A Framework for Recommendation Algorithms Using Knowledge Graph and Random Walk Methods

Takafumi Suzuki
Hokkaido University
szk.tkfm@frontier.hokudai.ac.jp

Satoshi Oyama
Hokkaido University
oyama@ist.hokudai.ac.jp

Masahito Kurihara
Hokkaido University
kurihara@ist.hokudai.ac.jp

2020 IEEE International Conference on Big Data (Big Data) | 978-1-7281-6251-5/20/$31.00 ©2020 IEEE | DOI: 10.1109/BigData50022.2020.9378103

***Abstract*—A number of knowledge graph (KG)-based recommendation algorithms have been introduced; KGs enable users and items and their attributes to be treated in an integrated way and structural information to be captured through graphs. There are many variations of the KG based recommendation algorithms. Among them, KG embedding is often used, but doing this does not take advantage of the meta-path-level proximity between users and items. This paper presents a flexible framework combining random walk and KG embedding methods. The random walk model is formulated on the basis of the similarity between nodes revealed by the KG embedding. This enables the meta-path level proximity of users and items to be efficiently utilized. Comparison testing demonstrated that the proposed framework performs better than random-walk-only methods and KG-embedding-only methods, and slightly better than the existing method we have extended.**

***Index Terms*—Recommendation, Knowledge Graph Embedding, Random Walk**

I. INTRODUCTION

The introduction of knowledge graphs (KGs) into recommendation algorithms has been attracting a great deal of attention. KGs are graphs in which the nodes act as entities representing users and items and their attributes, and the edges represent the relationships between the entities. KG-based recommendation algorithms can be roughly divided into embedding-based models and path-based models [1]. Embedding-based models use an embedding method similar to ones such as TransE [2] and ComplEx [3], which embed the nodes of the KG into a vector of continuous values. KG embedding enables use of the structure of the graph composed of the recommendation data. Most path-based models use methods that integrate the meta-path extracted from the KG with matrix factorization and deep learning (DL). Path-based methods can be used to investigate the fit between users and items on the basis of the meta-paths with two or more different lengths between users and items in a KG. However, it these methods are not scalable enough to extract meta-paths from a KG. In contrast, random- walk-based algorithms using a graph structure are scalable and can make use of meta-paths between user-items [4] [5]. Nikolakopoulos et al. [6] proposed using a framework that combining a random walk method with an arbitrary model that obtains item similarity from a graph consisting of users and items only. A model based on the Sparse Linear Method (SLIM) [7] was used to obtain item similarity, and the performance was reported to be better than that of a DL-based model. We have extended the framework of Nikolakopoulos et al. [6] to combine a random walk method with a KG embedding methods. The embedding method calculates the similarity of users and items, and the random walk method predicts the top-N recommendation items. Because any model can be used as the embedding method, we can select an appropriate embedding model in accordance with the structure of the KG data for the recommendation task. In addition, the random walk method can efficiently utilize the proximity between users and items at the meta-path level.

II. RELATED WORK

As mentioned above, KG-based recommendation systems are roughly divided into embedding-based models and path-based models, and there are various versions of each type [1]. Although there are no recommendation algorithms yet that combine random walk and KG embedding methods, the DeepWalk method [8] uses a random walk approach for general KG node embedding. Nikolakopoulos et al. [6] used a framework combining a random walk method with an arbitrary model to obtain item similarity. This framework models the process by which a random walker transitions on a bipartite graph consisting of users and items while teleporting to similar item nodes with a certain probability. SLIM [7] is used to calculate the similarity of items.

III. PRELIMINARY

A. Knowledge Graph

A knowledge graph is a directed graph in which the nodes are entities $\mathcal{E}$ and the edges denote their relations $\mathcal{R}$. Formally, we define a KG as $\mathcal{KG} = \{(h, r, t) \mid h, t \in \mathcal{E}, r \in \mathcal{R}\}$, where each triplet (h, r, t) indicates that there is a relationship from the head entity to the tail entity. Each entity belongs to a certain entity class $\mathcal{E}_C \subset \mathcal{E}$.

In this paper, users and items and their attribute information are represented by entities and relations. If there is an observed interaction between a user and an item (e.g., rate, click, purchase), we represent that interaction as a triplet $(u, interact, i)$ and $(i, interact^{-1}, u)$, where *interact* and $interact^{-1}$ are predefined relationship. If item i has a as an attribute, that fact is represented as a triplet (i, r, a), where r is a relationship defined in accordance with a.

B. Knowledge Graph Embedding

Knowledge graph embedding is the transformation of an entity on KG into a vector of low-dimensional continuous values. The embedded vectors contain structural information about the graph as represented by the distances and inner products on the embedded space. In this study, we used the TransE KG embedding methods [2]. Each embedding of triplet(h, r, t) in KG, $\mathbf{h}, \mathbf{r}$, and $\mathbf{t}$, is obtained by minimizing the following objective function with respect to the embedding vector.

$$L = \sum_{(h,r,t)\in D} \sum_{(h_{neg},r,t_{neg})\in D_{neg}} \gamma + d(\mathbf{h}+\mathbf{r},\mathbf{t}) - d(\mathbf{h}_{neg}+\mathbf{r},\mathbf{t}_{neg}),$$

where D is the training data, D_{neg} is the negative sample, $\gamma > 0$ is the margin hyperparameter, and $d(.,.)$ is the distance function. We use Euclidean distance as the distance function, so the vector embedding triplet(h, r, t) in KG is trained to be $\mathbf{h} + \mathbf{r} \approx \mathbf{t}$.

IV. FRAMEWORK

In this section, we describe the framework used for combining random walk method with KG embedding methods. The embedding method is used to calculate the similarity of users and items, and the random walk method is used to predict the top-N recommended items.

A. Random Walk

We use the Personalized PageRank (PPR) random walk algorithm to predict the top-N recommendation items. We consider a knowledge graph with N nodes consisting of users, items, and attributes. A random walker starts from the node of a user u, transitions randomly to neighboring nodes, and teleports to the node of another user u with probability $1 - \mu$. The stationary distribution of this process represents the importance of each node. Items of top-n value in the distribution are recommended. If the probability transition matrix of the graph is $P \in \mathbb{R}^{N\times N}$ and the damping factor is μ, then the stationary distribution $\mathbf{p}_u \in \mathbb{R}^N$ is

$$\mathbf{p}_u = \lim_{K\to\infty} \mathbf{e}_u^\top(\mu P + (1-\mu)\mathbf{1}\mathbf{e}_u^\top)^K,$$

where $\mathbf{e}_u \in \mathbb{R}^N$ is a vector that contains the element 1 on the position that corresponds to user u and zeros elsewhere. The probability transition matrix P is usually $P = diag(A\mathbf{1})^{-1}A$, where $A \in \mathbb{R}^{N\times N}$ is the adjacency matrix, but in this study we use a probability transition matrix that takes into account the similarity between nodes using node embedding.

TABLE I
NUMBER OF KG ENTITIES AND RELATIONS

entity	user	3819
	item	1581
	brand	8
relation	user *interact* item	34278
	item $interact^{-1}$ user	34278
	item *belong to* brand	2
	item *also view* item	454
	item *also buy* item	554

B. Random Walk with KG Embedding

We consider a model in which a random walker transitions to neighboring nodes and teleports to similar nodes with a certain probability. The similarity is the inner product of the embedding vector. For embedding vector $\mathbf{v}_i(i = 1, ..., \mid E_c \mid)$ of an entity belonging to entity class $\mathcal{E}_c$, we define matrix $M_c \in \mathbb{R}^{|E_c|\times|E_c|}$ as

$$M_c[i,j] = \mathbf{v}_i^\top \mathbf{v}_j.$$

Let C be the number of entity classes and define matrix M as

$$M = diag[M_1, M_2, ..., M_C].$$

M is a block diagonal matrix in which the diagonal components contain the inner product of nodes of the same entity class. We normalize M as follows.

$$M_n = \frac{1}{\|M\|_\infty} M + diag(\mathbf{1} - \frac{1}{\|M\|_\infty} M\mathbf{1})$$

where $\| M \|_\infty$ denotes the maximum row-sum of M. This definition retains the relative differences in similarity between entities, i.e. $M[i,j] \geq M[i',j'] \Rightarrow M_n[i,j] \geq M_n[i',j']$ for all i, j, i', j' [6]. Let A be the adjacency matrix of KG and calculate the probability transition matrix as

$$P = \alpha \cdot diag(A\mathbf{1})^{-1}A + (1-\alpha)M_n.$$

We perform a random walk in accordance with this transition probability matrix and make predictions using PPR. The proposed framework is an extension of RecWalk [6] and includes it.

V. EVALUATION

We evaluated the proposed framework with *5-core Luxury* Amazon review data [9]. This is a subset of the luxury product data for which all users and items have at least five reviews. Table I summarizes the entities and relations in a KG composed of this data. We used 1/3 of the data as test data and performed cross validation and hyperparameter tuning on the remaining 2/3.

We performed a top-n recommendation task using the following methods and compared their performances using the mean average precision (MAP) scores.

- **PPR** We use the naive random walk model, PPR, as a baseline.

- **BPR** [10] Bayesian Personalized Ranking(BPR) is a matrix factorization algorithm that optimizes the ranking of the recommended items.
- **NFM** [11] We use the Neural Factorization Model (NFM), which is a DL extension of the Factorization Machine(FM) [12], as a baseline for the DL-based models.
- **TransE** [2] This is the KG embedded model described above.
- **RecWalk** [6] This method uses SLIM [7] to calculate the similarity of items and perform a random walk.

The results of the comparison are shown in Table II. The proposed framework slightly outperformed RecWalk.

TABLE II
MAP score

PPR	BPR	NFM	TransE	RecWalk	**Proposed**
0.09589	0.2569	0.1210	0.1714	0.2703	**0.2708**

Since the proposed framework is an extension of RecWalk and includes it, this result is not surprising, but the difference in scores is small. This may be because the structure of the KG constructed from the dataset is not so different from that of a bipartite graph constructed of user items only. In this dataset, as shown in Table I, there is little information other than users, items, and their interaction relationships. The difference in scores between RecWalk and the proposed framework may be larger for more complex graphs that include more attributes and relationships. The proposed method outscored both PPR and TransE. This indicates that the proposed framework does a good job of learning the representation of nodes by KG embedding and utilizing proximity at the meta-path level by random walking.

VI. Conclusion

Our proposed framework combines KG embedding and random walk methods. It is a flexible framework that enables the use of any model as the embedding method and the choice of an appropriate embedding model in accordance with the structure of the KG data for the recommendation task. This framework is an extension of the RecWalk random-walk-based method [6] to a KG. Experimental results showed that the proposed framework outperformed the random walk and KG embedding methods alone and slightly outperformed RecWalk, thereby demonstrating the effectiveness of the proposed framework.

In future research, we will conduct experiments on a wider range of data and compare the results with other KG-based recommendation algorithms. We will also work on the explainability of the recommendation algorithms. One of the motivations for using KG as a recommendation algorithm is its high explainability [1]. An algorithm to give explainablity to RecWalk has been proposed [13], and our future work include developing a method to give explainability properties to the proposed algorithm.

Acknowledgments

This work was partially supported by JSPS KAKENHI Grant JP18H03337 and by the Global Station for Big Data and Cybersecurity, a project of the Global Institution for Collaborative Research and Education at Hokkaido University.

References

[1] Q. Guo, F. Zhuang, C. Qin, H. Zhu, X. Xie, H. Xiong, and Q. He, “A survey on knowledge graph-based recommender systems,” *ArXiv*, vol. abs/2003.00911, 2020.

[2] A. Bordes, N. Usunier, A. Garcia-Durán, J. Weston, and O. Yakhnenko, “Translating embeddings for modeling multi-relational data,” in *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, 2013.

[3] T. Trouillon, J. Welbl, S. Riedel, E. Gaussier, and G. Bouchard, “Complex embeddings for simple link prediction,” in *Proceedings of the 33rd International Conference on on Machine Learning*, 2016.

[4] F. Christoffel, B. Paudel, C. Newell, and A. Bernstein, “Blockbusters and wallflowers: Accurate, diverse, and scalable recommendations with random walks,” in *Proceedings of the 9th ACM Conference on Recommender Systems(RecSys ’15)*, 2015.

[5] C. Eksombatchai, P. Jindal, J. Z. Liu, Y. Liu, R. Sharma, C. Sugnet, M. Ulrich, and J. Leskovec, “Pixie: A system for recommending 3+ billion items to 200+ million users in real-time,” in *Proceedings of the 2018 World Wide Web Conference*, 2018.

[6] A. N. Nikolakopoulos and G. Karypis, “Recwalk: Nearly uncoupled random walks for top-n recommendation,” in *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, 2019.

[7] X. Ning and G. Karypis, “SLIM: Sparse linear methods for top-n recommender systems,” in *2011 IEEE 11th International Conference on Data Mining*, 2011.

[8] B. Perozzi, R. Al-Rfou, and S. Skiena, “Deepwalk: Online learning of social representations,” in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2014.

[9] R. He and J. McAuley, “Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering,” in *Proceedings of the 25th International Conference on World Wide Web*, 2016.

[10] “BPR: Bayesian personalized ranking from implicit feedback,” in *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, 2012.

[11] X. He and T.-S. Chua, “Neural factorization machines for sparse predictive analytics,” in *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2017.

[12] S. Rendle, “Factorization machines,” in *2010 IEEE International Conference on Data Mining*, 2010.

[13] A. Ghazimatin, O. Balalau, R. S. Roy, and G. Weikum, “PRINCE: Provider-side interpretability with counterfactual explanations in recommender systems,” in *Proceedings of the 13th International Conference on Web Search and Data Mining*, 2020.