EasyChair Preprint

№ 8246

CP Tensor Factorization for Knowledge Graph Completion

Yue Luo, Chunming Yang, Li Bo, Zhao Xujian and Hui Zhang

June 10, 2022

CP Tensor Factorization for Knowledge Graph Completion

Yue Luo[1], Chunming Yang[1,3], Bo Li[1], Xujian Zhao[1], Hui Zhang[2]

[1] School of Computer Science and Technology, Southwest University of Science and Technology, Mianyang 621010 Sichuan, China
[2] School of Science, Southwest University of Science and Technology, Mianyang 621010 Sichuan, China
[3] Sichuan Big Data and Intelligent System Engineering Technology Research Center, Mianyang 621010 Sichuan, China
yangchunming@swust.edu.com

Abstract. The problem of incomplete knowledge caused by the lack of relations in large-scale knowledge graphs increases the difficulty of downstream application tasks. Predicting the missing relations between entities according to the existing facts is the main means of knowledge graph completion. The triple of knowledge graph can be seen as a third-order binary tensor element that linearly transforms entities and relations into low-dimensional vectors through tensor decomposition to determine the probability that the triple of missing relations is true. However, the non-deterministic polynomiality in determining the tensor rank can lead to overfitting and unfavorable to the generation of low-rank models. Aiming at this problem, we propose to use CP decomposition to decompose the third-order tensor into the sum of multiple rank-one tensors, which is the sum of the outer products of the head entity embedding, relation embedding, and tail entity embedding for each triple, and convert it into a super-diagonal tensor product the factor matrix of each mode, and use scoring function calculate the probability that the triple of missing relation is true. Link prediction experimental results from four different domains of benchmarks knowledge graph datasets show that the proposed methods are better than other comparison methods, it also can express the complex relations of knowledge graph, and the decomposition has uniqueness, reduces the total amount of calculations and parameters, avoids overfitting.

Keywords: Knowledge graph completion, Tensor decomposition, CP decomposition.

1 Introduction

Knowledge Graphs (KGs) are large-scale semantic networks that store human knowledge in the form of graphs, where nodes represent entities and edges represent specific factual relations that connect two entities, usually represented as triples,

namely (head entity, relation, tail entity). KGs, which allow computers to model complex data in the form of structured storage of knowledge and have been widely used in automated question-answering, information retrieval, and recommendation systems [1,2,3]. At present, a large number of large-scale KGs have been constructed by manual and semi-automatic methods, such as Freebase [4], YAGO [5], NELL [6], etc., but there are problems such as the lack of entity attributes and relations between entities. For example, about 70% of people lack birthplace information, WordNet [7] and NELL also have different degrees of lack of relations such as race and part of speech [8]. The lack of data in KGs leads to data sparseness and knowledge incomplete, which increases the computational difficulty of downstream tasks [1,2,3]. Knowledge representation learning [9,10,12] maps entities and relations in KGs into a low-dimensional continuous space through representation learning, and uses a scoring model with low latitude embedding as input to score the triple, so as to determine the probability that the triple is true, and achieve the purpose of completing KGs quickly, which can effectively solve the data sparseness problem and improve the calculation efficiency of downstream tasks.

Knowledge representation learning can be divided into non-linear [10,12] and linear models [9]. Nonlinear models mainly include translation models and neural network models. The translation model [10] assumes that the relation is the translation from head entity to tail entity, and projects the head and tail entities into a vector space of the same dimension by the relations vector/matrix, and calculate the distance between the projection vectors to determine the confidence level that exists relations between entities. The translation model uses the distance between two entities to measure the rationality of a fact, and cannot accurately describe the semantic connection between two entities. Therefore, the synergy between entities is poor, cannot effectively solve the complex relation problem of triple, such as Trans E [11]. The neural network model [12] adopts the nonlinear model of a single-layer neural network to further describe the semantic correlation of entities under relations, but it is opaque, understandable and interpretable, and has high computational complexity. Linear models regard knowledge graph (KG) as a third-order binary tensor, where each element corresponds to a triple, *1* indicating a true fact and *0* indicating the unknown (either a false or a missing fact). The goal of KG completion is to linearly transform entities and relations into low-dimensional vectors through tensor decomposition [13], and then embed entities and relations to calculate the probability that triplets are true by product.

Tensor decomposition is the recovery of low rank components by approximating low-rank structure of data tensors, making full use of the information from all dimensions of data to effectively recover or predict the lost data. However, the non-deterministic polynomials that determining the rank of the tensor [14], such as TuckER [15], can lead to overfitting that unfavorable to the generation of low-rank models, which affects the accuracy of KG completion. To solve the above problems, we propose a method of using CP decomposition[16] for KG completion, which uses CP decomposition to decompose the third-order tensor into a sum of component rank-one tensors, which is the sum of the outer products of the head entity embedding, relation embedding, and tail entity embedding for each triple, and we convert it into a super-diagonal tensor product the factor matrix of each mode, and use the scoring function to calculate the score of

each triple to infer the probability that each triple is true. CP decomposition ensures the uniqueness of tensor rank, reduces the amount of calculation and the total number of parameters. We conduct a completion experiment on four different domains of standard datasets, the results show that this method has better prediction accuracy.

The rest of the paper is organized as follows: Section "Related Work" introduces the basic concepts of knowledge representation learning and existing embedding methods and their advantages and disadvantages. Section "Knowledge graph completion based on CP decomposition" describes the proposed CP model in detail. Section "Experiments and Results" compares CP with the most typical and state-of-the-art embedding models followed by a conclusion in the "Conclusions".

2 Related Work

Knowledge representation learning maps entities and relations in KGs into dense low-latitude real-valued vectors respectively while keeping the semantic relations unchanged, and then calculates complex semantic associations between entities and relations in low-dimensional space to improve the computational efficiency of downstream tasks. The mapping method mainly includes non-linear and linear models. Nonlinear models consist of translation models and neural network models.

Translation-based models regarding the relation as a mapping from head entity to tail entity, and measure the plausibility of a fact as the distance between two entities. The representative model is Trans E [11], Trans E models a real triple(h, r, t) as $\boldsymbol{h} + \boldsymbol{r} \approx \boldsymbol{t}$, where $\boldsymbol{h}, \boldsymbol{r}, \boldsymbol{t} \in \mathbb{R}^{\boldsymbol{k}}$ represents the head entity, relations, and tail entity respectively. $\boldsymbol{k}$ denotes vector dimensions, it can perform well on link prediction, but cannot handling 1-N, N-1, N-N relations. Trans R [17] embeds entities and relations into separate vector spaces, which is better in handling N-N relations, but has high time and space complexity. In addition, there are also translation models such as Trans M [18], Manifold E [19], and Tran Spare [20] that have made relevant improvements for complex relational problems, all of which have achieved better results. However, the translation model uses two different matrices to project the head and tail entities, so that the semantic connections between the entities cannot be accurately characterized, such as Trans H [21], so the synergy between entities is not good and it is still difficult to deal with the complex relations of KGs.

Neural network model uses neural networks to replace the bilinear transformation of linear models to calculate the scoring function of triple, NTN [22] portrays the semantic connections between entities and relations through nonlinear operations of single-layer neural networks, whose scoring functions is defined as a neural network-like output, it can portray complex semantic relations between entities more accurately, but too many parameters also bring an increase in complexity. Dong [23] et al proposed an ER-MLP model that can be considered a simplified version of NTN to reduce the number of parameters. Models such as R-GCN [24], Conv E [25] and Conv KB [26] use convolutional neural networks to learn vector representations of entities and relations in KGs, R-GCN uses relational graph convolutional neural networks to model relational

paths in KGs. Conv E combines entities and relations embedding into a two-dimensional matrix, and then performs a 2D convolution operation on this matrix, Conv KB uses only one-dimensional convolution. Although the prediction accuracy of the neural network model is considerable, the model is not transparent and understandable and high computational complexity.

Linear model treats each triple of KG as an element in a tensor (matrix) and uses tensor decomposition to decompose high-dimensional arrays into multiple low-dimensional ones for representation learning. Typical models include RESCAL [27], DisMult [28], ComplEx [29], SimplE [30], and TuckER [15], all of which apply different decomposition methods for this third-order binary tensor to solve the KG completion problem. RESCAL uses vectors to represent the potential semantics of entities, matrices to represent the relations between entities, and uses relation matrices to model potential factors with interactions, which makes the algorithm complex, parameter-rich and more prone to the risk of overfitting since each relation corresponds to a relation matrix. DisMult compensates for this risk by restricting the relation matrix to a diagonal matrix, which reduces complexity of the model but failing to establish triples of asymmetric relations. ComplEx represents entities and relations as complex vectors to capture triples of antisymmetric relations. SimplE learns each relation as two independent embedding, one for normal relations and the other for reverse ones. TuckER decomposes the tensor into a core tensor multiplied by the product of three factor matrices in three modes, each row of which represents subject entity, relation, and object entity respectively, and the core tensor characterizes the level of interaction between them and is a fully expressive model. However, TuckER decomposition is an approximation of *n*-rank and low rank, for fixed n-rank, the uniqueness of TuckER decomposition cannot be guaranteed, the core tensor is not constrained, and the correlation between entities and relations are not better handled, so it is prone to the risk of overfitting.

When we regard KGs as third-order binary tensors, the third-order KG tensor is decomposed into the sum of the outer product of embedding of each triple head entity, relation, and tail entity by CP decomposition, at which point we are able to build each triple that can express any relations of KGs (including complex relations such as many-to-many, multi-level, asymmetric, etc.). Meanwhile CP decomposition is an approximation of rank and low rank, decomposition has uniqueness, which can decompose high-dimensional tensors into the sum of component rank-one tensors, with each nucleus consisting of the outer product of vectors, reducing the rank of the weight, the amount of computation and the total amount of parameters, avoiding overfitting.

3 Knowledge graph completion based on CP decomposition

3.1 Problem Definition

For a given knowledge graph $KG = (\mathcal{E}, \mathcal{R})$, $\mathcal{E}$ denotes the set of all entities, $\mathcal{R}$ the set of all relations, and Δ denotes the set of ground truth triples in KG. The triple(h, r, t) represents a fact, $h, t \in \mathcal{E}$ denotes head entity and tail entity, $r \in \mathcal{R}$ the relations between head and tail entities. KG can be represented as a third-order binary tensor $\boldsymbol{\chi} \in$

$\mathbb{R}^{I\times J\times K}$, $\boldsymbol{h_i} \in \mathbb{R}^{I}$,$\boldsymbol{r_i} \in \mathbb{R}^{J}$,$\boldsymbol{t_i} \in \mathbb{R}^{K}$ represents head entity, relation, and tail entity of a triple, KG of the third-order binary tensor can be decomposed into:

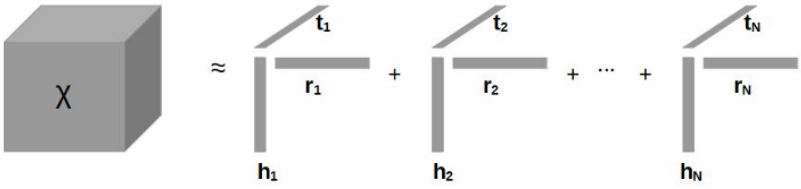

Fig. 1. CP decomposition of KG

Expanding the above CP decomposition:

$$\boldsymbol{\chi = h_1 \circ r_1 \circ t_1 + h_2 \circ r_2 \circ t_2 + \cdots + h_N \circ r_N \circ t_N \approx \sum_{i=1}^{N} h_i \circ r_i \circ t_i} \quad (1)$$

Where $\circ$ denotes the outer product of vector, *N* is a positive integer, and CP decomposition can also be expressed in terms of mod-*i* multiplication:

$$\boldsymbol{\chi = \mathcal{J} \times_1 H \times_2 R \times_3 T} \quad (2)$$

Where $\boldsymbol{\mathcal{J}} \in \mathbb{R}^{N\times N\times N}$ is the unit tensor, indicating the degree of reciprocity between different components, with all its super-diagonal elements being *1* and others being *0,* $\boldsymbol{H} = \{\boldsymbol{h_1}, \boldsymbol{h_2}, \ldots, \boldsymbol{h_N}\} \in \mathbb{R}^{I\times N}$, $\boldsymbol{R} = \{\boldsymbol{r_1}, \boldsymbol{r_2}, \ldots, \boldsymbol{r_N}\} \in \mathbb{R}^{J\times N}$, $\boldsymbol{T} = \{\boldsymbol{t_1}, \boldsymbol{t_2}, \ldots, \boldsymbol{t}_{\mathrm{R}}\} \in \mathbb{R}^{K\times N}$ are embedding matrix of head entity, relation and tail entity, and $\times_{\boldsymbol{n}}$ denotes the tensor product along pattern *n*. When left equals right, the rank of $\boldsymbol{\chi}$ is *N*. When $\boldsymbol{H}$,$\boldsymbol{R}$,$\boldsymbol{T}$ is orthogonal, it can be considered as the main constituent factor in each mode.

3.2 Model Definition

KG completion based on CP decomposition decomposes KG of a third-order binary tensor into a super-diagonal tensor product the factor matrix of each mode. Entity embedding matrix $\boldsymbol{E}$ that is equivalent for head and tail entities, i.e., $\boldsymbol{E = H = T} \in \mathbb{R}^{\boldsymbol{n_e}\times \boldsymbol{d_e}}$, and relation embedding matrix $\mathrm{R} \in \mathbb{R}^{\boldsymbol{n_r}\times \boldsymbol{d_r}}$, where $\boldsymbol{n_e}$ and $\boldsymbol{n_r}$ present the number of entities and d_e and d_r the dimensionality of entity and relation embedding vectors respectively.

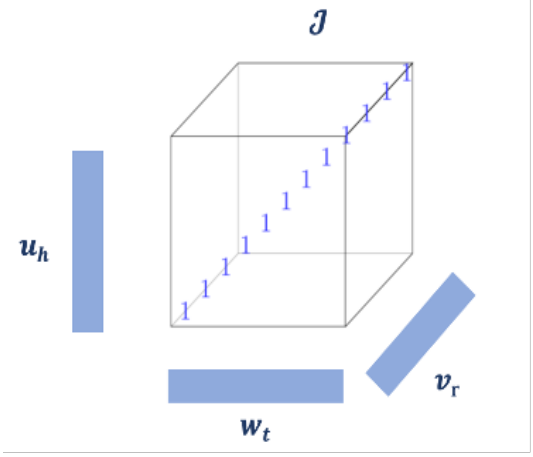

Fig. 2. Visualization of CP architecture

The scoring function for CP as:

$$f(h,r,t) = \mathcal{J} \times_1 \boldsymbol{u_h} \times_2 \boldsymbol{v_r^T} \times_3 \boldsymbol{w_t} \tag{3}$$

Where $\boldsymbol{d_e} = \boldsymbol{d_r}, \mathcal{J} \in \mathbb{R}^{\boldsymbol{d_e} \times \boldsymbol{d_r} \times \boldsymbol{d_e}}$ is the super-diagonal tensor of CP decomposition and $\times_n$ is the tensor product along the *n*-th mode. $\boldsymbol{u_h}, \boldsymbol{w_t} \in \mathbb{R}^{\boldsymbol{d_e}}$ are the rows of $\boldsymbol{E}$ representing the head and tail entity embedding vectors, $\boldsymbol{v_r}$ the rows of *R* representing the relation embedding vector. We first multiply $\mathcal{J}$-recombinant a matrix with $\boldsymbol{u_h}$, and the result forms a third-order tensor with the transpose matrix of relation vector for product, and finally multiply with w_t to get a score of the triple. We use the Sigmoid activation function for each triple score $\boldsymbol{f(h,r,t)}$ to get the prediction probability $\boldsymbol{p} = \boldsymbol{\sigma(f(h,r,t))}$ that each triple is true, and the space complexity of cp is $\boldsymbol{O(n_e d_e + n_r d_r)}$.

3.3 Model Learning

For learning CP, we assign the diagonal of decomposed third-order tensor to *1* and others to *0* (either a false or a missing fact). For each head entity $\boldsymbol{u_h}^{(i)}$, relation $\boldsymbol{v_r}^{(j)}$, tail entity $\boldsymbol{w_t}^{(k)}$ correspond to the *i*-th, *j*-th, and *k*-th elements of the three factor matrices respectively, and train this three matrices so that the positions (i, j, k) corresponding vectors $\boldsymbol{u_h}$, $\boldsymbol{v}_r$, $\boldsymbol{w_t}$ of true triple fall as far as possible on the super-diagonal of decomposed third-order tensor, as shown in Figure. 3:

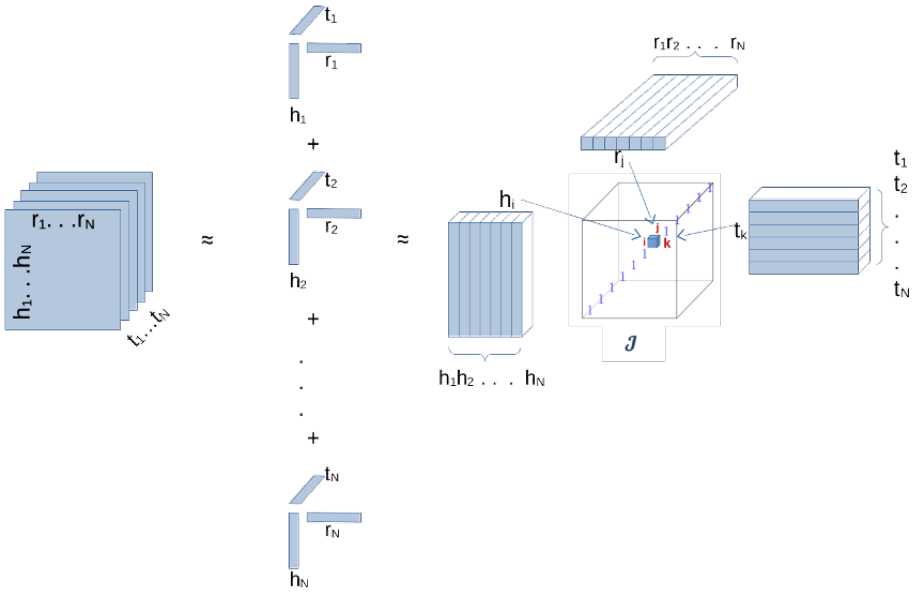

Fig. 3. Flow Chart of CP Learning

We take only the non-existent triples $(\boldsymbol{u_h}, \boldsymbol{v_r}, \cdot)$ and $(\cdot, \boldsymbol{v_r}, \boldsymbol{w_t})$ of the observed pairs $\boldsymbol{u_h}$, $\boldsymbol{v_r}$and $\boldsymbol{v_r}$, $\boldsymbol{w_t}$ respectively as negative samples and all observed triples as positive samples. To improve the training speed and accuracy of the algorithm, we refer to the training method of Dettmers et al [25] and use numerical method to train CP. Using the *1-N* scoring, that is, we simultaneously score a pair $\boldsymbol{u_h}$, $\boldsymbol{v_r}$ with all entities $\boldsymbol{w_t} \in \mathcal{E}$, in contrast to *1-1* scoring, where individual triples $(\boldsymbol{u_h}, \boldsymbol{v_r}, \boldsymbol{w_t})$ are trained one at a time. This way improves the training speed of algorithm significantly. We train our model to minimize the Bernoulli negative log-likelihood loss function:

$$\mathcal{L} = \sum_{(\mathrm{h,r,t}) \in \mathcal{B}(\mathrm{h,r,t})} l(h,r,t) log p(h,r,t) + (1 - l(h,r,t)) \log\,(1 - p(h,r,t)) \qquad (4)$$

Where $\mathcal{B}(\mathrm{h,r,t}) = \Delta \cup \Delta'$, Δ denotes the set of fact triples, $\Delta' = \{(\boldsymbol{u_h}, \boldsymbol{v_r}, \boldsymbol{w_t}')|\boldsymbol{w_t}' \in \boldsymbol{E}\} \cup \{(\boldsymbol{u_h}', \boldsymbol{v_r}, \boldsymbol{w_t})|\boldsymbol{u_h}' \in \boldsymbol{E}\}$ gets the set of negatively sample triples from positive triples, i.e., we replace head and tail entity of a fact triple randomly, then they may contain other fact triple, which not participate in computation. $\mathcal{B}(\mathrm{h,r,t})$ is the set consisting of fact triples and negatively sampled triples. The value of $l(h,r,t)$ depends on whether the triple is true or not.

$$\mathcal{L} = \; l(h,r,t) = \begin{cases} -1\ , & if\ (\boldsymbol{u_h}, \boldsymbol{v_r}, \boldsymbol{w_t}) \in \Delta' \\ 1\ , & if\ (\boldsymbol{u_h}, \boldsymbol{v_r}, \boldsymbol{w_t}) \in \Delta \end{cases} \qquad (5)$$

3.4 Model Analysis

From Equation (1) we can see that CP decomposition can completely and directly express every triple and relations between entities (many-to-many, multi-level, asymmetric, and other complex relations), so CP can model complex relations of KG.

Unconstrained decomposition in tensor decomposition leads to uniqueness of the results, which leads to unstable model training and reduces the robustness of the model and affects the prediction effect. the method of first determining the number of rank-one tensors before iterating ensures its uniqueness in CP decomposition. The proof is as follows:

$$\chi = \sum_{r=1}^{N} a_r \circ b_r \circ c_r = \{A, B, C\} \tag{6}$$

Where $a_r \in \mathbb{R}^{\boldsymbol{I}}, b_r \in \mathbb{R}^{\boldsymbol{J}}, c_r \in \mathbb{R}^{\boldsymbol{K}}$, and uniqueness means that the above decomposition may be a combination of single-rank matrices (A, B, C), and here the columns of the decomposed single-rank matrix are rearranged using the permutation matrix:

$$\chi = \{A, B, C\} = \{A\Pi, B\Pi, C\Pi\} \tag{7}$$

Where Π is the permutation matrix of $N \times N$, and scaling vectors in CP decomposition does not affect the outcome, for example:

$$\chi = \sum_{r=1}^{N} (\alpha_r a_r) \circ (\beta_r b_r) \circ (\gamma_r c_r) \tag{8}$$

Where $\alpha_r\ ,\ \beta_r\ , \gamma_r = 1, r = 1, \ldots, R$, so cp decomposition is unique, and the stability of training is ensured by orthogonal constraints on the three factor matrices.

CP decomposition is the sum of all rank-one tensors decomposed by high-dimensional tensors, reduces the rank of weights, which in turn reduce the total amount of calculations and parameters. We convert CP decomposition to obtain a super-diagonal tensor and three factor matrices, where most of elements in super-diagonal tensor are 0, eliminating the interaction between components in every dimension and reducing the overfitting problem of model training to a certain extent.

4 Experiments and Results

4.1 Datasets

To validate our model, we use four benchmark datasets from different domains for link prediction. Since the correlation prediction tasks in WN18 [25] and FB15K [11] are not affected by the inverse relation problems [31], we use FB15k-237 and WN18RR that have filtered out the inverted relations to better train our model. all datasets counted as in Table 1.

WN18RR [25] is an English vocabulary database that filtered out all inverse relations from WN18.

FB15k-237 [32] is a structured KG contributed by community members that filtered out all inverse relations from FB15K.

Table 1. Number of entities, relations, and the Dataset partition

dataset	Rel	Ent	Train	Test	Valid
FB15K237	237	14,541	272,115	20,466	17,535
WN18RR	11	40,934	86,835	3,134	3,034
Nell-995	200	75,492	149678	543	3992
Kinship	25	104	8544	1074	1068

Nell-995 [33] Contains information on coaches, universities, government agencies, etc.

Kinship [34] is a dataset about kinship。

4.2 Implementation and Evaluation

KG completion is to predict missing triples in KG, i.e., find the missing entities or relations in triples. we get the set of all true triples in KG, with the goal of training the scoring function $f(h,r,t)$ corresponding to each triple to guide whether the triples are true or not, and finally be able to accurately score all the missing triples. A positive score for a triple indicates that it is a fact triple and negative overwise. Scoring function is a specific form of tensor factorization.

We evaluate each triple from the test set. For a given triple. We create candidate triples by replacing the head or tail entities with all entities in the dataset, we then rank the scores obtained. We use the filtered setting, i.e., we remove all other true triples apart from the currently observed test triple.

For evaluation, we use two evaluation metrics used across the link prediction literature: mean reciprocal rank (MRR) and $hits@N$， $N \in \{1，3，10\}$, MRR is the average of the inverse of a mean rank assigned to the true triple over all $\boldsymbol{n_e}$ generated triples. $hits@N$ indicates the proportion of correct answers in the top *N* of all candidate sets. The aim is for our model to achieve high MRR and $hits@N$.

4.3 Experiment Setting

We select hyperparameters by random search algorithm, and choose learning rate $\gamma \in \{0.0005,0.001,0.003,0.005,0.01\}$, learning rate decay $\gamma' \in \{1,0.99,0.95,0.995\}$. We select the best parameters on each dataset by tuning parameters, as follows:

Table 2. Parameters Setting

	Train Times	Learning rate	Number of batches	dimensions
FB15K237	500	0.005	128	200
WN18RR	500	0.01	128	100
Nell-995	500	0.005	128	100
Kinship	500	0.0005	128	100

4.4 Link Prediction Result

We compared several typical linear and nonlinear KG completion models in experiments, link prediction results on all four datasets are shown in Table 3 and 4. In section *3. 4*, we analyze that CP can better solve the complex relations problem of KG, to test this, we experiment with extracting *n-n* relations as test-sets on KinShip. Table 5 shows the link predictions results for each model on test set with only complex relations on KinShip.

Table 3. Link prediction results on FB15K237 and WN18RR

	FB15K237				WN18RR			
	MRR	Hit@10	Hit@3	Hit@1	MRR	Hit@10	Hit@3	Hit@1
TransE	.279	.441	.376	.198	.243	.532	.441	.427
RotatE [35]	.338	.533	.375	.241	.476	.571	**.492**	.428
PairRE [36]	.351	.544	.387	.256	.452	.546	.467	.410
R-GCN	.248	.417	.264	.151	-	-	-	-
Conv E	.325	.501	.356	.237	.430	.520	.440	.400
InteractE [37]	.354	.535	-	.263	.463	.528	-	.430
DisMult	.281	.419	.263	.155	.430	.490	.440	.390
ComplEx	.278	.428	.275	.158	.440	.510	.460	.410
TuckER	.358	.544	.394	.266	.470	.526	.482	.443
CP	**.371**	**.552**	**.399**	**.272**	**.482**	**.547**	.484	**.455**

Table 4. Link prediction results on NELL-995和 and Kinship

	NELL-995				Kinship			
	MRR	Hit@10	Hit@3	Hit@1	MRR	Hit@10	Hit@3	Hit@1
TransE	.401	.501	.472	.344	.271	.623	.345	.090
RotatE	.460	.553	.493	.403	.811	.971	.891	.717
R-GCN	-	-	-	-	-	-	.880	.300
Conv E	**.491**	.613	.531	.403	.833	.981	.917	.738
InteractE	**-**	-	-	-	.777	.959	.870	.664
DisMult	.485	.610	.524	.401	.516	.867	.581	.367
ComplEx	.482	.606	.528	.399	.677	.963	.795	.526
TuckER	.411	.514	.459	.362	.843	.985	.915	.760
CP	.481	**.614**	**.541**	**.424**	**.880**	**.986**	**.941**	**.812**

From Table 3 and 4, we can see that CP outperforms all previous state-of-the-art models (except Conv E has a higher MRR on NELL-995 and RotatE has a higher Hit@3 on WN18RR), which indicates that CP has a very good performance, not only over other linear models, such as DisMult, ComplEx, and TuckER, but also over complex deep

neural networks and reinforcement learning architectures, such as R-GCN, Conv E, and InteractE, reflecting the strong performance of CP.

From the structural analysis of these four datasets, we find that when the number of entities increases sequentially (Kinship, FB15K237, WN18RR, NELL-995), the linear model has progressively superior expressiveness compared with translation models and neural network models. This is because the larger the number of entities, linear models with the parameterization of the relation have more expressive, the better it can handle the richness of the entity. Comparing Kinship, FB15K237 and WN18RR, the model is trained better when the factorization of the higher dimensional tensor is used, such as TuckER, CP. TuckER performed better on FB15K237, WN18RR and Kinship just behind CP, but TuckER did not perform well on NELL-995 and CP still performs the best, which shows that CP is not only more expressive than TuckER but also more stable.

Table 5 shows that for the test set containing only complex relations, CP remains the best and TuckER the second, which indicates that CP is able to handle triples with complex relations in KG excellently, while RESCAL cannot.

Table 5. Link prediction results with *n-n* relations on KinShip

	Kinship(only *n-n* relations)			
	MRR	Hit@10	Hit@3	Hit@1
TransR	.266	.769	.436	0
RotatE	.816	.972	.888	.722
Rescal	.011	.001	0	0
DisMult	.539	.873	.598	.391
ComplEx	.677	.963	.795	.526
TuckER	.843	.985	.912	.760
CP	**.874**	**.986**	**.945**	**.803**

To verify that CP can reduce the total number of parameters, that is, a lower embedding dimensionality (i.e., lower rank of the decomposition) compared to other models can also have better results. We trained ComplEx, TuckER, and CP for embedding sizes $\boldsymbol{d_e = d_r} \in \{\mathbf{20}, \mathbf{50}, \mathbf{100}, \mathbf{200}\}$ on KinShip, Table 4 shows the obtained MRR on the test set for each of the models.

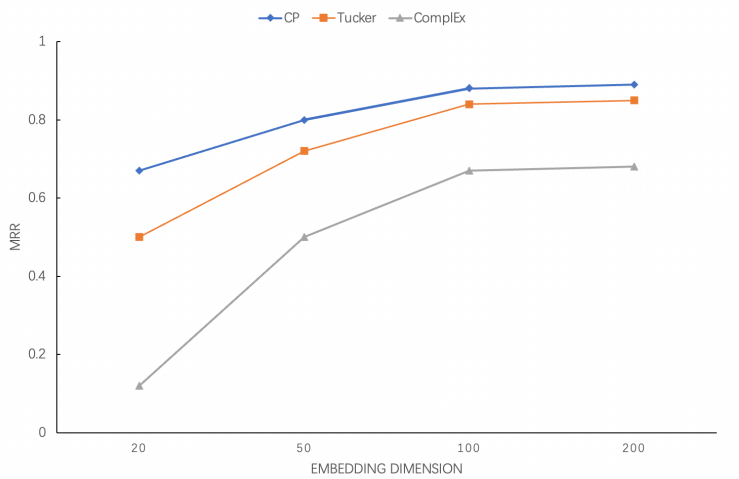

Fig. 4. MRR for ComplEx, TuckER and CP for embedding sizes $\boldsymbol{d_e = d_r} \in \{\mathbf{20}, \mathbf{50}, \mathbf{100}, \mathbf{200}\}$ on KinShip

We can see from Figure. 4 that CP outperforms Tucker and ComplEx in any embedding dimension, and the difference between the MRRs of ComplEx, TuckER and CP is approximately constant and performs best at embedding sizes 100 and 200. However, for lower embedding sizes, the difference between MRRs is larger. At the embedding size is 20, the performance of CP is the least different from that at the optimal embedding dimension compared to ComplEx and TuckER, which supports the hypothesis that CP can reduce the total number of parameters.

5 Conclusion

In this work, we have introduced CP, a simple but excellent linear model for link prediction in KGs based on CP decomposition, which decomposes KG represented by third-order tensor into a sum of multiple rank-one tensors, with the rank-one tensor consisting of the outer product of embedding of head entity, relation and tail entity of each triple for KG completion. CP is a higher-order expression of factorization models such as RESCAL, DisMult, and SimplE. We analyze that CP can model complex relations of KG, which solve the problem that most of the current KG completion models are difficult to deal with. And we prove that CP decomposition is unique, stable, reduces the total number of computations and parameters, and reduce overfitting. The experiments worked best on four different domains standard datasets and fill KG effectively. It is possible that it is of great practical significance to make link predictions of the knowledge graph of phenotypes (diseases, symptoms), drugs, genes and their relations to explore potential drug treatment mechanisms in the field of medicine.

Future work might include taking into account the semantic hierarchy and relational attributes between entities in triples, and combining the characteristics of the knowledge graph constructed by phenotype (disease, symptom), drug, and gene, to

achieve a more intelligent and high-precision knowledge graph completion, explore targeted drugs and disease-treating genes.

References

1. Jiao Jie,Wang Shujun,Zhang Xiaowang,Wang Longbiao,Feng Zhiyong,Wang Junhu. gMatch: Knowledge base question answering via semantic matching[J]. Knowledge-Based Systems,2021, 228. Author, F., Author, S.: Title of a proceedings paper. In: Editor, F., Editor, S. (eds.) CONFERENCE 2016, LNCS, vol. 9999, pp. 1–13. Springer, Heidelberg (2016).
2. Xiong C, Power R, Callan J. Explicit Semantic Ranking for Academic Search via Knowledge. Graph Embedding[C]//BARRETT R, CUMMINGS R, AGICHTEIN E, et al. Proceedings of the 26th International Conference on World Wide Web, WWW 2017, Perth, Australia, April 3-7, 2017. ACM, 2017: 1271–1279.
3. Zhou Z, Liu S, Xu G, et al. Knowledge-Based Recommendation with Hierarchical. Collaborative Embedding[C]//PHUNG D Q, TSENG V S, WEBB G I, et al. Advances in Knowledge Discovery and Data Mining - 22nd Pacific-Asia Conference, PAKDD 2018, Melbourne, VIC, Australia, June 3-6, 2018, Proceedings, Part II. Springer, 2018, 10938: 222–234.
4. Bollacker K, Evans C, Paritosh P, et al. Freebase: a collaboratively created graph database for. structuring human knowledge[C]//Proceedings of the 2008 ACM SIGMOD international conference on Management of data - SIGMOD ’08. Vancouver, Canada: ACM Press, 2008: 1247.
5. Mahdisoltani F, Biega J, Suchanek F M. YAGO3: A Knowledge Base from Multilingual. Wikipedias[C]//CIDR 2015, Seventh Biennial Conference on Innovative Data Systems Research, Asilomar, CA, USA, January 4-7, 2015, Online Proceedings. www.cidrdb.org, 2015.
6. Deepak Nathani,Jatin Chauhan,Charu Sharma,Manohar Kaul. Learning Attention-based. Embeddings for Relation Prediction in Knowledge Graphs. [J]. CoRR,2019, abs /1906.01195.
7. Miller G A. WordNet: a lexical database for English[J]. Communications of the ACM, 1995, 38(11): 39–41.
8. Ji Shaoxiong,Pan Shirui,Cambria Erik,Marttinen Pekka,Yu Philip S. A Survey on Knowledge. Graphs: Representation, Acquisition, and Applications. [J]. IEEE transactions on neural networks and learning systems,2021, PP.
9. Ji Shaoxiong,Pan Shirui,Cambria Erik,Marttinen Pekka,Yu Philip S. A Survey on Knowledge. Graphs: Representation, Acquisition, and Applications. [J]. IEEE transactions on neural networks and learning systems,2021, PP.
10. Han Xiao 0005, Minlie Huang,Yu Hao,Xiaoyan Zhu. TransA: An Adaptive Approach for. Knowledge Graph Embedding. [J]. CoRR,2015, abs /1509.05490.
11. Bordes A, Usunier N, Garcia-Duran A , et al. Translating Embeddings for Modeling Multi-relational Data. Curran Associates Inc. 2013.
12. Msahli M, Qiu H, Zheng Q, et al. Topological Graph Convolutional Network-Based Urban Traffic Flow and Density Prediction[J]. IEEE Transactions on Intelligent Transportation Systems, 2020, PP(99).
13. Hanie Sedghi,Ashish Sabharwal. Knowledge Completion for Generics using Guided Tensor. Factorization. [J]. CoRR,2016, abs /1612.03871.
14. Patents; "Polynomial Method of Constructing a Non-Deterministic (NP) Turing Machine" in. Patent Application Approval Process (USPTO 20160012339) [J]. Politics & Government Week,2016.
15. Ivana Balazevic,Carl Allen,Timothy M. Hospedales. TuckER: Tensor Factorization for. Knowledge Graph Completion. [J]. CoRR,2019, abs/1901.09590:

16. Tamara G. Kolda,Brett W. Bader. Tensor Decompositions and Applications[J]. SIAM. Review,2009,51(3).
17. Lin Y, Liu Z, Sun M, et al. Learning Entity and Relation Embeddings for Knowledge Graph. Completion[C]//BONET B, KOENIG S. Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA. AAAI Press, 2015: 2181–2187.
18. Fan M, Zhou Q , Chang E , et al. Transition-based Knowledge Graph Embedding with Relational Mapping Properties. 2014.
19. Xiao H, Huang M, Zhu X. From One Point to a Manifold: Knowledge Graph Embedding for. Precise Link Prediction[C]//KAMBHAMPATI S. Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, NY, USA, 9-15 July 2016. IJCAI/AAAI Press, 2016: 1315–1321.
20. Ji G, Liu K, He S, et al. Knowledge Graph Completion with Adaptive Sparse Transfer. Matrix[C]//SCHUURMANS D, WELLMAN M P. Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA. AAAI Press, 2016: 985–991.
21. Wang Z, Zhang J, Feng J, et al. Knowledge Graph Embedding by Translating on. Hyperplanes[C]//BRODLEY C E, STONE P. Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, July 27 -31, 2014, Québec City, Québec, Canada. AAAI Press, 2014: 1112–1119.
22. Socker R,Chen D,Manning C D,et al. Reasoning with neural tensor networks for knowledge. base completion[C]//Advances in neural information processing systems. 2013:926-934.
23. Nickel, Maximilian,Murphy, Kevin,Tresp, Volker,Gabrilovich, Evgeniy. A Review of. Relational Machine Learning for Knowledge Graphs[J]. Proceedings of the IEEE,2016,104(1).
24. Schlichtkrull M S, Kipf T N, Bloem P, et al. Modeling Relational Data with Graph. Convolutional Networks[C]//GANGEMI A, NAVIGLI R, VIDAL M-E, et al. The Semantic Web - 15th International Conference, ESWC 2018, Heraklion, Crete, Greece, June 3-7, 2018, Proceedings. Springer, 2018, 10843: 593–607.
25. Dettmers T, Minervini P, Stenetorp P, et al. Convolutional 2D Knowledge Graph. Embeddings[C]//MCILRAITH S A, WEINBERGER K Q. Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018. AAAI Press, 2018: 1811–1818.
26. Nguyen D Q, Nguyen T D, Nguyen D Q, et al. A Novel Embedding Model for Knowledge. Base Completion Based on Convolutional Neural Network[C]//WALKER M A, JI H, STENT A. Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 2 (Short Papers). Association for Computational Linguistics, 2018: 327–333.
27. Nickel M, Tresp V, Kriegel H-P. A Three-Way Model for Collective Learning on Multi-Relational Data[C]//GETOOR L, SCHEFFER T. Proceedings of the 28th International Conference on Machine Learning, ICML 2011, Bellevue, Washington, USA, June 28 - July 2, 2011. Omnipress, 2011: 809–816.
28. Yang B, Yih W, He X, et al. Embedding Entities and Relations for Learning and Inference in. Knowledge Bases[C]//BENGIO Y, LECUN Y. 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings. 2015.

29. Trouillon T, Welbl J, Riedel S, et al. Complex Embeddings for Simple Link. Prediction[C]//BALCAN M-F, WEINBERGER K Q. Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016. JMLR.org, 2016, 48: 2071–2080.
30. Kazemi S M, Poole D. SimplE Embedding for Link Prediction in Knowledge. Graphs[C]//BENGIO S, WALLACH H M, LAROCHELLE H, et al. Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, 3-8 December 2018, Montréal, Canada. 2018: 4289–4300.
31. Farahnaz Akrami, Mohammed Samiul Saeef, Qingheng Zhang, Wei Hu, Chengkai Li. Realistic Re-evaluation of Knowledge Graph Completion Methods: An Experimental Study[P]. Management of Data,2020.
32. Toutanova, K., Chen, D., Pantel, P., Poon, H., Choudhury, P., and Gamon, M. Representing Text for Joint Embedding of Text and Knowledge Bases. In Proceedings of the 2015Conference on Empirical Methods in Natural Language Processing, 2015.
33. Xiong W, Hoang T, Wang W Y. De eppath: A reinforcement learning method for knowledge graph reasoning[J].arXivpreprintarXiv:1707.06690,201
34. Xi Victoria Lin, Richard Socher, and Caiming Xiong. 2018. Multi-hop knowledge graph reasoning with reward shaping. In Proceedings of the 2018 Con- ference on Empirical Methods in Natural Language Processing (EMNLP), 2018.
35. Sun Z , Deng Z H , Nie J Y , et al. RotatE: Knowledge Graph Embedding by Relational Rotation in Complex Space[J]. 2019.
36. Chao L , He J , Wang T , et al. PairRE: Knowledge Graph Embeddings via Paired Relation Vectors[J]. 2020.
37. Vashishth S , Sanyal S , Nitin V , et al. InteractE: Improving Convolution-Based Knowledge Graph Embeddings by Increasing Feature Interactions[J]. Proceedings of the AAAI Conference on Artificial Intelligence, 2020, 34(3):3009-3016.