Advances in Artificial Intelligence, Big Data and Algorithms
G. Grigoras and P. Lorenz (Eds.)

doi:10.3233/FAIA230887

A Recommendation Model Fusing Knowledge Graph Convolutional Networks and Social Relationships

Jie WU[1], Lan SHEN[2], Yanhong JING[3], Chengxi GAO[4]
Dalian Polytechnic University, School of Information Science and Engineering, Dalian, Liaoning, China

Abstract—The knowledge graph's rich semantic information can solve some of the issues with conventional recommendation. Most current knowledge graph-based recommendation models solely take into account the prospective characteristics of items, ignoring the supporting function of user potential features for recommendation systems, despite approaches like data sparsity and cold start. We suggest a recommendation model that incorporates social relationships and knowledge graph convolutional networks to solve this issue. Once more, the KGCN model is utilized to get the item feature matrix. Next, the multilayer perceptron is fed with the user feature vector, user embedding representation, and item feature vector to determine projected scores and provide user recommendations. The proposed recommendation model outperforms other benchmark models, according to experimental results that were done on the publicly accessible datasets CiaoDVD, Epinions, and FilmTrust to verify the model performance.

Keywords-recommendation algorithm; knowledge graph; graph convolutional network; social relationship; multilayer perceptron

1. Introduction

In response to the Internet's and information technology's fast expansion, the global datasphere is growing significantly. Although the vast volume of information is convenient for consumers, it also causes them tremendous difficulty since they cannot easily extract the information they are interested in. In order to help users filter information effectively and bring personalized content to them, recommendation systems have been created. In order for consumers to easily locate useful information in the vast volume of data, recommender systems can connect users and products and actively investigate prospective user interests.

---

[1] Jie WU, Dalian Polytechnic University, School of Information Science and Engineering; e-mail: rejay95@163.com
[2] Corresponding author: Lan SHEN, Dalian Polytechnic University, School of Information Science and Engineering; e-mail: shenlan@dlpu.edu.cn
[3] Yanhong JING, Dalian Polytechnic University, School of Information Science and Engineering; e-mail: xiaohongjyh@163.com
[4] Chengxi GAO, Dalian Polytechnic University, School of Information Science and Engineering; e-mail: bzugcx@163.com

Due to its superior recommendation impact, the conventional collaborative filtering algorithm has emerged as one of the most popular recommendation algorithms. However, because it depends so heavily on historical user behavior data, it encounters two challenging issues: the cold start problem, which is brought on by providing too little initial information to new users, and the sparse interaction matrix data, which is brought on by having both too many items overall and too few items interacting with users. Researchers suggest adding various types of auxiliary information, such as contextual information, remark information, and knowledge graphs (KG), to augment the data in order to address these two issues. The knowledge graph's semantic information aids in the exploration of potential relationships between users and objects, the reasonable expansion of user interests, and the improvement of the diversity and precision of recommendation outcomes.

Despite these benefits, fusing these heterogeneous data and applying them to recommender systems still faces great challenges because knowledge graphs contain information with multiple different attributes (e.g., text, images, tags, etc.). Literature [1] proposed the DKN model, which uses graph neural networks to fuse semantic representations with knowledge-level news representation vectors on the basis of considering semantics, and also uses attention mechanisms to calculate user click history and candidate news to recommend news that may be of interest to users more precisely. However, DKN models are content-based recommendations, and entity embeddings need to be obtained in advance of training, and thus cannot be trained end-to-end. The RippleNet model was introduced in literature [2] and mimics the process of user preference propagation in the knowledge graph, focusing on the historical records of user interest and identifying the prospective hierarchical interests of users by diverging to the outer layers on the knowledge graph. However, the RippleNet model does not represent the relationship vector adequately, and the relationship matrix is difficult to be trained. The literature [3] suggests employing graph convolutional networks to implement the KGCN model to identify higher level structural and syntactic information in the knowledge graph. The model aggregates and aggregates the neighborhood information of entities to capture the higher-order user interest by expanding the representation domain of each entity in the knowledge graph. However, the KGCN model only constructs the knowledge graph by adding item auxiliary information, and hardly uses the auxiliary information brought by the users to be considered.

To address the aforementioned issues, this paper uses social relationships to extract potential features of users, and proposes a recommendation model combining of Knowledge Graph Convolutional Networks and Social Relationships (KGCNSR) model. The following are this paper's vital contributions:

(1) The Social Network based User Similarity (SNBUS) model is proposed. The approach is built on a matrix decomposition model that incorporates user trust, social similarity, and user rating similarity to derive a user feature matrix by computing.

(2) A recommendation model that fuses knowledge graph convolutional networks and social relationships (KGCNSR) is proposed. The above user feature matrix and embedding representation of users and feature matrix of items are input to the multilayer perceptron to obtain the predicted ratings of users for unevaluated items.

(3) Experimental validation on three real datasets (FilmTrust, CiaoDVD, Epinions) proves that the KGCNSR model outperforms other baseline models.

2. Related Jobs

2.1. Graph Convolutional Network

With the continuous development of Internet technology, people's ability to process Euclidean spatial data is increasing, but in life many data are stored through graph structures, such as citation networks, knowledge graphs, social networks, etc. For these graph data, traditional neural networks (e.g., CNN) are unable to extract features from them by convolutional operations. The graph convolutional network is a convolutional neural network that can act directly on the graph and use its structural information. Literature [4] proposed a multichannel hypergraph convolutional network, which depicts a common pattern of higher-order user relationships and uses higher-order user relationships to enhance social recommendations. The literature [5] investigates probable user preferences from the more complex connection structure of heterogeneous knowledge graphs and learns how to simultaneously represent users and items using a bidirectional information distribution technique. In the literature [6], after linking each user with an ID embedding, the embedding is propagated and improved on the U-I interaction graph. The final anticipated embedding information is then obtained by combining the weighted total with the embedding data from various propagation layers. Graph neural networks have powerful embedding representation capability for graph data, which can explore the potential features of users (items), expand their interests, and boost the recommendation system's quality of recommendations.

2.2. Knowledge Graph Convolutional Network

Knowledge graph convolutional networks (KGCN) were developed by Wang[3] proposed with the problem described as: set $U = \{u_1, u_2, \dots, u_M\}$ and $V = \{v_1, v_2, \dots, v_N\}$ stands for the set of users and the set of items, respectively. The user-item interaction matrix $Y = \{y_{uv} | u \in U, v \in V\}$ is defined based on the implicit feedback from the users, where $y_{uv} = 1$ indicates that there is implicit feedback between user u and item v, such as when the user clicks, browses, or makes a purchase; in addition, there is a knowledge graph G, which consists of a triad of entity-relationships (h, r, t). where $h \in \varepsilon$, $r \in R$ and $t \in \varepsilon$ denote the head entity, the relationship and the tail entity, respectively, and ε and R are the set of entities and the set of relationships in the knowledge graph, respectively. The model is given an interaction matrix Y and a knowledge graph G. The objective is to forecast user u's probable interest in item v.

3. KGCNSR Model

In this paper, we propose a recommendation model that fuses knowledge graph convolutional networks and social relationships. The model is broken down into three parts: First, the user trust, social similarity and user rating similarity are aggregated by weighting to get the user similarity based on social network, and the matrix decomposition model determines the user's eigenvector representation. Second, we use the graph convolutional network to characterize the social network and get the embedding representation of the user. Third, the knowledge graph is characterized using KGCN model to get the feature vector representation of items. Finally, the predicted ratings of items by target users are obtained by fusing the three parts of data through a multilayer perceptron.

3.1. User Trust

Literature [7] Research shows that in real life, each user's interest is more or less influenced by the friends around them, so in the recommendation system, friends' interest also affects the recommendation results. In this study, the trust connections between users are divided into direct and indirect connections, with direct trust degree and indirect trust degree being utilized to quantify each connection, respectively. The direct trust degree is 1 if there is a direct trust relationship between users. If there is a connection of indirect trust between users, the indirect trust degree decreases with the increasing length of the indirect relationship. In a social network, if user a knows user b, then a and b have a direct trust connection, and the edge is written as $E_{(a,b)} = \{< a, b >\}$ and the trust degree at this time is $Tr_{(a,b)} = 1$. There exists an indirect trust connection between user a and user c if user a knows user b and user b knows user c, and the edge is $E_{(a,c)} = \{< a, b >, < b, c >\}$ and the trust degree is $Ut_{(a,c)} = (Tr_{(a,b)} + Tr_{(b,c)}) / (e^1)$.

User trust is defined as follows: in a social network, the social relationship graph G={U,E,R}, U is the set of all users, E is the set of all edges, and R is the set of all direct trust degrees (Tr) between users. Where:
$U = \{u_1, u_2, u_3, \ldots, u_k, u_m\}$;
$E = \{< u_1, u_2 >, < u_2, u_3 >, \ldots, < u_k, u_m >\}$;
$R = \{Tr(u_1, u_2), Tr(u_2, u_3), \ldots, Tr(u_k, u_m)\}$.

Set u_1 with u_m there is an indirect trust relationship. The shortest path is $Path(u_1, u_m) = \{u_1, u_2, u_3, \ldots, u_k, u_m\}$, then the trust degree Ut(User Trust) is :

$$Ut(u_1, u_m) = \frac{(Tr(u_1,u_2) + Tr(u_2,u_3) + \cdots + Tr(u_k,u_m))}{e^i} \tag{1}$$

where i is the shortest path length between u_1 and u_m minus one. This definition considers how the user's friends may have an impact on the present user. and is consistent with the trend that user trust decreases gradually with the decreasing relationship.

3.2. Social Similarity

Literature [8] showed that the greater the number of common friends among users, the more similar their interests are. Therefore, the number of common friends among users is also an important factor that affects users' interests. Social similarity is defined as Ss (Social Similarity), equation (2) displays the formula for the computation:

$$Ss(u, v) = \left|\frac{F(u) \cap F(v)}{F(u) \cup F(v)}\right| \tag{2}$$

Where $|F(u) \cap F(v)|$ denotes the number of common friends of user u and user v, and $|F(u) \cup F(v)|$ denotes the sum of the number of friends of each of user u and user v.

3.3. Similarity of User Ratings

User Score Similarity (Uss) is similar to the collaborative filtering algorithm in that it is based on historical user behavior data to calculate similarity. In this study, we first get the user's feature vector through matrix decomposition using the "user-item" scoring matrix, and then we assess user similarity using the cosine similarity. The following is the calculating formula:

$$\text{Uss}\,(u, \text{v}) = \frac{\sum_{i=1}^{n} R_{ui} \times R_{vi}}{\sqrt{\sum_{i=1}^{n} R_{ui}^2} \times \sqrt{\sum_{i=1}^{n} R_{vi}^2}} \tag{3}$$

Among them, the R_{ui} and R_{vi} represent how users u and v rated item i, respectively.

3.4. User Similarity Based on Social Networks

The user trust, social similarity, user rating similarity and user embedding representation obtained above can all have an impact on the recommendation results, so they should all be included in the calculation of User Similarity (US). In this paper, we define the user similarity formula as follows: $US(u, v) = \alpha\, Tr(u, v) + \beta\, Ss(u, v) + \gamma\, Uss(u, v)$. Where, the fusion factor $\alpha, \beta, \gamma \in [0,1]$ is the parameter that regulates the weight of user similarity, and $\alpha + \beta + \gamma = 1$.

3.5. SNBUS Model

In order to incorporate user similarity into a multilayer perceptron, it is necessary to convert user similarity into user feature vectors. In this study, the conversion will be carried out using the matrix decomposition method, and the objective function will be optimally solved using the gradient descent approach. The equation reads as follows:

$$L = \frac{1}{2} \sum_{u=1}^{m} \sum_{i=1}^{n} w_{ui} (R_{ui} - U_u I_i^T)^2 + \frac{\lambda_1}{2} \parallel U \parallel_F^2 + \frac{\lambda_2}{2} \parallel I \parallel_F^2 + \frac{\lambda_3}{2} \sum_{u=1}^{m} \sum_{v \in F(u)} US(u, v) \| u - \text{v} \|^2 \tag{4}$$

The first term is a matrix decomposition model, where R_{ui} is the real rating of item i by user u, the $U_u I_i^T$ is user u's prediction score for item i; the second and third terms are regularization terms, which are used to avoid overfitting of the feature vector during the training process; the fourth term is the user constraint term, which indicates that the features between similar users should also be similar, and $US(u, v)$ is the user similarity, and $F\,(u)$ denotes the user u of k is the set of similar users.

3.6. User's Embedded Representation

Social networks are graph data structures, so they can be processed using graph convolutional networks. The user's embedded representation (Er) is created by combining and updating the data from the user's neighbors. The graph convolution formula is shown in equation (5):

$$H^{(l+1)} = \sigma\left(\widetilde{D}^{-\frac{1}{2}}\tilde{A}\widetilde{D}^{-\frac{1}{2}}H^{(l)}W^{(l)}\right) \tag{5}$$

Where $\tilde{A}$ is the adjacency matrix with self-connection, and $\widetilde{D}$ is the $\tilde{A}$ the degree matrix, and $H^{(l)}$ is the output of the l is the output of the layer, and $H^{(0)} = \mathrm{X}$ is the user feature matrix. $W^{(l)}$ is the l layer's weight matrix, and $\sigma(\bullet)$ is the nonlinear activation function.

The user's feature vector input in this study is the user's history behavior data, which is represented by a m n interaction matrix, where m is the quantity of users and n is the quantity of items, and where 0 is present if user u does not rate item i. Each user node is aggregated with information about its neighbors to obtain a representation of the user embedding vector with potential features.

Each row vector in the user feature matrix that results from convolutional aggregation represents a user, and the user similarity may subsequently be determined using the cosine similarity approach, which is calculated as shown in equation (6):

$$\mathrm{Er}\,(u, \mathrm{v}) = \frac{\sum_{i=1}^{n} u_i \times \mathrm{v}_i}{\sqrt{\sum_{i=1}^{n} u_i^2} \times \sqrt{\sum_{i=1}^{n} \mathrm{v}_i^2}} \tag{6}$$

Among them, the u_i and v_i are the i-th dimensional vectors of users u, v, respectively.

3.7. KGCNSR Model

The input of KGCNSR model is "user-item" scoring matrix, social network and knowledge graph, and the user similarity, user embedding representation and item embedding representation based on social network are obtained respectively after the training of the above model. After training, the user's score of the target item is output.

The KGCNSR model is displays in Figure 1:

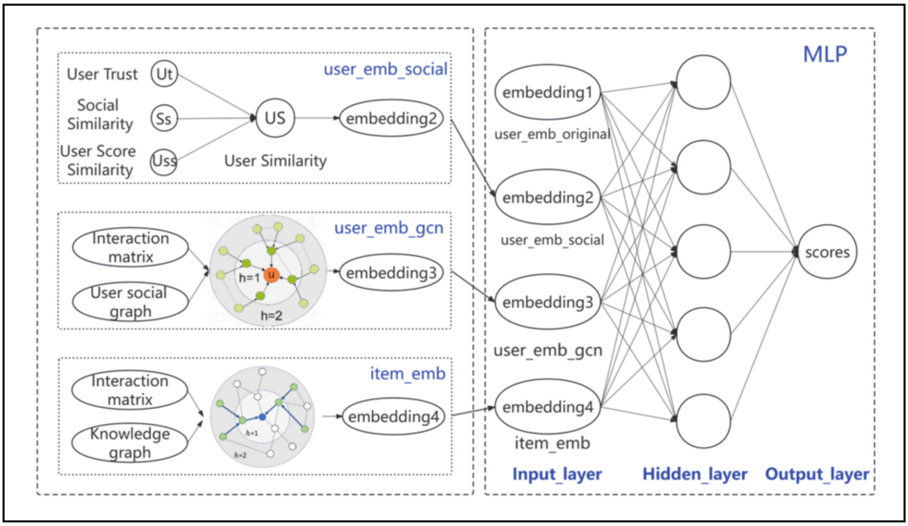

Figure 1. KGCNSR model diagram

4. Experimental Analysis

4.1 Data Sets and Evaluation Metrics

The experiment uses three publicly available datasets containing user rating information and social connections, FilmTrust[9] , CiaoDVD[10][11] and Epinions[12] They not only contain users' rating information but also their social information, which can be used to build a social relationship graph of users.

4.2 Parameter Setting and Analysis

The parameters α, β, and γ are set to 0.2, 0.2, and 0.6, respectively. d represents the dimension of the user-item representation vector, and too small a dimension will lead to the vector not being able to fully describe the user and the item, and vice versa, making it difficult for the model to converge. The parameter η represents the total number of node neighbors sampled, and too small η will lead to the model not being able to comprehensively extract the influence of neighbors on the central node, and vice versa will introduce noisy neighbor nodes, resulting in poor results. Experimentally, when d=16 and η=8 are set, the AUC and NDCG evaluation metrics achieve better results.

4.3 Contrast Model

The improved algorithm KGCNSR proposed in this paper is compared with the following baseline model:

1) LibFM: A feature-based factorization model for click-through rate (CTR) scenarios.

2) MKR: A recommendation model that uses multi-task learning using knowledge graph analysis of auxiliary information.

3) RippleNet: A memory network recommendation model by propagating user preferences over a knowledge graph.

4) KGCN: A recommendation model using graph neural networks with knowledge graphs as auxiliary data.

5) KGCNSR: A recommendation algorithm fusing knowledge graph convolutional networks and social relationships proposed in this paper.

TABLE I. PERFORMANCE COMPARISON OF DIFFERENT ALGORITHMS

Model	**CiaoDVD**		**Epinion**		**FilmTrust**	
	AUC	***NDCG***	***AUC***	***NDCG***	***AUC***	***NDCG***
LibFM	0.635	0.897	0.702	0.921	0.918	0.968
MKR	0.658	0.941	0.732	0.958	0.955	0.993
RippleNet	0.648	0.914	0.715	0.937	0.931	0.981
KGCN	0.654	0.936	0.728	0.955	0.948	0.992
KGCNSR	0.721	0.970	0.759	0.962	0.974	0.996

As shown in Table 1, the improved algorithm KGCNSR proposed in this paper has all improved compared with the baseline model. Among them, the highest improvement in AUC evaluation is 13.54%, 8.12% and 6.10% on the datasets of, CiaoDVD Epinions and FilmTrust; the highest improvement in KGCNSR is also 8.14%,4.45% and 2.89% respectively on the evaluation criteria of NDCG.

5. Conclusion

In this paper, we present the KGCNSR recommendation model, which combines social relationships and knowledge graph convolutional networks. The model determines the feature matrix for the user using a matrix decomposition model, obtains the embedding representation for the user using a social network, and determines the feature matrix for the object using a knowledge graph convolutional network. In order to obtain the anticipated scores of things from the target users and provide suggestions for the users, the previous three types of data are fed into a multilayer perceptron. Experiments on three publicly available datasets, CiaoDVD, Epinions and FilmTrust, and comparisons with several benchmark models validate that the recommendations of the KGCNSR model outperform other benchmark models.

Both the knowledge graph and the social network in the KGCNSR model are static, and various information of users in real life may change, so the focus of future research should be on how to deal with the characteristics of items and users dynamically.

References

[1] Wang H, Zhang F, Xie X, et al. DKN: Deep knowledge-aware network for news recommendation[C]. Proceedings of the 2018 world wide web conference, 2018: 1835-1844.

[2] Wang H, Zhang F, Wang J, et al. RippleNet: Propagating User Preferences on the Knowledge Graph for Recommender Systems[J]. ACM, 2018.

[3] Wang H, Zhao M, Xie X, et al. Knowledge Graph Convolutional Networks for Recommender Systems[J]. arXiv, 2019.

[4] Yu J, Yin H, Li J, et al. Self-Supervised Multi-Channel Hypergraph Convolutional Network for Social Recommendation, 2021.

[5] Yang Z X, Dong S B. HAGERec: Hierarchical Attention Graph Convolutional Network Incorporating Knowledge Graph for Explainable Recommendation[J]. Knowledge-Based Systems, 2020, 204: 11.

[6] He X, Deng K, Wang X, et al. LightGCN: Simplifying and Powering Graph Convolution Network for Recommendation: ACM, 2020.

[7] Jamali M, Ester M. A matrix factorization technique with trust propagation for recommendation in social networks[C]. Proceedings of the fourth ACM conference on Recommender systems, 2010: 135-142.

[8] Guo G, Jie Z, Thalmann D. Merging trust in collaborative filtering to alleviate data sparsity and cold start[J]. Knowledge-Based Systems, 2014, 57(feb.): 57-68.

[9] Guo G, Zhang J, Yorke-Smith N. A novel Bayesian similarity measure for recommender systems[C]. IJCAI, 2013: 2619-2625.

[10] Tang J, Gao H, Liu H. mTrust: Discerning multi-faceted trust in a connected world[C]. Proceedings of the fifth ACM international conference on Web search and data mining, 2012: 93-102.

[11] Tang J, Gao H, Liu H, et al. eTrust: Understanding trust evolution in an online world[C]. Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining, 2012: 253-261.

[12] Massa P, Souren K, Salvetti M, et al. Trustlet, open research on trust metrics[J]. Scalable Computing: Practice and Experience, 2008, 9(4)