Multiary Relational Knowledge Base Completion via Tensor Decomposition

Yu Liu[1,2] , **Quanming Yao**[3] , **Yong Li**[1,2]
[1]Department of Electronic Engineering, Tsinghua University, China
[2] Beijing National Research Center for Information Science and Technology, China
[3]4Paradigm Inc, Hong Kong
liuyu2419@126.com, yaoquanming@4paradigm.com, liyong07@tsinghua.edu.cn

Abstract

As the generalization of knowledge graph, the multiary relational knowledge base (KB) with binary and beyond-binary relational facts, is closer to the real-world knowledge, but still underexplored. In this short paper, we investigate the multiary relational knowledge base completion problem, and propose a generalized model based on Tucker decomposition and Tensor Ring decomposition. Compared to the state-of-the-art methods, the proposed model obtains a relative improvement of 7% and 9% on two benchmark multiary relational KB datasets respectively.

1 Introduction

Knowledge graphs (KGs) have been an extremely productive research direction in recent years, which enable binary relational facts [Wang *et al.*, 2017]. However, with binary and beyond-binary relational facts therein, the generalized multiary relational knowledge bases (KBs) are less studied. Especially, multiary relational KBs face the serious issue of incompleteness, and a fundamental problem is multiary relational knowledge base completion (KBC).

As such, several studies have been done on learning low-dimensional representations of entities and relations for multiary relational KBC. Especially, a scoring function is designed to measure the plausibility of multiary relational facts. For instance, both m-TransH [Wen *et al.*, 2016] and RAE [Zhang *et al.*, 2018] inherit translation idea in KGs, and calculate the distance-based score between entities on relation-specific hyperplanes. Besides, NaLP [Guan *et al.*, 2019] leverages neural network for scoring function design. However, few works investigate the potential of tensor decomposition approaches for multiary relational KBC, which are the most popular and powerful approaches in KGs [Wang *et al.*, 2017; Balažević *et al.*, 2019].

In this work, based on the state-of-the-art KG model TuckER [Balažević *et al.*, 2019] and a recent n-ary relational KBC model GETD [Liu *et al.*, 2020], we generalize tensor decomposition for multiary relational KBC, termed as m-GETD. Specifically, m-GETD models the interaction between entities and relations with Tucker decomposition [Tucker, 1966], and the core tensor therein is decomposed by Tensor Ring (TR) decomposition [Zhao *et al.*, 2016] for scalable model complexity. Especially, the entity and relation embeddings are shared across arities, and a group of TR tensors are introduced as the base space for mixed arity representation. Furthermore, extensive evaluations on two representative multiary relational KB datasets demonstrate the superior performance of m-GETD.

2 Methodology

2.1 Background

First, we introduce the definition of multiary relational KBs,

Definition 2.1 *Given the set of relations $\mathcal{R}$ and the set of entities $\mathcal{E}$, the multiary relational KB is defined as $\mathcal{B} = \{\mathcal{E}, \mathcal{R}, \mathcal{F}\}$, where $\mathcal{F} = \{(r, e_1, e_2, \cdots, e_n) | e_{i=1,2,\cdots,n} \in \mathcal{E}, r \in \mathcal{R}, n = 2, 3, \cdots\}$ is the set of multiary relational facts, and n denotes the arity of relation.*

Hence, the multiary relational KBC problem is defined as,

Problem 1 *Given an incomplete multiary relational KB $\mathcal{B} = \{\mathcal{E}, \mathcal{R}, \mathcal{F}\}$, the multiary relational KBC problem aims to infer missing facts based on $\mathcal{B}$.*

As introduced in Section 1, the recent work GETD [Liu *et al.*, 2020] focuses on n-ary relational KBC, where the outer layer is based on TuckER [Balažević *et al.*, 2019] and the inner layer is based on Tensor Ring decomposition [Zhao *et al.*, 2016]. For an n-ary relational fact $(r, e_1, e_2, \cdots, e_n)$, the scoring function is defined as,

$$\begin{aligned}&\phi(r, e_1, e_2, \cdots, e_n) = \\ &\boldsymbol{TR}(\mathcal{Z}_1, \cdots, \mathcal{Z}_{n+1}) \times_1 \boldsymbol{r} \times_2 \boldsymbol{e}_1 \times_3 \boldsymbol{e}_2 \cdots \times_{n+1} \boldsymbol{e}_n, \end{aligned} \quad (1)$$

where $\boldsymbol{TR}(\cdot)$ is TR decomposition, $\boldsymbol{TR}(\mathcal{Z}_1, \cdots, \mathcal{Z}_{n+1})$ is an $(n + 1)$-th order tensor, playing the role of core tensor in Tucker decomposition. $\boldsymbol{r}$ and $\boldsymbol{e}_i$ are embeddings for relation r and entity e_i. However, due to the predetermined arity n, this scoring function is only for single arity relational KBs.

2.2 m-GETD: Design and Model

To model general multiary relational KBs, in this section, we introduce the m-GETD model for arbitrary arity of relations.

According to (1), if we extend GETD to multiary relational KBs, the corresponding scoring function for m-ary relational facts can be defined as,

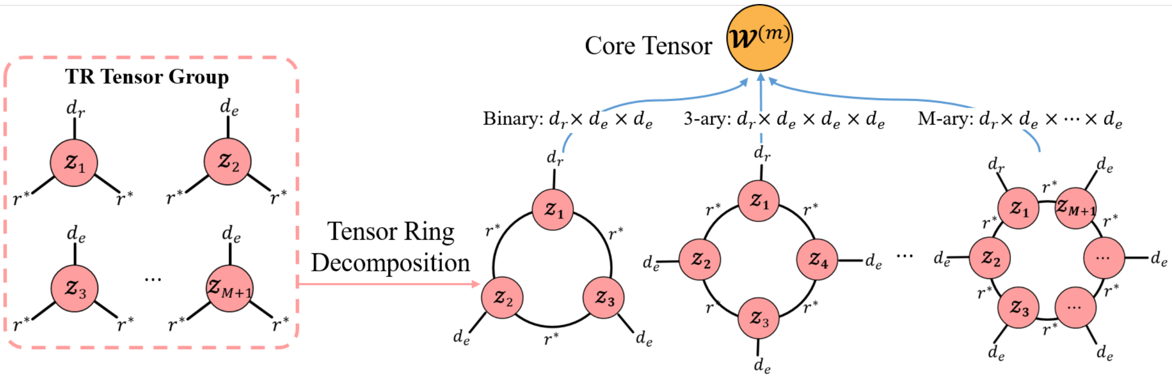


Figure 1: The framework of m-GETD. M is the maximum arity of the multiary relational KB. r^* is the rank of TR tensors.

Table 1: Dataset Statistics of Multiary Relational KBs. Here "#Arity-5+" denotes the number of facts whose relation is 5-ary and above.

Dataset	#Entities	#Relations	Arity	#Train	#Valid	#Test	#Arity-2	#Arity-3	#Arity-4	#Arity-5+
WikiPeople	47,765	702	2-9	305,725	38,223	38,281	337,914	25,820	15,188	3,307
JF17K	28,645	322	2-6	61,104	15,275	24,568	54,627	34,544	9,509	2,267

$$\phi^{(m)}(r, e_1, e_2, \cdots, e_m) =$$
$$\mathcal{W}^{(m)} \times_1 \boldsymbol{r} \times_2 \boldsymbol{e}_1 \times_3 \cdots \times_{m+1} \boldsymbol{e}_m, \ \forall m = 2, \cdots, M, \quad (2)$$

where $\mathcal{W}^{(m)} \in \mathbb{R}^{d_r \times d_e \cdots \times d_e}$ is $(m+1)$-th order core tensor, and M is the maximum arity of the multiary relational KB.

To make the best of mutual information across different arities, we introduce M 3rd-order TR tensors in m-GETD, termed as TR tensor group $\{\mathcal{Z}_i | \mathcal{Z}_1 \in \mathbb{R}^{r^* \times d_r \times r^*}, \mathcal{Z}_{i \neq 1} \in \mathbb{R}^{r^* \times d_e \times r^*}\}_{i=1}^{M+1}$, where d_e and d_r are the embedding dimensionality of entity and relation respectively, and r^* is the rank of TR tensors, TR-rank. Subsequently, with TR decomposition applied, m-GETD utilizes first three TR tensors in the group to recover the 3rd-order core tensor $\mathcal{W}^{(2)}$ for binary relational facts, and utilizes first four TR tensors in the group to recover the 4th-order core tensor $\mathcal{W}^{(3)}$ for 3-ary relational facts and etc. The whole group of TR tensors is utilized to recover the $(M+1)$-th order core tensor $\mathcal{W}^{(M)}$ for M-ary relational facts. The framework of above process is shown in Fig 1, which indicates the solution to multiary relational KBs. It can be observed that, TR tensor group as well as shared embeddings in the outer layer encode the mutual information across different arities.

Thus, the scoring function of m-GETD in multiary relational KBs can be expressed as,

$$\phi^{(m)}(r, e_1, e_2, \cdots, e_m) =$$
$$\boldsymbol{TR}(\mathcal{Z}_1, \cdots, \mathcal{Z}_{m+1}) \times_1 \boldsymbol{r} \times_2 \boldsymbol{e}_1 \times_3 \cdots \times_{m+1} \boldsymbol{e}_m,$$
$$\forall m = 2, 3 \cdots, M. \quad (3)$$

The loss function and corresponding algorithm follow the successful practice in [Liu *et al.*, 2020].

3 Experiments and Results

3.1 Experiment Settings

We evaluate our model with standard KBC task, i.e., link prediction task, on two real multiary relational KB datasets, WikiPeople [Guan *et al.*, 2019] and JF17K [Zhang *et al.*, 2018]. Table 1 presents the dataset statistics.

The standard metrics of mean reciprocal rank (MRR) and Hits@k, $k \in \{1, 3, 10\}$ are used for evaluation. Besides, we compare m-GETD with the baselines: RAE [Zhang *et al.*, 2018], NaLP [Guan *et al.*, 2019], n-CP [Hitchcock, 1927], and n-DistMult [Yang *et al.*, 2015].

The embedding sizes as well as TR latent tensor dimensions on WikiPeople are set to $d_e = d_r =$50 with r^* equal to 10. The settings on JF17K are $d_e = d_r = r^* = 25$.

3.2 Link Prediction in Multiary Relational KBs

The link prediction results on two multiary relational KB datasets are shown in Table 2. Based on the results, m-GETD outperforms all baselines on most metrics, which proves the efficiency. Specifically, compared with NaLP on WikiPeople, m-GETD improves MRR by 0.07 and Hits@10 by 9%. Besides, m-GETD improves MRR by 0.037 and Hits@10 by 7% for n-DistMult on JF17K. It is worth mentioning that the best baselines including NaLP and n-DistMult perform differently between two datasets. In contrast, the best performance of m-GETD on both datasets indicates its robustness.

According to the design of n-CP, n-DistMult and m-GETD, these tensor decomposition models similarly share the entity and relation embeddings across arities. However, the results in Table 2 show quite large gaps (over 7% with MRR) between the performance of m-GETD and other tensor decomposition models, which owes to the gains of mutual information encoded by the TR tensor group. Simply sharing the embeddings in n-CP and n-DistMult may not be able to overcome the noise caused by different arities of relational facts, while the TR tensor group plays a role of the base space. By selecting certain tensors in the group for different arities, m-GETD successfully captures the positive effect across arities with noise removed.

We present the breakdown performance on both datasets in Table 3. It can be observed that, compared with baselines on JF17K, m-GETD increase Hits@10 by 6% and 8% for 2-ary

Table 2: Link prediction results on WikiPeople and JF17K. Results of NaLP on WikiPeople are copied from original paper.

Model	WikiPeople				JF17K			
	MRR	Hits@10	Hits@3	Hits@1	MRR	Hits@10	Hits@3	Hits@1
RAE	0.253	0.463	0.343	0.117	0.396	0.561	0.433	0.312
NaLP	0.338	0.466	0.364	**0.272**	0.310	0.450	0.334	0.239
n-CP	0.313	0.451	0.372	0.223	0.400	0.542	0.431	0.324
n-DistMult	0.318	0.478	0.391	0.213	0.452	0.599	0.482	0.375
m-GETD	**0.345**	**0.510**	**0.415**	0.237	**0.489**	**0.643**	**0.521**	**0.409**

Table 3: Breakdown performance across relations of different arities on WikiPeople and JF17K.

Model	WikiPeople						JF17K					
	Arity-2		Arity-3		Arity-4		Arity-2		Arity-3		Arity-4	
	MRR	Hits@10	MRR	Hits@10	MRR	Hits@10	MRR	Hits@10	MRR	Hits@10	MRR	Hits@10
RAE	0.265	0.494	**0.225**	**0.365**	0.158	0.272	0.207	0.348	0.414	0.593	0.627	0.751
NaLP	0.351	0.465	-	-	-	-	0.095	0.201	0.313	0.468	0.612	0.754
n-CP	0.357	0.511	0.191	0.298	0.047	0.079	0.236	0.367	0.411	0.550	0.640	0.773
n-DistMult	0.359	0.536	0.220	0.350	0.062	0.090	0.240	0.399	0.506	0.647	0.669	0.794
m-GETD	**0.369**	**0.540**	0.180	0.300	**0.330**	**0.520**	**0.267**	**0.422**	**0.562**	**0.707**	**0.709**	**0.857**

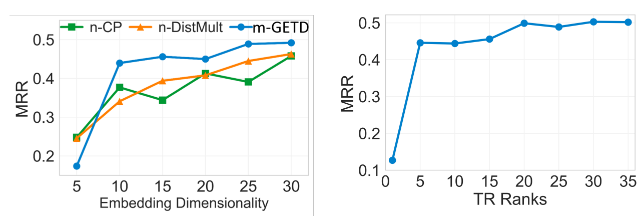
and 4-ary relational facts, respectively. Besides, the prediction performance of all models on JF17K increases with the arity increasing. Note that the number of entities and relations involved in higher-ary relational facts are much less, which reduces the learning difficulty and thus increases the performance. However, a reverse trend is inferred from WikiPeople due to its distinctive composition. We note that 3-ary and 4-ary relational facts in WikiPeople always involve with entities about time/space points, which are quite difficult for link prediction and lead to lower accuracy for higher arity. Even so, m-GETD increases MRR by 0.172 and Hits@10 by over 90% for 4-ary relational facts compared with baselines, which shows the capability of joint learning in multiary relational KBs as well as strong expressiveness for time/space points.

3.3 Influence of Parameters for m-GETD

The performance of tensor decomposition models under different embedding sizes are evaluated on JF17K with TR-rank equal to embedding sizes. The results are shown in Fig 2(a). The MRR of three models increases with the increase of embedding sizes, and converges at large embedding sizes. m-GETD outperforms baselines significantly when embedding sizes are over 10. For instance, m-GETD increases MRR by over 6% for baselines at embedding size 30. Especially, m-GETD with embedding size 10 reaches comparable performance to n-CP and n-DistMult with embedding size 30, which demonstrates the strong expressiveness.

We show the performance of m-GETD with different TR-ranks in Fig 2(b). The MRR increases slowly when TR-rank exceeds 5, which indicates that even small TR-rank for m-GETD are able to recover the core tensor with complex interactions between entities and relations. Also, we can reduce TR-rank of m-GETD for storage space saving as well as training acceleration.

According to the results in Fig 2, it can be concluded that embedding sizes play a much more important role in m-GETD than TR-rank. A possible explanation is that TR-rank only determines the inner layer with TR tensor groups, while embedding sizes determine the representation space of entities, relations and the recovered core tensor, which are closely related to the interaction and expressiveness.



(a) Embedding sizes d_e , d_r. (b) TR-ranks $\boldsymbol{r}^*$

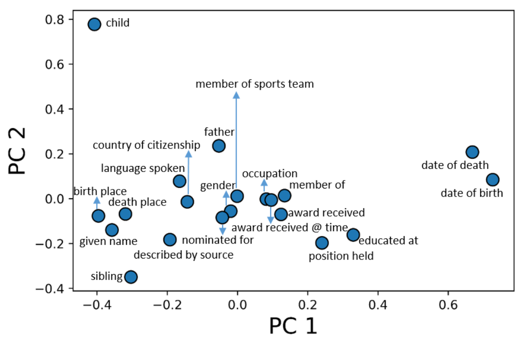
Figure 2: Influence of embeddings sizes and TR-ranks for m-GETD on JF17K

3.4 Embedding Visualization

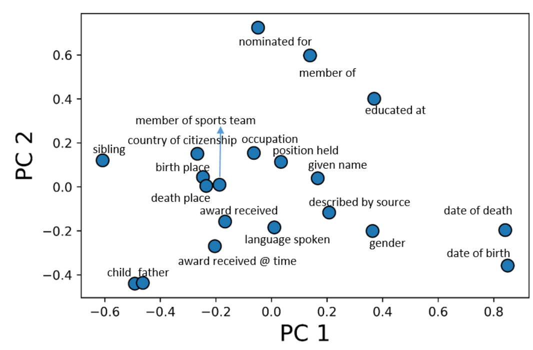
The top 20 relation/entity embeddings of m-GETD and n-DistMult on WikiPeople are visualized in Fig 3 through principal component analysis (PCA).

From the above two figures we can observe that most similar relations such as birth/death place, award received (@ time), and date of birth/death are correctly clustered by both models. However, the anti-symmetric relation pair of father and child is visualized differently. Specifically, the child (-0.4, 0.8) and the father (0.0, 0.2) relations are opposed spatially in m-GETD, while assigned closely (-0.5, -0.4) in n-DistMult. DistMult based models cannot represent anti-symmetric relations due to the same entity embeddings at different positions. Hence, n-DistMult learns the close embeddings for child/father relations due to their sharing semantics, which leads to wrong representation in practical use. In contrast, m-GETD correctly learns the opposite relations, which validates the strong expressiveness.

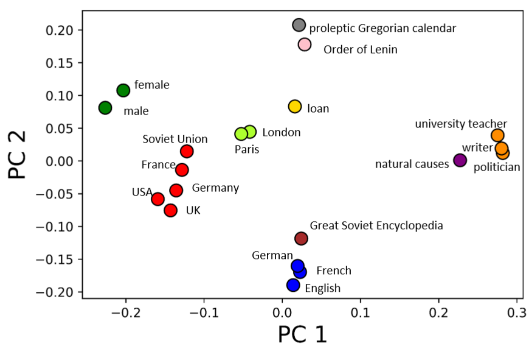
As for entity embedding visualization, there are three



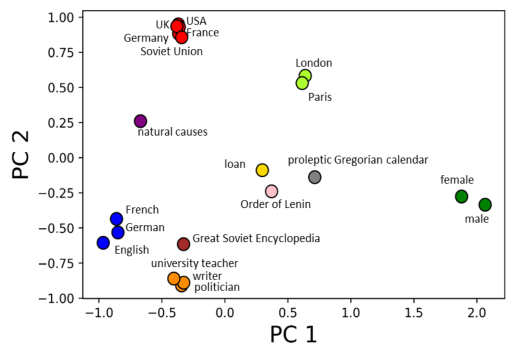
(a) Relation embeddings of m-GETD.



(b) Relation embeddings of n-DistMult.



(c) Entity embeddings of m-GETD.



(d) Entity embeddings of n-DistMult.

Figure 3: Entity embedding visualization of m-GETD and DistMult through PCA on WikiPeople dataset.

groups of entities with close connections, the country group (red), the capital group (green) and the language group (blue). Moreover, these groups of embeddings are all shown in the third quadrant of m-GETD embedding visualization, while shown in a distance in n-DistMult embedding visualization. Considering the co-occurrence of these entities in KBs, the learnt embeddings of m-GETD successfully capture this.

4 Conclusion and Future Work

We generalized tensor decomposition for multiary relational knowledge base completion. Considering the uneven distribution of different arities of relational facts, we argue that better modeling the mutual effect across arities is crucial in multiary relational KBs. Besides, incorporating background knowledge such as logical rules and entity properties may also bring performance enhancement.

References

[Balažević *et al.*, 2019] Ivana Balažević, Carl Allen, and Timothy M Hospedales. Tucker: Tensor factorization for knowledge graph completion. In *EMNLP*, 2019.

[Guan *et al.*, 2019] Saiping Guan, Xiaolong Jin, Yuanzhuo Wang, and Xueqi Cheng. Link prediction on n-ary relational data. In *WWW*, pages 583–593, 2019.

[Hitchcock, 1927] Frank L Hitchcock. The expression of a tensor or a polyadic as a sum of products. *Journal of Mathematics and Physics*, 6(1-4):164–189, 1927.

[Liu *et al.*, 2020] Yu Liu, Quanming Yao, and Yong Li. Generalizing tensor decomposition for n-ary relational knowledge bases, 2020.

[Tucker, 1966] Ledyard R Tucker. Some mathematical notes on three-mode factor analysis. *Psychometrika*, 31(3):279–311, 1966.

[Wang *et al.*, 2017] Quan Wang, Zhendong Mao, Bin Wang, and Li Guo. Knowledge graph embedding: A survey of approaches and applications. *TKDE*, 29(12):2724–2743, 2017.

[Wen *et al.*, 2016] Jianfeng Wen, Jianxin Li, Yongyi Mao, Shini Chen, and Richong Zhang. On the representation and embedding of knowledge bases beyond binary relations. In *IJCAI*, pages 1300–1307, 2016.

[Yang *et al.*, 2015] Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. Embedding entities and relations for learning and inference in knowledge bases. *ICLR*, 2015.

[Zhang *et al.*, 2018] Richong Zhang, Junpeng Li, Jiajie Mei, and Yongyi Mao. Scalable instance reconstruction in knowledge bases via relatedness affiliated embedding. In *WWW*, pages 1185–1194, 2018.

[Zhao *et al.*, 2016] Qibin Zhao, Guoxu Zhou, Shengli Xie, Liqing Zhang, and Andrzej Cichocki. Tensor ring decomposition. *arXiv preprint arXiv:1606.05535*, 2016.