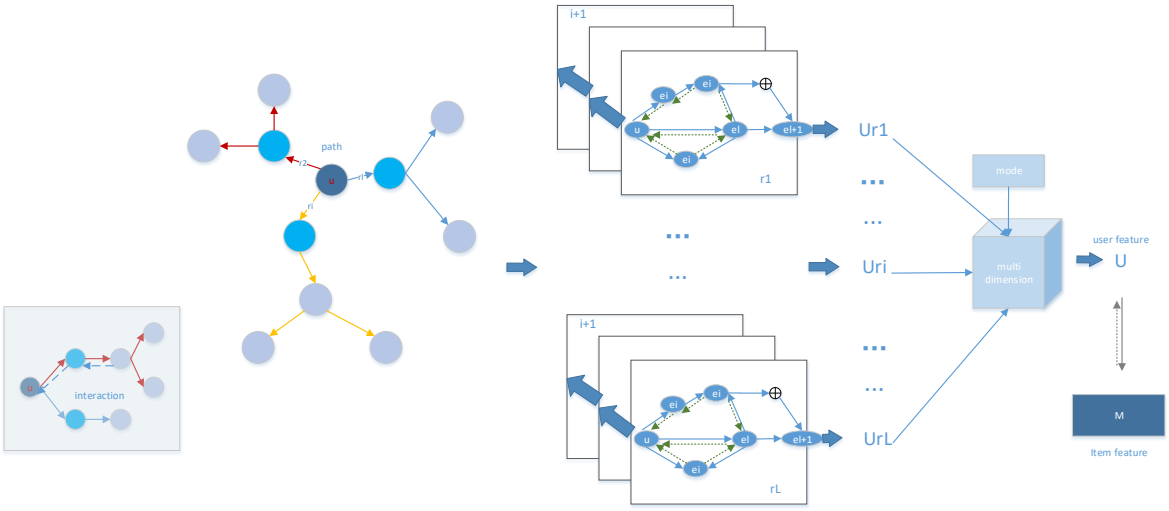


Fig. 2: Schematic diagram of our model.

where h represents the length of path r and the total number of iterations to be performed on path r. GPRE aggregates each of the expanded points in turn along the direction of expansion. The user is the first point to perform the aggregation operation, and the nodes expanded by the user are the subsequent points to perform the aggregation operation. The nodes expanded by user are the subsequent points to be aggregated. The aggregation process is repeated h times, on J_1 to J_h. In the i+1 iteration, the entities in the set J_{i+1} are updated by applying the aggregation operation. The state e of entity e in J_{i+1} is changed from e_i to e_{i+1}.

$$e_{i+1} = c(e_i \oplus Avg(e_i^1, \ldots, e_i^y)) \tag{9}$$

$$c(x) = \delta(W_a x + b) \tag{10}$$

$$Avg(e_i^1, \ldots, e_i^y) = \frac{1}{y}\sum_{a=1}^{y} e_i^a \tag{11}$$

where e is a one-dimensional vector of dimension d, δ is a nonlinear function such as Sigmoid, and e_{i+1} is the new state obtained by entity e after the i+1 update. After h iterations, the feature representation obtained by user u under path r is denoted as U_r^h, which is a one-dimensional vector of dimension d.

The feature representation of the central node is constructed by aggregating the feature representations of neighboring nodes to the central node and imposing a nonlinear activation function. Meanwhile, GPRE enables the central node to receive information from distant, non-directly connected nodes by repeating the aggregation and nonlinear activation process.

3.4 Features in Multiple Dimensions

Suppose that L paths $\{r_1, r_2, \ldots, r_L\}$, the L feature representations of user u are obtained. The final representation U of user u is given by the following equation.

$$U = LW = [U_{r_1}^{h_1}, U_{r_2}^{h_2}, \ldots, U_{r_L}^{h_L}]W \tag{12}$$

where h is the length of path r. W is a weight vector of size $L \times 1$. L is the representation matrix obtained under the L paths, of size $d \times L$.

The loss function is defined as follows.

$$loss = \frac{1}{N}\sum_{i=1}^{N}(l_i - q(U_i^T M_i)^2) + \mu||W||_2 \tag{13}$$

where U_i, M_i and l_i are the user feature representation, item feature representation and label, respectively. The label is 1 if there is an interaction between the user and the item, and 0 if the opposite. μ is the hyperparameter of L_2 regularization. The size of M_i is $d \times 1$. q is a nonlinear function, such as sigmoid.

4 EXPERIMENTS

4.1 Datasets

In this paper, we chose two real-world datasets: Last.FM, MovieLens-1M, which correspond to the two domains of music and movies respectively. Last.FM is the world's largest music social platform, with more than 20 million users using the site every month, and with its large user base, the site provides real, reliable and high-quality music recommendation data. In this paper, we use the publicly available dataset of 1871 users, 3864 music tracks and 42346 user-music interactions. MovieLens-1M is a widely used recommendation dataset in the movie domain, published by the GroupLens Research Program at the University of Minnesota. The dataset contains 6036 users, 2445 items and 753772 user interactions with movies.

The complete statistics of the two datasets are presented in Table 1, where users represents the number of users, items represents the number of items, interactions represents the number of interactions between users and items, entities represents the number of entities in the knowledge graph, and triples represents the number of triples in the knowledge graph.

4.2 Metrics

In this paper, we use AUC and F1 to evaluate the performance of the model in a clickthrough rate prediction

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 29 November 2022 doi:10.20944/preprints202211.0534.v1

TABLE 1: Statistical information of datasets.

	MovieLens-1M	Last.FM
users	6036	1872
items	2445	3864
interactions	753772	42346
entities	182011	9366
triples	1241995	15518

TABLE 2: Model Performance Comparison on Movielens-1M.

Model	AUC	F1
SVD	0.823	0.762
CKE	0.795	0.736
PER	0.846	0.803
KGCN	0.865	0.815
GPRE	0.887	0.833

TABLE 3: Model Performance Comparison on Last.FM.

Model	AUC	F1
SVD	0.774	0.735
CKE	0.745	0.724
PER	0.796	0.773
KGCN	0.802	0.781
GPRE	0.821	0.803

(CTR) scenario, which measure the ability of the model to determine whether a pair of users and items interact with each other. The performance of the model in the topK recommendation scenario is also evaluated using hits@k and ndcg@k ($k \in \{5, 10\}$), which measure the ability of the model to discover items of interest to users among the top-K recommended items.

AUC, the size of the area under the subject operating characteristic curve (ROC, Receiver Operating Characteristic Curve). the horizontal coordinate of the ROC curve is the rate of false positives and the vertical coordinate is the rate of true positives.

$$AUC = \frac{\sum r_i - \frac{M(1+M)}{2}}{M \times N} \tag{14}$$

where M is the number of positive class samples, and N is the number of negative class samples, the meaning of the formula is equivalent to evaluating the number of positive sample scores greater than negative sample scores among $M \times N$ positive and negative sample pairs.

TABLE 4: Model Performance Comparison on Movielens-1M.

Model	AUC	F1
SVD	0.315	0.506
CKE	0.236	0.375
KGCN	0.403	0.522
GPRE	0.428	0.545

TABLE 5: Model Performance Comparison on Last.FM.

Model	AUC	F1
SVD	0.357	0.541
CKE	0.187	0.304
KGCN	0.378	0.573
GPRE	0.405	0.603

4.3 Baselines

In this paper, we chose SVD, CKE, PER, and KGCN as comparison methods. The rationale for the selection of these methods, a brief description and hyperparameter settings are as follows.

- SVD [24] is a representative of the classical collaborative filtering-based recommendation algorithm. It uses the user and item feature vectors as the inner product of the recommendation probabilities. For implementation, this paper uses an unbiased version of SVD, where The dimensionality of both the user feature vector and the item feature vector is 8.
- CKE [12] is a representative of using entity embeddings only to design recommendation algorithms. In the original paper of CKE, it combines two types of knowledge - structured knowledge, textual knowledge and visual knowledge - into a collaborative filtering algorithm for recommendation. In the test dataset used in this paper, textual knowledge and visual knowledge are not available in this paper, so in terms of implementation, this paper follows the setup in the original paper and uses TransR to obtain the embedding of structured knowledge and combines the structured knowledge into the collaborative filtering algorithm as the CKE algorithm. The dimensionality of the entity embeddings generated by TransR is 64.
- PER [15] is a representative of meta-path-based recommendation algorithms. This algorithm treats the whole knowledge graph as Heterogeneous Information Network (HIN), and calculates the meta-paths by similarity based on meta-paths to perform recommendation prediction.
- KGCN [23] is a representative of aggregation-based recommendation algorithms, and this algorithm uses graph convolutional neural networks for aggregation. In this paper, we use the source code of this paper published on Github to obtain the performance of this model on the test dataset. In this paper, we use the source code of this paper published on Github to obtain the performance of this model on the test data set, and the hyperparameters are referred to the default settings in the paper and the code.

4.4 Result Analysis

GPRE achieved the best AUC and F1 on the Last.FM, MovieLens1M datasets. GPRE has stronger recommendation prediction ability. CKE's AUC and F1 on Last.FM and MovieLens-1M datasets are weaker than the algorithm SVD which does not use knowledge graph at all. The AUC and F1 metrics of PER on Last.FM and MovieLens1M are the worst among all methods. In this paper, we argue that this shows the high dependence

 doi:10.20944/preprints202211.0534.v1

of meta-path-based methods on the quality of manually constructed meta-paths, and without high-quality meta-paths, the effectiveness of such algorithms is greatly discounted. The PER algorithm for automatic path mining leads PER in AUC and F1 on all two datasets, which also indicates that the automatic mining rule algorithm has better robustness than the algorithm using meta-paths.

Compared to the rest of the algorithms, PER for automatic path mining has more outstanding AUC and F1 on each dataset, and the result on MovieLens-1M is second only to GPRE, which shows that the algorithm for automatic path mining has the ability to get rid of the dependence on human resources. However, the algorithm has much lower hit@k and ndcg@k values on all two datasets than the other algorithms lower, which is a result of the difference in testing strategies.

5 Conclusion and Future Work

Collaborative filtering is one of the most widely used recommendation algorithms. The main idea of collaborative filtering is to discover the correlation between users based on their preferences for products and make recommendations based on the correlation, i.e., users with high similarity tend to have similar preferences. The advantage of CF algorithm is very obvious, it does not need to know the details of users and items, only requires a record of user-item interactions to complete the entire recommendation process. The greatest advantage of CF is its simplicity and universality. However, despite the excellent performance of CF algorithms, the algorithm has been very successful in dealing with cold recommendations. The cold start problem refers to when the number of users is small and the user-item interaction is low in the early stages of a recommendation system. Since the CF algorithm essentially recommends a new item to a user based on the user's history of interaction with the item, in a cold start situation, the history of user-item interactions is very sparse, so it is difficult for the CF algorithm to play its role. In recent years, the development of deep learning has further improved the performance of recommendation systems. Knowledge graphs, which are structured data, have become the choice of many algorithms due to the high quality and wide scale of the data, and therefore many recommendation algorithms combined with knowledge graphs have emerged as a popular new direction in recommendation systems. The knowledge graph embedding model represented by TransE models the knowledge graph structural relationships in a way that the embeddings of entities are obtained using the well-established knowledge graph embedding model. And the embeddings of the entities corresponding to the items in the knowledge graph are used as a new input feature, either by using existing mature algorithms or by designing new algorithms for recommendation prediction. The biggest problem with these types of approaches using metapaths is that their effectiveness is highly dependent on the quality of the metapaths themselves, which are often constructed by humans in order to ensure quality, which requires considerable expertise of the constructors. Methods such as GCN, GAT, MMGCN, KGAT and KGCN sequentially aggregate neighboring entities around an entity at different distances to that entity so that the entity can obtain information about the surrounding nodes. These algorithms are able to preserve the rich connections between different entities. Moreover, when constructing the features of an entity, the entities that are far away from the central entity can also be utilized. Entities are no longer only directly connected to each other. To address the shortcomings of existing recommendation algorithms, this paper designs the recommendation algorithm GPRE using graph neural networks. GPRE focuses on expressing the user's features. The graph neural network provides GPRE with a strong generalization capability for modeling, which can provide long-range semantics between users and entities, as well as selective entity selection in the auxiliary graph neural network. Explicit semantic links are established between remote and central nodes to reduce the introduction of noise. In this paper, experiments are conducted on real-world datasets and the results are compared with baselines. The experimental results show that GPRE performs well on the experimental dataset.

6 Conflict of Interest Statement

All authors have no conflict and declare that: (i) no support, financial or otherwise, has been received from any organization that may have an interest in the submitted work ; and (ii) there are no other relationships or activities that could appear to have influenced the submitted work.

References

[1] J. Bobadilla, F. Ortega, A. Hernando, and A. Gutiérrez, "Recommender systems survey," *Knowledge-based systems*, vol. 46, pp. 109–132, 2013.

[2] J. Lu, D. Wu, M. Mao, W. Wang, and G. Zhang, "Recommender system application developments: a survey," *Decision Support Systems*, vol. 74, pp. 12–32, 2015.

[3] Y. Deldjoo, M. Schedl, B. Hidasi, Y. Wei, and X. He, "Multimedia recommender systems: Algorithms and challenges," in *Recommender systems handbook*. Springer, 2022, pp. 973–1014.

[4] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, vol. 42, no. 8, pp. 30–37, 2009.

[5] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Item-based collaborative filtering recommendation algorithms," in *Proceedings of the 10th international conference on World Wide Web*, 2001, pp. 285–295.

[6] G. Linden, B. Smith, and J. York, "Amazon. com recommendations: Item-to-item collaborative filtering," *IEEE Internet computing*, vol. 7, no. 1, pp. 76–80, 2003.

[7] S. Rendle, W. Krichene, L. Zhang, and J. Anderson, "Neural collaborative filtering vs. matrix factorization revisited," in *Fourteenth ACM conference on recommender systems*, 2020, pp. 240–248.

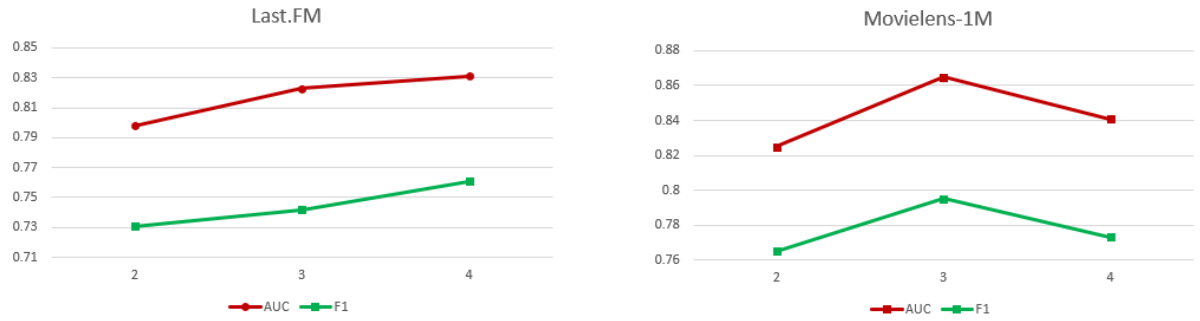
[8] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua, "Neural collaborative filtering," in *Proceedings of the 26th international conference on world wide web*, 2017, pp. 173–182.

[9] X. Wang, X. He, M. Wang, F. Feng, and T.-S. Chua, "Neural graph collaborative filtering," in *Proceedings of the 42nd international ACM SIGIR conference on Research and development in Information Retrieval*, 2019, pp. 165–174.

[10] Y. Wei, X. Wang, W. Guan, L. Nie, Z. Lin, and B. Chen, "Neural multimodal cooperative learning toward micro-video understanding," *IEEE Transactions on Image Processing*, vol. 29, pp. 1–14, 2019.

[11] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, and O. Yakhnenko, "Translating embeddings for modeling multi-relational data," *Advances in neural information processing systems*, vol. 26, 2013.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 29 November 2022 doi:10.20944/preprints202211.0534.v1



(a) Path length (b) Path length

Fig. 3: AUC and F1 under different maximum length limits of the path.

[12] F. Zhang, N. J. Yuan, D. Lian, X. Xie, and W.-Y. Ma, "Collaborative knowledge base embedding for recommender systems," in *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, 2016, pp. 353–362.

[13] H. Wang, F. Zhang, M. Zhao, W. Li, X. Xie, and M. Guo, "Multi-task feature learning for knowledge graph enhanced recommendation," in *The world wide web conference*, 2019, pp. 2000–2010.

[14] Y. Sun, J. Han, X. Yan, P. S. Yu, and T. Wu, "Pathsim: Meta path-based top-k similarity search in heterogeneous information networks," *Proceedings of the VLDB Endowment*, vol. 4, no. 11, pp. 992–1003, 2011.

[15] X. Yu, X. Ren, Y. Sun, Q. Gu, B. Sturt, U. Khandelwal, B. Norick, and J. Han, "Personalized entity recommendation: A heterogeneous information network approach," in *Proceedings of the 7th ACM international conference on Web search and data mining*, 2014, pp. 283–292.

[16] Y. Dong, N. V. Chawla, and A. Swami, "metapath2vec: Scalable representation learning for heterogeneous networks," in *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, 2017, pp. 135–144.

[17] Z. Sun, J. Yang, J. Zhang, A. Bozzon, L.-K. Huang, and C. Xu, "Recurrent knowledge graph embedding for effective recommendation," in *Proceedings of the 12th ACM conference on recommender systems*, 2018, pp. 297–305.

[18] X. Wang, D. Wang, C. Xu, X. He, Y. Cao, and T.-S. Chua, "Explainable reasoning over knowledge graphs for recommendation," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 33, no. 01, 2019, pp. 5329–5336.

[19] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.

[20] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," *stat*, vol. 1050, p. 20, 2017.

[21] Y. Wei, X. Wang, L. Nie, X. He, R. Hong, and T.-S. Chua, "Mmgcn: Multi-modal graph convolution network for personalized recommendation of micro-video," in *Proceedings of the 27th ACM International Conference on Multimedia*, 2019, pp. 1437–1445.

[22] X. Wang, X. He, Y. Cao, M. Liu, and T.-S. Chua, "Kgat: Knowledge graph attention network for recommendation," in *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, 2019, pp. 950–958.

[23] H. Wang, M. Zhao, X. Xie, W. Li, and M. Guo, "Knowledge graph convolutional networks for recommender systems," in *The world wide web conference*, 2019, pp. 3307–3313.

[24] Y. Koren, "Factorization meets the neighborhood: a multifaceted collaborative filtering model," in *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2008, pp. 426–434.

[25] C. Shi, X. Han, L. Song, X. Wang, S. Wang, J. Du, and S. Y. Philip, "Deep collaborative filtering with multi-aspect information in heterogeneous networks," *IEEE transactions on knowledge and data engineering*, vol. 33, no. 4, pp. 1413–1425, 2019.

[26] Y. Wei, X. Wang, Q. Li, L. Nie, Y. Li, X. Li, and T.-S. Chua, "Contrastive learning for cold-start recommendation," in *Proceedings of the 29th ACM International Conference on Multimedia*, 2021, pp. 5382–5390.

[27] M. Schlichtkrull, T. N. Kipf, P. Bloem, R. v. d. Berg, I. Titov, and M. Welling, "Modeling relational data with graph convolutional networks," in *European semantic web conference*. Springer, 2018, pp. 593–607.

[28] D. Wang, P. Cui, and W. Zhu, "Structural deep network embedding," in *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, 2016, pp. 1225–1234.

[29] M. M. Bronstein, J. Bruna, Y. LeCun, A. Szlam, and P. Vandergheynst, "Geometric deep learning: going beyond euclidean data," *IEEE Signal Processing Magazine*, vol. 34, no. 4, pp. 18–42, 2017.

[30] P. Cui, X. Wang, J. Pei, and W. Zhu, "A survey on network embedding," *IEEE transactions on knowledge and data engineering*, vol. 31, no. 5, pp. 833–852, 2018.

[31] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," *Advances in neural information processing systems*, vol. 29, 2016.

[32] M. Gao, L. Chen, X. He, and A. Zhou, "Bine: Bipartite network embedding," in *The 41st international ACM SIGIR conference on research & development in information retrieval*, 2018, pp. 715–724.

[33] Y. Wei, X. Wang, L. Nie, X. He, and T.-S. Chua, "Graph-refined convolutional network for multimedia recommendation with implicit feedback," in *Proceedings of the 28th ACM International Conference on Multimedia*, 2020, pp. 3541–3549.