A Recommended Method Based on the Weighted RippleNet Network Mode

Yutai Luo, Baocheng Sha and Tao Xu

Chinese National Language and Character Information Technology Laboratory, Northwest Minzu University, Lanzhou, Gansu, China
Email: xutao@xbmu.edu.cn

Abstract. User preferences were modeled by the RippleNet network and successfully applied in the recommender systems, but the weight of the entity was not considered. This paper proposes a RippleNet model incorporating the influence of complex network nodes. After the construction of complex networks based on knowledge Graphs, we build the maximum subnet model and calculate the influence of nodes in the graph network. We added it to the RippleNet as the weight of entities. The experimental results showed that new method increased the AUC and ACC values of RippleNet to 92.0% and 84.6%, solve the problem that entity influence was not considered in the RippleNet network, and made the recommended results more in line with users’ expectations.

Keywords. RippleNet; recommender systems; complex network; node influence.

1. Introduction
The traditional recommendation system method is based on collaborative filtering. That guess their possible common interests through the analysis of users’ historical interactive data [1]. But this method has the problems of cold start and sparse matrix. Researchers have proposed many solutions, such as combining knowledge graphs and social networks [2]. They add side information to the matrix. But there are still have some limitations.

There are two recommendation methods based on knowledge graph in recommender systems. They are embedding-based method and path-based method.

Zhang et al. proposed a collaborative knowledge base embedding (CKE) method and combined the collaborative filtering model with embedding [3] in 2016. Wang et al. proposed deep knowledge perception Network (DKN) in 2018. He divided entity embedding and word embedding into two different channels, and designed a CNN model to combine them [4]. Yu et al. proposed the heterogeneous information network method [5] in 2014. Zhao et al. proposed the recommendation method based on meta-graphs [6] in 2017. Both of them represent the relationship between users and items by creating heterogeneous information networks and extracting meta paths or meta graphs based on the potential characteristics of the network. Wang et al. Proposed RippleNet [7] by combining embedding and path based methods. Wang simulated the propagation mode of water ripple and the user preference and got very good results. However, RippleNet does not consider the weight of entities in the limited domain knowledge graph which cannot get the recommendation results of entities with different weights and does not focus on the more important entities. So it reduces the accuracy of recommendation.

Scholars have done many researches on the evaluation of node propagation ability in complex network science. Currently, the commonly used methods are Degree Centrality [8], Betweenness...
Centrality [9] and Closeness Centrality [10], etc. We use the method of complex network science to solve the problem of RippleNet. Entities of the triple in the knowledge graph were taken as nodes and relations as edges to calculate the node influence of entities in the complex network, which was put into the RippleNet network as the weight of entities for calculation.

2. Improved RippleNet Algorithm

2.1. Building Complex Networks Based on Knowledge Graph

We use the film and book knowledge Graphs in RippleNet [7]. The graph data is a pure text file and the data of the file is triple (entity-relation-entity). For example, all entities are regarded as nodes of the network and the relationships between entities are regarded as edge to represent the association between nodes. When we constructing the complex network of film graph, the established complex network of film Knowledge Graph is a no-directional and unweighted network. It has the following characteristics:

- The network size is huge. Contains 169,366 nodes and 333,543 edges. So we establish an unweighted network to reduce the computational complexity and facilitate feature analysis.
- It only describes the relationship between entities and the distance between them and ignoring the direction of the relationship.

Because the network constructed by knowledge graph is not necessarily a fully connected network and some nodes cannot participate in the calculation of their node influence. So we need to eliminate redundancy and calculate maximum connected subnet of film knowledge graph.

We use subnet extraction algorithm based on set operation [11]. The method takes Graph, a film knowledge Graph file, as the input. The node set MaxSubNet are output of maximum connected subnet that obtained by subnet extraction algorithm.

The algorithm process is as follows:

1. We select a triple from the Graph whose node have the highest degree value and use the two entities in the triple as the central nodes. SubNet, (i =1), the core layer of the maximum connected subnet is added to MaxSubNet.
2. Find the adjacent nodes of SubNet. Traverse the triples (head, relation, tail) in the Graph to determine whether the head and tail exist in the SubNet. If there is, it means that the corresponding head or tail is the adjacent node of the SubNet. Then add these adjacent nodes to NeighborsSet.
3. Compare NeighborsSet with MaxSubNet. If the node of NeighborsSet is not in MaxSubNet, NeighborsSet is added in MaxSubNet and SubNet will be replaced with the difference set of NeighborsSet and MaxSubNet as the new SubNet, i = i +1, then Jump to (2). If no new node exists, proceed to (4).
4. Return MaxSubNet.

The key step of the algorithm is traversing Graph triples. Each time we traverse the Graph, we get all the adjacent nodes of the node in the current layer. Because the nodes with higher degree value are more likely to exist in the maximum connected subnet. Therefore, the node with the highest degree value can be selected as the central node to obtain the maximum connected subnet with a higher probability.

After finding all entities of the maximum connected subnet, extract the set of edges of the maximum subnet from the movie knowledge Graph file Graph according to MaxSubNet and store it in GraphLink. Finally, MaxSubNet and GraphLink generate the maximum connected subnet as shown in figure 1. The storage format is still in the form of triples. It can be seen that the core layer is composed of two points. The adjacent entities of the two points as the second layer and the adjacent entities of the second layer as the third layer. Keeping recursion to get the entire subnet, until there are no new adjacent entities.
2.2. Node Influence Calculation

A complex network usually contains important nodes. Some important nodes generally affect most of the nodes in the network. Node influence refers to the analysis of all nodes in a complex network and the sorting of influential or important nodes in the network. In different networks, researchers will establish measurement indexes of node influence from different scales, directions and experimental conditions. Trying to find the most influential nodes in the most accurate and rapid way, then sort them [12].

We use the maximum connected subnet as a complex network to calculate the influence of all triple nodes. In this paper, the degree centrality method [8] is used to calculate the node influence of the film knowledge graph network to find the node with greater influence in the film knowledge graph.

In degree centrality, the degree value of a node refers to the number of other nodes connecting the node. Degree centrality is calculated according to the degree value of a single node and the total number of nodes. The greater the degree value is, the greater the degree centrality will be. In the knowledge graph of limited domain, the degree distribution of most networks is power-law distribution. So the degree value of a few nodes is large, while the degree value of a large number of nodes is small. Therefore, we use degree centrality to calculate node influence, which can accurately distinguish nodes with different degree values and calculate their centrality for all entities as their weight information.

Since the network composed of film knowledge graph is a directionless network, the specific calculation formula is as follows:

$$C_D(N_i) = \frac{\sum_{j=1}^{g} x_{ij} (i \neq j)}{g-1}$$

In a directionless network, \(g\) is the total number of nodes. \(i\) is a single node. \(C_D(N_i)\) is degree centrality of \(i\). The final result is a scale, ranging from 0 to 1. 0 means it has no relationship to any node and 1 means it has a direct relationship to each node. We calculate the degree centrality of all nodes in the knowledge graph network and add to the entities in the graph. We use dictionary to save, which is marked as \(C\).

2.3. RippleNet Model of Joint Complex Network

The RippleNet is a propagation method that simulates water ripples. It takes the user’s click history as the seed and uses the seed as the initial point in the limited domain knowledge graph to spread outward round by round. This process is called the user’s preference propagation. The model considers...
that the items in the seed outer ring still belong to the user’s potential preference, so it should be considered. The RippleNet model is improved as follows:

The degree centrality of head in the current Hop of Ripple set (triple) is obtained by the above experiment and is used as the weight of the head. Then multiply the embedding matrix of the Ripple set with the weight matrix of head to obtain the Ripple set embedding with weight. Finally, user embedding and item embedding are calculated to get the final result with entity weight.

The framework of improved RippleNet is shown in figure 2:

**Figure 2.** Improved RippleNet model of joint complex network.

In the RippleNet, user \( u \) and item \( v \) are respectively taken as the input of the model. The predicted probability that user \( u \) may click item \( v \) is taken as the final output. Each user \( u \) builds his Ripple set based on his click history \( v_u \), whose click history acts as a seed for preference propagation. Then, the Ripple set \( S_u^k (k = 1, 2, ..., H) \) corresponding to Hop is formed along the relationship of knowledge graph. Ripple set \( S_u^k \) is a triple after user clicks history \( v_u \) to propagate preference. Ripple set interacts with item embedding to fuse the information of user \( u \) and item \( v \). Then combine these information to form the final user embedding. Finally, the embedding of user \( u \) and item \( v \) are used to calculate the final prediction probability \( \hat{y}_{uv} \).

First, we use the user’s historical click data \( v_u \) in the knowledge graph to represent the preference-related entities based on the user’s historical \( v_u \). A set of preference-related entities is created for user \( u \) by recursive method, as shown below:

\[
\sigma_u^k = \{t \mid (h, r, t) \in \mathcal{G}, h \in \sigma_u^{k-1}\}, k = 1, 2, ..., H.
\]

These entities can be viewed as extensions of the user’s preferences in the knowledge graph based on the historical click \( v_u \). After defining the relevant entities, we define the K-hop Ripple set for user \( u \) as follows:

\[
S_u^k = \{(h, r, t) \mid (h, r, t) \in \mathcal{G}, h \in \sigma_u^{k-1}\}, k = 1, 2, ..., H.
\]

In each Hop, the degree centrality \( c_i \) of the head entity in the Ripple set is taken as the weight and multiplied by the embedding matrix \( \text{head}_i \) of the Ripple set to get the weighted \( h_i \). As shown below:
Then we calculate the correlation probability $p_i$ of each triple $(h_i, r, t_i)$ by comparing item $v$ with the header $h_i$ and relationship $r_i$ in triples:

$$p_i = \text{softmax}(v^T R h_i) = \frac{\exp(v^T R h_i)}{\sum_{(h, r, t) \in S_i} \exp(v^T R h)}$$

The correlation probability can be viewed as a measure of $R_i$ in the relational space between item $v$ and head entity $h_i$. Because different types of relationships will definitely calculate different similarities. Relational data $R_i$ also participates in the calculation of correlation probability. After obtaining the correlation probability, all tail entities $tail$ in $S_i$ are multiplied by the corresponding correlation probability and the vector $o_i^u$ is returned. $o_i^u$ can be regarded as the first response of user to the historical click record and used to form the embedding of user $u$.

$$o_i^u = \sum_{(h_i, r, t_i) \in S_i} p_i t_i$$

$$U = o_i^1 + o_i^2 + \ldots + o_i^{H}$$

Finally, combined with user embedding and item embedding, the predicted click probability is output:

$$\hat{y}_{uv} = \frac{1}{1 + \exp(U^T v)}$$

3. Experimental Setup and Result Analysis

3.1. Data Set

In our experiment, we used the following two data sets for experiments:

MovieLens-1M and Book-Crossing are data sets often used in movie recommendation and book recommendation respectively. The movie data set MovieLens-1M contains millions of ratings from real users. Book-Crossing also contains millions of books which are rated by users in the Book Exchange community. This paper uses the knowledge graph constructed for each data set in RippleNet model to conduct experiments. We can get the entity, relationship, and triples of the two data sets in table 1.

| Table 1. Basic statistics of two data sets. |
|-------------------------------------------|
| **MovieLens-1M** | **Book-Crossing** |
| User          | 6036          | 17860          |
| Item          | 2445          | 14967          |
| Relational interaction | 753772   | 139746          |
| Triples of 1-hop | 20782    | 19876           |
| Triples of 2-hop | 178049    | 65360           |

3.2. Experimental Environment

We set the hop number $H=2$ for MovieLens-1M/Book-Crossing. According to the experimental results, many hops are not only difficult to improve performance, but also leads to greater calculation power. The complete parameter setting is given in table 2, where $d$ represents the embedding dimension of the item and knowledge graph. $\eta$ represents the learning rate.
We will process each data set separately and divide the whole data set into training set, evaluation set and test set. The proportion of data is 6:2:2. Five experiments were carried out for each data set, and the average value was taken as the final result.

| Table 2. Hyperparameters for the movie and book data sets. |
|----------------------------------------------------------|
| MovieLens-1M: $d = 16, H = 2, \lambda_1 = 10^{-7}, \lambda_2 = 0.01, \eta = 0.02$ |
| Book-Crossing: $d = 4, H = 3, \lambda_1 = 10^{-5}, \lambda_2 = 0.01, \eta = 0.001$ |

3.3. Valuation Methodology
ACC (accuracy) and AUC (area under curve) are used as the evaluation indexes of the experiment. AUC comes from ROC, which is the area of the lower part of the ROC curve. The value of AUC is less than 1. The ROC curve graph is used to judge the advantages and disadvantages of binary classifiers. The horizontal coordinate of the curve graph is FPR (false positive rate) and the vertical coordinate is TPR (true positive rate). These coordinate pairs are connected to form the ROC curve graph. Finally, the area of AUC is calculated to get a value of 0.5 to 1. The larger the AUC, the better the classification effect.

3.4. Experimental Results
This paper constructs the knowledge graph of movies and books and uses the subnet extraction algorithm based on set operation to construct the complex network. Part of the triple data of the maximum connected subnet is shown in table 3. For the convenience of use and statistics, entities and relationships are represented by serial numbers.

| Table 3. Partial data of maximum connected subnet. |
|--------------------------------------------------|
| Head | Rel | Tail   |
| 49695 | 4   | 73697  |
| 36508 | 9   | 9892   |
| 34029 | 4   | 73698  |
| 73699 | 9   | 73700  |
| 6511  | 1   | 57474  |
| 4159  | 4   | 73701  |
| 73702 | 2   | 2466   |

By calculating the influence of nodes, we get the degree centrality of each entity in the subnet. The results of degree centrality of some film knowledge graph entities are as follows in table 4:

| Table 4. Partial centrality of maximum connected subnet. |
|----------------------------------------------------------|
| Entity | Degree Centricity |
| 2458   | 0.216795780       |
| 2755   | 0.100912037       |
| 35906  | 0.000137355       |
| 52673  | 0.000005494       |
| 73289  | 0.000038495       |
| 74102  | 0.000009884       |
| 144090 | 0.000027471       |
Table 5. Results of AUC and Accuracy in different models.

| Model       | MovieLens-1M | Book-Crossing |
|-------------|--------------|---------------|
|             | AUC         | ACC          | AUC | ACC |
| Cn-RippleNet| 0.920       | 0.846        | 0.745| 0.681|
| RippleNet   | 0.911       | 0.834        | 0.721| 0.661|
| CKE         | 0.796       | 0.739        | 0.674| 0.635|
| SHINE       | 0.778       | 0.732        | 0.668| 0.631|
| DKN         | 0.655       | 0.589        | 0.621| 0.598|
| PER         | 0.712       | 0.667        | 0.623| 0.588|

The results obtained when different parameters are tested. The Cn-RippleNet model performs best when parameters dim=16, lr=0.02, kg_wight=0.02 and both AUC and ACC obtained satisfactory results.

As shown in table 5, compared with other mainstream recommendation models based on knowledge graph, Cn-RippleNet achieves the best performance on both datasets. The reason is that Cn-RippleNet combines path based and embedding based methods. It uses the side information and node influence of knowledge graph and gives weight to the entities in the network which can give better recommendation to the recommendation system and get more accurate recommendation results.

According to the results of table 5, we can find that compared with the performance of CKE, Cn-RippleNet is respectively 12 and 11 percentage points higher in the evaluation index of AUC and ACC. Because CKE combines knowledge graphs with collaborative filtering method. It only makes use of the structural knowledge characteristics of graphs and does not consider the path of graphs.

DKN failed to perform satisfactorily in terms of movie and book recommendations, being increased by 27 percentage points and 25 percentage points respectively by Cn-RippleNet model. Because DKN pays more attention to the text content and requires multiple entities to ensure the accuracy of prediction. For DKN, the entity name of data set is not long enough to provide useful information and the result is more inaccurate.

Cn-RippleNet overcomes this problem by combining path-based method and gaining node influence. Therefore, Cn-RippleNet can also achieve high performance in the case of insufficient data length.

As user-defined meta-paths are hardly the best, PER’s performance in film and book adaptations is not satisfactory. While Cn-RippleNet’s path combining knowledge graph entities achieves higher performance. In these two data sets, Cn-RippleNet performs better.

In contrast to Ripplenet, the improved Cn-RippleNet in the MovieLens-1M data set was 1 percentage point higher in both AUC and ACC. While the index in the BookCrossing data set was 2 percentage points higher. This shows that the recommendation effect of RippleNet considering the influence of nodes is more accurate. Therefore, this paper believes that the effect of Cn-RippleNet is better after the method of integrating node influence according to the path.

At the same time, it can be seen from figure 3 that the result is the best when the value of Hop is 2 or 3. The result of Hop number on performance is not that the higher value, the better the effect. Shown in figure 4, the result of Dim is also similar to Hop. At Dim 16, the model performance is optimal.
4. Conclusion

We proposed a RippleNet recommendation model combining the influence of complex network nodes in this paper. We networked the domain knowledge graph and calculated the influence of the entities in the knowledge graph. And integrated into the RippleNet. So that the recommendation result is more in line with people's interest. The problem that the RippleNet does not consider the influence of key nodes on the recommendation result is solved. Thus increasing the recommendation accuracy. Experiments show that the improved RippleNet algorithm by node influence can improve the effect of the recommended algorithm. The next step is that consider the maximum connected subnet of the knowledge graph as a directed network. Then refine the relationships and assign different weights to different relationships. More new node influence algorithms can also be considered.
Acknowledgment
The work is supported by the Fundamental Research Funds for the Central University (NO.319202101017), Gansu Province Archives Science and Technology Project (GS-2020-X-07), Gansu Province Youth Science and Technology Fund Project(21JR1RA211), and Major National R&D Projects (NO.2017YFB1002103).

References
[1] Koren E, Bell R and Volinsky C 2009 Matrix factorization techniques for recommender systems Computer 42 8 (2009).
[2] Jamali M and Ester M 2010 A matrix factorization technique with trust propagation for recommendation in social networks Proceedings of the 4th ACM Conference on Recommender Systems (ACM) pp 135-142.
[3] Zhang F Z, Nicholas J Y and Lian D F 2016 Collaborative knowledge base embedding for recommender systems Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (ACM) pp 353-362.
[4] Wang H W, Zhang F Z and Xie X 2018 DKN: Deep knowledge-aware network for news recommendation Proceedings of the 2018 World Wide Web Conference pp 1835-1844.
[5] Yu X, Ren X and Sun Y Z 2014 Personalized entity recommendation: A heterogeneous information network approach Proceedings of the 7th ACM International Conference on Web Search and Data Mining pp 283-292.
[6] Zhao H, Yao Q M and Li J D 2017 Metagraph based recommendation fusion over heterogeneous information networks Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining pp 635-644.
[7] Wang H W, Zhang F Z and Wang J L 2019 Exploring high-order user preference on the knowledge graph for recommender systems ACM Transactions on Information Systems (TOIS) 37 (3) 1-26.
[8] Freeman L C 1978 Centrality in social networks conceptual clarification Social Networks 1 (3) 215-239.
[9] Freeman L C 1977 A set of measures of centrality based on betweenness Sociometry 1977 35-41.
[10] Sabidussi G 1966 The centrality index of a graph Psychometrika 31 (4) 581-603.
[11] Ding L H, Sun B and Shi P 2019 Empirical study of knowledge network based on complex network theory Journal of Physics 68 (12) 324-338.
[12] Fan Y N, Liu S Y and Bai Y G 2020 Identifying critical nodes in complex networks based on multi-scale centrality algorithm The Practice and Understanding of Mathematics 50 (10) 159-167.